

分布系统与分布对象计算

蔡希尧 刘西洋 边定平

(西安电子科技大学软件工程研究所 西安 710071)

TP338.8

摘要 In this paper, the necessity of the development of distributed systems, the state-of-the-art, the new requirements, and the future trends are discussed, the problems faced after the introducing of objects into the distributed systems are analyzed, some possible solutions of the distributed object computing is given. At last, the authors give their own solution on distributed object computing with inheritance.

关键词 Distributed system, Distributed object computing, Inheritance.

一、分布系统——发展的必然

分布系统是当前计算机系统发展的主要模式。以计算机网络为依托,把多个同时工作的分散计算单元、不同的数据库、不同的操作系统联结成一个整体的分布式系统,为多种用户提供一个统一的工作环境。这种分布式系统能够把系统的吞吐能力提高几个数量级,同时提高了系统的生存能力和适应能力,满足多种应用的要求,是当前许多现代化企事业的运行所依靠的主要工具,也可以说是它们赖以发展和生存的重要基础。

分布系统不是一种选择,而是发展的必然。有的组织,如银行系统、连锁商店等,本身就是分布的,自然要求分布式计算;工业生产的体系结构从树形变为网状、贸易的全球化,一个世界范围市场的形成,对分布系统的发展起了强大的推动作用;另一推动力量是人们对资源共享要求的普遍化,如工程设计队伍的合作、图书情报的远距离使用、医疗会诊、个人间的协作交流等等。集中式系统不能反映人们的这些要求,也不能与生产和经济格局的改变相适应,只有开放的、可互操作的、多平台的、多应用的、多数据库、多语言的分布式计算环境才能最终满足各种用户的要求。因此,从用户的应用角度来看,可以把分布系统定义为:“在一个允许的范围,任何用户只要具有关于所用系统的最基本的知识,就可以在任何地点、任何时候,以应用目的可接受的响应时间,访问并使用系统中的任何资源,得到所需要的服务。”

从技术的角度来看,则可以定义为:“一个分布式计算系统是由多种计算资源、以一定的互联方式

组成的、开放式的、多平台的、可互操作的、合作的系统,能够为用户提供一定范围的服务。”

国民经济信息化的普及、企业结构的改变、工程方法的更新(如并发工程)、管理效率的提高等等,正是在具有资源共享和多种服务的分布计算系统的支持之下发展起来的。近年来,便携式计算机和“个人数字助理”(PDA)的迅速推广和无线计算机网络的兴起,以及应运而生的“虚拟办公室”的出现,使得分布计算更具有活力,应用的范围更加扩大普及。

二、分布计算的现状与发展趋势

一个分布计算系统要解决两个最基本的问题:(1)逻辑互联方式;(2)系统的互操作性,当前的技术主流如下:

1. 逻辑互联方式

一个集成的、大规模的、基于网络的分布式系统的首要问题是采用什么逻辑的互联方式以实现进程间的通信。目前发展中的主要互联方式是客户/服务器模式^[1,2]。这是一种典型的以服务为中心的計算方式,把分布式计算系统组织成一个具有多个分布的服务进程的系统,根据请求,通过网络为用户提供所需要的服务。

在客户/服务器模式中,有一个或多个客户和一个或多个服务器,用户通过客户进程使用系统,得到服务。服务器进程的任务是为客户提供服务,它们只响应客户进程的查询或命令,完成所请求的服务,并返回结果。客户与服务器之间有时候加一个中间环节,称为“服务器中介者”(server broker),简称中介者,它为所有的平台上可用的服务、或通过某个平台能够得到的可用服务提供一个接口,还可以向其他

中介者提供一个特定调用的历史信息,以及在不同服务器之间的协商。这样,一个客户/服务器系统的一般组成可以用图1来表示:

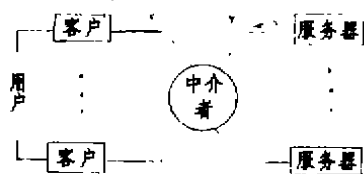


图1 客户-服务器系统的一般结构

客户/服务器系统的工作情况与原来的 PC-LAN 有明显的不同之处。在原来的系统中,用户的查询或命令是由 PC 机自己来处理的,这就要求 PC 机配以较大的存储容量;当应用需要文件服务时,要通过网络把文件取到 PC 机的内存,因此要求网络有较大的数据传送能力。在客户/服务器系统中,当客户响应用户的请求时,把请求翻译成标准的查询语言送给服务器,一般不需要客户自己处理;服务器按请求经过处理,把结果送回。这样的合作处理,克服了 PC-LAN 的上述缺点,更好地利用了系统资源,降低了开销,并且便于移植和维护。支持这种合作的基本机制是“远程过程调用”(RPC)^[3~5],这是异种机组成的分布式处理的事实上的通信标准,使得一个客户进程可以调用远程过程为自己服务。Sun 工作站采用的 ONC 和 Apollo 工作站采用的 NCA 是两种典型的广泛使用的 RPC 结构。

在应用中发现了客户/服务器工作方式一些有待克服的缺点,主要在远程管理和安全防护的可靠性等方面,此外,降低成本,提高效率,也是值得注意的问题。

2. 系统的互操作性

分布系统是异种技术(不同结构的子网、不同的操作系统、不同的通信协议等)构成的,但又彼此提供服务,因此,必须解决互操作问题。我们说两种或更多种资源是可以互操作的,是指它们能够交互地共同执行任务,互相提供服务。为了做到这样,系统的设计及实现必须服从开放式标准。开放式是当代计算机和通信技术发展的主要潮流,是在异构信息系统之间实现可移植性和互操作性的基本策略,它的基础是通用的标准。

已经制定了一些支持分布系统互操作的标准,有的则正在加紧进行之中。国际标准化组织 ISO 所制定的有名的 OSI 开放式系统互连,是网络的开放

式标准;IEEE 制定的 POSIX,是应用程序与操作系统之间的标准接口。OMG 所制定的一个基于对象模型的客户/服务器标准 CORBA^[6],采用中介者作为客户与服务器之间的接口,其关键部分是“对象请求中介者”的核和接口。服务器是一个对象,对象则是一个密封的实体,能够提供一种或多种服务。

内容最丰富的分布式计算环境是 OSF 所开发的 DCE^[7],被称之为“计算的联合国”^[8]。OSF 同时开发另一个分布环境 DME,用以统一异种环境下的系统管理。DCE 基于三种分布计算模型:客户/服务器、远程过程调用、和数据共享;最后一项分两部分:分布式文件服务和目录服务。DCE 的开发建立在五个方面考虑的基础之上:(1)提供分布应用的工具和服务。(2)服务是集成的、综合的,不是就事论事。(3)为异种平台提供可互操作性和可移植性;(4)支持数据共享;(5)参与全局的计算环境,如全局的命名服务和域名服务。

ISO 和 CCITT 正在联合制定的一个标准,称“开放式分布处理”(ODP)^[9],其目的是为宽范围的分布系统提供一个参考模型,使得整个系统能保持一致。ODP 包括四个部分:(1)概论及参考模型使用指南;(2)描述模型:定义概念和分析框架,以及分布处理系统规范描述的记法;(3)法规模型:鉴定分布处理系统是否开放的特征说明;(4)结构语义:给 ODP 在描述模型中定义的基本造型和规范概念以形式化,提供规范技术的需求,评估标准化的形式描述技术是否符合需求。ODP 的参考模型已于 1987 年开始制定,四个部分的草案于 1994 年提交。

目前网络发展很快,为以网络为依托的分布计算的发展提供了条件,但分布计算软件的发展不能适应需求,因此,不能充分地利用网络资源,更好地发挥分布计算的作用。应用要求尽快解决这个问题,并且要求分布计算做到:

(1)解决方案必须是开放的,任何部分都是可替换的和可修正的。

(2)必须坚持标准,用标准的部件使之可以重用,接口也必须服从标准。

(3)实时应用的分布系统,必须能够适应系统的演变。

(4)语言中立。

(5)原有系统的码和数据必须能尽量继续使用。

(6)给系统管理以充分的支持。分布系统的管理比集中系统要复杂,需要开发有效的工具。

(7)提供共同的标准服务和协议以支持应用集

成。

要达到这些要求,计算机企业界的共同结论是,唯一有效的解决办法是采用面向对象技术^[10]。发展的趋势将是在分布系统中引入一种可分布的、可互操作的对象机制,使得不同的对象和非对象应用可以集成在一起;要基于主流的对象环境,以二进制方式提供类库,不需要重新编译或重新连接客户。分布计算机软件的开发要以对象部件为基础;管理也要基于类库,而且是可剪裁的。

三、分布对象计算

把对象引入分布系统是很自然的。对象是一个自备的具有离散特点的实体,对象之间采用消息传递作为唯一的通信方式,特别适合于分布应用。但是,分布式的对象应用却不容易做到,特别是继承性的使用。通常的面向对象程序设计语言(OOPL)都是采用指针作为程序构造的基本设施的,在分布的情况下,不同机器之间的能见度低,使用指针很困难;语言一般缺乏对分布程序的足够的语义支持;分布环境下对于对象的有效管理也是件不简单的事情。正是由于这些困难,所以在面向对象技术的发展过程中,不少人把对象及继承性在分布系统中的应用视为难题,有的则知难而退。例如欧洲的 Esprit 计划中,为并行处理机 DOOM 而设计的面向对象语言 POOL,开始采用了继承性,由于困难较大,在第二个版本 POOL-T 中放弃了继承性^[11]。在研究分布对象计算的过程中,提出过排除继承性的其他一些机制,如代理(delegate)^[12]和演员(actor)^[13],它们的特点是灵活,但控制权要在代理者或演员之间不断转移,这给系统的管理带来极大的困难;另外,都缺乏语言的有力支持,未能流行。因此,在大多数采用对象为分布计算的基本构造的系统中,大多以基于对象的为主,而不是面向对象的。

我们认为,在分布对象计算中引入继承是必要的。分布系统一个基本的要求是资源共享,数据库系统能够提供的是静态的资源共享,而应用同时要求有动态的资源共享,解决分布系统的动态资源共享的主要措施是对象的继承机制。异构分布环境下的继承机制是跨越不同的体系结构、不同的操作环境、不同的网络协议、不同的地址空间的层次式动态资源共享系统,它的引入,保证了对象系统的完整性。

现在已经提出的解决方案大体上可以分为二类:

1. 基于程序设计语言的分布对象计算:这主要

是把现有的面向对象语言加以扩充,如 Smalltalk 扩充为 Distributed Smalltalk^[14],C++ 扩充为 DC++^[15]和 COOL^[16],或设计新的语言,如 DOWL^[17]。这种解决方案的局限性较大,推广特定的语言是不容易的事情。

2. 基于操作系统的分布对象计算:80 年代中期以来,陆续出现过一些基于对象或面向对象的操作系统,以支持分布计算,如 Amoeba、Nexus、Eden、Argus、Clouds 等,我们在文[18]中对它们有较详细的介绍。最新的操作系统如 Windows NT Cairo, Pink, WorkPlace 等,采用面向对象技术,有丰富的应用程序接口(API),可以更好地支持分布对象计算。这类新的操作系统采用一些共同的新技术,包括:(1)微内核化(只有存储管理、进程间通信等),其他功能分成模块,按用户需要提供;(2)装入“操作系统个性”功能,可仿真运行其他操作系统的应用程序,以 WorkPlace 为例,可仿真的应用程序所用的操作系统包括 DOS、Windows、UNIX、AIX、OS/2、Mac、Windows NT、和 Pink 等。(3)有用于服务器的能力。(4)适用于 CISC 和 RISC 芯片,适用于单处理机系统,也适用于多处理机系统,因此,具有很强的伸缩性;(5)采用面向对象技术,如 Windows NT Cairo, Pink。

3. 定义统一的对象模型,以支持分布计算:前面提到的 DCE、DME、ODP、CORBA,以及 Microsoft 公司开发的 OLE、COM 和 IBM 公司开发的 DSOM 等。

在以上三类方案中,第三类比较容易达到前面提出的分布对象计算的七点要求,是目前几个大计算机公司普遍采用的发展策略。

四、一种新的分布计算方案

我们在分布计算的研究工作中,提出一种新的含有继承机制的分布计算方案,其特点是:

1. 采用系统镜象的概念以实现分布继承

这一概念是在 RPC 的启发之下形成的。RPC 的最大优点是:应用通常的过程调用语义,自然而直接地表达并实现分布式应用,最大限度地屏蔽体系结构、网络协议、操作系统等的细节。但现有的 RPC 工业产品对类型系统的表达能力不足(如不能把函数作为参数使用),因而难以完整地实现过程调用的语义。调用者和被调用者的镜象关系如图 2 所示。

2. 引入接口继承

继承性依照其语义特征可分为基于代码共享的

继承和基于概念共享的继承。通常的 OOPL 所提供的继承主要是代码共享,是操作(行为)的重用,不是概念的重用,接口是一个对象的外特征,是概念实现的提供者和它的客户之间的契约,描述所提供的服务;接口继承是概念的共享。引入接口继承为异构分布环境下继承机制的实现提供了方便。

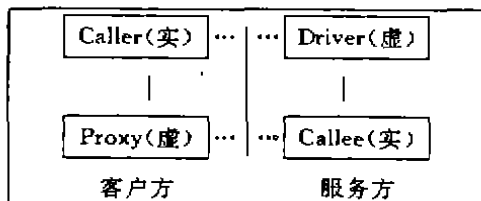


图2 镜像关系

3. 原型

在以上两个概念的基础上,我们开发了一个分布对象计算系统的原型,所用的环境是 Windows NT 3.1, Visual C++ for Windows NT, Borland C++ 4.0, 证实了概念的正确性。

有关这一新方案的详细情况,将另有专文论述。

五、结束语

自主和协作(独立和共享),是分布系统的一对内在的矛盾。在分布系统的概念形成的初期,比较强调自主,在集中处理占统治的情况下,这是比较自然的具有革新意义的想法。但实际上人们需要分布系统,更主要的目的是资源共享,是任务协同。技术的发展已经使得个人计算机的处理能力和存储容量达到甚至超过七十年代主机的水平,自主的问题基本上不复存在,因此,现在更需要强调的是协作和共享,强调软件重用,在这方面有大量的问题有待解决。对象引入分布计算系统,给分布计算增添了新的活力,也向计算机从业者提出新的挑战。这方面已经作的工作,都还只是初步的,有待深入和完善。如同异构分布环境下的管理(包括对象管理)一样,需要新的理论、新的思路、新的方法和新的工具,才能取得新的进展。

参考文献

- [1] 蔡希尧、陈平,面向对象技术,西电出版社,1993
- [2] Sinha, A., Client-Server Computing, CACM, Vol. 35, No. 7, 1992
- [3] Andrews, G. R. et al., Concepts and Notations for Concurrent Programming, Computing Surveys, Vol. 15, No. 1, 1983

- [4] 蔡希尧,多处理机系统的逻辑分析与设计,西电出版社,1987
- [5] 蔡希尧、边平定、金益民、王小民, The Design and Implementation of Remote Procedure Calls, Proc. of the 3rd Pan Pacific Computer Conference, 1989
- [6] Nicol, J. R. et al., Object Orientation in Heterogeneous Distributed Computing Systems, IEEE Computer, Vol. 26, No. 6, 1993
- [7] Open Software Foundation, Introduction to OSF DCE, Prentice Hall, 1992
- [8] Keuffel, W., DCE: The United Nations of Computing? Database Programming & Design, Vol. 6, No. 9, 1993
- [9] Bowen, D., Open Distributed Processing, Computer Networks and ISDN Systems, Vol. 23, No. 1-3, 1991
- [10] A Panel Discussion, including the experts from IBM, DEC, HP and other companies, ACM SIGPLAN Notices, Vol. 29, No. 10, 1994
- [11] America, P., POOL-T: A parallel Object-Oriented Language, in (Object-Oriented Concurrent Programming), ed. by A. Yonezawa and M. Tokoro, The MIT Press, 1987
- [12] Lieberman, H., Using Prototypical Objects to Implement Shared Behavior in Object-Oriented Systems, ACM SIGPLAN Notices, Vol. 21, No. 11, 1986
- [13] Aghe, G. et al., Concurrent Programming Using Actors, in (Object Oriented Concurrent Programming), ed. by A. Yonezawa and M. Tokoro, The MIT Press, 1987
- [14] John K. Bennett, The Design and Implement of Distributed Smalltalk, OOPSLA'87
- [15] Alexander B. Schill, et al., DC++ Distributed Object-Oriented Support on Top of OSF DCE, Distributed Systems Engineering, Vol. 1, No. 2, 1993
- [16] Chabdra, R., et al., COOL: An Object-Based Language for Parallel Programming, IEEE Computer, Vol. 27, No. 8, 1994
- [17] Achauer, B., The DOWL Distributed Object-Oriented Language, CACM, Vol. 36, No. 9, 1993