

# 开放系统中移动性研究

钟国祥

(重庆教育学院 重庆400067)

**摘要** 开放系统技术在信息系统的集成方面已得到了广泛应用。应用开放系统技术可以方便地开发、实现、升级和维护各种应用系统,降低构建应用系统的代价,并提高其集成效率。但是,当前的开放系统技术更多关注的是功能方面的问题(如可互操作性、可移植性等),而对于有些重要的性能方面的问题,如移动性的保证(可利用它来提高合作性,优化移动网业务)等,尚未涉及。然而近年来,移动性应用的范围却日趋扩大。这些应用系统都存在这样的要求,即以开放系统的方式集成,以缩短开发周期、降低构建系统的费用。因此产生了开放系统包容移动性应用的问题。本论文以此为目标,力求从理论和实践两个方面着手,探索开放系统中引入移动性的解决方案。文中分析了开放系统中的移动性,提出了开放系统中融入移动性的实现思路,给出了其基于中间件的概念模型。并对支撑技术中三大核心成分——分布式对象技术、新型计算模式及网络协议、智能合作技术进行了深入的研究。分析了当前中间件用于移动性领域中的局限性,给出了面向移动性的中间件的方案。

**关键词** 开放系统,互操作性,移动性,合作性,移动 agent,移动中间件,无线 CORBA, Agent 通信语言(ACL)

## Research of Mobility Issue in Open System

ZHONG Guo-Xiang

(Chongqing Education College, Chongqing 400067)

**Abstract** Currently, open system technology is widely used in the area of system integration. Applying this technology can develop, implement, upgrade and maintain application system conveniently, that causes the cost of building system to be reduced and its productivity to be improved. However, existing open system technology cannot support applications with some important performance requirements, such as mobility, which may be used to improve collaboration or to develop high efficiency mobile network business. But recently, the coverage of mobility application is greatly increased. These systems are required to be designed with open system technology in order to shorten the development cycle and to reduce the cost. Therefore, it is necessary to study how to integrate mobility-oriented application with open system architecture. The dissertation is devoted to showing the research on this subject from both theoretical and practical points of view. In this paper, according to the features of mobility-oriented systems and characteristics of open system, the influence that mobility behaviors impact on system's openness is discussed in detail. Based on the research, a conceptual model built on middleware is proposed. Then three core technologies, which are distributing object technology, modern computing model and network protocol technology and intelligent collaboration technology are focused. Limitations of current middleware in mobility domain are analyzed, and mobility-oriented middleware solution is given out.

**Keywords** Open system, Interoperability, Mobility, Collaboration, Mobile agent, Mobile middleware, Wireless CORBA, Agent communication language (ACL)

## 1 引言

随着计算机软硬件技术的飞速发展,网络技术的普及,客户/服务器技术、分布式技术和高性能 RISC 计算机的广泛应用,以及 PC 技术不断向高端领域发展,用户的应用环境变得异常复杂;许多组织有着种类繁多的硬件系统,包括:PC 机,各种类型的工作站,可能还有各种类型的大中型机,甚至近几年迅速出现的各式各样的嵌入式设备;同时,在这些硬件系统上,还运行着不同的操作系统和应用软件,依赖不同的网络结构;然而在很多情况下,却要求运行在这些异种平台上的应用之间协同地完成工作。用户环境的复杂性、多样性和多变性,导致了开放系统技术的出现。

开放系统已经成为当前计算机界中的一个流行名词,尽管它不存在精确的定义,但对它却存在着公认的“必具特征”<sup>[1]</sup>,它们是:1)具有可移植性(Portability);2)具有可互操作性(Interoperability);3)具有可伸缩性(Scalability);4)具有

易获得性(Availability)。

开放系统技术在信息系统的集成领域已经得到了广泛应用。应用开放系统技术可以方便地开发、实现、升级和维护各种应用系统,大大降低了构建应用系统的代价,并极大地提高了集成效率。

文[1]在澄清了开放系统的内涵后,为其建立了一个较为实用的工作定义:对于一个计算机环境,通过对其轮廓(AUSPICE)选用适当的标准并配以相应的软件(与硬件),使其能实现应用软件和使用者可移植性,实现异种机网络内多节点间的互操作性,以及实现全局性的安全性,则此环境就进化为一个开放系统。

由于开放系统短暂的历史(计算机领域出现开放系统的名称只有十多年的历史)和多变的有关技术环境(用户的用机条件从单机变为局域网,又从局域网变为广域网;应用从简单的单机应用变为网上应用,特别是网上多媒体应用),使开放系统的内涵迅速地变化和丰富。以下几种发展趋势促使我们

有必要对开放系统中的移动性进行深入的探讨。

### 1.1 无线、移动网络的发展

从全球通信的发展趋势看,通信网络的发展趋势之一是无线化。根据 IDC、ITU、MII 的资料统计,1990~1999年全球电信平均业务增长率为10%;干线电话平均增长17%;移动电话平均增长率49.2%;移动电话及互联网用户的增长速度远远高于固定电话的。

当前,全球运营商实现增值业务的移动智能网技术正在成为热点。据英国 ARC 集团预计,在未来5年(从2000年起)内,7.5亿人将使用移动终端,6.7亿人使用有线。一个值得注意的趋势是:移动数据通信的发展将使移动数据业务从通信业务发展为信息服务业务,而这种变化必须得到高性能的开放系统体系结构的支持。

### 1.2 信息组织模式的进步

信息组织模式正在逐渐进化。明显的例子是无所不在的计算(ubiquitous computing 或 pervasive computing)的出现。这一模式将打破传统意义上的计算机的概念,即大量使用更易移动的各种具有计算能力的设备,并且在用户需要时可以随时将其连接到信息网络中,使用户可高效地交换信息和获得各种信息服务。

同时深入分析现有 Internet 信息组织模式和其技术基础,可发现确实还存在着一系列亟待解决的挑战,将它们可概括为以下几点:

1) 信息提供者和信息需求者之间缺乏有效的沟通手段。从而使 Internet 上的信息的及时性难以得到保证。

2) 大量存在于现有信息系统中的信息,难以有效地通过 Internet 加以利用。由于缺乏方便的有效技术手段,使得大量的现有系统难以与 Internet 有机结合,进而发展新的基于 Internet 的业务需求。

3) 移动计算设备的出现和其广阔的应用前景,使现有的 Internet 所采用的计算模式受到挑战。宽带主干网与带宽有限且费用昂贵的接入网之间的矛盾,难以有效得到解决,对于移动设备,这一矛盾更为突出。

深入进行信息合作技术和新一代分布式计算模式的研究,乃是解决以上挑战、取得进步的有效途径,而且这两项技术的研究必须紧密地结合在一起进行。前者从信息需求和使用的角度,研究如何高效地组织和利用信息,以克服 Internet 现有的信息组织和利用模式的缺陷;后者从既要允许远程分布而又要具有紧密结合的角度,研究如何解决目前所常用的分布式模式的不足,以满足对新一代开放式计算环境的要求。

### 1.3 移动计算扩大,大规模分布式移动系统的涌现

移动应用领域的不断扩大,促使大规模分布式移动系统日益涌现和广泛发展。电信<sup>[2]</sup>、交通、作战指挥等信息系统都是常见的大规模分布式移动系统。这些庞大的移动系统除了面临移动运行的问题外,同时也面临其它任何大规模软件系统所遇到的问题,如开发困难、维护困难等。传统的移动系统的规模通常较小,它一般都是经过精心定制以便能在专用系统上运行。虽然这类定制的系统为移动应用的运行提供了所需条件,但是它也创建了一个不灵活的结构,存在不容易修改、升级和与第三方产品集成的缺点。用这样的方法来原因上述庞大的移动系统的开发、实现和维护,其代价极大。唯一的解决办法是使用开放系统技术,发展面向移动计算的中间件及其软件工程,以解决开发大规模分布式移动系统所面临的问题。

## 2 研究方向和现状

### 2.1 分布式中间件

分布式面向对象中间件将面向对象的思想用于分布式计算环境中。这种中间件实现了真正的通用软件总线,具有优良的互操作性和应用程序集成能力。这些应用程序可以位于网络的任何地方,彼此实现透明协作,即使是向不同供应商购买的产品也可以协同工作。分布式对象中间件可以采用的标准和规范有:ISO、IEC 和 ITU-T 联合制定的国际标准 RM-ODP<sup>[3]</sup> 和 OMG 制订的规范 CORBA<sup>[4]</sup>,非规范的有 Microsoft 公司的 DCOM 和 Sun 公司的 RMI。就目前实际应用的情况来看,RM-ODP 主要对其它规范起指导作用,起着元标准(即标准的标准)的角色。CORBA 在市场的占有率最高,究其原因它是技术较为成熟、支持的厂商多,用户易于为自己的平台找到适用的产品。DCOM 则主要在 Windows 平台上使用。虽然存在不同的分布式面向对象中间件系统,但是,(1)它们只支持 TCP/IP 协议,对 WAP、蓝牙等新兴协议并不支持,从体系结构上讲,它们都属于传统的 C/S 结构,因为需要频繁交互,在低带宽网络中效率低下。(2)同时分布式计算环境从原来仅仅适用于科学计算、数据交互、信息检索、远程控制等应用,发展成为适用于各种类型应用(如实时计算、移动计算等业务)的通用平台,这些变化使得原来的分布式面向对象中间件系统的内涵需要进一步丰富,所以适用于新一代移动计算的分布式对象中间件系统的研究被提上日程并开始变得十分重要。

### 2.2 移动 agent 方案

为了解决高性能和智能合作问题,有学者采用了移动 agent 技术<sup>[5]</sup>。移动 agent 的概念是20世纪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。

移动 agent 方式与 Client/Server (RPC) 方式的区别之一在于:在 C/S 方式中,客户方的请求和服务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交互。而采用移动 agent 方式时,请求服务的客户方 agent 移动到服务方所提供的 agent 上下文环境中,客户 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,客户 agent 把处理结果传回给客户方,它本身则可在服务方上下文环境中消亡,而不必移回客户方。此外,传统的 RPC 客户与服务器间的交互需要连续性的通信支持;而移动 agent 可以迁移到服务器上,与之进行本地高速通信,显然这种本地通信不再占用网络资源,也即允许不连续性通信的支持。总之,利用移动 agent 所形成的是一种高性能的计算模式。

移动 agent 本质上是 AI 与分布式计算技术的混合结晶。AI 的基因使移动 agent 具有智能性,实践表明我们可以利用 agent 通信语言来解决智能合作问题。

值得注意的是,单纯采用移动 agent 方案,属于另起炉灶性质,这将面临以下的棘手问题:一、不同厂商之间的移动 agent 系统,由于设计的差异,并不能互操作;二、不易集成遗留系统。所以,这种做法并不实际。

### 2.3 专用协议方案

针对移动网络中无线信道的突发性错误、较大的延时变

化,以及移动主机的越区切换,使得传统的 TCP 拥塞控制和丢包恢复机制并不能有效适应无线移动环境。国外研究网络的学者,提出各种新型协议,如 I-TCP<sup>[6]</sup>和 M-TCP<sup>[7]</sup>等来提高移动网络性能。但这些协议不易兼容遗留程序,而且系统成本较高。

以上三者从不同的角度来探索对付系统中引入移动性的办法,但它们都不够全面,而且也没有上升到从开放系统的高度来对付移动性这一问题。但是这些前人的工作也为我们启示了“在开放系统中包容移动性”的研究思路,即中间件可能是解决开放系统中移动性的一条有效途径,但是它必须辅之以新的计算模式——移动 agent;这样既可利用移动 agent 支持高性能计算和智能合作计算,又可利用中间件解决互操作性和集成遗留系统的特长。

### 3 面向移动性开放系统的相关技术

正如前面所述,开放系统中支持移动性(利用移动性来达到高度合作,利用移动性来达到性能优化)面临很大的困难。为了解决此困难,这里论述了几种可能用到的技术。

#### 3.1 新型计算模式及网络协议

这方面技术中的一类主要用来解决移动环境所带来的低性能等问题。1988年 Van Jacobson 指出了 TCP 在控制网络拥塞方面的不足,并提出了“慢启动”(Slow Start)、“拥塞避免”(Congestion Avoidance)的算法。1990年出现的 TCP Reno 版本增加了“快速重传”(Fast Retransmit)、“快速恢复”(Fast Recovery)算法,避免了网络拥塞不严重时采用“慢启动”算法而造成过大地减小发送窗口尺寸的现象。这样 TCP 的拥塞控制就由这四个核心部分组成。最近几年又出现了 TCP 的改进版本,如 New-Reno、SACK 等。但这些协议需要修改传统的 TCP/IP,兼容性不好,不易融入现有的系统中。

这方面技术的另一类则是通过新型计算模式来达到高性能,如采用移动 agent。它不需要修改传统的 TCP/IP,兼容性好,易于融入现有的系统中,而且能按自主意愿将代码和数据迁往它处执行,高性能地完成任

#### 3.2 开放系统技术

开放系统技术作为系统集成的主要技术,有着丰富的内容,如系统管理、用户界面、安全性、编程服务、互操作服务、通信服务和信息实体(即所谓的开放系统轮廓 AUSPICE 的七个方面),各有其关注的对象。实现一个开放系统,通常是首先根据本系统或本单位的实际需要,修改这一通用的轮廓,据此为各成份域选择相应的标准。有了这一具体化的标准集,就可以按此选购相应的软件和硬件。将符合标准的各成份集成,组成所需的开放系统。

在实现开放系统时,较易解决的是可移植性问题,因为在操作系统方面,POSIX 和 X/Open 等已经作了大量标准化工作,厂商正开始按此开发并提供着产品,这使人们对轮廓选用标准时所遇的障碍减少。一个完善的开放系统的建立,往往并非依靠“弃旧立新”的策略,而更重视“建新蓄旧”的做法,也就是新旧应用组建共存,并且它们能够协同工作,这使得要实现一个高层次的开放系统,必须解决可互操作功能,后者可以依靠定制或现成的可互操作中间件来实现。而与此相关的技术中,以 ISO 和 ITU 的 RM-ODP 标准最为全面,最具有指导意义;而 OMG 的 CORBA 规范是 RM-ODP 在现实中的简化和妥协的结果,因而最具有现实意义。

开放系统技术提供了分布、扩展和联合等功能。一个开放分布式系统环境提供了多种分布式透明性机制来支持多个应

用成分的透明协同工作。分布式透明性屏蔽了由系统的分布所带来的复杂性。分布式透明性使应用编程者不必关心系统是分布的,还是集中的,从而可以集中精力设计具体的应用,这大大减少了分布式编程的复杂性。另外,应用编程者可以选择自己所需要的透明性。例如:RM-ODP 定义了八种透明性:访问透明性、位置透明性、迁移透明性、失败透明性、重定位透明性、复制透明性、持久透明性、事务处理透明性。

RM-ODP 作为开放分布式处理的一个国际标准,由于制订这一标准的起点高,面向的对象大,所具备的功能全面,对其它相关的规范起着重要的指导作用,成为标准的标准(即元标准)。而 OMG 的 CORBA 规范则成为事实上的应用标准,它已经被用户广泛接受。CORBA 为应用系统提供了大量的好处:开发简单性、可互操作性、灵活性、可维护性、可重用性等。

#### 3.3 智能合作技术

其中一类为传统的智能技术,如传统的人工智能、知识工程,但是它们不易集成到开放系统;另一类则来自于分布式人工智能领域,如移动 agent,它既有分布系统中通信的能力,又有智能系统中的合作能力。

以上三种独立的技术并不是互斥的,开放系统中移动性的实现方案就是要设法有机结合以上三种技术。首先从中选择一个最适于担任集成载体的技术,随后精选其余的技术在此载体上进行扩展,这将是有效的结合途径。

### 4 面向移动性开放系统的概念模型

随着移动应用的日益广泛使用,移动系统之间,甚至移动系统与非移动系统之间,必然会出现更广泛的协同工作的需求。特别是,由于各种原因,如系统的功能、性能及系统的成本等,这些系统会采用多种技术。这就会造成应用系统之间,甚至同一个应用系统内,存在很大的差异。这时如果使用层次较低的通信协议(如 TCP/IP)来实现协同工作,其开发代价较大。应用开发者不但要关注具体应用的问题,更是要花费大量的精力去了解下层平台的特性,并解决所处平台之间的差异。

而在当前的桌面和企业应用系统中,这样的协同工作问题已经得到很好的解决,那就是使用一类支撑性软件——中间件(有人也称之为软总线 software bus)来屏蔽平台、网络之间的差异,为应用之间的协同工作奠定了基础。通过中间件,应用系统的对象能达到透明合作的效果。这样的应用系统,具有易于开发、易于维护、易于升级的特点。为此,我们的思路是:在中间件的基础上来实现开放系统的移动性。当然,这一中间件应有其自己的特殊性,为此我们称之为面向移动性的中间件,或简称为移动性中间件。

**结束语** 随着技术的提高和应用发展的需要,高度合作和性能优化正在同时成为开放系统中的重要的内涵之一。当前人们对如何将“开放”与“合作和性能优化”相结合的研究,尚处于起步阶段。本文通过对相关技术的分析,提出了利用移动 agent(借助其高性能计算模式)和开放系统技术(借助其中间件的软总线)来达到性能优化,利用智能合作技术(借助移动 agent 的合作能力)和开放系统技术(借助其中间件的互操作能力)来达到合作。最后给出了开放系统中如何融入移动性的概念模型。

### 参考文献

- 1 刘锦德,唐雪飞. 开放系统中互操作技术的发展和前景. 计算机科学, 2000, 27(10): 27~31

(下转第147页)

析等特点,效率与原始的事务监控器 XATMI 库程序有较大的差距,必须针对特点做相应改进。表1是在 ISTX2上的 ToUpper 服务两种调用方式的延迟时间对比,ToUpper 服务用于将参数字符串改为大写。SOAP 客户端采用 Delphi 编写,广域网环境是通过局域网模拟得到,每次调用加上200~500毫秒的随机延迟,为减少误差,取循环运行100次的总时间。

表1 两种调用方式延迟时间对比

延迟时间(单位:秒)	SOAP	XATMI
局域网	5.4	0.2
广域网	39.7	34.8

SOAP 调用方式下,整个延迟时间  $t = \text{SOAP 处理时间} + \text{通信延迟} + \text{服务处理时间}$ ,比例系数  $p = (\text{SOAP 处理时间}) / (\text{SOAP 处理时间} + \text{通信} + \text{服务处理})$  代表了 SOAP 处理在一次调用延迟中占用的比例。

具体到本例,局域网环境下,由于实验中采用的 ToUpper 是种计算量极小的服务,通信延迟也极小,经测量该比例系数约为0.91,绝大部分时间消耗在 SOAP 解析上,导致延迟时间远远大于 XATMI 方式。广域网情况下,由于通信代价的增大,该比例系数急剧变小,在本例中约为0.12,两种方式的性能差别就很小。比例系数越小,对 Web 服务调用方式越有利。

我们无法改变 SOAP 处理时间和通信延迟,但可以通过增大服务的粒度来降低比例系数。而事务监控器作为传统的交易处理环境,存在大量细粒度、计算量小的服务。所以在引入 Web 服务后,除了 TP-WS 本身的优化外,事务监控器必须根据 SOAP 协议的特点做相应的优化,如谨慎选择公布服务的类型,合并服务以增大粒度等,才能达到较满意的效果。

## 5 相关工作

目前市场上应用较广泛的事务监控器系统有 BEA 公司的 Tuxedo,IBM 公司的 CICS,Encina 等。目前它们对 Web 服务的支持均有不足之处。

Tuxedo 8.0 以前不支持 Web 服务<sup>[9]</sup>,在2003年发布的最新版本 8.1 中宣称支持 Web 服务,但是其支持是通过外接一个 J2EE 服务器(Weblogic)来实现的,从 Web 服务接口到真正的 Tuxedo 服务调用要经过 WebLogic 和 WebLogic Tuxedo Connector(WebLogic 到 Tuxedo 的连接程序)<sup>[7]</sup>,这样一方面非常低效,另一方面,如果需要使用 Web 服务,必须再另外使用 WebLogic 服务器,很大程度上增加了用户的成本负担。

IBM 提出一套解决方案<sup>[8]</sup>,即“SOAP for CICS”来使得部署在 CICS 上的应用可以包装为 Web 服务。SOAP for CI-

CS 的传输层可以使用 HTTP 协议或 IBM 消息队列 MQ Series。截至2003年4月份,该方案仍处于调研的阶段。而且目前只支持 OS/390,z/OS 这两个操作系统平台上的 CICS Server 版本,不够完备。

Encina 由 Transarc 开发,后被 IBM 收购,其开发工作早已停止,不支持 Web 服务。

目前这些传统的事务监控器的解决方案都是与对应厂商的产品具体相关的,不具备通用性和可复用性。

**小结** 事务监控器是开发部署运行分布式应用的重要平台,在银行、证券、航空等关键行业得到广泛应用。随着新的应用需求的发展,传统事务监控器暴露出很多问题,如不能部署基于广域网的应用,对 Internet 环境下的多平台和多语言支持不够,不能与其他分布式系统互操作等。Web 服务是近年来提出的一种分布式应用组件标准,如果事务监控器支持 Web 服务则可以解决上述问题。然而迄今为止,主流的事务监控器在 Web 服务的支持方面均显不足。我们提出了一个 Web 服务支持框架 TP-WS,它可以移植到任何符合 X/Open 标准的事务监控器上,目前该设计已经成功应用于我们自主开发的事务监控器 ISTX2.0 上。

TP-WS 应用于事务监控器后,事务监控器就可以融入面向服务的体系结构(Service-Oriented Architecture)中,作为服务提供者和服务请求者出现,可以解决引言中提到的三个问题以及其它一些新问题。但事务监控器与 J2EE, CORBA 等其它分布式系统的互操作目前还只是一种浅层次的互操作,这其中的事务上下文和安全上下文的跨系统传播等,还有待进一步的研究。

## 参考文献

- 1 Cray J, Reuter A. Transaction Processing: Concept and Techniques. Morgan Kaufmann Publishers, Inc. 1993
- 2 Bernstein P A, Newcomer E. Principles of Transaction Processing. Morgan Kaufmann Publishers, Inc. 1997
- 3 Apshankar K, Ayala D. Professional Open Source Web Services. Wrox Press Ltd, 2002
- 4 X/Open Company Ltd, Distributed Transaction Processing: The XATM Specification, 1991
- 5 X/Open Company Ltd, Distributed Transaction Processing: The TX Specification, 1991
- 6 Apache Group. <http://ws.apache.org/axis/index.html>
- 7 BEA. <http://edocs.bea.com/tuxedo/tux81/overview/webaces.htm>
- 8 IBM. <http://www-3.ibm.com/software/http/cics/soap/>
- 9 Andrade J M, et al. The Tuxedo System: Software for Constructing and Managing Distributed Business Applications. Addison-Wesley Publishing Company, 1996
- 5 Cockayne W R, Zyda M. Mobile Agents. Manning Publications Co., <http://flower.ce.cnu.ac.kr/~dkkang/mobile-agent-books/HTML/Front.htm>, 1998
- 6 Bakre M, Badrinath B. I-TCP: Indirect TCP for mobile hosts. In: Proc. 15th Intl. Conf. on Distributed Computing Systems, Vancouver, Canada, 1995
- 7 Brown K, Singh S. M-TCP: TCP for mobile cellular networks. ACM Computer Communications Review, 1998, 27(5)

(上接第143页)

- 2 Breugst M, Magedanz T. On the Usage of Standard Mobile Agent Platforms in Telecommunication Environments. available via [www.fokus.gmd.de/research/cc/ecco/climate/intro-climate-cluster.html](http://www.fokus.gmd.de/research/cc/ecco/climate/intro-climate-cluster.html)
- 3 ISO/IEC IS 10746-1 | ITU-T X.901, ODP-RM Part 1: Overview, 1995
- 4 OMG. CORBA 3. Specification. <http://www.omg.org/>.