

一种移动 Agent 通信算法*

王忠群¹⁺, 陶先平^{2,3}, 冯新宇^{2,3}

¹(安徽工程科技学院 计算机科学与工程系,安徽 芜湖 241000)

²(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

³(南京大学 计算机软件研究所,江苏 南京 210093)

A Mobile Agent Communication Algorithm

WANG Zhong-Qun¹⁺, TAO Xian-Ping^{2,3}, FENG Xin-Yu^{2,3}

¹(Department of Computer Science and Engineering, Anhui University of Technology and Science, Anhui 241000, China)

²(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 21009, China)

³(Institute of Computer Software, Nanjing University, Nanjing 21009, China)

+ Corresponding author: Phn: 86-553-2871235, E-mail: zqwang@auts.edu.cn

<http://www.cs.auts.edu.cn>

Received 2002-12-04; Accepted 2003-03-05

Wang ZQ, Tao XP, Feng XY. A mobile Agent communication algorithm. *Journal of Software*, 2003,14(7): 1292~1299.

<http://www.jos.org.cn/1000-9825/14/1292.htm>

Abstract: An algorithm based on communication algorithm in the Mogent system is presented in this paper. In this algorithm, address registration and multicasting are used by means of address-book so that it is more efficient, and suitable for many kinds of patterns of migration and communication and has a better solution to the troubles with mobile Agent communication.

Key words: mobile Agent; communication; location-transparency; multicast; address-book

摘要: 在 Mogent 系统所实现的通信算法基础上,借助通讯录再次提出一种基于组播和地址注册的通信算法,它更加有效,能适应多种迁移和通信模式,可以较好地解决移动 Agent 通信所面临的难题。

关键词: 移动 Agent;通信;地址透明性;组播;通讯录

中图法分类号: TP393 文献标识码: A

移动 Agent 技术对于未来的分布式系统的设计、实现和维护具有重要意义。一般来说,移动 Agent 是一个运行于开放、动态网络环境中的封装良好的计算实体。它代表用户在网络中移动,完成指定的任务。通常,移动 Agent 由数据、操作和行为规则封装而成。虽然移动 Agent 目前还没有 Application Killer,但与传统的 RPC 相比,

* Supported by the National High-Tech Research and Development Plan of China under Grant No.863-306-ZT02-01-4 (国家高技术研究发展计划(863)); the Key Project of the Natural Science Foundation of Education Department of Anhui Province of China under Grant No.2003kj007zd (安徽省教育厅自然科学基金重点基金)

第一作者简介: 王忠群(1965—),男,安徽芜湖人,副教授,主要研究领域为分布对象计算,软件工程,移动 Agent 技术。

它具有支持网络断接操作、减轻网络负载、动态适应网络等优点,将广泛应用于并行处理、电子商务、信息发布、监视和通知、分布信息查询等领域。

要想使 Agent 系统实用,必须提供有效而方便的透明定位和消息路由机制.文献[1,2]认为,在目前众多的移动 Agent 系统中,并没有对以上机制给出完善的解决方案;有些算法较复杂,实用性较差.在有些系统中仅提供了基本的消息发送原语,而把定位和消息发送交给编程人员去完成^[3,4],即使是目前提出的定位和消息路由算法^[5,6],也给系统带来了较大的额外开销,继而影响通信的效率.在文献[1]中,我们对文献[3~6]的通信方式进行了详细的分析,给出了它们的特点和不足,限于篇幅,在此不再赘述.

文献[2]的算法思想是利用 HomeAgent 作为消息传递的中介,并且提供对 MA(mobile Agent)位置的管理,所有需要传递给 MA 的消息都先发送给相应的 HomeAgent 暂存,由 MA 和 HomeAgent 交互实现消息提交,同时将提交方式由 HomeAgent 主动发送改为由 MA 主动来取,并且通过消息通知和主动查询消息相结合来保证 MA 能及时、可靠地获取每个消息.该算法的特点是:(1) 对 HomeAgent 的过分依赖以及出现通信瓶颈问题.(2) 消息通知和主动查询消息解决了消息对 Agent 的追逐.虽然消息是一次可靠提交,但是对一个消息的接收面临着多次 MA 和 HomeAgent 交互,导致较多的额外通信开销,其效率较低.(3) 当通信消息数和消息体积较大时,HomeAgent 的通信瓶颈问题尤为突出.(4) 当网络带宽等资源在动态不定的情况下,MA 迁移过程所需时间最大值 $T_{\text{threshold}}$ 是难以确定的,从而使得当 MA 处于迁移状态时,“消息到达”这个通知消息可能发送多次,甚至因为 $T_{\text{threshold}}$ 设置过小而错误判断 Agent 意外死亡,这一点使该算法的实用性值得怀疑.通过分析,我们认为该算法存在与文献[7]中的算法同样的一些不足.例如,对 HomeAgent 的过分依赖、出现通信瓶颈、地址注册开销大、通信效率较低等问题.

在 Mogent 系统的设计和开发中,我们参考 Mobile IP 方案提出了一种基于同步思想的通信算法^[7](以下称为原算法).该算法思想是,移动 Agent 在每次迁移之前和到达目的地之后,都要分别向其 Home 注销和注册其地址.它采用“先寻址再发送”的消息传送方式,如果移动 Agent A1 要与 Agent A2 通信,首先 A1 要从 A2 的 Home 获得 A2 的当前地址,然后将消息直接发往该地址.该算法在目标 Agent 的 Home 上实现了目标 Agent 的迁移和通信的同步,从根本上解决了通信失效现象,但也存在地址注册开销大、Agent 迁移受到限制和访问 Home 时可能出现拥挤现象、效率不高等不足.文献[1]中的算法(以下称为改进算法)在原算法的基础上,引入了信箱 Mailbox,并且把它从 Agent 分离出来,通过 Agent 的 Home 和其 Mailbox 的配合,实现了 Agent 之间可靠且有效的通信.实验结果表明,改进算法在那些 Agent 迁移频繁且通信较少的应用中对原算法优化尤其有效,但是,在那些 Agent 间通信量大和通信次数较多的应用中,优化效果不明显,并且仍然具有系统对 Home 依赖较强、Home 可能成为通信瓶颈等特点.另外,改进算法对决定 Mailbox 留在原地的因素阐述得也不够全面.例如,没有阐述自己接收到的消息次数较少是决定其 Mailbox 留在原地的因素之一.Agent 间通信量大、通信次数多恰恰是引起整个系统的性能、带宽和网络服务质量下降的一个重要原因,而改进算法没有对它们给予解决.

在基于 Agent 的系统中,Agent 何时迁移以及 Agent 间何时通信呈现出不确定性,依据迁移的频繁程度、通信的频次,我们可以把通信模式划分为以下几种:(1) 迁移和通信都频繁;(2) 迁移和通信都不频繁;(3) 迁移频繁而通信不频繁;(4) 迁移不频繁而通信频繁.一个好的通信算法应该是可靠的,能够适应较多的通信模式,并且都能取得较高的通信效率.

本文在 Mogent 系统的原算法和改进算法基础上提出一种更加完善的新颖算法.它通过引入通讯录来记载那些给自己发送消息的发送者,把地址注册和组播两种方式结合起来,使得发送消息 Agent 在通信的大部分情况下直接和目标 Agent 通信,只有在其通信失效时才需要到 Home 处去寻址定位目标 Agent 的地址,然后再和目标 Agent 通信.新算法具有文献[1,7]中算法的可靠性以及比它们能够适应更多的通信模式,并且取得更高的通信效率等优点,较好地解决了移动 Agent 通信所存在的问题.

本文第 1 节描述系统模型和消息发送过程.第 2 节详细叙述我们的算法.第 3 节给出算法的分析和实验证明.最后给出结论.

1 系统模型

在我们实现的移动 Agent 系统(Mogent 系统)中,系统包括 Mogent,Home,Communicator 及其 Mailbox,Adr-book(通讯录)和 Out-set 等 6 种实体,如图 1 所示.(1) 在 Mogent 系统中,移动 Agent 被称为 Mogent.(2) 每个 Mogent 都有其出生地(创建结点),它被称为 Mogent 的 Home.(3) Communicator 一方面负责通信以及把目标 Mogent 的地址通知给发送消息 Mogent,另一方面同时也负责将本结点上运行的 Mogent 所接受到的消息提取给对应的 Mogent 体.(4) Mailbox 是该 Mogent 的通信信箱,所有发送给 Mogent 的异步消息都被暂时存放在其信箱中,当用到这些消息时,其 Mogent 才从信箱中取出来.(5) 每个 Mogent 都设置一个对应的通讯录 Adr-book.Mogent 通讯录负责记录发送给自己消息的所有其他 Mogent 的名称、地址以及所接收到消息的次数等.通讯录用于当 Mogent 迁移时,它确定到底向哪些 Mogent 组播通知其最新通讯地址,以及其信箱 Mailbox 是否迁移(见第 2.1 节中 Mailbox 和 Adr-book 请求迁移算法). (6) Out-set 是指那些最近未曾联系过的 Mogent,不需要进行记录,只是为了叙述问题方便而引入的.

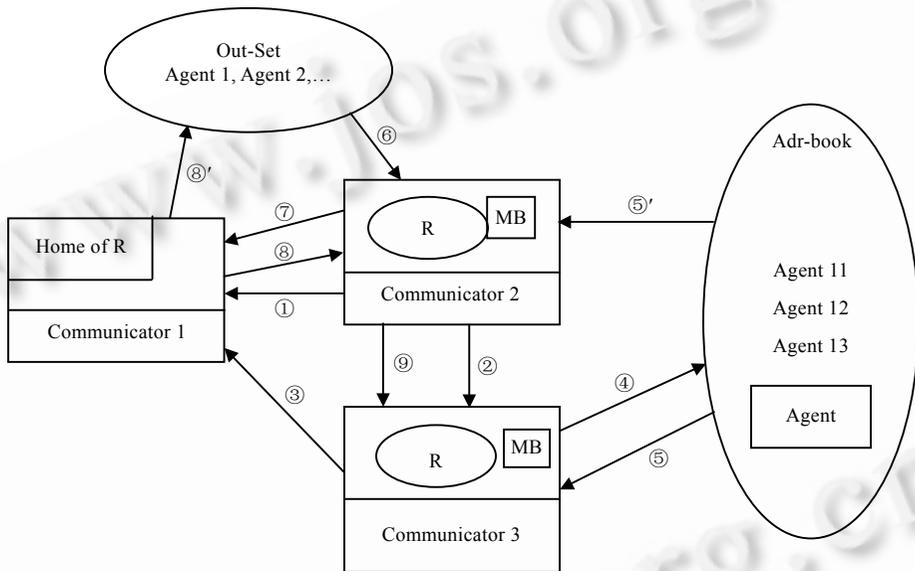


Fig.1 System model and process of message delivering
图 1 系统模型以及消息发送过程

我们采用的是 Mogent 体和 Mailbox,Adr-book 分离的思想^[1],但是 Adr-book 和 Mailbox 绑定在一起,同步迁移和驻留(以下我们称 Mailbox 迁移是指 Mailbox 和 Adr-book 同时迁移,Mailbox 停留是指 Mailbox 和 Adr-book 同时驻留).把 Mailbox 作为该 Mogent 的消息代理,它不仅存放其对应的 Mogent 所接受的消息,还可以主动要求迁移到 Mogent 体所在的结点上.一般情况下,Mogent 及其 Mailbox,Adr-book 处于同一结点,三者同时迁移,但是必要时可以分处于不同的结点并且分别迁移,因为 Mogent 在迁移频繁且传输消息少的时候只需要迁移 Mogent 体,而将其 Mailbox 留在原地充当其消息代理,接受异步消息.由于所有发送给 Mogent 的消息都存放在其 Mailbox 中,所以我们只关心所发送的消息是否传递到目标 Mogent 的 Mailbox 中,而不管 Mogent 体和其 Mailbox 是否在同一结点,都是由 Communicator 负责把 Mogent 体所需要的消息在适当的时候提取出来给 Mogent 体处理.下面我们来叙述模型中每一步骤的含义.

①表示 Mogent R 通知其 Home 准备迁移,如果其 Mogent 体和 Mailbox 同时迁移,则注销其地址,否则只迁移 Mogent 体,则不通知其 Home,当然,不存在注销其地址.②表示迁移 Mogent R,可能同时迁移 Mogent 体和其 Mailbox,也可能只迁移 Mogent 体.③表示如果 Mogent 体和其 Mailbox 同时迁移,则到其 home 注册新地址.④表示 Mogent R 通知那些最近给它发送过消息的 Mogent 它所迁移的新地址,甚至可以根据它们发送消息的次数进行选择某些 Mogent 来通知.当然,如果只迁移了 Mogent 体,则也不需要通知它们.⑤'表示通讯录 Adr-book

中的 Mogent 在最近一段时间向 Mogent R 发送过消息。⑤表示 Agent 直接向 Mogent R 发送消息,不需要到 Mogent R 的 Home 寻址。当然,如果 Mogent R 只迁移了 Mogent 体,那么 Agent 发送消息的地址仍然是其留下的 Mailbox 的地址。⑥表示通讯录中没有记录的 Mogent 或者说是最近未向 Mogent 发送过消息的 Mogent 现在需要向 Mogent R 发送消息。如果 Mogent R 的体与其 Mailbox 一同迁移走了,则发送消息失败,它需要与 Mogent R 的 Home 联系,经过⑦,⑧获得 Mogent R 的最新地址,⑧'给消息发送者通知 Mogent R 以免它第 2 次发送消息时失败。如果 Mogent R 的体与其 Mailbox 没有一同迁移走,发送消息成功。⑨表示在 Out-set 中的 Mogent 发送消息给 Mogent R 失败后获取了 Mogent R 的最新地址再次重发消息给目标 Mogent R。

2 新颖的移动 Agent 算法以及通讯录机制描述

2.1 新颖的移动 Agent 算法

新算法包括 Mogent 体通信、迁移、Mailbox 迁移算法 3 部分。在新算法中我们引入了两个控制变量:(1) 消息失败在途标记;(2) 失败在途消息信号量。其中,前者是为每个消息设置的逻辑变量,它表示一个发送消息在预发送阶段是否到达目标结点;后者是为每个 Agent 设置的信号量,它表示其他 Agent 给自己已发出的但处于失败在途的所有消息总数。算法过程分别见下。

注:Mailbox 和 Adr-book 绑定在一起,它们同步迁移和驻留。

2.1.1 Agent 体通信算法

•消息发送者

(1) 如果目标 Agent 和发送者 Agent 在同一结点,则直接把消息(消息中包含目标 Agent 的 Home 地址)放入目标 Agent 的 Mailbox 即可,算法结束。

(2) 预发送阶段。如果目标 Agent 和其 Mailbox 在同一结点(如果不在同一结点,由于 Agent 体迁移时没有通知其通讯录上 Agent 的新地址,故 Mailbox 的地址仍然是发送 Agent 的发送地址),则发送 Agent 根据其地址直接发送,如果发送成功,算法结束。如果发送不成功,消息则保留在当前结点转发,并且给该消息的失败在途标记设置为“真”,转步骤(3)。

(3) 依据消息中的目标 Agent 的出生地 Home 地址,询问目标 Agent 的迁移新地址以后,同时利用失败在途消息信号量机制取得消息发送和接受者同步。如果 Mailbox 为“迁移”状态,则阻塞,等待;若 Mailbox 为“非迁移”状态,则 Home 通知 Mailbox 所在结点给该 Agent 的失败在途消息信号量增加 1,获得目标 Agent 地址以后,继续在失败所在结点转发该消息,并且 Home 向原消息发送者通知其接受者的最新地址。如果询问得知目标 Agent 还处于在途中,发送者则等待目标 Agent 的出生地 Home 主动通知其目标 Agent 所迁移的新地址(注意,我们只关心 Mailbox 的状态,因为所接收的信件都存放在其中)。

•消息接受者

对于该消息发送者,如果此次为给该接受者第 1 次发送消息,则接受者在其通讯录上记载该发送者的通信地址以及通信次数为 1;如果不是第 1 次,则在该发送者的通信次数上增加 1。判断该消息的失败在途标记是否为“真”,若为“真”,则给 Mailbox 的失败在途消息信号量减 1,否则不减 1。

2.1.2 Agent 体迁移请求算法

(1) 检查 Agent 体和 Mailbox 是否在同一结点,如果在同一结点,则转移到步骤(2),否则转移到步骤(4)。

(2) 检查其通讯录上主动与其通信的 Agent 以及通信次数,如果通信的 Agent 数目以及通信的次数很少,则本 Agent 直接迁移,而不需要向其出生地 Home 注销通信地址以及迁移到目标结点后再向其出生地 Home 注册新地址,也不必向其通讯录上记载的那些 Agent 组播通信其最新通信地址,算法结束;否则,检查失败在途消息信号量是否为 0,如不为 0,等待直至为 0,将 Agent 状态设定为“迁移”,Agent 体、Mailbox 一同迁移,并且要求注销其当前地址。转移到步骤(3)。

(3) 到达目标结点以后,Agent 向其出生地 Home 注册新地址并且要求把其状态设置为“非迁移”,同时依据通讯录机制所描述的向那些与其联系过的 Agent 组播通知其最新地址。然后清除刷新通讯录。算法结束。

(4) 发送消息给其 Mailbox 检查其通讯录,如果通讯录上主动与其通信的 Agent 以及通信次数很少,则本 Agent 直接迁移,而不需要向其出生地 Home 通信要求注销其当前地址以及迁移到目标结点后再向其出生地 Home 注册新地址,也不必向其通讯录上记载的那些 Agent 组播通知其最新通信地址,算法结束;如果通讯录上主动与其通信的 Agent 以及通信次数很多,则转移到步骤(5).

(5) 迁移 Agent 发送消息给其 Mailbox,要求它迁移到目标结点(调用 Mailbox 迁移算法),同时 Agent 体本身也申请迁移.

2.1.3 Mailbox 和 Adr-book 请求迁移算法

Mailbox 定期检查自己,决定是否迁移.

(1) Mailbox 检查是否与其 Agent 在同一结点,如果是在同一结点,算法结束.否则,转移到步骤(2).

(2) 检查其通讯录向该地址发送消息的 Agent 数目和消息是否较多,如果较多,则主动要求迁移到其对应的 Agent 体所在结点,转移到步骤(3);如果不多,则算法结束.

(3) Mailbox 分别通知其出生地 Home 和 Agent 体所在结点,本 Mailbox 要求迁移到和 Agent 体同一结点,同时其 Home 将其状态设置为“迁移”态.要求其 Agent 暂时冻结不能迁移以及检查失败在途消息信号量是否为 0,如果为 0,则 Mailbox 实施迁移,到达目标结点以后再向其出生地 Home 申请注册新地址,并且 Home 将其状态设置为“静止”,如果失败在途消息信号量不为 0,则等待直至为 0 再实施 Mailbox 迁移.然后,依据通讯录描述机制将其最新地址组播通知给其通讯录上记载的那些 Agent.算法结束.

2.2 通讯录机制描述

当一个 Mogent 迁移以后,它要把其最新地址通知一些最有可能需要与它通信的 Mogent.为此,我们引入通讯录机制,其结构如图 2 所示.其中 Sender-name 表示给自己(指该通讯录所属的 Agent)发送消息的发送者名称,Address 表示发送者的当前地址,Time-stamp 表示最近所接收到消息的时间戳,Com-count 表示一段时间内给自己发送的消息数.

Sender-name	Address	Time-stamp	Com-count

Fig.2 Structure of address-book

图 2 通讯录结构

考虑到通讯录 Adr-book 在迁移时要花费一定的系统代价以及预测的准确性,因此其大小设置要求适中,不能过大.当有新的 Agent 在向自己发送消息时,受通讯录体积的限制,需要选择通讯录已记载的消息发送者予以淘汰.淘汰消息发送者可以借用内存页面调度算法思想,依据最新和最常联系的策略进行.针对此策略,当有新的 Agent 给自己发送消息时,淘汰时间戳最久未发送消息的或者发送次数最少的 Agent.另外,每隔一段时间对 Time-stamp 和 Com-count 进行一次刷新.显然,引入通讯录达到了以下目的:(1) 记载了与自己通信次数较多的 Agent;(2) 可以知道是否存在若干固定的 Agent 与自己通信;(3) 由于引入了信箱分离技术,依据通讯录上记载的通信次数较少可以决定其 Mailbox 留在原地.达到上述目的使得新算法具有在 Agent 间直接通信,提高了通信效率以及适应多种通信模式等特点.注意:我们把通知地址的消息和一般通信消息区分开,这里记载的消息为一般通信消息,否则难以得到真正通信消息的时间戳和通信次数.

3 算法特性分析

原算法和改进算法在解决 Agent 的寻址透明性、通信失效性以及通信效率方面都取得了较好的效果.虽然新算法在实现机制上也使用了 Home 以及 Mailbox 与 Agent 体分离的思想,但是解决问题的思路有别于原算法和改进算法.再者,它采用了通讯录和组播思想,使新算法在通信效率方面能够取得比原算法和改进算法更好的效果.新算法与原算法以及改进算法相比,具有如下显著的特点:(1) 一般情况下,新算法由于通过组播通知了需要和自己联系的那些 Agent 的最新地址,通信过程是直接通信方式,不需要先寻址再通信.只有在通信失败的情

况下,才借助目标 Agent 的出生地 Home 寻址后再次通信.允许该消息第 1 次发送失败且仅失败一次,它既部分解决了由于对 Agent 的完全依赖所给系统引进的性能瓶颈以及寻址的在线限制和健壮性,又解决了基于广播方法而给系统带来的较大额外通信开销.对于一个迁移的 Agent M 来说,作为目标 Agent 的使用组播究竟通知哪些 Agent(因为这些 Agent 可能会发送消息给 Agent M)其最新的迁移地址,我们采取的策略是依据其 Adr-book 上最近和自己联系的那些 Agent 以及通信的次数,选择通讯录上全部或者部分 Agent 来通知其最新地址.(2) 依据最近通信次数的记载可以方便地决定组播成员范围的大小以及迁移时决定是否把 Agent 体与其 Mailbox 一同迁移.这样可以减少迁移过程的数据量以及 Agent 体向其出生地 Home 注销和注册地址的开销.(3) 目标 Agent 在迁移到新结点而未通知某 Agent 其新地址,该 Agent 在第 1 次给目标 Agent 发送消息时通信失败,但是以后在目标 Agent 没有迁移新结点之前,该 Agent 都可以采用第一次通信的新地址而直接和目标 Agent 通信(由于借助 Home 记录下第 1 次和目标 Agent 成功通信的地址),在这种情况下,只要发送者多次和同一目标 Agent 通信,其通信效率将大幅度地提高,但是,原算法和改进算法由于没有具备直接通信机制,则只能仍然先向目标 Agent 的 Home 寻址然后再通信.

为了验证新算法对通信开销的优化以及分析结果的正确性,我们在一个简化的移动 Agent 环境中实现了新算法.在实际系统中,上述特点的 3 种情况都是随机出现的,我们的新算法依据其通讯录上记载的和消息发送者的联系情况进行预测.以随机数代替实际出现的情况,模拟实验方案如下:3 个发送消息的 Mogent 静止不动,以随机的时间间隔发出消息,接受消息 Mogent 在几个结点上循环迁移,并且接受消息发送者的消息.在实验中,假定接受 Mogent 的迁移的次数为 100 次,它在每个结点的驻留时间呈指数分布,其期望值为 10s.消息发送的时间间隔 t 也服从指数分布,期望值为 λs ,则“每个结点的平均消息数”为 $10/\lambda$.接受 Mogent 每次迁移前估计在下一结点的停留时间并计算接受消息的数目,如果接受消息数目为 0,则进行无通信迁移,即 Mogent 体留下其 Mailbox 自己迁移到下一个结点;否则,Mogent 体和其 Mailbox 一同迁移,新算法还向其他结点组播其最新通信地址.

实验结果表明,3 种算法在消息接受者不停地迁移时,都可以接受来自消息发送者的消息而不会发生消息丢失现象.新算法分别与原算法以及改进算法的通信开销对比,如图 3 和图 4 所示.在这里,我们已接受 Mogent 在 100 次迁移中系统发送的所有的消息,包括 Mogent 消息、寻址消息、地址注册与注销消息等.由于 Mogent 迁移消息和其他几种控制消息在通信数据量上的不同,我们设定 Mogent 的消息通信开销为单位 1,而控制类消息其通信开支为 1/4.从图 3 和图 4 我们可以得到以下结论:(1) 当 Mogent 迁移频繁时,很少有通信的情况下,新算法只迁移其 Mogent 体,将其 Mailbox 保留在原结点,不随自己迁移,并且 Mailbox 作为自己的通信代理,Mogent 在结点间迁移时不向其 Home 注销和注册地址,从而大大减少了额外开销.(2) 当有较多的 Mogent 与自己通信并且其中通信次数较多时,通讯录记载下来这些情况,组播通知了那些 Mogent(最近联系(LRU)或者最经常联系(MFU))自己的最新地址,让它们和自己直接通信,避免了通信时要求先寻址再通信.每次通信减少一次寻址的通信开销,显然,通信效率的改善将非常可观.(3) 和自己通信的 Mogent 个数以及其中通信次数适中的情况,依据第 2 种情况分析,通信效果依然有改善.对于第 1 种情况,改进算法已经实验证明了其优化效果明显,对于第 2 种情况,本算法优化效果明显.综合起来看,新算法在两种情况都具备明显的优化结果,并且借助于通讯录记载通信次数,使其能够自动调节适应更多不同的通信模式,从而全面提高了系统通信效果,改善了对 Home 有较强的依赖性以及由此带来的通信瓶颈等问题.

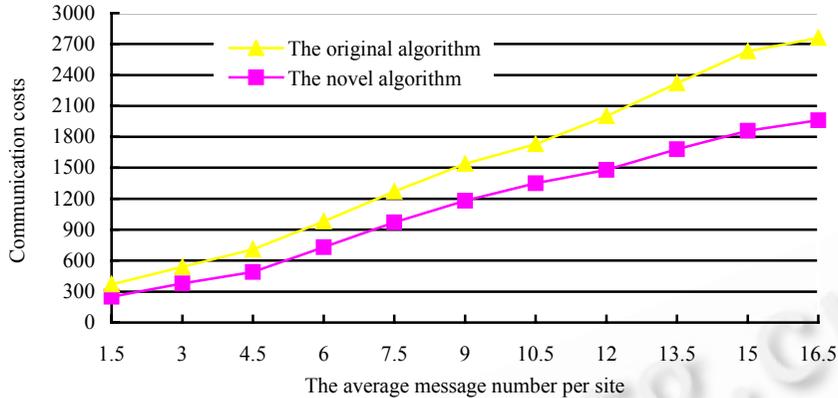


Fig.3 The experimental results of the novel algorithm and the original algorithm
图3 新算法和原算法通信开销比较

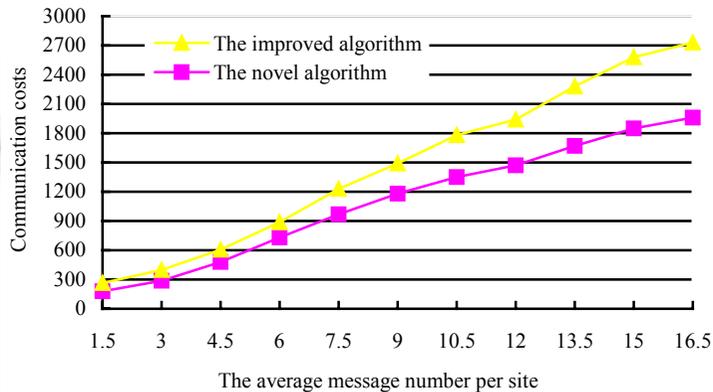


Fig.4 The experimental results of the novel algorithm and the improved algorithm
图4 新算法和改进算法通信开销比较

4 结 语

Agent 间的通信是 Agent 系统必备的基础设施,透明寻址和通信可靠性是移动 Agent 系统必须解决的两个难题.一个好的通信算法首先要解决透明寻址和通信的可靠性,再者,对于一些非功能性需求应给予足够的重视.例如,引入较少的额外通信开销、具有较好的健壮性、尽可能减少对 Agent 迁移的限制以及降低 Agent 对其出生地 Home 的依赖.当然,有些非功能性需求难以同时达到,但是我们应该有一个较满意的权衡.原算法和改进算法在上述方面已经取得了较好的效果.本文在它们的基础上引入了通讯录记载那些发送消息和自己通信的 Agent 地址以及通信次数,依此向它们组播自己最新迁移的地址,从不同的思路使用了 Agent 的出生地 Home,从而实现了部分直接寻址,使得新算法在通信模式的适应性、系统效率以及对于 Home 的依赖性等方面都有了进一步的改善.如果 Agent 的消息发送者过于随机,本算法则达不到对原算法性能优化的效果.

致谢 在此,我们向对本文的工作给予支持和建议的南京大学计算机软件新技术国家重点实验室的吕建老师表示衷心的感谢.同时,第一作者还向其重点实验室访问研究期间给予其帮助的其他老师和研究生表示感谢.

References:

- [1] Feng XY, Tao XP, Cao C, Li X, Zhang GQ, Lü J. An improved mobile Agent communication algorithm. Chinese Journal of Computers, 2002,25(4):357~364 (in Chinese with English abstract).

- [2] Wu G, Wang HM, Wu QY. An algorithm on location management and reliable communication for mobile Agents. Journal of Software, 2002,13(2):269~273 (in Chinese with English abstract).
- [3] StraBer M, Baumann J, Hohl F. Mole—A Java based mobile Agent system. In: Mühlhäuser M, ed. Special Issues in Object Oriented Programming. Dpunkt-Verlag, 1997. 301~308. <http://mole.informatik.uni-stuttgart.de/papers.html>.
- [4] Lange DB, Oshima M. Mobile Agents with Java: Aglet API. World Wide Web, 1998,1(3):111~121.
- [5] Murphy A, Picco GP. Reliable communication for highly mobile Agents. In: Proceedings of the Agent Systems and Architecture/Mobile Agents (ASA/MA)'99. 1999. 141~150. <http://www.cs.Rochester.edu/Murphy/papers>.
- [6] Belle WV, Verelst K, D'Hondt T. Location transparent routing in mobile Agent systems merging name lookups with routing. In: Proceedings of the 7th IEEE Workshop on Future Trends of Distributed Computing Systems. 1999. 207~212. <http://bovy.rave.org/research.html>.
- [7] Tao XP, Feng XY, Li X, Zhang GQ, Lü J. Communication mechanism in Mogent system. Journal of Software, 2000,11(8):1060~1065 (in Chinese with English abstract).

附中文参考文献:

- [1] 冯新宇,陶先平,曹春,张冠群,吕建.一种改进的移动 Agent 通信算法.计算机学报,2002,25(4):357~364.
- [2] 吴刚,王怀民,吴泉源.一个移动智能体位置管理与可靠通信的算法.软件学报,2002,13(2):269~273.
- [7] 陶先平,冯新宇,李新,张冠群,吕建.Mogent 系统的通信机制.软件学报,2000,11(8):1060~1065.

oo

2003 全国软件与应用学术会议(NASAC 2003)

征文通知

由中国计算机学会软件工程专业委员会主办,上海交通大学计算机系承办,北京大学、北京航空航天大学、复旦大学、国防科技大学协办的 2003 全国软件与应用学术会议将于 2003 年 11 月 14 日~16 日在上海召开。届时将进行软件工程等方面的技术与应用交流,会议将出版正式论文集,并将优秀论文推荐到核心学术刊物(EI 检索源)发表。欢迎大家踊跃投稿。

一、征文范围(包括但不限于)

需求工程、软件过程、质量保障、软件工具与环境、软件工程实践、软件工程教育、操作系统、中间件、软件复用、软件语言、应用软件,等。

二、论文要求

1. 论文未曾在其他杂志、会议上发表或录用
2. 论文长度: 每篇限定在 6 页(A4)内
3. 请以 PDF 或者 PS 格式提交论文。有关文章的版心、字号、题目、各级标题、格式及参考文献格式与《软件学报》相同,具体模板请参阅如下网址 <http://www.jos.org.cn> 中的“相关网站”一栏

三、重要日期

文稿截止日期:2003 年 8 月 15 日

论文录用通知日期:2003 年 9 月 20 日

四、联系方式

200030 上海交通大学计算机系 李明禄

E-mail: li-ml@cs.sjtu.edu.cn

关于会议更详细内容请访问:<http://www.cs.sjtu.edu.cn/nasac2003/>