

普适计算环境中基于移动 Agent 的数据访问原型^{*})

张向刚 张云勇 刘锦德

(电子科技大学计算机科学与工程学院微机所 成都610054)

摘要 普适计算^{**}(pervasive computing)不仅包含无处不在的计算设备和网络连接,更重要的是需要实现无处不在的数据访问。如何使用户及时、可靠、安全、方便地获取信息是这种计算环境的目的,同时也是该环境中数据管理的核心和目标。本文以移动 agent 为基础,实现了适合于普适计算环境的数据访问原型。重点阐述了实现原理、原型体系结构、工作流程,并通过实验数据以体现其优越性。

关键词 Pervasive computing, 数据管理, 中间件, 移动 agent, 推

A Mobile Agent-Based Data Access Prototype in Pervasive Computing

ZHANG Xiang-Gang ZHANG Yun-Yong LIU Jin-De

(Micro Computer & Network Technology Research Institute, College of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, 610054)

Abstract The pervasive computing means not only ubiquitous computers and connectivity, but also ubiquitous data access. How to access data in pervasive computing freely, robustly and real-timely has become an exigent problem. In the paper, we implement a data access prototype based on mobile agent in pervasive computing. We expatiate the idea and principle of mobile agent-based data access, as well as the architecture and work flow of the prototype, then compare its performance with C/S mode to exhibit its advantage.

Keywords Pervasive computing, Data management, Middleware, Mobile agent, Push

1 基于移动 Agent 的数据访问概述

随着计算机和通信技术的发展,各种小型、智能和移动设备(如:移动电话和 PDA 等)正在成为计算的主流。计算组件通过各种方式嵌入到各种各样的设备中,并且通过各种形式存在于我们生活和工作的方方面面。这就是所谓的无处不在的计算环境^[1,2](pervasive computing, 简称普适计算环境)。在普适计算环境中,各种设备通过网络连接到一起,能够让用户更简单、更普遍、更可靠、更安全地访问和使用各种资源和信息。

我们可以想象这样的环境给我们的工作和生活带来的便利,在医院里,装配了各种计算设备(包括医院的网络、中心数据库、安装在病房的监控设备和穿戴在病人身上的传感器)。不论医生在医院或在家里或在汽车上,他都能够从自己随身的 PDA 或其他设备上了解到病人当时的状态,如体温、心跳和血压等(通过病人身上的传感设备),同时从记录数据库中了解到这些信息的历史日志,还能够从医院的中心数据库中了解到该病人的病历,以及各种药品指标。医生能够利用这些信息及时全面地了解病人情况和药品的最新信息,从而能够更准确地对病人做出判断和治疗。对于商务活动者,这样的环境更加使他们如虎添翼,当他们到达一个城市后,能够随时从自己随身的移动设备上了解到公司产品的最新消息和性能指标,以及这个城市潜在的客户和他们的情况,同时还能得到旅店和餐馆的方位及交通情况等。

从上面的情景中,我们可以看到在信息瞬息万变的时代,

如何及时、方便地获取最新信息,已经成为这种新型计算环境中数据管理^[3]的一个核心的问题。良好的信息获取技术不仅能够随时提供给用户最新信息,而且能够减少网络和服务器负载,提高用户的请求响应速度,同时减少用户的干预,实现自动化。

传统数据访问的通信模式是 C/S 或 B/S 模式。由于普适计算环境新的特点,这些模式已不再适合于它。在这样的环境中不仅存在着大型设备,而且存在大量的小设备,这些小设备只拥有有限的计算能力和电源,不适合建立持久稳定的连接;其次,网络连接中存在大量误码率高、时断时续的无线连接;再有设备不仅多,而且移动性强。具体而言,编在计算环境中采用传统的 C/S 模式主要存在以下问题:

(1) 传统 C/S 模式是一种客户端主动的 pull 模式,它不能及时反映数据源的最新信息,尤其不适合实时信息采集,如果通过频繁地访问数据源,则需要持久的网络连接和相当的带宽,以及人为干预。

(2) 普适计算环境中,数据源众多。用户并不需要所有信息,在传统的 C/S 模式中需要人为地进行数据筛选,因此一个有用信息的获得,往往需要客户端与数据源多次的反复才能完成。

(3) 在普适计算环境中,异构数据源众多。很多数据源不具有数据需求端的语义和工作逻辑,如何在数据源执行用户的工作逻辑,通过传统的 C/S 模式无法解决。

新兴的移动 Agent 技术以其特有的移动性和智能性集于一身的特点^[3],为解决普适计算环境中的数据访问提供了

^{*})本课题得到总装备部军事预研项目《新型系统体系结构技术》(41315010103)资助。张向刚 博士生,主要研究领域为:中间件、普适计算和数据管理。张云勇 博士后,主要研究领域为:中间件、agent 技术和软交换。刘锦德 教授,博士生导师,主要研究领域为:开放系统与中间件技术。

^{**})普适计算,英语为 pervasive computing 或 ubiquitous computing,意思为无处不在的计算,作者以前文献译为“遍在计算”,现根据国家专家意见统称为“普适计算”。

新思路。基于移动 Agent 的数据访问技术的思想在于通过 Agent 的智能性表达用户的需求和客户端的工作逻辑,同时通过将 Agent 迁移到不同的数据源,与之进行本地高速通信,自动进行数据的收集和筛选,避免其间的交互过程在网络中传输,最后将结果数据推给用户。这样既能够有效减少对终端小设备和网络资源的需求,同时也能完成数据的自动收集,以及实时访问数据和跟踪数据源移动,达到信息交换的各方协调一致地进行实时、可靠、安全、自适应地交换信息。

本文第2节首先介绍了基于移动 Agent 的数据访问原型的体系结构,以及各组成部分的功能;第3节说明原型的工作流程;第4节与传统 C/S 模式进行性能比较;最后进行全文的总结。

2 体系结构

根据普适计算的特点,我们运用移动 Agent 技术实现了一个数据访问原型,其体系结构如图1所示,整个系统由终端设备、中心服务器和数据源系统三大部分构成,各部分功能如下:

(1) 终端设备包括移动设备和固定设备,它们代表信息的消费者。用户可以通过终端设备设置感兴趣的信息,以及接受中心服务器推回的信息。

(2) 中心服务器是整个系统的核心,它管理数据源,接受

终端设备的接入,分派和管理 Agent,提供安全服务。它包括:

·管理子系统:向上管理终端设备的接入,向下管理与各种数据源的连接,以及接受终端用户的信息定制文件等。

·移动 Agent 中间件:中心服务器的移动 Agent 中间件,除了提供 Agent 工作的上下文外,还承担了 Agent 的产生、分派和销毁任务,以及将用户定制文件转换为 Agent 内部语言。系统中一个 Agent 实体实际上是一个可移动的 Java 对象。在到达一个网络主机后,它自主地启动线程执行任务,并且能够对接受到的消息作出快速反应。

·安全子系统:提供终端接入的安全认证,网络传输的数据安全和 Agent 的安全管理。

(3) 数据源系统是整个普适计算环境的数据提供者,数据的来源可能是数据库系统,也可能是文件系统,或者直接是传感器接受的数据。它包括:

·移动 Agent 中间件:提供 Agent 运行的上下文;

·本地系统:这是一个设备之间互不相同的部分,提供本地数据源的管理。

·安全系统:本地的安全系统由两个部分组成,一是移动 Agent 中间件中提供的 Agent 安全机制,另一个是本地系统提供的安全机制,包括本地操作系统的安全机制,数据库的安全机制等。

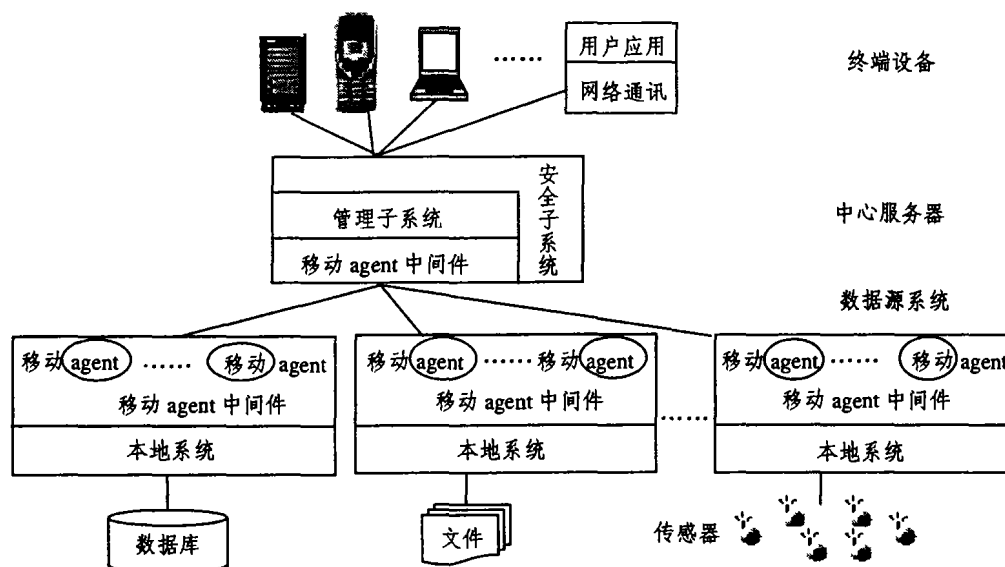


图1 基于移动 Agent 的数据访问原型体系结构

3 基于 push 的数据访问流程

当用户需要新的信息数据的时候,通过终端设备(如普通的 PC 客户终端、PDA 和手机)向中心服务器送出有关的信息定制文件(信息定制文件表达用户感兴趣的信息,可以简单地使用关键词流表达,也可以使用 profile 文件^[4]表达复杂的信息定制),中心服务器接收到信息定制文件后,将其转换为 Agent 内部语言,通过移动 Agent 中间件系统向已注册的数据源派送数据检索 Agent,数据检索 Agent 在数据源本地按照信息定制文件检索数据,同时将满足条件的数据推回中心服务器,中心服务器再将数据推回用户终端。以后信息检索 Agent 将一直驻留于数据源,一旦有满足条件的信息,将立即推回中心服务器,期间中心服务器可以根据新的定制文件,更改检索 Agent 的检索条件,直至撤销。

在具体实现时,存在两种数据推回路径。其一是把数据从

数据源推回中心服务器,之后推回用户终端;另一种是直接数据推回用户终端。在本系统实现中采用了前者,主要原因存在以下三点:一是直接将数据推回用户终端,需要在终端上驻留移动 Agent 中间件系统,这对终端的资源 and 计算能力提出了较高的要求,不具有广泛的适应性;二是当用户进入一个环境的时候,往往不知道有哪些数据源,以及它们的位置。这时往往需要中心服务器提供统一的服务。其三,采用推回路径通过中心服务器的方式,系统的网络通信部分比较规范、易实现。

在系统的具体实现中,为了满足平台无关性、安全性、面向对象特性、多线程特性,对于中心服务器和数据源服务器部分,我们采用了纯 Java 技术:采用 Java 的 JDBC 数据库技术完成对数据库操作;采用了基于 Java 的移动 Agent 技术来实现有关 Agent 的移动性和安全性要求;Agent 之间的通信,采用了与 KQML (Knowledge Query and Manipulation Lan-

guage)^[6]语言兼容的语意表达形式,利用移动 Agent 系统的消息机制加以实现。终端系统可以用各种语言,在各种平台上进行实现。

4 性能比较

4.1 理论分析

移动 Agent 方式和 Client/Server(C/S)方式的区别之一在于:在 C/S 方式中,信息的请求方和提供方的处理结果都要在网上传递,每次应答的信息都要在网上传交互。而采用移动 Agent 方式时,信息请求方将 Agent 移动到信息提供方的 Agent 上下文环境中,请求方 Agent 和服务方的交互都在此上下文环境中进行。同时,中间结果和冗余信息不必在网络上传输,这样一方面可以减少网络开销,另一方面又可以克服网络传输延时的影响。最后,请求方 Agent 把处理结果传回给客户方,它本身在服务方上下文环境中消亡,不必移回请求方。以下,我们对传统 C/S 分布式计算模式和我们所采用的移动 Agent 计算模式的性能进行比较。我们假设请求方的某个信息请求,需要在请求方和提供方之间进行 N 次交互才能完成,并假设 T_{req} 为一次交互请求在网上传送的平均时间; T_{queue} 为一次交互请求在服务方队列中的平均等待时间; T_{proc} 为一次交互请求的平均处理时间; T_{rtpn} 为一次交互结果在网上的平均传送时间,于是得到: C/S 方式完成服务所需要的时间为:

$$T_{C/S} = N * (T_{req} + T_{queue} + T_{proc} + T_{rtpn}) \quad (1)$$

移动 Agent 方式完成服务所需要的时间为:

$$T_{agent} = N * (T_{queue} + T_{proc}) + T_{req} + T_{rtpn} \quad (2)$$

由以上分析可以推导出二者的时间差为:

$$T_{C/S} - T_{agent} = (N-1) * (T_{req} + T_{rtpn}) \quad (3)$$

由上面(3)式可得:基于 Agent 的数据访问相对于 C/S 模式节约了多次交互的时间。

4.2 实验比较

在对系统进行性能试验测试时,我们采用了简化体系结构的策略,目的在于消除其他环境因素对性能的影响,着重突出移动 Agent 相对于传统的 C/S 模式的优越性。测试模型体系结构约定如下:

- (1)将数据源结构,统一为桌面 PC+SQL server;
- (2)中心服务器和数据源之间的网络分为三种,10M 的以太网,GPRS 移动分组接入(最大理论带宽为13.4kbps)和 GSM 接入(最大理论带宽为9.6kbps);
- (3)从每个数据源读取40条记录,每条记录40字节。

在测试系统时,我们通过同时使用2个、3个、4个、5个和6个数据源子系统,分别测试完成数据获取操作的时间情况。表1给出了所得的实验结果。

表1 基于 Agent 和基于 C/S 的数据访问对比测试结果

数据源数量	基于 Agent			基于 C/S		
	10M(秒)	13.4K(秒)	9.6K(秒)	10M(秒)	13.4K(秒)	9.6K(秒)
2	1.60	28.46	47.73	4.67	68.46	168.90
3	1.73	29.79	57.75	6.07	102.76	253.35
4	2.04	38.73	86.05	7.04	143.48	405.36
5	2.35	52.66	120.43	8.80	200.14	620.01
6	2.63	54.25	177.00	10.92	296.01	998.23

图2是表1的折线示意图,从图中可以看到:基于移动 Agent 中间件的数据访问方式比普通 C/S 结构模式具有更高

的性能,而且随着网络带宽的恶化,系统基本性能无明显变化,特别是在无线低带宽情况下,系统仍稳定可靠。而普通 C/S 结构模式性能则有大幅下降。同时也看到,随着数据源的动态增加,系统并发机制发挥了良好的性能,具有良好的可动态扩展能力;而普通 C/S 结构模式随着子系统的增加,其计算时间会成指数级增长。

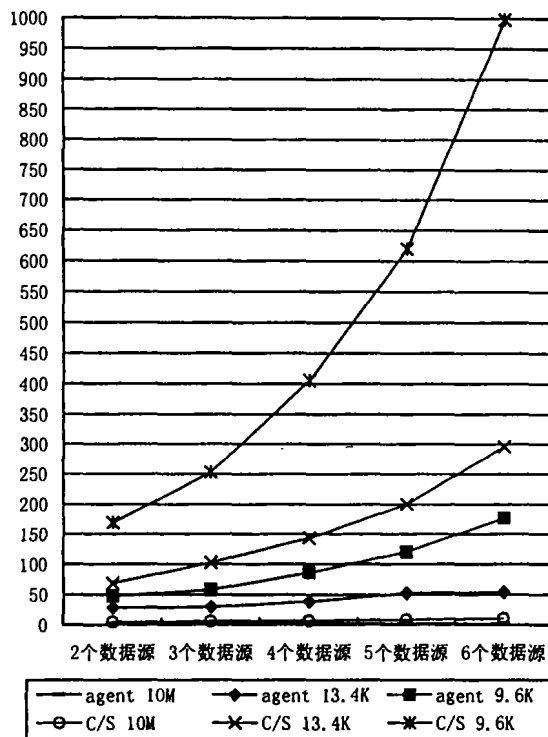


图2 测试结果折线图

小结 普适计算包含无处不在的计算设备、无处不在的网络连接和无处不在的数据访问三层含义。而最后一点是这种新兴环境的本质和目的所在。如何及时、可靠、安全、方便地获取数据已经成为普适计算环境中数据管理的核心问题之一。由于这种新的计算环境有别于传统的分布式环境,打破了传统计算技术对计算环境的假定,使得传统的技术和算法在性能和可用性等方面不能满足这种新型计算环境的要求。本文希望通过移动 Agent 技术解决这种环境中的数据访问问题,并通过构造原型进行验证。实验表明:通过移动 Agent 的智能性和移动性,能够表达用户的各种需求,减少对终端资源和网络资源的要求,提高数据访问的速度,满足应用对系统可靠性和安全性的要求。

参考文献

- 1 Weiser M. The computer for the twenty-first century. Scientific American, 1991, 265(3): 94~104
- 2 Grimm R, et al. Systems Directions for Pervasive Computing. In: Proc. of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Elmau, Germany, May 2001. 128~132
- 3 Cockayne W R, Zyda M [EB/OL]. Mobile Agents. <http://flower.ce.cnu.ac.kr/~dkkang/mobile-Agent-books/HTML/Front.htm>, 1998
- 4 Cherniack M, Franklin M J, Zdonik S. Expressing user profiles for data recharging. IEEE Personal Communications: Special Issue on Pervasive Computing, 2001, 8(4): 32~38
- 5 Cherniack M, Franklin M J, Zdonik S B. Tutorial: Data Management for pervasive computing. VLDB 2001
- 6 Vieira S. FIPA-ACL [EB/OL]. <http://www.cs.helsinki.fi/u/kraatika/Courses/Agents/fipa-acl.html>, 1998