

Mogent 系统的通信机制

陶先平 冯新宇 李新 张冠群 吕建

(南京大学计算机软件新技术国家重点实验室 南京 210093)

(南京大学计算机软件研究所 南京 210093)

E-mail: txp@softlab.nju.edu.cn

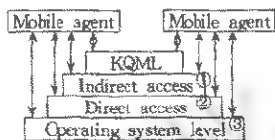
摘要 移动 agent 技术是新型软件构件框架的基础技术之一,而通信机制是其重要的组成部分.该文结合移动 agent 系统 Mogent (mobile agent) 平台的研制,在分析了 agent 的移动特性对通信机制的影响后,提出并实现了一种用于 Mogent 平台的一套系统的解决方案.该方案主要包括移动 agent 命名机制、移动 agent 寻址机制以及通信失败解决方法等,具有易于使用、效率高、可靠性好等优点.

关键词 移动 agent, Mogent 系统, 通信机制, 通信失败.

中图法分类号 TP311

一般来讲^[1],移动 agent 是一个独立运行的计算机程序,它可以代表用户完成特定的任务,具有自主性、移动性、安全性、协作性和智能性等特性.基于移动 agent 的分布计算模式具有一系列的优点,应用前景十分广阔,因此,越来越多的学者认为,移动 agent 技术将提供一种统一的、灵活的分布计算模式,人们可以使用该模式搭建各种分布计算平台和应用^[2-4].目前,我们正在进行的新型主动构件框架和远程高性能计算协同环境研究进一步表明了移动 agent 技术在软件构件技术和高性能计算环境方面应用的潜力和前景^[4-7].然而,通信机制是移动 agent 的关键技术之一,因此,对它的研究具有十分重要的意义.

Agent 的协同体现为若干个移动 agent 可在网络中相互协作并合作完成某一项任务,主要包括功能互通、协作联盟/模式和通信机制这 3 个层次.由于功能互通涉及语义层次,目前只能在理论研究方面进行工作.协作联盟/模式对于多 agent 系统的构造具有重要意义,面向应用领域的协同模式的研究可能是一条有效的途径.由于上述原因,目前在移动 agent 系统中,对软件 agent 的协同性的支持主要集中于对通信机制的研究.代表性的研究工作有两类,一类是基于知识交换的 KQML (knowledge query and manipulation language)^[8]等工作,另一类是基于消息传递的 aglets^[9]系统等.然而,我们发现上述两类工作实际上是分离的,尚未见到同时支持 KQML 和



①间接访问,②直接访问,③OS通信层.

Fig.1 Hierarchical multi-mode communication framework

图1 分层多模式通信框架

消息传递两种协作方式的通信机制.而在实际的应用中,信息的交流是多级别、多层次的,因此,我们的思路是,一个系统的移动 agent 通信机制应该支持 agent 之间开展不同层次的协作通信,其体系结构应该呈现如图 1 所示的分层多模式结构^[10].其中,KQML 支持移动 agent 间进行高层的基于知识交换的 KQML 的协作,间接访问模式以消息传递为通信手段,直接访

• 本文研究得到国家自然科学基金(No. 69873021)、国家“九五”重点科技攻关项目基金(No. 96-729-1-08)、国家 863 高科技项目基金(No. 863-306-ZT02-01-4)和国家杰出青年基金(No. 61525204)资助.作者陶先平,1970 年生,博士生,讲师,主要研究领域为对象技术,移动 agent 技术.冯新宇,1978 年生,硕士生,主要研究领域为分布计算,移动 agent 技术.李新,1976 年生,硕士,主要研究领域为对象技术,移动 agent 技术.张冠群,1977 年生,硕士生,主要研究领域为对象技术,移动 agent 技术.吕建,1960 年生,博士,教授,博士生导师,主要研究领域为形式化方法,对象技术,分布计算技术,构件技术.

本文通讯联系人:陶先平,南京 210093,南京大学计算机软件研究所

本文 2000-01-17 收到原稿,2000-06-19 收到修改稿

问模式允许 agent 在直接进行基于网络协议的数据传输. 移动 agent 可以同时任意使用其中的一种模式进行通信协作.

从技术支撑的角度出发, 我们采用自底向上的方法构建整个通信机制. 每一底层都向紧邻上层提供实现支撑. 我们目前的工作主要集中在间接访问层的建设, 拟在 PVM 的基础上, 遵循 MPI(message passing interface) 标准, 设计 Mogent(mobile agent) 系统的通信机制. 然而, 移动 agent 环境下的消息传递机制有其特性, 集中表现在实现如何在 Internet 范围内有效、可靠地支持 agent 之间的移动通信, 如 agent 命名唯一性问题、移动 agent 的寻址问题以及保证消息传递可靠性的移动通信失效问题等. 这方面的工作在 Mole^[11,12], Aglets 等系统中也有体现, 但皆不尽如人意. 如 Mole 虽然提供了同步通信、异步通信和远程方法调用等多种通信原语, 但却缺乏层次支撑, 其寻址机制和通信可靠性保证问题的处理也较为简单, 系统性和有效性不足. 而 Aglets 虽然提供了灵活的基于消息传递的同步和异步通信方式, 保证了 agent 命名的唯一性, 并通过 Aglet Proxy 实现了 agent 的透明寻址. 但系统在通信失效解决方案等提高通信可靠性方面的工作所见较少, 在通信保障的系统性方面存在不足, 较难满足实用的需要.

基于上述分析, 在 Mogent^[3] 系统的通信机制的设计中, 我们针对 agent 的随机移动对 agent 通信的影响这一问题作了深入研究, 并给出了一套包括命名、寻址和可靠性保证的系统性的解决方案. 本文首先简要介绍 Mogent 的基于 MP(message passing) 的间接访问模式框架, 然后分析 agent 移动带来的对 agent 通信的影响, 最后结合 Mogent 的设计和实现, 提出一个系统的解决方案.

1 Mogent 系统通信框架

Mogent 系统以 Internet 为基础网络环境, 基于 Java 的移动 agent 系统. 从概念上讲, 系统中存在着 Mogent 和 Host 两个实体. Mogent 是一个移动 agent 的概念模型, 而 Host 则是 Internet 上的某个节点的抽象, 由其 IP 地址或相应域名来标识. Mogent 环境中每个 Host 安装有一个统一的执行环境 MogentServer, 维护及管理活动在本地的 Mogent. MogentServer 主要由迁移子系统、通信子系统、安全子系统和开发监控系统这 4 部分组成.

通信子系统中设置了双层名空间命名机制, 为 agent 通信提供了透明寻址, 采用基于消息传递的通信方式, 仿照 MPI, 提供了一组完整的常用同步、异步通信原语, 支持 Mogent 进行点对点、广播或组通信. 其基础构架采用一种基于 Home-Communicator 的通信模型. 每个 MogentServer 中拥有一对上述构件. 其中, Home 构件负责纪录该 Host 上“出生”的 mogent 的动态信息, 与 Communicator 合作完成透明寻址和通信失效解决工作. Communicator 构件纪录该 Host 上所有 Mogent 的信息, 负责 Mogent 之间通信的目标寻址、信件转发和通信失效排除等工作. 这既为处于同一台主机的 agent 提供了高效的本地通信方式, 也支持不同主机上 agent 之间的异地通信, 为 agent 用户提供了较大的灵活性. 如图 2 所示.

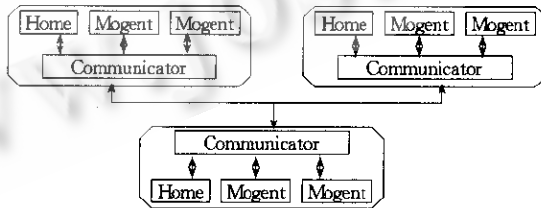


Fig. 2 Mogent system communication model
图2 Mogent系统通信模型

2 Mogent 系统中移动通信

移动 agent 可以自主地在异构网络上移动, 因此, 它们之间的通信面临以下问题: (1) 分布环境中 agent 标识的唯一性; (2) 动态移动的 agent 的定位; (3) 排除移动通信失效现象.

2.1 Agent 的命名问题

Agent 的命名机制产生的可动态移动 agent 的名称应该满足以下几条: (1) 在 agent 运行的分布环境中必须

同信息,从而导致协作的失败,甚至是错误的协作. 一个实用的移动 agent 系统必须解决这一问题.

2.3.2 分析

通信失效是由信件接收者的随机移动造成的,主要症结在于通信和移动所共享的“位置”信息的同步使用问题. 为此,我们从区分 agent 的移动状态和非移动状态的角度出发,将一个移动 agent 的生命周期划分为如图 5 所示的状态.

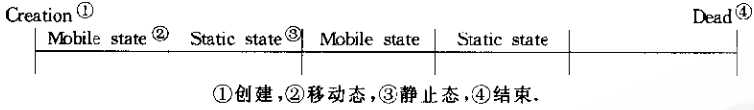


Fig. 5 Agent states analysis
图5 Agent状态划分

其中,agent 处于“静止态”是指 agent 从到达某个 Host 被恢复执行开始,到该 agent 离开该 Host 开始向另一个 Host 迁移为止的这段时间内所表现出的物理位置的无变化状态. agent 处于“移动态”是指 agent 从开始迁移直到在某一 Host 被恢复执行为止这段时间内所表现出的物理位置的不确定状态. 显然,如果向一个处于“移动态”的 agent 发送信件,或者在信件传输过程中通信对象 agent 发生了从“静止态”到“移动态”的状态变化,则必定会出现通信失效现象. 因此,接收者的“状态”在通信失效问题中具有决定性的意义. 当通信和移动相矛盾时,该“状态”就成了一个必须互斥使用的“资源”. 解决了这个互斥问题就能解决通信失效问题. 所以,在一个能够避免通信失效的移动 agent 系统中,必须且只须做到以下 3 条: (1) 准确纪录 agent 的状态信息; (2) 只能向一个处于“静止态”的 agent 发送信件; (3) 信件发送过程中必须限制接收者从“静止态”向“移动态”的状态转换.

基于以上分析, Mogent 系统采用集中式同步控制策略^[13], 通过 Home-Communicator 集中管理对目标 Mogent 地址信息的互斥访问,从根本上避免了通信失效现象的发生.

2.3.3 信号量设置

系统为每个 Mogent 引入了两个信号量: 迁移状态和在途信件数目, 其中, (1) 迁移状态: 保存在 Mogent 对应的 Home 结构中. 它记录 Mogent 是否处于迁移过程中, 其值可能为“迁移态”或“静止态”. 当一个 Mogent 在某个 Host 上执行时, 其状态值定义为静止态, 当一个 Mogent 按旅行计划准备向另一个 Host 迁移时, 当前 MogentServer 必须通知该 Mogent 的 Home 将状态值改为移动态; 当一个 Mogent 到达某个新 Host 时, 新 Host 上的 MogentServer 必须向 Mogent 的 Home 通知该 Mogent 的到达并将状态值改为静止态. (2) 在途信件数目: 保存在 Mogent 当前所处的主机上. 该值记录了当前时刻以该 Mogent 为通信对象的在途信件数. 显然, Home 在回复了寻址信件后, 应及时向目标 Host 提交“增加在途信件”通知, 目标 Host 在完成信件入箱后应及时将该值减 1. 实际上, “在途信件数”非 0 反映了目前“通信”功能已经在使用“位置”资源. 所以, 当一个 agent 准备离开某个节点时, 它所对应的“在途信件数”必须为 0, 否则, 移动要求将被暂缓.

2.3.4 算法

算法包括通信, 迁移请求和地址注册 3 个部分, 以图 6 为例.

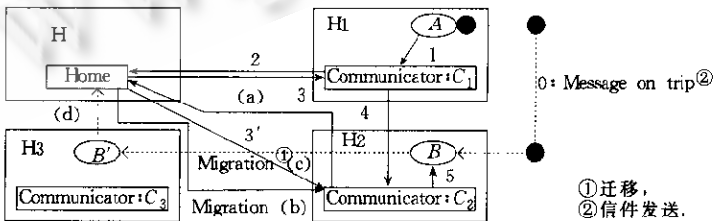


Fig. 6 Communication failure solution
图6 通信失效解决方案示意图

迁移状态初始化为“静止态”, 在途信件数目为 0.

(I) 远地通信

(0) A 向 B 发出信件.

- (1) H1 上的 Communicator 截获通信要求, 检查 B 是否在 H1 上.
- (2) 向目标 Mogent 的 Home 发送寻址消息.
- (3) H 上的 Home 收到寻址消息后, 检查该 Mogent 的迁移状态. 若状态为“静止态”, 向 H1 上的 C₁ 返回目标 Mogent 的当前地址 H2, 同时通知 B 所在主机的 Communicator 将其在途信件数目加 1 (如图 6 中的 3'). 若状态为“迁移态”, 表明目标 Mogent B 正在迁移过程中, 不返回物理地址, 将寻址信件放入通信阻塞队列中, 直至 B 到达新的 Host 并向 Home 登记后, 才将寻址信件从阻塞队列中释放, 并返回更新后的地址.
- (4) C₁ 获得 B 的地址 H2 后, 将信件发往 C₂.
- (5) C₂ 将消息直接放入 B 信箱, 然后将 B 的在途信件数目减 1.
- (II) 迁移请求
 - (a) B 通过 Communicator 向其 Home 提出迁移请求.
 - (b) Home 收到请求后, 修改 B 的迁移状态置为“迁移态”, 并返回确认信息.
 - (c) Mogent 收到确认信息后, 检查在途信件数目是否为 0. 若为 0, 说明没有信件在发送途中, 可以立即迁移; 若不为 0, 则等待直至所有途中的信件到达, 在途信件数目减为 0 为止. Mogent B 进入实际迁移状态.
- (III) 地址注册
 - (d) Mogent B 到达新的 H3 后, 向 Home 注册其当前物理地址. Home 收到注册信息后, 更新其保存的 B 的物理名, 将 B 的迁移状态置为“静止态”. 释放通信阻塞队列中所有等待与 B 通信的其他寻址信件, 并一一通知 H3 的 Communicator B 的“在途信件数目加 1”.

3 结论及与相关工作比较

移动 agent 间的协同是一个非常具有挑战性的移动 agent 关键技术, 其中的通信机制在移动 agent 研究领域受到广泛的关注, 代表性的工作有基于知识交换的 KQML 通信语言以及 IBM 公司研制的 Aglets 系统, 他们分别从 agent 通信语言研究和 agent 通信的消息传递系统支撑的角度展开了深入的研究. 然而, 迄今为止, 我们尚未见到同时支持 KQML 和消息传递两种协作方式的 agent 通信机制, 而在实际的应用中, 信息的交流是多级别、多层次的. 鉴于此, 本文提出了一种分层多模式的移动 agent 通信机制, 并介绍了关键的间接访问模式的设计和实现工作.

Mogent 系统的通信机制充分考虑了在移动 agent 环境下基于消息传递的 agent 移动通信所面临的困难, 有效地解决了移动 agent 系统中特有的 agent 标识唯一性问题、agent 通信的透明寻址问题以及保证 agent 通信可靠性的通信失效等问题, 一方面支持多种形式的 agent 协作通信, 另一方面保证了 agent 通信的可靠性和高效性. 总的来讲, Mogent 通信机制具有结构灵活、使用方便、效率较高、通信坚定性较强等优点, 为在其上进行各种后继研究奠定了一个良好的基础.

参考文献

- 1 Tao Xian-ping, Lü Jian, Li Xin *et al.* Mobile agent; one of the main paradigm of future distribute computing. *Computer Science*, 1999, 26(2): 1~6
(陶先平, 吕建, 李新等. 流动 agent: 一种未来的分布计算模式. *计算机科学*, 1999, 26(2): 1~6)
- 2 Lange D B. Mobile objects and mobile agents; the future of distributed computing?. In: Eric Jul ed. *Proceedings of the ECOOP'98. Lecture Notes in Computer Science 1445*, Berlin: Springer-Verlag, 1998. <http://ecoop98.vub.ac.be/technicalprogramme.html#Wednesday>
- 3 Milojevic D, Johansen D, Kotz D *et al.* Mobile agent application. *IEEE Concurrency*, 1998. 80~90. <http://www.computer.org/concurrency/pd1999/pdf/p3080.pdf>
- 4 Lü Jian, Tao Xian-ping, Dong Huan *et al.* The design and implementation of mobile agent system mogent1.0. Technical

- Report, State Key Laboratory for Novel Software Technology, Nanjing University, 1998
(吕建, 陶先平, 董桓等. 移动 agent 系统 Mogent 的设计和实现. 南京大学计算机软件新技术国家重点实验室技术报告, 1998)
- 5 Lü Jian. Some research on componentware frameworks based on mobile agent technology. Technical Report, State Key Laboratory for Novel Software Technology, Nanjing University, 1999
 - 6 Li Ying-jun. Research and implementation of object oriented parallel application framework [Ph. D. Thesis]. Nanjing University, 1999
(李英军. 面向对象并行应用框架的研究与实现[博士学位论文]. 南京大学, 1999)
 - 7 Lü Jian, Zhang Ming, Liao Yu *et al.* Cogent: the software component framework based on the mobile agent technology. Technical Report, State Key Laboratory for Novel Software Technology, Nanjing University, 1999
(吕建, 张鸣, 廖宇等. 基于移动 Agent 技术的构件软件框架 Cogent 的设计. 南京大学计算机软件新技术国家重点实验室技术报告, 1999)
 - 8 Labrou Y, Finin T. A proposal for a new KQML specification. 1997. <http://www.cs.umbc.edu/kqml/papers/kqml97.pdf>
 - 9 Lange D B, Oshima M. Programming and Deploying Mobile Agents With Aglets. New York: Addison-Wesley Press, 1998
 - 10 Dong Huan, Ding Jun-hua, Li Xin *et al.* On open communication frameworks for software agents. In: Chen Jian, Li Ming-shu, Meyer B eds. Proceedings of the TOOLS Asia'97. Los Alamitos, CA: IEEE Computer Society Press, 1997. 188~195
 - 11 Rothermel K, Schwehrm M, Hohl F *et al.* ATOMAS; a transaction-oriented open multi agent-system. Technical report, The Distributed Systems Group at the IPVR, University of Stuttgart, 1998. http://www.informatik.uni-stuttgart.de/cgi-bin/ncstrl_rep_view.pl?/inf/ftp/pub/library/ncstrl.ustuttgart-fi/TR-1998-11/TR-1998-11.bib
 - 12 Straßer M, Baumann J, Hohl F. Mole-a Java based mobile agent system. In: Mühlhäuser M ed. Special Issues in Object Oriented Programming. Herdelberg: Dpunkt-Verlag Press, 1996. 327~334

Communication Mechanism in Mogent System

TAO Xian-ping FENG Xin-yu LI Xin ZHANG Guan-qun LÜ Jian

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

(Institute of Computer Software Nanjing University Nanjing 210093)

Abstract Mobile agent technology can be used as one of the key technologies for new componentware frameworks. The communication mechanism of mobile agent plays a very important role in mobile agent systems. Based on the design of the mobile agent system—Mogent, after analyzing the mobility's influence on its communication mechanism, the authors propose a systematic solution to communication mechanism used in Mogent system in this paper. This mechanism involves agent naming, agent addressing and agent communication failure solutions, and its high availability, performance and robustness are demonstrated.

Key words Mobile agent, Mogent system, communication mechanism, communication failure.