移动 Agent 综述

Overview of Mobile Agent

陈 松

（重庆交通学院计算机系 重庆400074）

Abstract The history and background of the mobile Agent is reviewed. The technology of mobile Agent is expatiated from the followed three aspects: structure, key technology and security. After some mobile Agent orgnizations and their specifications are introduced, the common application and its development are discussed. In the end, the conclusion and prospect are made.

Keywords Mobile Agent, Agent Communication Language, Security, Interoperability, MASIF FIPA

1. 引言

移动 Agent 的概念是90年代初由 General Magic 公司在推出商业系统 Telescript 时提出的，简单地说，移动 Agent 是一个能在异构网络中自主地从一台主机迁移到另一台主机并与其它 Agent 或资源交互的程序，它实际上是 Agent 技术与分布式计算技术的混血儿。传统的 RPC 客户和服务器间的交互需要连续的通信支持；而移动 Agent 可以迁移到服务器上，与之进行本地高速通信，这种本地通信不再占用网络资源。移动 Agent 迁移的内容既包括其代码也包括其运行状态。运行状态又可分为执行状态和数据状态。执行状态主要指移动 Agent 当前运行时状态，如程序计数器、运行栈内容等；数据状态主要指与移动 Agent 运行有关的数据堆的内容。按所迁移的运行状态的内容，移动 Agent 的迁移可以分为强迁移和弱迁移。强迁移同时迁移移动 Agent 的执行状态和数据状态，但这种迁移的实现较为复杂；弱迁移只迁移移动代理的数据状态，其速度较强迁移快，但不能保存移动 Agent 的完整运行时状态。

移动 Agent 不同于远地执行，移动 Agent 能够不断地从一个网络位置移动到另一个位置，能够根据自己的选择进行移动；移动 Agent 不同于进程迁移，一般来说进程迁移系统不允许进程选择什么时候和迁移到哪里，而移动 Agent 带有状态，所以可以根据应用的需要，可以在任意时刻移动，可以移动到它想去的任何地方；移动 Agent 也不同于 Applet，Applet 智能从服务器向客户机单方向移动，而移动 Agent 可以在客户机和服务器之间双向移动。

移动 Agent 具有很多优点，移动 Agent 技术通过将服务请求 Agent 动态地移到服务器端执行，使得此 Agent 较少依赖网络传输这一中间环节而直接面对要访问的服务器资源，从而避免了大量数据的网络传送，降低了系统对网络带宽的依赖；移动 Agent 不需要统一的调度，由用户创建的 Agent 可以异步地在不同节点上运行，待任务完成后再将结果传送给用户；为了完成某项任务，用户可以创建多个 Agent，同时在一个或若干个节点上运行，形成并行求解的能力；此外它还具有自治性和智能路由等特性。

2. 移动 Agent 系统结构及其关键技术

2.1 移动 Agent 系统结构

移动 Agent 系统由两部分组成：移动 Agent 和移动 Agent 服务设施（或称移动 Agent 服务器）。移动 Agent 服务设施基于 Agent 传输协议（Agent Transfer Protocol）实现 Agent 在主机间的转移，并为其分配执行环境和服务接口，Agent 在服务设施中执行，通过 Agent 通信语言 ACL（Agent Communication Language）相互通信并访问服务设施提供的服务。

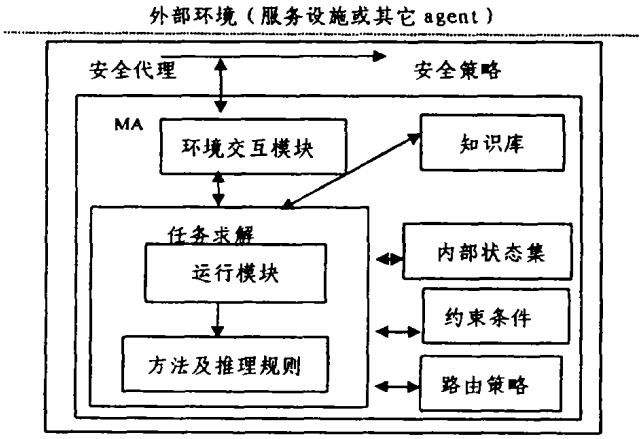

图1 移动 Agent 的结构模型

移动 Agent 体系结构定义为以下相互关联的模块：安全代理、环境交互模块、任务求解模块、知识库、内部状态集、约束条件和路由策略。体系结构的最外层为安全代理，它是 Agent 与外界环境通信的中介，执行 Agent 的安全策略，阻止外界环境对 Agent 的非法访问。Agent 通过环境交互模块感知外部环境并作用于外部环境。环境交互模块实现 ACL 语义，保证使用相同 ACL 的 Agent 和服务设施之间的正确通信和协调，而通信内容的语义与 ACL 无关。Agent 的任务求解模块包括 Agent 的运行模块及 Agent 任务相关的推理方法和规则。知识库为 Agent 所感知的世界和自身模型，并保存在移动过程中获取的知识和任务求解结构。内部状态集是 Agent 执行过程中的当前状态，它影响 Agent 的任务求解过程，同时 Agent 的任务求解又作用于内部状态。约束条件是 Agent 创建者为保证 Agent 的行为和性能而作出的约束，如返回时间、站点停留时间及任务完成程度等，一般只有创建者拥有对约束条件的修改权限。路由策略决定 Agent 的移动路径。路由策略可能是静态的服务设施列表（适用于简单、明确的任务求解过程），或者是基于可规则的动态路由满足复杂和

非确定性任务的求解。

服务设施提供移动 Agent 的基本服务(包括创建、传输、执行等),移动 Agent 的移动和任务求解能力很大程度上决定于服务设施所提供的服务。一般来讲,服务设施应包括以下的基本服务:

·生命周期服务:实现 Agent 的创建、移动、持久化存储和执行环境分配。

·事件服务:包括 Agent 传输协议和 Agent 通信协议,实现 Agent 间的事件传递。

·目录服务:提供定位 Agent 的信息,形成路由选择。

·安全服务:提供安全的 Agent 执行环境(在下一节中阐述)。

·应用服务:是任务相关的服务,在生命周期服务的基础上提供面向特定任务的服务接口。

2.2 关键技术

移动 Agent 利用先进的思想提供智能化的服务和任务规划求解,为实现这个目标,必须解决好以下关键技术:

2.2.1 *移动 Agent 理论模型* 目前一般基于 BDI 系统。BDI 系统,也称为意识系统,是把 Agent 看作理性主体,通过信念(Belief)、愿望(Desire)、意图(Intention)属性来预测 Agent 的行为。把主体看作意识系统的主要好处有:(1)对于设计者和分析者来说,这样是自然的;(2)对于描述复杂系统的行为提供了简洁的表示,有利于理解和解释;(3)不依赖于具体物理实现就可以得到许多主体的规则和模式;(4)可被 Agent 自身用来相互推理。

目前大多数学者利用模态逻辑理论来研究 BDI 理论模型,并得出了一些有益的结论。

2.2.2 *Agent 通信语言* ACL ACL 基于语言-行为理论(speech act),定义了 Agent 及服务设施间协商过程的语法和语义。移动 Agent 的 ACL 应具有应用的普遍性、简捷一致的语法和语义、通信内容的独立性等。目前常用的 ACL 有 KQML (Knowledge Query and Manipulation Language) 和 FIPA(Foundation of Physical Intelligent Agent)ACL,它们的格式非常接近,这里只讨论 KQML。

KQML 被分为三层:内容层、消息层和通信层。内容层包含消息的实际内容,KQML 可以携带任何语言表达的内容,包括表达为 ASCII 码或二进制代码的语言;通信层描述低级的通信参量,如发送者、接收者和与通信有关的唯一标识符;消息层是 KQML 语言的核心,它的主要作用是识别传输消息所采用的网络协议,给出发送者对内容的态度或意图,即语言行为或原语(performatives)。行为原语定义了可作用于 Agent 的知识库和目标库(KQML 支持基于 BDI 的 Agent 模型)的各种许可的操作,常用的原语有基本操作原语(Tell、Deny)、基于知识数据库的操作原语(Insert、Delete)、基本响应原语(Error、Sorry)、基本查询原语(Evaluate、Reply、Ask-If)、能力宣告原语(Advertise)、网络操作原语(Register、Forward)和协调器操作原语(Broker-One)等,开发者可以自己扩充 KQML 来实现既定系统。

目前虽然出现了一些 KQML 的开发包(如 KAPI、Jkqml 等),但 KQML 及其类似语言还有一定的局限性,它们缺少一个精确的语义系统,通信级上基本上无语义。

2.2.3 *Agent 传输协议* 定义了移动 Agent 传输的语法和语义,具体实现了移动 Agent 在服务设施间的移动机制。IBM 提出的 ATP 框架结构(ATP framework)定义了一组原语性的接口和基础消息集,可以看作是一个 Agent 传输协议的最小实现,其基本操作如图2所示。目前研究的重点是可靠而实时的传输。

图2 ATP 的示意图

2.2.4 *路由策略* 移动 Agent 的效率很大程度上决定于路由策略的优化。可行的路由策略有两种,分别为固定路由及基于规则和目录服务的动态路由。目前,在路由策略中引入 QoS(Quality of Service)是一个研究重点。

2.2.5 *系统性能影响及其测试工具* 当前的移动 Agent 系统虽然可以减少网络负载和克服网络延迟,但却增加了服务方主机的负载,基于可移植性和安全方面的原因,Agent 通常都是采用相对较慢的解释性语言,并且当到达目的地后,必须置入相应的运行环境中才能执行。这些原因使得移动 Agent 的执行速度低于普通程序。所幸的是,以 Java 为代表的即时编译(Just-in-time compilation)取得了很大的进步,使得移动代码的执行速度得以显著提高。

性能测试工具方面的研究目前还很不成熟,基本上没有好的测试工具。实际评估性能时,一般利用现有的理论如随机 Petri 网(把系统协议用随机 Petri 网形式化描述,然后等价成 Markov 链,利用 Markov 更新方程或者随机 Petri 网工具分析出系统性能)、着色 Petri 网。

2.2.6 *容错策略* 移动 Agent 系统必须考虑到移动过程中可能存在网络故障、服务设施故障、长时间停机等情况造成的移动 Agent 破坏和失败。常见的容错策略有:(1)创建相同任务的多个备份在网络中独立运行,在任务结束后比较结果;(2)集中式容错:特定的服务器保留移动 Agent 的原始备份并实施跟踪,通过重发原始备份恢复失效的移动 Agent;(3)分布式容错:将容错责任分配到网络中多个非固定的站点进行。

容错机制将是将来移动 Agent 系统性能的重要评价标准,也是移动 Agent 优势得以体现的重要手段。

2.2.7 *互操作性* 目前市场上移动 Agent 系统非常多,国外一些主要的厂商都分别推出了 Agent 系统,由于它们采用的技术及设计框架的差异,使得它们难于很好地协调工作。正是在这种背景下,OMG 组织制定了 MAF(Mobile Agent Facility)规范,随后改为 MASIF(Mobile Agent System Interoperability Facility),并于98年三月正式推出,它为解决不同厂商间 Agent 系统的互操作性提出了一些基本建议。随着移动 Agent 逐渐在智能领域里的应用,有必要在 MASIF 规范中体现智能性,实现和符合 FIPA 规范的智能 Agent 系统的互操作。MASIF 和 FIPA 的兼容,以及基于语义(底层)的互操作性是将来研究的重点。

2.2.8 *控制策略* 必须对移动实施有效的控制,避免移动 Agent 失控(如不停地复制,迁移等)。另外为了保证系统性能,引入负载均衡的机制很有必要。

2.2.9 安全机制 在第3节阐述。

3. 移动 Agent 中的安全

现有的基于 Java 的移动 Agent 系统基本上都采用了 Java 的沙箱(sand box)安全模型作为其安全机制实现基础,但是 Java 安全模型本身就存在不完善的地方,而且移动 Agent 系统对安全性有着特殊的要求,系统地进行移动 Agent 系统安全性研究,有着重要的意义。

移动 Agent 系统的安全涉及到三个方面:(1)移动 Agent 之间通讯的安全保护;(2)保护执行环境免受潜在的恶意 Agent 的损害(保护主机);(3)保护移动 Agent 免受潜在的恶意服务器和环境的攻击。下面对上述问题进行讨论。

3.1 移动 Agent 通讯的安全保护

为保护通讯安全,要对所有信息提供通讯认证,对任何可能的安全危害进行检测,目前有一种基于"加密信道--权限控制"的 Agent 系统通讯安全性实现方法,提供了多层次检查机制。用基于 RSA 和 Rabin 算法的加密信道来提供底层的签名和加密服务,而在高层提供权限控制机制 DSM。

3.2 保护主机

机器应当能够认证 Agent 的所有者,然后根据执行过程的资源请求和相关的安全策略给予相应的权限。为了防止破坏敏感数据和拒绝服务 DoS 攻击,资源的限制应包括读取某个文件的权限、总 CPU 的最大消耗时间等。目前的方法有如下几种:

·沙箱(sandbox):类似于 Applet 的安全机制,Agent 运行环境限制移动 Agent 访问本地资源的权限,就好象运行在一个特定的箱子中(称为沙箱),Agent 只能对运行环境施加有限的影响,破坏有限的资源。

·认证、授权:运行环境通过数字签名(每一个移动 Agent 都有自己的护照 passport,其中记录了移动 Agent 的创建时间、制造者等数字签名信息)来对移动 Agent 进行身份认证,判断它是否合法。如果通过认证,就通过存取控制表分配相应的资源操作权限给移动 Agent。

·检验-传输代码(Proof-carrying code):由卡内基·梅隆大学提出,核心思想是移动 Agent 的运行环境能够验证某个移动 Agent 是否遵守由运行环境提供的一套安全规则。

3.3 保护移动 Agent

与保护主机技术相比,移动 Agent 的保护技术仍处于起步阶段。目前主要有两种方法:

·基于检测的安全性:根据移动 Agent 的执行结果检测判断其是否受到攻击、破坏,这是一种事后的被动的方法。另外也可以进行事前检测,即不让移动 Agent 到不信任的运行环境上去运行,但要求提供一种检测和评价各个主机可信任程度的一种手段。

·主动的保护措施:基于检测的方式是被动的,只有主动保护才能更好地保护 Agent。虽然要让移动 Agent 在不信任的站点上完全安全是非常困难的,但是对这个问题的部分解决方案还是存在的。如 Stuttgart 大学的 Hohl 提出的黑匣子(Blackbox)(从一个给定的 Agent 规范生成执行代码,如果在任何时间内 Agent 不会受到攻击,且只有它的输入和输出行为被观察到,则一个 Agent 可被看作为一个黑匣子)算法,以及基于硬件的保护方案等。

当然,对于基于 CORBA 的移动 Agent 系统,可以利用 CORBA 的对象安全服务来实现系统的全局安全性要求,这里不再赘述。

4. 移动 Agent 系统组织及其规范

为了加快移动 Agent 技术的发展和推动移动 Agent 的具体应用,与移动 Agent 技术研究相关的各研究机构和企业,形成了一些标准化组织,展开了关于移动 Agent 技术标准化的工作。目前推动移动 Agent 标准化工作的最有影响力的国际组织有 OMG 下属的 Agent Working Group、FIPA 等。

OMG 的 MASIF 规范建议对 Agent 管理、Agent 迁移、Agent 以及 Agent 系统名称、Agent 系统类型以及位置语法标准化。定义了 MAFAgentSystem 和 MAFFinder 接口。其中 MAFAgentSystem 负责接收 Agent、列出 Agents、获得 MAFFinder 接口、获得 Agent 系统类型、获得 Agent 状态等。MAFFinder 提供注册、去消、查询等服务,它实际上为一名字服务。

FIPA 是由来自多个国家的活跃于 Agent 领域的大学和公司组成的非盈利组织,其宗旨在于"促进基于 Agent 的应用、业务和设备的成功"。目前 FIPA 制定了 FIPA 97、FIPA 98、FIPA 99等规范。FIPA 97提供了有关基本 Agent 技术的规范说明,包括 Agent 管理、ACL、Agent 软件集成、个人旅游助手、个人助理、音像娱乐及广播、网络管理及供应等七节构成。FIPA 97只研究了静态 Agent,FIPA 98则开始为移动 Agent 技术制定规范。

1999年3月,OMG 和 FIPA 正式成立了联络机构(OMG-FIPA 联络处),以协调两个组织关于 Agent 技术的工作。

值得一提的是,另外还有一些相关的组织,如 Agent Society(成立于1996年,主要目的是为了帮助与 Internet 相关的移动 Agent 技术和市场的开发)、CLIMATE(目标是协调有关移动 Agent 技术项目的研究、信息交换及合作)、DARPA、AgentLink、Active Group 在积极推动移动 Agent 技术的研究。

5. 移动 Agent 的应用及开发

电子商务:电子交易对股票价格等实时信息非常敏感,由于 Agent 降低了网络负担,很适合电子商务领域。

个人助理:移动 Agent 具有迁移到远地机器执行的能力,所以可以调度其它 Agent 来共同协商,完成会议。

安全代理:为了防止网络中的不安全性,程序可以先派其 Agent 去完成任务,而不至于太危险。

分布式信息检索:这个是移动 Agent 常见的应用领域,Agent 可以被派遣到远地去搜索信息,并创建搜索索引,以后再把索引返回到本地,免去了数据的大量传输。

电信网络服务:提供动态网络的高级管理。

工作流系统:移动 Agent 的自治性在此很好地体现。

另外还有并行处理系统、基于移动 Agent 的入侵检测系统、GIS 系统、移动数据库系统以及用移动 Agent 来求解一些数学问题等。

总之,Agent 特别适合于解决传统方法要么代价过于昂贵,要么就解决不了的问题,如数据、控制、专家知识或资源分布问题,使大量的数据处理可在数据源进行(因为 Agent 可以移动),只需交换少量的高层信息,减少了大量原始数据传送到远地的操作,提高了网络的利用率;如需要人性化的问题:Agent 具有观察能力、主动适应能力,而不是通过一些预

先严格确定的接口函数与外界进行交互作用，能根据目标主动规范化自己的行为，使用户界面达到"人性化"；如需要集成的问题：通过给旧系统上包装一层 Agent 外壳，其它系统可以调用旧系统的功能。

运用移动 Agent 技术开发应用系统一般有以下的步骤：

·分析系统的特点，选择合适的实现技术：决定哪些采用移动代理实现，哪些采用其它方法实现。

·移动 Agent 的功能设计：确定系统中决定用 Agent 技术实现部分的数据和功能，移动 Agent 间明确分工后，应当根据各自功能确定内部数据，注意，移动 Agent 的内部数据应尽可能少，减少由于移动带来的网络负担。

·Agent 的接口设计：Agent 的接口设计非常关键，它往往是系统性能好坏的关键。这时既要考虑系统中的 Agent 间交互方式又要考虑 Agent 与非 Agent 部分的交互方式。

·Agent 的详细设计：应当了解目前移动 Agent 平台各自的优缺点，选择合适的平台。

·Agent 的运行与维护：目前由于移动 Agent 可靠性还不高，维护工作特别重要。

总结和展望 移动 Agent 不同于基于过程的 RPC（如 OSF/DCE 中的），也不同于面向对象的对象引用（如 OMG/CORBA，OLE/DCOM 和 Java/RMI 中的），其独特的对象传递思想和卓越的特性给分布式计算乃至开放系统带来了巨大的革新。但 Agent 平台间的互操作还不是非常理想，容错性、可靠性、安全性还应当加强，面向 Agent 的软件工程也应该规范化。而且要正确理解面向 Agent 的方法，而不能有偏差。须知 Agent 方法不是万能的，Agent 的不成熟限制了它的应用领域，如整个系统不可预测，不可确定，哪个 Agent 将与其它哪个 Agent 以什么方式交互实现什么目标，这是不能事先确定的；而且由于 Agent 可以自由作出决策，我们不能保证 Agent 之间的依赖关系能得到有效管理，由此可能导致死锁；又如整个系统的性质和行为在设计阶段不能确定，整体行为只有在运行时才能体现。其二、Agent 受理论和硬件的限制。Agent 由于自身带有大量的精神状态，在某些情况下，性能反而不高。其三、在应用 Agent 方法进行具体设计时必须考虑异质性问题、不确定、不完备和矛盾信息问题、实时操作问题以及适应性问题。但我们相信，移动 Agent 技术将会日趋成熟，Internet 与移动 Agent 技术的结合将给远程教学、电子商务带来无限发展机遇，并在移动计算、信息服务等新兴应用领域里找到用武之地。

参 考 文 献

1 Joshi N, Ramesh V C. On Mobile Agent Architectures. avialable at: http://vcr.ece.iit.edu/papers/mobileAgents/mobile Agents.html

2 Cockayne W R, Zyda M. Mobile Agents. Manning Publications Co., http://flower.ce.cnu.ac.kr/~dkkang/mobile_Agent_books/HTML/Front.htm, 1998

3 张云勇. 面向 Agent 的软件工程，电子科技大学计算机学院博士生研究报告，2000年8月

4 刘锦德、张云勇. 一个实用的移动 Agent 系统(Aglet)的综述. 计算机应用

5 胡健. 开放式分布协作信息技术. 电子科技大学计算机学院博士论文，2000年10月

6 马俊涛，刘积仁. 移动 Agent 体系结构及关键技术探讨. 小型微型机系统，1998，19(2)

7 刘建勋、李仁发、张申生，移动 Agent 及其安全性问题，计算机工程与应用，2000年第7期

8 王济勇、温涛、李春光等，一个基于 Java 的移动 Agent 安全体系，计算机工程与应用，2000，(10)：141～144

9 董红斌、石纯一，移动 Agent 系统的安全问题，计算机科学，2000，27(10)

10 张云勇、刘锦德，移动 Agent 互操作性研究，计算机科学，2001，28(9)

11 张云勇、刘锦德，一种基于移动 Agent 的主动网的框架方案，计算机应用，2001，21(7)

12 印鉴、刘斌、邹胜等，用演化 Agent 方法处理整数线性规划问题，小型微型机系统，2000，(6)：608～610

13 王柏、王红漫、邹华，分布计算环境，北京邮电大学出版社，2000.8

14 Danny B. Lange, Mobile Objects and Mobile Agents: The Future of Distributed Computing available via http://www.generalmagic.com/asa/danny/Ecoop98.pdf

（上接第126页）

变量 Position 为该数组 ID，访问状态 AccessState 为1（指针指向块的最后位置）。

(12)将该图像数组信息写入图像数组引用链表。

(13)如果有待分析的语句，读入下一语句，转到3。

(14)算法结束。

结论 图像处理过程中包含大量独立的重复运算，因此作为提高图像处理速度的重要途径—并行图像处理一直是图像处理研究领域中的热点问题。本文在对当前图像数据分布研究的基础上，提出了图像数据有序覆盖和有序划分的概念，将图像处理过程中图像数据的访问描述为一组相似有序覆盖访问的过程。给出了基于不同数据空间的图像数据划分方法。在对 LS MPP 数据块访问地址的计算方法进行了深入分析的基础上，建立了实现图像覆盖块连续访问的两种模式，并详细讨论了其中一种模式的图像数据访问地址计算及数据访问实现算法。这些方法在面向图像处理的 LS MPP C 编译器中得到了实现。

参 考 文 献

1 沈绪榜. MPP 嵌入式计算机设计. 清华大学出版社，1999

2 Lee Cheolwhan. Global Optimization for Mapping Parallel Image Processing Tasks on Distributed Memory Machines: [Technical Report]. University of Califonia. 1997

3 Copty N. A Data Parallel Algorithm for Solving the Region Growing Problem on the Connection Machine: [Technical Report]. Syracuse University, 1994

4 Bevilacqua A. Parallel image restoration on parallel and distributed computers. Parallel Computing, 2000, 26(4): 495～500

5 Potter J L. Image Processing on the Massively Parallel Processor. IEEE Computer, 16(1), 1983

6 Ibrahim H A. Low-Level Image Analysis Tasks on Fine Grained Tree-Structured SIMD Machines. Journal of Parallel and Distributed Computing, 4, 1987

7 王晖，等. LS SIMD C 编译器的数据通信优化算法. 计算机科学，2001.9.

8 王晖. LS MPP C 编译器的研究与实现. 西北工业大学博士学位论文，2001.4