

TP335.8

开放分布式处理的新发展—CORBA 规范

汪芸 兑继英 顾冠群
(东南大学计算机系 南京 210096)

摘要 With the analysis of the development of open distributed processing from the procedure-oriented RPC(Remote Procedure Call) to DCE(Distributed Computing Environment), this paper researches on the importance and certainty of CORBA(Common Object Request Broker Architecture) specification and its release. The structure and processing of CORBA are also discussed. At last, the paper points out the potential applications and uses of CORBA.

关键词 CORBA specification, RPC, DCE, Object-oriented, Open distributed processing, C/S model

1 引言

公共对象请求代理体系结构 CORBA(Common Object Request Object Architecture)是由 OMG(Object Management Group)组织制定的一个工业规范。OMG 组织成立于 1989 年,至今已拥有超过 600 名成员,许多著名的计算机公司如 IBM、DEC、HP 等都是其成员。该组织的中心任务是基于实用的对象技术,建立一个体系结构和一组规范,在分布式环境下实现应用的集成,使得基于对象的软件成员在分布异构环境中可重用、可移植和可互操作。

CORBA 规范是开放分布式处理发展一定程度,在现有技术和认识条件下的产物,它的出现具有必然性,更是具有重要性。本文旨在通过剖析分布式处理发展的历程和 CORBA 规范本身,来阐述作者对 CORBA 规范的认识。

2 RPC(远程过程调用)

RPC 是网络上的高层协议,允许网络上的应用通过特定的网络上的过程调用来实现,从而隐藏了底层网络的细节。RPC 实现的是一个客户方和服务方之间通信的接口系统,用于支持网络上的应用开发。利用 RPC,客户方可以发送过程调用的请求给服务器方,服务器方在接收到这些请求后调用本地例程执行客户方请求的操作,并将执行结果返回给客户方。

RPC 模型类似于本地过程调用,调用方(客户方)发送一个调用消息给被调用方(服务器方),之后

调用方阻塞,等待被调用方的回答。调用消息包括调用过程的参数等信息;被调用方的回答包括调用过程的结果等信息。一旦接收到被调用方的回答,调用方取出返回结果,结束阻塞继续运行。

在服务器方有一个专门的进程守候调用消息的到来。一旦接收到调用消息,服务器方的守候进程将取出调用进程的参数,运算并得出结果,发送返回信息并等待下一个调用的到来。如图 1 所示。

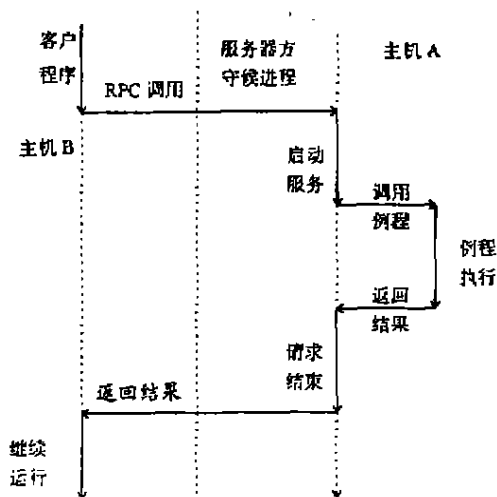


图 1 RPC 调用时序

3 DCE(分布式计算环境)

DCE 是开放软件基金会(Open Software Foundation)提出的标准,它通过在网络上提供增值服务

为应用开发人员提供具有良好的互操作性的开发环境,开发人员通过 DCE 可以广泛地使用所有的系统和设施而无需考虑用户、应用程序和其它所需资源的位置,从而使得用户可以更有效、更充分地使用网络上的计算资源。

DCE 包括一组集成设施,用来支持分布式应用的开发、使用和维护。网络上各种设施的集成使得它们扩大了使用的范围,从而可以物尽其用。同时 DCE 还是一种独立于网络的操作系统,为用户提供现有环境下的移植功能,其可移植性是通过一组成型的分布式设施的调用来实现的。OSF 分布式计算环境体系结构如图 2 所示。

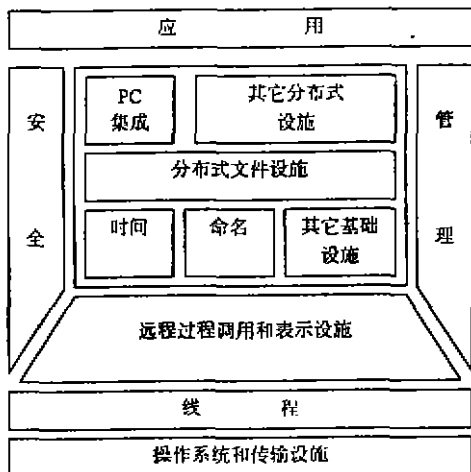


图 2 OSF 分布式计算环境体系结构

DCE 包括两组设施:

(1)基础的分布式设施为软件开发人员提供了一组工具,用来为终端用户创建分布式应用所需的服务,包括:远程过程调用(RPC)、目录服务、时间服务、安全服务、线程服务。

(2)数据共享设施为终端用户提供了基于基础分布式设施的服务。这些服务无需终端用户编制程序,从而可以更方便地使用信息。包括:分布式文件系统、无盘服务。

4 CORBA 规范

通过前面的分析,我们可以看到,无论是 RPC,还是基于 RPC 机制的 DCE,它们都存在着一一些明显的不足之处:(1)虽然是按照客户/服务器模式构建的系统,但是客户程序和服务器程序之间的调用关系是静态的,例如客户程序必须事先知道所需服务

由哪个服务器提供,调用的具体方式是怎样的。(2)调用关系的静态性质导致了所构建的系统也是静态的,也就是说,一旦有程序发生改变(尤其是服务器方程序),那么整个系统必须重新编译生成,原有的系统就会被抛弃。(3)客户程序和服务器程序必须用 RPC 或 DCE 的实现语言来编制,因为这些程序需要同 RPC 或 DCE 提供的库函数或接口界面捆绑在一起,这样就大大限制了用户的可选语言范围,在某些情况下,用户不得不去学习一种新的程序设计语言,为的是能够使用 RPC 或 DCE 提供的服务。

这些缺点限制了 RPC 和 DCE 的进一步发展,而 CORBA 规范则能够在这些方面提供完美的解决方案。

4.1 CORBA 规范的组成

OMG 组织在成立后不久就制定了对象管理体系结构 OMA (Object Management Architecture),其参考模型如图 3 所示。它由以下四个部分组成:

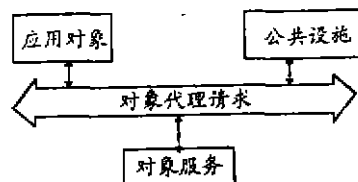


图 3 OMA 参考模型

(1)对象请求代理 ORB (Object Request Broker),使对象在分布式环境中透明地收发请求和响应,它是构造分布式对象应用,使应用在不同层次的异构环境下互操作的基础。

(2)对象服务 (Object Services),是为使用 and 实现对象而提供的基本服务集合。对于分布式应用这些是基本服务,它们独立于应用领域,如生命周期服务定义了对象创建、删除、拷贝和移动的方式,它不考虑对象在应用中如何实现。

(3)公共设施 (Common Facilities),是为许多应用提供的共享服务集合,如系统管理、电子邮件等。

(4)应用对象 (Application Objects),相应于传统的应用表示,可由单个厂商生产,是参考模型中的最高层,OMG 对此不作规定。

1991 年,OMG 组织为 ORB 制定了规范,即 CORBA 规范,经历了 1.1 和 1.2 版本,现行版本为 2.0 版。

CORBA 规范定义了 IDL 语言及映射、单个

ORB 体系结构和 ORB 间互操作机制。

(1)IDL 语言是 CORBA 规范中定义的一种中性语言,它用来描述对象的界面,而不涉及对象的具体实现。使用 IDL 界面的客户对象可以根据 IDL 界面中的信息来决定发出请求和接收响应,从而使得客户对象完全独立于具体对象实现所在的位置,使用的变成语言及其他与对象界面无关的方面。IDL 的词法、语法规则遵从 C++ 的规定,但加入了一些关键词以支持分布概念。

IDL 完全是一种描述性语言,它不能用于实现客户程序。为了使对象能够统一使用 IDL 界面信息,CORBA 规定了 IDL 到其他具体语言的映射,不同的语言有不同的映射结果。在 CORBA 中定义了 IDL 语言到 C、C++ 和 SmallTalk 语言的映射。

(2)单个 ORB 的体系结构如图 4 所示。ORB 定义了客户方程序和对象实现方程序间静态和动态两种调用方式。

(3)ORB 间互操作是为了使不同实现的 ORB 系统能够共存于同一个环境下而提供的相互识别和转换的机制。从策略上分,有间接桥接和直接桥接两种方法,从实现层次上分,则有线内桥接(in-line bridging)和请求层桥接(request-level bridging)两种机制。CORBA 规范还定义了 GIOP(General Inter-ORB Protocol)协议和 IIOP(Internet Inter-ORB Protocol)协议。把 ORB 互操作规范加入到 ORB 体系结构后,会引起 ORB 内部执行过程发生一些改变。

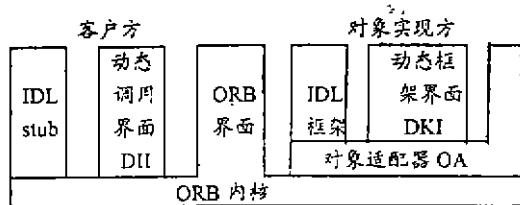


图 4 ORB 体系结构

4.2 CORBA 规范的特点

CORBA 规范以其自身的优势领导着开放分布式处理的发展,这些具有里程碑意义的特点是:

(1)在 CORBA 规范中引入了代理(broker)的概念。一个代理至少可以有三个方面的作用:完成对客户方提出的抽象服务请求的映射;自动发现和寻找服务器;自动设定路由实现到服务器方的执行。这样用户在编制客户方程序时就可以避免了解很多的

细节,而只要完整地定义和说明客户需要完成的任务及其目标。

(2)增加了代理机制后,实现了客户方程序与服务器方程序的完全分离,这是与面向过程调用机制为基础的客户/服务器模式是根本不同的。客户将不再同服务器发生直接的联系,而仅仅需要同代理进行交互。客户与服务器之间就可以有更加灵活的关系存在。

(3)CORBA 规范定义的基础是面向对象的设计思想和实现方法。把分布式计算同面向对象的概念相互结合后可以带来诸多的好处,如能够将冗余度控制在最低的程度、一个对象既能被客户程序也能被服务器程序使用、对象实现的修改不会影响到双方实现程序、提高软件重用率等等。

(4)提供了软件总线的机制。所谓软件总线是指 CORBA 规范定义了一组接口规范,任何应用程序、软件系统或工具只要具有与该接口规范相符合的接口定义,就能方便地集成到 CORBA 系统中,而这个接口规范是独立于任何实现语言和环境。任意被集成到 CORBA 系统中的成员间都是可以使用对方所提供的服务。

(5)分层的设计原则和实现方式。CORBA 规范只是针对 OMA 体系结构中的 ORB 制定的标准,因此在 CORBA 规范仅定义了 ORB 中需要用到的基本对象类,封装了相应的数据结构和方法。而面向应用的对象定义则可以在 OMA 的应用对象或应用开发环境中逐步分层定义和实现。

CORBA 规范充分利用了现今各种技术发展的最新成果,将面向对象的概念揉合到分布式计算中,定义了一组与实现无关的接口方式,引入代理机制分离客户和服务器,使得 CORBA 规范成为开放的、基于客户/服务器模式的、面向对象的分布式计算的工业标准。

5 CORBA 规范的应用前景

自从 CORBA 规范颁布以来,世界范围内有很多的计算机商和科研部门对它进行了深入的研究和实现,已有的较为著名的实现是:

(1)IONA 的 Orbix 是一个基于库的 CORBA 规范实现,它提供了 C++ 语言映射器和 C++ 代码产生器,在 Windows 和主要的 UNIX 平台上运行。

(2)Digital 的 ObjectBroker 完全实现了 CORBA 1.1 规范,它包含动态、静态例程调用、IDL 语言实现、界面和实现仓库和上下文对象,它可以运行于

多个平台之上,实现 IDL 语言到 C 语言的映射。

ObjectBroker 作为一个集成工具来实现,它允许用户将 CORBA 技术很容易地集成到现有体系结构中,使其相应客户/服务器环境迁移,已有应用通过所写的 C 语言封装方式集成到 ObjectBroker 中去。

(3)IBM 的 DSOM 基于其 SOM(System Object Model)之上。SOM 以一种库工具箱的形式提供,它可以用来方便地创建应用需要的二进制类库。DSOM 提供了一个分布式框架,使用 SOM 工具箱来实现 ORB。DSOM 支持两种类型的分布式处理:支持单机上的多进程操作的工作站 DSOM 和支持网络环境下的多机系统的工作组 DSOM,其中工作组 DSOM 才是遵从 CORBA 规范的实现。

(4)东南大学研究现状 东南大学计算机系网络研究室依靠它在网络、分布式处理和系统集成方面长期从事研究的基础和经验,现在正在研究 CORBA 规范及其实现。基于 CORBA 规范面向并行工程的集成框架可行性论证报告在国内多家大学和研究所的合作下,已经通过了国家“863”CIMS 专家组组织的评审,计划实施亦已开始,相信不久将会有这方面的系统实现问世。

基于 CORBA 规范实现的系统集成和应用开发环境在我国企业中将会有潜在的巨大的应用前景,它能够在企业逐步实现生产和管理的自动化、提高劳动生产率、快速响应市场需求等方面起到积极的推动作用。并且基于 CORBA 规范实现的集成系统在处理已有应用和系统方面有其独特的优势,用户无需抛弃已有的应用,只要在已有的应用之上实现一些适当的接口,就能集成到 CORBA 系统中,从而对企业已有的投资给予充分的保护。

未来的基于 CORBA 规范的系统应该是一个能跨越不同地理位置、穿越不同网络系统、屏蔽实现细

节,实现透明传输、集成不同用户特长的基于客户/服务器模式、面向对象、开放的分布式计算集成环境。其组成如图 5 所示。

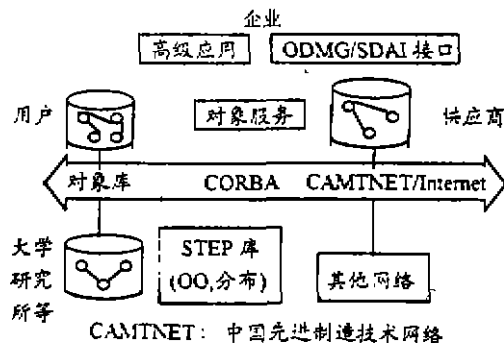


图 5 未来可能的基于 CORBA 规范实现系统的组成

结束语 CORBA 规范是开放分布式处理发展过程中的一个新的里程碑,它继承了 RPC 和 DCE 的长处和分布处理设计思想,改变了实现手段和方法,补充和完善了开放分布处理的体系结构,引入了代理的新概念,使得开放分布式处理能够适应当前新的应用需求,为进一步的集成、共享以及合作问题提供良好的解决基础。随着对 CORBA 规范的认识和 OMG 组织对其他 OMA 组成部分制定规范,CORBA 规范会在更多的场合发挥其巨大的作用。

参考文献

- [1] Randy Otte, Paul Patrick, Mark Roy, Understanding CORBA, Prentice Hall, 1995
- [2] Common Object Request Broker Architecture Specification, Version 2.0, 1994
- [3] Technology Enabling Agile Manufacturing-TEAM, USA, 1995

(上接第 49 页)

结束语 本文论述了采用 Client/Server 结构的智能信息集成和工作流技术来实现的智能决策支持原型。该原型的原理和方法在中国温州正泰集团公司 CIMS 工程和决策管理系统的设计实施

过程中已得到证明和运用。理论和实践都证明,采用 Client/Server 结构的智能化领导决策支持原型来辅助实现企业 CIMS 信息的集成和智能管理可以正规范有效地利用企业 CIMS 中的信息,以解决智能化计算机辅助决策支持中的许多难题。