

## 网络化移动应用的全局适应性数据预取机制<sup>\*</sup>

张晓薇<sup>1,2</sup>, 曹东刚<sup>1,2+</sup>, 田刚<sup>1,2</sup>, 陈向群<sup>1,2</sup>

<sup>1</sup>(北京大学 计算机科学技术系,北京 100871)

<sup>2</sup>(高可信软件技术教育部重点实验室(北京大学),北京 100871)

### Globally Adaptive Data Prefetching Mechanism for Mobile Network Applications

ZHANG Xiao-Wei<sup>1,2</sup>, CAO Dong-Gang<sup>1,2+</sup>, TIAN Gang<sup>1,2</sup>, CHEN Xiang-Qun<sup>1,2</sup>

<sup>1</sup>(Department of Computer Science and Technology, Peking University, Beijing 100871, China)

<sup>2</sup>(Key Laboratory of High Confidence Software Technologies (Peking University), Beijing 100871, China)

+ Corresponding author: E-mail: caodg@sei.pku.edu.cn

Zhang XW, Cao DG, Tian G, Chen XQ. Globally adaptive data prefetching mechanism for mobile network applications. *Journal of Software*, 2010,21(8):1783-1794. <http://www.jos.org.cn/1000-9825/3617.htm>

**Abstract:** This paper presents an approach named Multiple Goals Oriented Data Prefetching (MGODP) to satisfy the data prefetching requirements from different users. MGODP does not only take the user preferences into account to prefetch appropriate amounts of data, but also adopts global coordination for Client/Server data access model to greatly improve the quality of service from the server's perspective. Moreover, MGODP balances the workload between the mobile client and the backend server to achieve proper allocation of resources and to guarantee the system performance. Experimental results demonstrate that MGODP could satisfy diverse user requirements, and reduce the consumption of battery energy and network bandwidth through global coordination and workload balance.

**Key words:** data prefetching; user preference; context-aware; mobile computing; global coordination

**摘要:** 提出一种多目标的数据预取方法(multiple goals oriented data prefetching,简称MGODP)来满足不同用户的数据预取需求.MGODP 不仅从用户偏好出发为其预取合适量的数据,而且从服务器角度出发,对于 Client/ Server 模式下的数据访问提出全局合作的方法,以大幅度提高服务质量.另外,MGODP 提供了移动客户端和服务器之间平衡工作负载的合作机制,合理分配系统资源,保障系统性能.通过一系列实验可以看出,MGODP 方法能够很好地满足不同用户的需求,并通过全局合作和负载均衡机制在保证用户性能需求的前提下,尽可能地减小对电池电量和网络带宽的消耗.

**关键词:** 数据预取;用户偏好;上下文可感知;移动计算;全局协作

中图法分类号: TP316 文献标识码: A

\* Supported by the National Natural Science Foundation of China under Grant Nos.60503029, 90412011 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2007AA01Z133, 2009AA01Z139 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2005CB321805 (国家重点基础研究发展计划(973))

Received 2008-10-20; Revised 2009-01-14; Accepted 2009-03-31

随着移动设备和移动网络的飞速发展,移动应用呈现出网络化趋势,以移动设备作为客户端的 Client/Server(C/S)模式和 Browser/Server(B/S)模式开始普及,移动设备可以充分利用服务器的计算资源和存储资源,以及移动网络提供的快速数据传输服务来满足用户更丰富的应用需求.但是,移动环境的动态变化性和移动设备的资源受限性会严重影响网络化移动应用的使用效果和服务质量<sup>[1]</sup>.一方面,移动应用所处的上下文环境是不断变化的,例如,移动设备所处的物理环境、移动网络、电池电量等都是动态变化的;另一方面,移动设备通常具有有限的电池电量供应、CPU 处理能力和内存空间等.

在不稳定的移动网络条件下,本地数据缓存机制能够较为有效地提高网络化移动应用的数据访问效率和执行效率.通过本地缓存,用户可以在真正使用数据之前将其缓存在本地,对其访问并进行修改,并在适当的时刻与服务器同步对数据的修改.但是,本地数据缓存会占用较多的移动网络带宽,同时为移动设备带来更大的能量消耗.因此,本地数据缓存涉及到数据访问效率与资源消耗之间的权衡.

然而,不同用户对于不同应用有着不同的需求偏好.例如,能耗偏好型的用户一般希望应用程序尽可能地节能,从而延长电池的使用时间;而性能偏好型的用户则希望尽可能地提高应用程序的执行效率,从而缩短程序的执行时间.因此,在移动设备和移动环境资源受限的条件下,如何通过预取缓存合适的数据,使得移动应用更好地满足不同用户的不同需求偏好,成为有效提高网络化移动应用服务质量的关键方法.

本文提出了多目标的数据预取机制(multiple goals oriented data prefetching,简称 MGODP)将用户对于性能和能耗的偏好考虑进去,使得移动应用可以预取缓存合适的数据量,与用户的偏好达成匹配.网络化移动应用场景通常由很多移动客户端和若干服务器组成,而多个客户端同时对服务器提出数据预取请求可能导致其对移动网络带宽和服务器计算资源的竞争,从而带来移动客户端预取行为的相互干扰,影响整个系统的运行效率.对此,MGODP 采用全局优化策略为网络中的各个移动客户端预取合适量的数据,尽可能地降低各移动客户端之间的行为干扰.本文采用成本指标(cost index)和利益指标(benefit index)来衡量预取数据带来的成本和好处,而其权重值则反映用户对于性能和能耗的偏好.若成本参数的权重值高,则意味着用户偏好于节约能耗;采用惩罚指标(penalty index)来衡量数据预取操作对网络中其他移动节点的数据访问服务所造成的影响,并在惩罚指标中加入数据预取服务与数据访问服务的优先级区别,使得数据预取服务不会对移动客户端正常的数据访问造成过多影响.最后,移动客户端与服务器端的合作,可以对数据进行符合用户偏好的适当处理(如图片压缩、数据合并等).

本文的主要贡献在于:

- (1) 通过引入用户偏好,可以为用户预取合适量的数据,从而达到数据访问效率与资源消耗的权衡与用户目标之间的匹配.
- (2) 通过量化不同客户端之间数据预取行为的干扰以及数据预取服务与数据访问服务之间的干扰,使得数据预取达到系统网络范围内的全局优化,并保证服务器的服务响应速度.
- (3) 通过移动客户端与服务器端的合作,使得数据处理更加有效,同时更好地满足用户偏好.

## 1 相关工作

缓存管理是移动计算中非常重要的研究内容,很多研究人员对缓存服务<sup>[2-6]</sup>和数据预取<sup>[7-12]</sup>进行了深入的研究和探讨.本文的工作主要借鉴了其衡量数据预取的若干因素变量,并将其整合到网络化移动应用的特定应用背景中,通过成本指标、利益指标和惩罚指标的量化比较,对数据预取量进行合理调整,从而有效提高网络化移动应用的服务质量.

文献[7]提出了一个满足移动多媒体应用 QoS 请求的数据预取模型,利用马尔可夫链对未来访问每个数据项的可能性进行预测.为了保证整个系统的性能指标,选择最需要的数据进行预取,一般认为,未来最有可能被访问到的数据是应用程序当前最需要的;另外考虑到移动设备的资源限制,会估算预取所耗资源,从而判断当前的资源情况是否能够满足预取需要.相比之下,本文的工作选取更多数据属性和环境属性对预取行为进行考察,从而能够更加准确地判断数据对于应用的效用性.

文献[9]注意到现有的数据预取技术没有考虑电池剩余电量、数据大小、数据被访问频率、数据更新频率等因素对数据预取的影响,因此提出一个可感知电量变化的预取机制 VAP(value-based adaptive prefetch).通过构建预取代价函数,对其最小化来动态确定所预取的数据及预取量.我们注意到,最小化预取代价的方法并不能使数据预取满足不同用户的目标,只能使其满足占用资源最小化,也就是我们提到的节约能耗这一方面的需求;而有些用户则希望通过数据预取获得最优性能,VAP 方法不能很好地满足此类需求.

文献[10]提出了一个考虑能源和带宽占用情况的数据缓存机制 GreedyDual Least Utility,可以提高无连接状态时的数据可获得性和一致性,并在性能和资源占用之间进行权衡,但是它没有考虑每个移动节点进行数据预取时对网络带宽的占用对其他移动节点所造成的影响.相比之下,本文提出的数据预取机制从全局视角对数据预取进行优化,以提高网络化移动应用的整体运行效果.

文献[11]主要关注在并发控制系统体系结构下利用数据预取技术,在网络空闲时进行数据预取操作,达到缩短平均数据访问时间,满足即时数据访问需求的目的,从而为用户提供快捷、有效的数据访问服务.相比之下,本文加入了用户偏好和全局协作机制,能够对数据访问的整体效用进行更准确的判断,从而更好地提高数据访问服务质量.

文献[12]考虑了内存大小、电池电量以及数据传输成本等因素对预取的影响,从而决定是否为用户自动下载网页内容,但其没有考虑网页内容未来被访问的可能性及其对用户效用的影响.相比之下,本文提出的数据预取机制从客户端和服务端根据应用特性来获取数据的被访问可能性信息,从而有效地提高了预取数据效用.

对于与位置服务相关的移动应用而言,往往需要预测移动设备的未来位置并依据此位置信息提供相关的服务,这涉及到预测移动路径,预取位置相关数据等一系列问题.文献[5]对位置相关服务中的数据预取效用问题进行了探讨,提出一种权衡响应时间和占用带宽、内存的数据预取算法 Benefit Oriented Prefetching(BOP),从而在有效利用网络带宽和内存资源的前提下,尽可能地提高数据对于客户的有用性.

与这些相关工作相比,本文的工作更侧重于网络化的移动应用场景,对其应用特性进行更全面的建模,从而对全局的数据预取策略进行优化,并对客户端与服务端的合作机制进行整合,使得两端目标协同一致,从而提高资源的使用效率和数据效用.

## 2 系统模型

在本文提出的数据预取方法 MGODP 中,移动客户端和服务端是其参与方,其中网关和数据源是服务器端的两个重要组成部分.图 1 展示了 MGODP 的系统体系结构.移动客户端了解当前的上下文信息(如电池能量、网络带宽等)、用户偏好(偏好节约能耗或偏好提高性能)和应用程序特性,并据此确定预取原则(参见第 2.2 节);网关为移动客户端从数据源获取其所需数据,依据用户偏好和当前上下文对数据进行适当处理(如压缩、译码等),并将数据发送给移动客户端;另外,多个移动客户端同时进行数据访问对整个系统所造成的影响,将通过全局协作方法加以解决(参见第 3.3 节).因此,我们的系统模型具有以下特点:

- (1) 可配置性.根据应用特性对预取原则进行配置,根据用户需求对用户偏好进行配置.
- (2) 可扩展性.客户端、服务器端的各模块均采用构件化技术进行设计和实现,从而保证对多种网络应用的扩展能力以及对各模块的演化替换.
- (3) 按需性.MGODP 为用户预取满足其偏好和需求的数据,使得这些数据在符合用户目标的前提下,为用户带来最大的效用.
- (4) 全局协作性.移动客户端与服务端协作,为用户提供满意的数据预取服务.同时,从整个系统角度出发,在当前网络带宽和计算资源的限制条件下尽可能地满足所有移动用户的数据需求.

在 MGODP 中,我们主要试图解决以下几个问题:

- (1) 需要为移动用户预取哪些数据?
- (2) 需要为移动用户预取多少数据?

- (3) 移动客户端和网关如何根据应用特性进行合作,对数据进行适当处理,更好地满足用户体验?
- (4) 如何进行移动客户端和服务器端的全局资源协调,从而保证整个应用系统的性能?

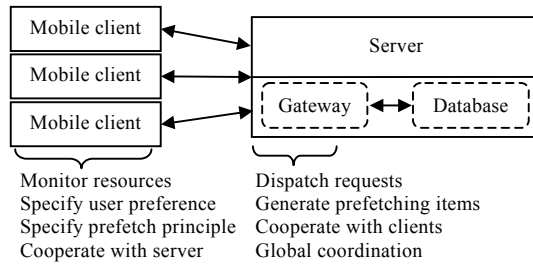


Fig.1 System architecture of MGODP

图 1 MGODP 系统体系结构

2.1 应用实例分析

为了集中展示 MGODP 的特点,本文选择以移动相册作为网络化移动应用的典型示例.图 2 展示了基于 MGODP、移动相册进行数据预取的基本步骤.当移动设备向服务器端请求数据时,服务器端除立即响应请求外,还要根据预取原则和预取算法为用户确定需要预取的数据(参见图 2 步骤(3)~步骤(5)),其中,相片 B,C,D 与相片 A 最为相关,在未来最有可能被访问到.

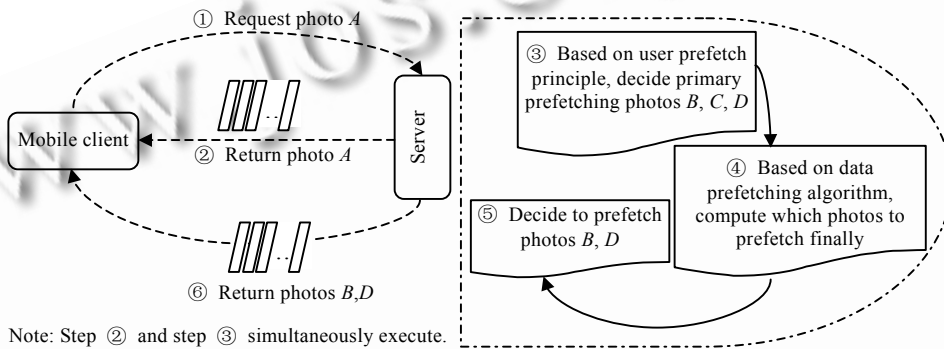


Fig.2 Steps to prefetch data for mobile album using MGODP

图 2 基于 MGODP 的移动相册数据预取步骤示例

2.2 客户端体系结构

图 3 展示了 MGODP 客户端的体系结构,由应用逻辑模块和预取模块两部分组成.应用逻辑模块主要包含完成应用业务子模块以及与服务器端交互的子模块(数据请求和数据接收子模块).

上下文感知子模块负责完成对移动客户端所处的当前上下文环境信息的采集,包括电池电量、网络带宽、内存大小等,并会将感知到的信息整合在数据请求中一并发送给服务器.这些上下文信息对于预取数据项的选择具有关键作用,从而影响用户效用及使用体验.

用户偏好配置子模块负责完成与用户偏好信息配置相关的任务.用户偏好决定了需要预取多少数据——对于性能偏好者需要为其预取较多数据.用户偏好反映为成本指标和利益指标的权重值,从而决定预取数据量(参见第 3.2 节).另外,用户偏好也会决定服务器端对数据的处理方式.例如,若用户偏好节约能耗,则服务器会将真彩图片压缩为灰度图片,使得移动客户端的处理更为简单,从而起到节约能耗的效果.开发人员可以通过如下 API 来设置用户偏好:

```
setPreference (double preferToEnergy){
```

```

energyPrefer=preferToEnergy;
performancePrefer=1-preferToEnergy;
}
    
```

预取原则配置子模块指明从何处获取数据未来被访问的可能性信息.预取数据时应尽可能地预取那些未来最可能被访问到的数据.在 MGODP 方法中,可通过两种方式来获得数据的被访问可能性信息:一种方式是基于用户的个人访问习惯.例如,某用户经常使用 PDA 访问个人移动相册,他浏览照片 A 之后通常会浏览照片 B.这时,应用会将他的访问习惯收集存储存储在本地.下次当他访问照片 A 时,应用会自动为其预取照片 B,从而减少访问延迟提高应用性能;另一种方式则通过服务器提供的推荐信息来获得未来最可能访问到的数据.例如,某服务器提供多种食品信息,每天都有很多用户访问该服务器.因此服务器收集到不同食品信息之间的相关性信息,如巧克力与糖果之间有很强的相关性.这样,当用户访问某食品信息时,服务器可以自动为其推荐最相关、最有可能在未来被访问到的其他信息.

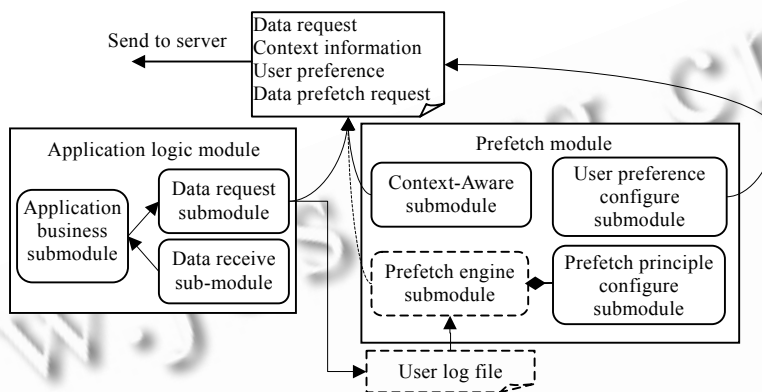


Fig.3 Architecture of MGODP mobile client

图 3 MGODP 移动客户端体系结构

如何获取数据的被访问可能性信息由应用特性决定,而预取原则反映了该信息获取方式的不同,开发人员可以根据应用的特点利用以下 API 为其指定不同的预取原则:

```

setPrefetchPrinciple (int prefetchMode, RecordStore filename)
//prefetchMode=1→基于用户的个人访问习惯,应指定存储相关性信息的文件
//prefetchMode=0→基于服务器端的推荐
    
```

另外,若采用第 1 种方式从移动客户端获取数据的未来被访问可能性信息,则在客户端应具有预测引擎子模块,它负责从用户使用轨迹文件中选择与当前数据请求项最为相关的若干备选预取数据项,这些预取请求也将与数据请求整合,发往服务器端.

### 2.3 服务器端体系结构

图 4 展示了 MGODP 服务器端的体系结构,由应用逻辑模块、优化引擎模块和预取引擎模块组成.应用逻辑模块包含完成服务流程的业务子模块以及与客户端交互的子模块(请求接收和数据发送子模块).优化引擎模块(包含合作数据处理子模块和全局协作子模块)主要从客户端与服务器端的合作角度以及应用系统的全局角度出发,对原始请求数据和预取数据进行优化处理.

合作数据处理子模块基于用户偏好对数据进行合适的处理(参见第 2.2 节中用户偏好配置子模块),移动客户端将用户偏好通过 HTTP 消息发送给服务器端,服务器考虑用户偏好和当前上下文环境对数据进行诸如压缩、色彩调整等处理.

全局协作子模块基于网络带宽负载及服务器资源,在不同客户端之间以及数据预取服务和数据访问服务

之间进行合理的资源分配,从而达到系统整体性能最优化:一个客户端进行数据预取,会占用网络带宽及服务器资源,从而影响其他客户端的服务请求,造成客户端之间的冲突(参见第 3.2 节);而数据预取服务和数据访问服务也会相互竞争网络带宽和服务器资源,从而造成不同服务之间的冲突(参见第 3.2 节).

若预取原则中的数据未来被访问可能性信息来自服务器端的推荐信息,服务器端应具有预取引擎模块来完成对预取数据项的选择任务.这些被选择的预取数据项经过优化引擎模块的处理返回给客户端.

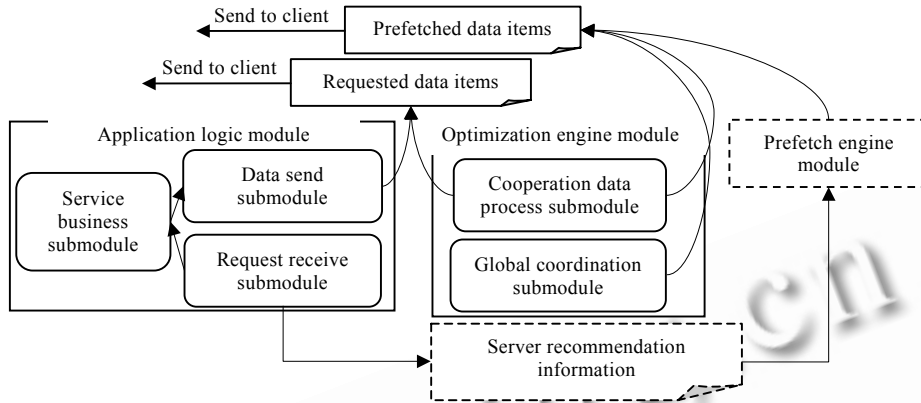


Fig.4 Architecture of MGODP server

图 4 MGODP 服务器端体系结构

### 3 数据预取算法

数据预取算法作为 MGODP 方法的核心,解决了备选预取数据的择取及最终预取决定的计算等关键问题.其中,备选预取数据  $S_{AL(k)}$  由若干未来最有可能被访问的数据项组成,  $S_{AL(k)} = \{i | \text{数据 } i \text{ 很可能被客户端 } k \text{ 预取}\}$ .

备选预取数据的择取因预取原则的不同而有所区别.若备选预取数据信息来源于客户端,其数据访问模型如图 5(a)所示,此时,客户端掌握数据访问可能性的信息.本文第 2.1 节中展示的移动相册应用的数据预取步骤所对应的预取原则就是依据客户端信息来确定可能需要预取的数据项.在如图 5(b)所示的数据访问模型中,该信息则来源于服务器端,对应于本文第 2.2 节中预取原则的第 2 种获取数据被访问可能性信息的方式.因此,不同的预取原则使得备选预取数据信息来源不同,带来不同的数据访问模型和数据预取流程.

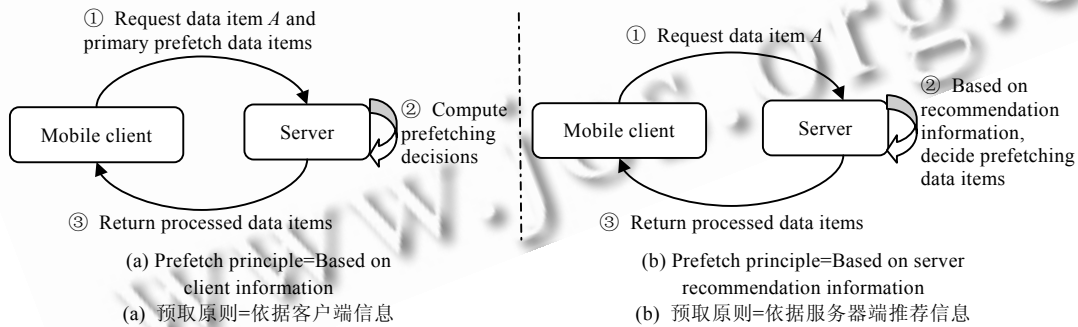


Fig.5 Data access model

图 5 数据访问模型

#### 3.1 备选预取数据的择取

请求数据项与备选预取数据之间的相关性可以表述为函数关系  $\varphi: S \rightarrow R$ , 其中,集合  $S$  由所有数据项对组成

$S = \{(A, B) | A, B \in \text{数据项集合}\}$ , 集合  $R$  表示两个数据项之间的相关性  $R = [0, 1]$ ,  $r = 0, r \in R$  表示两个数据项之间不存在前后顺序访问关系, 也就是访问数据项  $A$  之后不会紧接着访问  $B$ ,  $r = 1, r \in R$  则表示数据项之间存在必然的前后顺序访问关系。

选择备选预取数据时, 根据与所请求数据的相关性大小选择最相关的若干数据作为备选, 具体采用堆排序算法选择前  $k$  个数据项(其中,  $k$  值可配置), 算法复杂度为  $O(k \log n)$ 。

备选预取数据  $S_{AL(k)}$  确定之后, 则通过计算来判断  $S_{AL(k)}$  中的每个数据项是否进行最终预取。考量每个数据项时, 主要通过计算其成本指标(利益指标和惩罚指标)来进行综合评定。

为了得到最终的预取决定, 数据预取算法需要经过如图 6 所示的若干步骤。

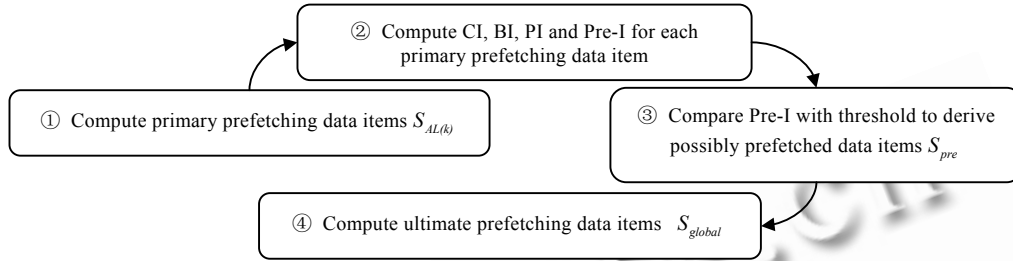


Fig.6 Major steps of data prefetching algorithm

图 6 数据预取算法的主要步骤

### 3.2 成本指标(cost index, 简称CI)、利益指标(benefit index, 简称BI)、惩罚指标(penalty index, 简称PI)、预取指标(prefetch index, 简称Pre-I)

很多因素会影响是否预取某数据项的决定, 下面列举 MGODP 方法中与计算预取决定相关的因素变量:

- $s_i$ : 数据项  $i$  的大小.
- $e$ : 剩余电池电量.
- $p_i$ : 数据项  $i$  未来被访问到的概率.
- $n$ : 当前网络带宽.
- $ur_i$ : 数据项  $i$  的更新频率, 单位是次数/分钟.
- $ar_i$ : 数据项  $i$  的访问频率, 单位是次数/分钟.
- $\alpha$ : 传输每比特数据所消耗的能量.
- $\beta$ : 从本地缓存访问数据所需的时间.

定义成本指标为预取所消耗的电池电量占整个剩余电池电量的比例,

$$CI_i = \frac{\alpha \times s_i}{e} p_i.$$

定义利益指标为与不进行数据预取相比, 进行预取提高性能的比例,

$$BI_i = p_i \frac{ar_i}{ur_i} \times \frac{s_i - n\beta}{s_i}.$$

服务器端同时为移动客户端提供数据访问服务和数据预取服务. 两种服务都会消耗服务器端的计算资源并占用网络带宽, 因此会造成资源竞争情况的出现. 相对于数据预取服务而言, 数据访问服务应该有更高的优先级, 因为其服务质量对整个应用的性能有更重要的影响. 为了反映数据访问服务和数据预取服务的优先级差别, 本文引入优先级权值(priority weight, 简称  $PriW$ ), 对其加以反映,  $PriW_a$  和  $PriW_p$  分别表示数据访问服务和预取服务的优先级, 且满足如下的条件:  $PriW_a > PriW_p, PriW_a + PriW_p = 1$ .

定义惩罚指标  $PI_j$  为与不进行数据预取相比, 移动客户端  $j$  进行预取给其他移动节点带来的时间延迟,

$$PI_j = \frac{\sum_{i \in S_{AL(j)}} isPrefetch(i) \times s_i}{PriW_p \times \left[ \sum_{k \in S_{active}, k \neq j} \left( \sum_{i \in S_{AL(k)}} isPrefetch(i) \times s_i \right) \right] + PriW_a \times \sum_{k \in S_{active}} ts_k}$$

其中,  $ts_k$  是移动客户端  $k$  所请求数据的总大小(不包括其预取的数据),  $S_{active} = \{i | \text{移动客户端 } k \text{ 正从服务器端请求数据}\}$ ,  $isPrefetch(i) = \begin{cases} 1, & \text{若数据项 } i \text{ 被预取} \\ 0, & \text{若数据项 } i \text{ 不被预取} \end{cases}$ .

基于成本指标和利益指标,可以定义预取指标来反映预取操作对于应用程序的效用:

$$Pre-I_i = w_p \times BI_i - w_e \times CI_i,$$

其中,  $w_p$  和  $w_e$  分别表示利益指标和成本指标的权重,且  $w_p + w_e = 1$ .

通过比较  $Pre-I_i$  和阈值(关于阈值的确定参见第 4.1 节),对  $S_{AL(k)}$  进行进一步筛选,从而得到一个由满足阈值条件的可能被预取数据项组成的集合  $S_{pre}: S_{pre} = \{i | k \in S_{active}, \forall i \in S_{AL(k)}, Pre-I_i > threshold\}$ , 且  $S_{pre} \subseteq S_{AL(k)}$ .

为了便于与阈值比较,  $BI_i$  和  $CI_i$  的取值都将归一化到  $[0, 1]$  之间. 只有当  $Pre-I_i$  大于阈值时, 该数据项才有可能被预取, 最终是否被预取还要从全局协作的角度衡量其惩罚指标是否过大(参见第 3.3 节).  $w_p$  越大, 表明用户越偏好提高性能, 因此  $Pre-I_i$  的取值会越大, 从而预取更多的数据项, 用户偏好恰好得到满足; 相反,  $w_e$  越大, 表明用户越偏好节约能耗, 使得  $Pre-I_i$  的取值越小, 从而减少预取数据量以帮助达成节能目标. 可以通过 `API setPreference(double preferToEnergy)` 来设置  $w_p$  和  $w_e$ .

### 3.3 全局协作

为了优化整个系统的综合运行效果, 使来自不同移动客户端的并发数据访问请求和数据预取请求在考虑相互影响的前提下得到最好的满足, 本文采用全局协作的方式决定最终预取哪些数据.

之所以需要考虑数据预取行为对整个系统的影响, 是因为数据预取会占用服务器的计算资源和网络带宽, 导致其他数据访问服务的延迟. 本文的全局协作算法会对预取之于应用程序的效用和预取之于整个系统的延时代价进行权衡, 因此全局协作算法可以表述为如下的最优化问题:

$$\max \left\{ \sum_{i \in S_{pre}} isPrefetch(i) \times BI_i - \sum_{k \in S_{active}} PI_k \right\}.$$

全局协作问题的解空间是使上述表达式取值最大化的  $isPrefetch(i)$  的值. 通过全局协作的处理可以得到一个由最终预取数据项组成的集合  $S_{global}: S_{global} = \{i | isPrefetch(i) = 1\}$ , 且  $S_{global} \subseteq S_{pre}$ .

虽然整数规划问题是一个 NP-Hard 问题, 但是由于上述最优化问题的取值范围为 0 或 1, 因此其算法复杂度为  $O(n)$ , 并不会给服务器端带来过重的计算负担. 虽然全局协作处理过程会消耗服务器计算资源, 但考虑到对于一对多的网络化移动应用服务模式, 服务器端的网络带宽往往成为系统性能瓶颈, 网络带宽资源相对于计算资源更为宝贵. 因此, 本文的全局协作算法使服务器资源在整个系统网络中得到更为合理的权衡和分配.

## 4 实验

本文基于 Java ME 平台实现了 MGODP, 并以 DELL X51 PDA 作为移动客户端, 以 DELL PowerEdge 1600/SC 作为服务器, 以移动个人相册作为实例(参见第 2.1 节)进行实验测试. 移动用户需要从远端数据服务器获取其所需的照片, 并按照用户偏好预取适量的照片数据. 根据第 2.2 节的分析, 配置预取原则如下:

$$setPrefetchPrinciple(1, predictFile).$$

另外, 在我们的数据预取算法中, 参数  $PriW_a$  和  $PriW_p$  也需要进行配置, 在以下实验中配置如下:  $PriW_a = 0.8$ ,  $PriW_p = 0.2$ .  $PriW_a$  和  $PriW_p$  的取值相对大小反映了数据访问服务和数据预取服务之间的优先级高低(参见第 3.2 节).  $PriW_a$  和  $PriW_p$  的取值只会影响实验中变量关系的大小, 并不会对实验所呈现的变量关系的性质产生影响. 例如,  $PriW_a$  越大, 数据访问服务的优先级就越高, 则预取数据的可能性以及预取为用户带来的效用就会随之降



低.因此,本文的实验选择较大的  $PriW_a$ ,使实验结果相对保守,从而具有更大的说服力.

由于移动个人相册需要依据客户端采集的用户使用习惯得到备选预取数据,本文采用文献[7]中所采用的预测未来可能行为的方法.图 7 展示了其访问预测模型,其中:

$$freq(i)++, \text{ 若数据项 } i \text{ 被访问,}$$

$$\exists link(i, j) \wedge freq(i, j)++, \text{ 若数据项 } i \text{ 被访问后数据项 } j \text{ 立即被访问.}$$

可以看出,该访问预测模型实际上是一个一阶马尔可夫链模型.当用户访问数据项  $i$  时,将从  $(i,j)(i,k)(i,m)$  这 3 个分支中选择  $freq(i,*)$  最大的若干数据项作为备选预取数据.

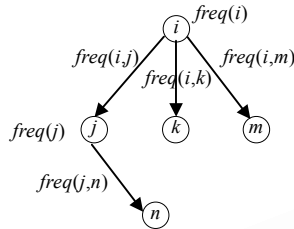


Fig.7 Access prediction model

图 7 访问预测模型

### 4.1 预取阈值

预取指标  $Pre-I_i$  需要与阈值比较,以确定该数据项是否可以放入集合  $S_{pre}$  中(参见第 3.2 节).因此,阈值的设定对于用户最终的满意程度有很大的影响.图 8 显示了阈值、本地缓存命中率与用户偏好之间的关系.随着阈值的增大,预取的数据量会减少,导致本地缓存命中率下降,不同用户偏好下的命中率曲线都随着阈值的增加而向下倾斜(如图 8 所示).不同的用户偏好对命中率的要求有所不同,如果用户非常偏好提高性能,则需要高命中率(对应于图 8 中的 A 点),随着用户偏好性能程度的降低,所要求的命中率也会不断降低(对应于图 8 中的 B 点),最终可以得到随着  $w_p$  增加而所对应的阈值减小的阈值曲线(参见图 8 中由若干圆点连接而成的向下倾斜的虚线),其中,每个圆点表示此  $w_p$  值所对应的阈值取值.例如,当  $w_p=0.1$  时,所选取的阈值为 0.82.因此,阈值曲线描述了用户偏好与阈值选择之间的对应关系,为下一步实验提供了阈值选择标准.

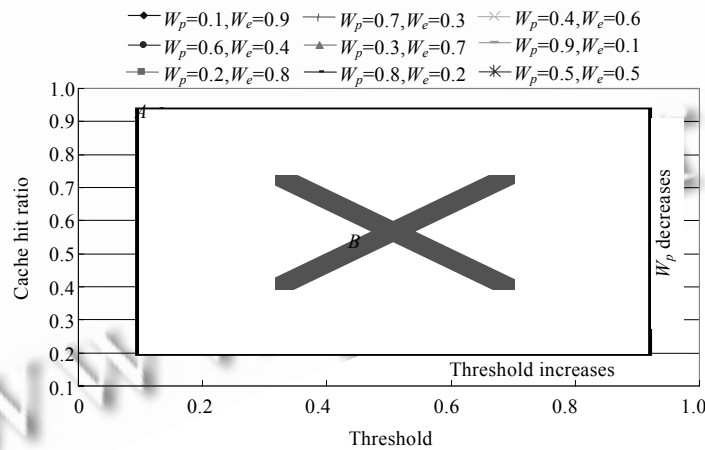


Fig.8 Relationship among threshold, cache hit ratio and user preference

图 8 阈值、命中率与用户偏好的关系

### 4.2 考虑用户偏好和移动客户端/服务器端合作的实验比对

图 9 显示了用户偏好、预取率、响应时间和剩余电池电量之间的关系.在实验过程中,我们选择不同的  $w_p$

值(0.1~0.9)来反映用户偏好, $w_p$ 值越大,表明用户越偏好性能.在每个 $w_p$ 取值上进行10次实验,每次实验包含100个数据请求,对每次实验计算其预取率和响应时间.最后对10次实验结果取算术平均,从而保证实验结果的可靠性和稳定性.从图9可以看出,随着 $w_p$ 的增加,预取率增加,请求的响应时间缩短,剩余电池电量减少.

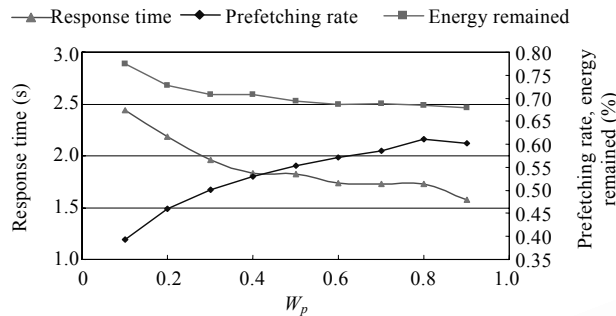


Fig.9 Relationship among user preference, prefetching rate, response time and energy remained  
图9 用户偏好、预取率、响应时间及剩余电量的关系

由于考虑了用户偏好,可以更好地满足用户需求,例如,为能耗偏好者节约更多电量.另外,本文考虑了移动客户端和服务端之间的合作,可以对数据进行符合用户偏好的适当处理,更好地帮助用户达成目标.图10显示了用户偏好和移动客户端/服务器端合作对数据预取的影响.

为了考察用户偏好和移动客户端/服务器端合作在节约能耗方面所起到的作用,我们选择不同的 $w_e$ 值(0.1~0.9)进行实验.在每个 $w_e$ 取值上进行10次实验,每次实验包含100个数据请求,每次实验完成后记录其剩余电量,并对10次实验结果取算术平均,从而保证实验结果的准确性.从图10(a)中可以看出,考虑用户偏好可以有效节约能耗,考虑合作可以进一步提高用户满意程度.

同样,为了考察用户偏好和移动客户端/服务器端合作在提高性能方面的影响,针对不同的 $w_p$ 值(0.1~0.9)进行类似的实验,从而得到图10(b)所展示的实验结果.可以看出,考虑用户偏好可以明显缩短响应时间,考虑合作可以进一步提高性能.

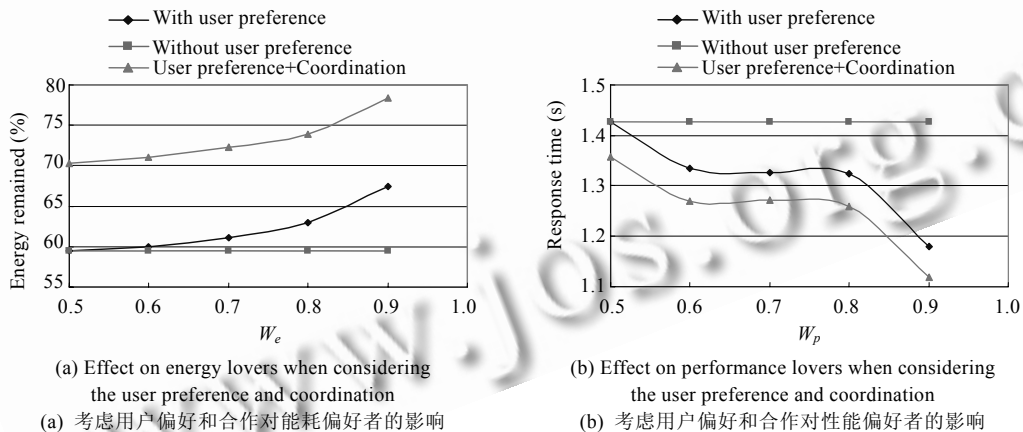


Fig.10 Experimental results when considering user preference and cooperation  
图10 考虑用户偏好和合作的实验结果

4.3 全局协作

本文利用 EasyEclipse Mobile Java<sup>[13]</sup>提供的模拟器来仿真移动客户端的并发访问.全局协作可以有效减少由多移动客户端的并发访问带来的数据访问延迟;同时,由于减少了那些可能导致系统整体性能下降的预取操

作,全局协作可以在保证系统性能的前提下,显著减少对电池电量的消耗。

图 11、图 12 分别显示了 MGODP 与 BOP<sup>[5]</sup>,VAP<sup>[9]</sup>,GreedyDual Least Utility<sup>[10]</sup>,DPT<sup>[11]</sup>和 Auto-Update<sup>[12]</sup>在平均响应时间和剩余电量上的实验结果对比。

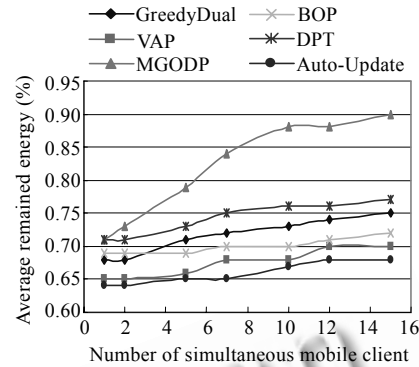
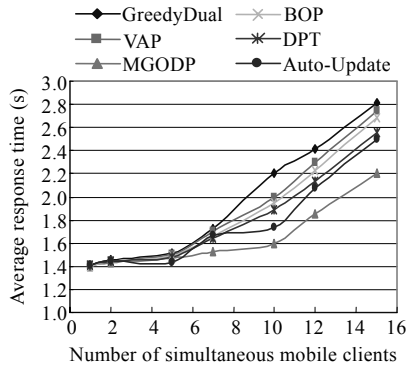


Fig.11 Global coordination effect on response time Fig.12 Global coordination effect on energy remained

图 11 全局协作对响应时间的影响

图 12 全局协作对剩余电池电量的影响

为了避免实验结果的偶然性,我们设定不同的并发移动客户端数目(分别为 1,2,5,7,10,12,15),进行了 7 组实验,在每组实验中对每个  $w_p$  取值(0.1~0.9)进行 3 次实验,每次实验包含 100 个数据请求,每次实验后计算其响应时间和剩余电量,然后对 3 次实验结果取算术平均,完成针对所有  $w_p$  的实验后,对每个  $w_p$  的响应时间和剩余电量取算术平均,从而得到此并发移动客户端数目下的平均响应时间和平均剩余电量。

从图 11 可以看出,MGODP 方法对并发数据访问带来的冲突进行了较好的解决,使得系统平均响应时间明显缩短,从整体上提高系统运行效率。从图 12 可以看出,MGODP 方法可以显著节约能耗。随着并发客户端的增多, MGODP 会限制一部分可能降低系统性能的预取操作,因此可以在保证系统整体性能的前提下,为移动客户端更好地节约能耗。

由对比实验可以看出,与不考虑用户偏好的数据预取机制相比,MGODP 可以节约大约 18%的电量(如图 10(a)所示),缩短大约 19%的响应时间(如图 10(b)所示)。与 VAP,Greedy Least Utility 等没有考虑全局协作的数据预取机制相比,MGODP 可以缩短大约 25%的响应时间(如图 11 所示),并节约能耗 22%左右(如图 12 所示)。

## 5 总 结

数据预取服务对于移动环境而言,可以有效减少由网络、电源等易变因素带来的数据访问延迟过大、无连接状态服务中断等问题。本文试图从用户偏好、移动客户端与服务器端的合作以及全局协作 3 个方面提高移动用户对数据预取服务的满意程度。由第 4 节的实验结果可以看出,MGODP 能够更好地满足用户的不同需求,并从全局角度更合理、有效地利用了服务器资源。

在未来的工作中,希望将数据同步时机、缓存内容替换等机制加入到 MGODP 中,多侧面地提高网络化移动应用的数据服务质量。数据同步的时机择取对于整个系统性能有重要影响,因为过于频繁的同步会消耗移动客户端资源并占用系统网络带宽;但是过少的同步又会使数据有效性难以得到保证,因此希望在未来找到合适的方法来解决数据同步时机的选择问题。而缓存内容的替换方法对于内存有限的移动设备而言,会影响整个系统数据服务的质量。这些后续工作可以保证数据预取所获得的本地缓存数据更好地为用户提供服务。

## References:

- [1] Forman GH, Zahorjan J. The challenges of mobile computing. IEEE Computer, 1994,27(4):38-47.
- [2] Cao GH. Proactive power-aware cache management for mobile computing systems. IEEE Trans. on Computers, 2002,51(6): 608-621. [doi: 10.1109/TC.2002.1009147]

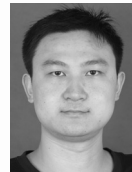
- [3] Jing J, Elmagarmid A, Helal A, Alonso R. Bit-Sequences: An adaptive cache invalidation method in mobile client/server environments. *Mobile Networks and Applications*, 1997,2(2):115-127. [doi: 10.1023/A:1013616213333]
- [4] Yuen JCH, Chan E, Lam KY, Leung HW. Cache invalidation scheme for mobile computing systems with real-time data. *ACM SIGMOD Record*, 2000,29(4):34-39. [doi: 10.1145/369275.369282]
- [5] Chen CC, Lee C, Wang CC, Chung YC. Prefetching LDD: A benefit-oriented approach. In: *Proc. of the 2006 Int'l Conf. on Wireless Communications and Mobile Computing*. New York: ACM Press, 2006. 1103-1108.
- [6] Huang SN, Xi JQ. Distributed data cache systems. *Journal of Software*, 2001,12(7):1094-1100 (in Chinese with English abstract). [http://www.jos.org.cn/ch/reader/view\\_abstract.aspx?file\\_no=20010718&flag=1](http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20010718&flag=1)
- [7] Tuah NJ, Kumar M, Venkatesh S. Resource-Aware speculative prefetching in wireless networks. *Wireless Networks*, 2003,9(1): 61-72. [doi: 10.1023/A:1020829124143]
- [8] Swaminathan N, Raghavan SV. Intelligent prefetch in WWW using client behavior characterization. In: Williams AD, ed. *Proc. of the 8th Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Washington: IEEE Computer Society Press, 2000. 13-19.
- [9] Yin LZ, Cao GH, Das C, Ashraf A. Power-Aware prefetch in mobile environments. In: Rodrigues LET, Raynal M, Chen WSE, eds. *Proc. of the 22nd Int'l Conf. on Distributed Computing Systems*. Washington: IEEE Computer Society Press, 2002. 571-578.
- [10] Shen HP, Kumar M, Das SK, Wang ZJ. Energy-Efficient data caching and prefetching for mobile devices based on utility. *Mobile Networks and Applications*, 2005,10(4):475-486. [doi: 10.1007/s11036-005-1559-8]
- [11] Wang HP, Zhang TJ, Yang XZ. A data access method in mobile computing environment: Using data prefetch technology in parallel control system architecture. In: *Proc. of the 2006 Int'l Conf. on Machine Learning and Cybernetics*. 2006. 4425-4429.
- [12] Vartiainen E, Roto V, Popescu A. Auto-Update: A concept for automatic downloading of Web content to a mobile device. In: *Proc. of the 4th Int'l Conf. on Mobile Technology, Applications*. Singapore: Research Publishing, 2007. 683-689.
- [13] EasyEclipse. <http://www.easyeclipse.org/site/home>

#### 附中文参考文献:

- [6] 黄世能, 奚建清. 分布数据缓存体系. *软件学报*, 2001,12(7):1094-1100. [http://www.jos.org.cn/ch/reader/view\\_abstract.aspx?file\\_no=20010718&flag=1](http://www.jos.org.cn/ch/reader/view_abstract.aspx?file_no=20010718&flag=1)



张晓薇(1983-),女,黑龙江黑河人,博士生,CCF 学生会员,主要研究领域为软件工程,嵌入式构件技术.



田刚(1980-),男,工程师,主要研究领域为嵌入式软件,软件工程.



曹东刚(1975-),男,博士,副教授,CCF 会员,主要研究领域为中间件技术,软件工程.



陈向群(1961-),女,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,操作系统,嵌入式软件.