

移动 Agent 技术研究^{*}

The Study of Mobile Agent Technology

董红斌

石纯一

(哈尔滨师范大学计算机系 哈尔滨150080) (清华大学计算机系 北京100084)

Abstract Mobile Agent is a new distributed computing model. This article introduces the concepts, origin, behavior, development language, general structure of mobile agent, and attempts to present an account of current research efforts.

Keywords Internet, Mobile agent, Mobility, Autonomous

1 引言

Internet 和 WWW 已成为世界范围内的电子商务工具,Internet 用户的不断增长将在远程学习、电子商务和多媒体通讯等领域带来无限商机。同时,网络低带宽、高延迟越来越成为限制网络应用发展和推广的一个瓶颈。人们希望通过软件来解决这个矛盾,移动 Agent 就是在这种形式下提出的一种新的分布式计算模型^[1]。移动 Agent 包含两个不同的含义:移动性和代理。移动 Agent 是一段可执行的程序代码,它可以从一台机器通过网络移动到另一台机器运行,并根据需要可生成 Agent,子 Agent 具有同父 Agent 相同的性质,对于移动 Agent 没有一种公认的定义,有些学者将移动 Agent 归入 Agent 的特殊情况^[2];其他人则认为 Agent 是移动代理,尽管概念上不同,但大多数研究文献表明移动 Agent 有两个基本目标:减少网络传输和异步交互。移动 Agent 程序将网络看作多 Agent 友好环境,从一台机器移动到另一台机器为用户执行任务,Agent 间彼此独立工作或相互合作进行任务求解。移动 Agent 作为一种新技术在各种研究中是争论的焦点。目前,Agent 研究朝着移动 Agent 系统结构的设计和实现的最优化方向进行^[3],这包括这样一些内容:Agent 移动机制;命名、地址、一个移动 Agent 的位置;移动 Agent 的控制;移动 Agent 状态输出;移动 Agent 数据传输;通讯;安全;秘密和隐私 procedure;协作;稳定性,实现;可扩展性;资源管理和发现;权威;义务;道德规范。

这些问题的一部分属于过程移动和分布式操作系

统的领域,其它来自于人工智能的领域。前12个问题是技术上的内涵,权利、义务和道德规范是社会问题。本文将对移动 Agent 的概念、起源、行为、开发语言和研究状况等主要内容进行阐述,并给出了一种移动 Agent 的通用结构。

2 移动 Agent 起源

移动 Agent 起源于三个早期技术:过程移动、远程计算和移动对象,其目标是提高分布式程序设计的远过程调用(RPC)^[2]的能力。

过程移动:早期系统支持过程移动允许全部地址空间从一台计算机移动到另一台计算机,其目标之一是当多个远程调用执行一个应用程序时减少网络带宽。当过程移动允许全部过程被传输到远端主机,在全部过程未返回的情况下这种机制不允许数据返回源点(见图1)。

远程计算:远程计算程序允许一个计算机以程序的形式向另一个计算机发送请求,远程计算机接受这样一个请求,参照其局部地址空间的请求执行程序并返回结果给发送计算机。远程计算系统在没有从源主机到目标主机传输过程控制数据的情况下通过远程编程提高过程移动效率。

移动对象:尽管减少了网络带宽需求,但远程计算系统缺少在远程主机封装更多状态信息到可执行程序的能力,移动对象基于面向对象编程技术,在移动对象内通过获取更多程序行为扩大远程计算能力,当传送执行代码,特别是对象特性数据,和潜在其它嵌入执行对象时,这种对象可以从点到点移动。在八十年代,大

^{*} 本课题得到国家自然科学基金资助,董红斌 副教授,清华大学访问学者,研究方向为分布式人工智能。 石纯一 教授,博士生导师,研究方向为人工智能应用基础。

量的移动对象系统流行,并直接导致了移动 Agent 作为嵌入系统的开发。

移动 Agent 具有自主性,自己能动态决定在哪里、在什么时候基于嵌入元数据去传送到一些特殊的目的地。移动 Agent 为分布式程序的设计提供执行代码、程序状态信息和其它被传输数据的方法以提高这些早期技术。

移动 Agent 容易修改其日常行为以适应程序状态和网络环境的变化,移动 Agent 给用户提供一个自然异步的交互模式。在早期的分布式程序设计中,移动 Agent 的其它特性如容错和安全方面是非常有用的。

3 移动 Agent 的行为

Agent 和多 Agent 都工作在软件系统中,在 DAI 中形成了一个主要的研究领域。研究的热点一方面集中于所谓的 Agent 社会,如何能更好地联盟、协调和使用,另外的热点是 Agent 的特性、描述一个 Agent 的行为、研究 Agent 的结构和用于描述它们的语言^[4]。后一方面发展较快。

通常情况下,Agent 是执行一个或多个共同任务,按预定方式工作的软件程序。Agent 的设计者通常决定 Agent 可以作什么,但它的行为也可以由其它的软件程序——一种超级 Agent 来建立,当需要更多帮助时它可以产生新的 Agent。从面向对象的观点看 Agent 的定义可以描述为对象类。本质上 Agent 有五种性质:移动性、反应性、自主性、学习能力、和合作,由此五种特性能导致其它多种特性。

·移动性:这是一个移动 Agent 的基本特性,是指 Agent 能或不能在不同的环境中移动。Agent 可表示为“静态”和“动态”。“静态”指 Agent 限定在单一状态,不能在网络中移动,早期的 Agent 都是静态的,工作在一个稳定的环境中。Agent 的动态是指它能从一台主机移动到另一台主机,工作在不同的主机环境中。

移动 Agent 可以接受的描述是一个软件过程,它可以把其状态从一个环境移到另一个环境,在新环境中仍能正常工作。早期移动 Agent 的活动是一个移动程序,在自主控制下从一台机器移到另一台机器。1995 年在 Dartmouth 学院 Agent TCL 的开发提出了第一个移动 Agent 模型^[5],Agent 可在任意点暂停执行,将它的代码和状态传送到另一台机器,在新机器中重新执行。

·自主性:如果移动 Agent 是自主的,则它可以控制自身的行为。移动 Agent 具有独立的局部于自身的知识与知识处理方法,能够解决给定的问题,以特定的方式适应周围的环境。它能在自主的情况下按照用户的需求去工作,能对自己的行为作出规划。例如它可以

通过访问服务数据库搜索信息,对用户需求作出反应。

·学习能力:学习指的是 Agent 具有利用获得的关于环境的信息,调整修改自己行为的能力。

·反应性:指的是 Agent 对需求做出的反应,Agent 分为主动的和反应的,主动的 Agent 有一个在任何环境下都可存在模型,它了解打印、通讯、显示能力等资源在哪里找到,这种能力使它推断出能胜任的工作。因此主动 Agent 可以规划并与其它 Agent 协商完成任务。

反应 Agent 没有所处环境的模型,只能以预测的方式对环境做出反应,主动 Agent 比反应 Agent 更具适应性。

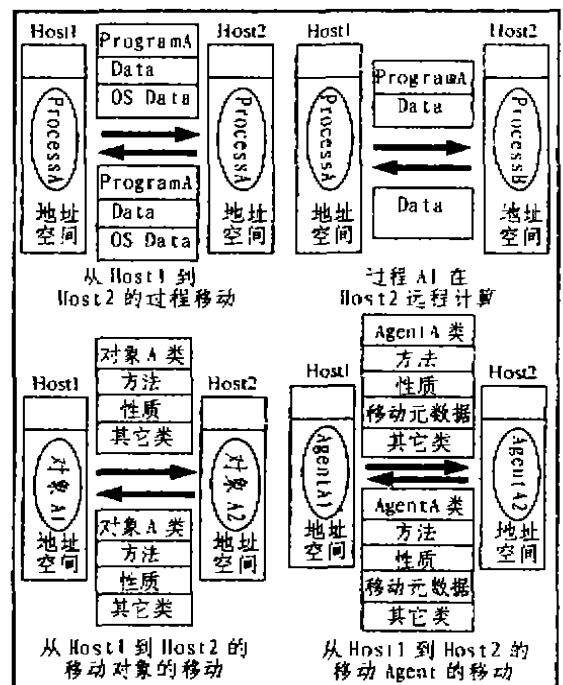


图1 分布计算过程的进化

·合作:是 Agent 具有合作求解和管理通讯的能力。例如:给定合作 Agent 一项查找某人重要日期的任务,一个 Agent 找到一个生日信息,立即将查询结果通知其它 Agent,其它 Agent 将不再查找生日,而查找其它日期,如结婚日期、工作日期等,避免其它 Agent 重复查找,减少了通讯调用。

自主性、学习能力和合作是 Agent 的重要特性。如果 Agent 具备三项能力之一,Agent 就具备了它的基本能力;如果 Agent 具有自主性和学习能力或具有自主性和合作能力,则 Agent 就具有高级能力;如同时具有上述三项能力,则 Agent 是理想的 Agent。目前

移动 Agent 研究的主要领域之一就是它的行为,大多数 Agent 只具有其中两种能力,Agent 同时具有三种能力是困难的。

4 移动 Agent 的开发语言

Java 是移动 Agent 的首选语言。Java 的跨平台特性和书写一次随处可用的特点使 Java 极好地适用于移动 Agent 技术。更进一步,通过 Internet,Java 的虚拟机简化移动 Agent 的传播。

Java 有几个其它语言所不具备的直接支持移动 Agent 实现的特性。例如,Agent 移动性要求将一个 Agent 及其状态方便地转换为适于网络传输的形式,并在接收端,允许远程系统重构 Agent。

一些基于 Java 的移动 Agent 系统提供持续的 Agent 状态信息。持续(persistent)是通过一个连续的 Agent 状态写入持续存储器实现的,然后检索这个状态并利用它重构 Agent。当一个移动 Agent 处于连续状态后,它可以传送到其它主机并在到达的地方重构。Java 的网络支持包括 socket、URL 通讯和称为远程方法调用(RMI)的分布对象协议。在 Java 中,对分布对象的语义访问是容易实现的;RMI 是一个通过局部代理或与实际远程对象交互的端口的操作,是 Java 体系中新的一门技术。Java 是一种面向对象的语言,每一个程序实体都是以对象的形式提交的。为使分布式环境下的 Java 编程保持这一特性,Sun 公司提出了分布式对象的概念,并为分布式对象之间互相引用提供了适合对象语义的基于抽象的对象层次的机制,这就是 RMI。它使得分布式环境下的 Java 编程或者说 Java 程序之间的网络通讯,不再涉及底层协议和数据的编解码,从而更加简单自然。

此外,Java 的代码和状态移动特性是通过其类调用机制实现的。Java 类调用者动态调用包含在局部 Java 类路径(目录列表)中的应用类。为动态调用移动 Agent 代码和类,一个具体的类调用提供这样几种操作:一个 Agent 的序列化形式包含任何访问类;一个 Agent 类可以被 Web 服务器调用或被其它服务器调用;一个 Agent 类可以被类路径调用。

所有被 Java 类调用的代码受到安全的限制,这对移动 Agent 系统是非常有用的,它能保护 Agent 免受未经认证的访问。Java 的安全管理员支持细粒度的开发、高可配置安全策略。Java 也支持移动 Agent 与 Web 的集成开发。Applet 能从 Web 浏览器创建移动 Agent 并接受完成远程执行后创建的 Agent。Java 也提供类似 Applet 服务器技术。所谓的 servlet 函数有点象公共网关接口脚本,它能创建和接受移动 Agent。Java 的命名和目录接口(JNDI)允许通过统一访问命

名目录服务器无缝连接商业信息。例如,移动 Agent 能利用 JNDI 服务提供它们需要的局部服务,然后连接遗留系统。移动 Agent 软件以它突出的特点表现了对 Internet 环境的高度适应性,移动 Agent 可以从一个网络结点转移到另一个结点,逐渐地完成它的任务。Agent 首先暂停执行,将自己的执行代码、数据和运行状态传送到网络的其它结点,然后从暂停点继续执行,在它转移过程中,可以启动新的 Agent 来传送需要的信息给客户或者某个子任务。这比 Client/Server 模式有显著的优点,在计算机之间传送的不是请求或者调用,而是过程本身和参数。利用移动 Agent 可将 Internet 的网络上的服务器变成一个公共的软件平台,集成各种软件产品。移动 Agent 的关键是要具备允许进程在其它计算机上执行的软件结构。

5 移动 Agent 的通用结构

普通基于 Java 的移动 Agent 结构由八个主要部分组成:一个 Agent 管理员;一个中间 Agent 通讯管理员;一个安全管理员;一个可靠性管理员;一个自适应单元;一个合作模块;一个应用网关和一个目录管理员。它们分别支持健壮性、可靠性、自适应性、合作、安全性和实际 Agent 应用的开发(见图2)。

• Agent 管理员发送 Agent 到远程主机,并接收在局部主机执行任务的 Agent。Agent 管理员连续传输 Agent 和它的状态信息,通过连续传输在目标主机形成它的副本。在一个高可靠性结构中,通过 Agent 来关闭可靠性管理员,这保证 Agent 被远程主机的 Agent 管理员接受。一收到 Agent,Agent 管理员就重构 Agent 并以它作为参考对象,产生它的执行文本。在 Agent 执行前安全管理员认证 Agent,此后,移动 Agent 系统(Java 虚拟机制)自动启动安全管理员,利用系统资源认证操作。当这个 Agent 准备移动到其它主机时,它要求 Agent 管理员将它传送到正确位置。

• 安全管理员保护主机和移动 Agent 防止非认证的访问。所有其它移动 Agent 系统部件与它交互认证并认证移动 Agent。安全管理员通过传输前和保存到持续存储器前编码的方法保护 Agent。在高安全系统,安全管理员能数字化标记 Agent,移动 Agent 系统彼此间通过交换许可进行认证。安全管理员允许经过认证的 Agent 通过防火墙。

• 可靠性管理员保证移动 Agent 系统的坚固性。在高可靠性系统,它保护 Agent 免受服务器和系统故障的影响。它的主要任务之一是保证 Agent 和移动 Agent 系统相关的状态的持续性。此外,可靠性管理员甚至在系统出现故障时能保证 Agent 实现其目标。

• 自适应单元保证 Agent 不断适应网络环境条件

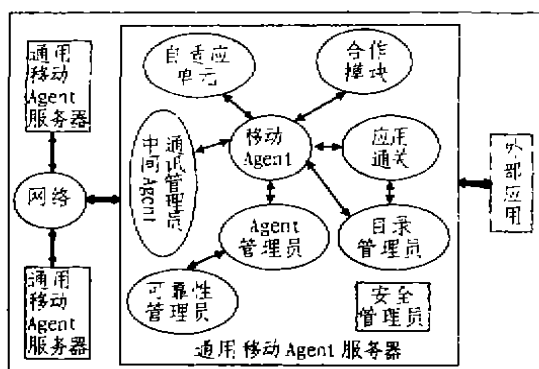


图2 通用移动 Agent 系统结构

的变化,满足用户对服务质量的需求。

·合作模块实现 Agent 与其它移动 Agent 之间的通讯合作,当移动 Agent 接受的任务超出它的求解能力时,移动 Agent 通过合作模块完成多个移动 Agent 间的任务求解。

·中间 Agent 通讯管理员在 Java 和其它系统中促进分散在网络各处的移动 Agent 的通讯,最简单的应用是实现多 Agent 计算和多相关 Agent 集成通讯,移动 Agent 系统典型工作是提供信息和分布事件。

·应用网关服务作为 Agent 与应用服务器(如遗留数据库)进行交互的安全入口。Agent 可利用基于 JNDI 的目录管理员验证应用服务器的位置并移动到服务器所在的主机,通过网关访问常驻服务器,安全管理利用网关认证 Agent 和应用服务器。这个通用结构对于多数应用领域是足够的,相信扩展和提高分布计算技术将使移动 Agent 更有效。当前 Agent 框架支持具有移动性、自主性、传递性和适应性的分布系统的实现。

结束语 目前,国际上一些高等院校和研究机构对移动 Agent 技术进行大量的研究,并开发了应用领域的移动 Agent 系统,如 Dartmouth 学院开发的

Agent TCL 移动 Agent 系统^[6],IBM 公司用 Java 语言开发的 Aglet 移动 Agent 模型^[11],Stuttgart 大学基于 Java 开发的 Mole^[10],Ottawa 大学多媒体和移动 Agent 研究室的 SHIP-MAIL^[12]等。随着这些实用系统的研究与开发,移动 Agent 技术将日趋成熟,Internet 与移动 Agent 技术的结合将给远程学习、电子商务和多媒体通讯领域带来无限发展机遇。

参考文献

- 1 Pham V A, Karmouch A. Mobile Software Agents An Overview. IEEE Commun. Magazine, 1998(7): 26~37
- 2 Wong D, et al. Java-based Mobile Agents. CACM, 1999, 42(3): 92~102
- 3 Jamah N, et al. An Actor-Based Architecture for Customizing and Controlling Agent Ensembles. IEEE Intelligent Systems, April, 1999
- 4 Sandholm T. Agents in electronic commerce. Component technologies for automated negotiation and coalition formation. In: Proc of the Second Workshop on Cooperative Information Agents (Paris, France, July 4-7). Springer-Verlag, Berlin, 1998. 113~134
- 5 Shebory O, et al. Agent Cloning an Approach to Agent Mobility and Resource Allocation. IEEE Commun. Magazine, 1998(7): 58~66
- 6 Kotz D, et al. AGENT TCL: Targeting the Needs of Mobile Computers. IEEE Internet Comp., 1997, 4(1): 58~67
- 7 Rothermel K, et al., eds. Mobile Agents, Lecture Notes in Comp. Sci., Series, vol. 1219, Springer, 1997
- 8 Morreale P. Agents on the Move. IEEE Spectrum, April, 1998
- 9 Arisha K A, et al. Impact: A Platform for Collaborating Agents. IEEE Intelligent Systems, April, 1999
- 10 Available at: <http://www.informatik.uni-stuttgart.de/ipvr/vs/projecte/Ara/index.e.html>
- 11 Available at: <http://www.trl.ibm.com/aglets>
- 12 Available at: <http://www.deneb.genie.uottawa.ca>

(上接第94页)

DM3系统中我们实现了外联接,并利用文[1]中提出的算法思想,实现了外联接的优化,在联接顺序的选择上,采用了折衷的方法,进行局部优化,减小搜索空间。在这个优化系统中,待改进的地方在于因所用的算法要求联接的谓词拒绝 NULL,并不适用于所有的查询,因此需要更新算法来减少约束。

参考文献

- 1 Galindo-Learia, Arnon Rosenthal. Outerjoin Simplification and Reordering for Query Optimization[J]. ACM Transactions on Database Systems, 1997, 22(1)
- 2 Selinger P G. Access path selection in a relation database system[C]. In: Proc. of ACM SIGMOD 1979 Intel. Conf. on Management of Data
- 3 周龙骧. 数据库管理系统实现技术[M]. 中国地质大学出版社