

开放分布式处理

软件接口

新计算机 (7)

27-31

开放分布式处理及其实例

汲化 陈军 谢立

(南京大学计算机科学系 南京 210093)

TP335.8

摘要 ODP(Open Distributed Processing) is the tendency of distributed computing, which is a new technology to solve the distributed software interface problems. Its goal is to define a framework of distributed systems to achieve portability, interoperability, and distribution transparency. This paper presents the main characteristics of open distributed processing. The Reference Model of Open Distributed Processing RM-ODP and some support environments for ODP are also introduced.

关键词 Open distributed processing, Open system, Distributed system.

迅速发展的计算机与通信技术使得构造大型分布式系统成为可能。尽管如此,互连系统特别是分布式操作和应用接口的异构性,严重影响了系统间的互操作性。随着应用的推广和深入,如信息高速公路,要求实现全球范围内的网络,在这样的网络系统中,需要连结不同的计算机、不同的计算机网络、不同的软件和不同的用户等。为此,需要研究多项新的关键技术。

开放分布式处理 ODP (Open Distributed Processing) 是一个试图解决分布环境下软件接口问题的一项新兴技术, ODP 不仅刻画了一个利用公共交互模型来支持组织内部和组织之间的异构型分布式处理的开放系统,而且提出了一个构造分布式系统的框架,其范围包括:不同的厂商支持;使用异构型技术实现;可扩展到全球范围;系统可以平滑地更新;不同应用之间可以互相操作和共享数据;开发和操作开销减少。

开放分布式处理参考模型 RM-ODP 是 ISO 和 CCITT 开发中的一个国际标准 (ISO 10746 和 CCITT X.900 系列)。它的目标是:1)异构型平台间应用的可移植性。分布式应用的可移植性提供应用程序在系统中迁移的潜力并且不破坏应用所提供或正使用的服务。这种迁移包括静态的系统重构或

重新安装和动态的系统重构;2)ODP 系统间的互操作性,提供不同系统之间信息的有意义交换,即信息的语义交换而不仅仅限于语法交换。此外,它还提供系统间功能服务的方便使用,特别是资源的动态发现和动态类型检查;3)分布透明性,使得程序员和用户在构造分布式应用时,不必关心潜在的硬件、操作系统及通信机制的差异。

目前世界上很多研究者都在开展 ODP 方面的研究工作,已经推出了一些支持 ODP 的开发环境,如 ANSAware, OSF/DCE 和 OMG/CORBA。但与 RM-ODP 相比,还有很多问题需要进一步研究和解决。本文综述了开放分布式处理的特点,介绍了开放分布式处理参考模型 ISO RM-ODP 的主要内容以及现有的、支持开放分布式处理的分布式计算环境实例。

1. 开放分布式处理的特点

1.1 透明性

分布式系统的一个重要问题就是透明性,使得用以克服分布问题和细节的机制相对应用无关。开放分布式处理的目标就是对所有这些不同的机制提供一个共同的界面。

在开放分布式系统中,典型的透明性包括:

汲化 博士生 研究方向为分布式系统。谢立 副校长,教授,博士生导师,研究方向为分布式系统。

存取透明性;隐蔽表示中的相异性;

并发透明性;隐蔽执行中的重叠性;

位置透明性;隐蔽访问路径的相异性;

资源透明性;隐蔽运行和存贮环境中对象的主动和被动性;

迁移透明性;隐蔽位置的变化;

配置透明性;隐蔽系统构成的变化。

1.2 分离性

由于分布式系统各构件的潜在分离性,所有构件间的访问必须基于请求和响应信息的交换。这就是典型的顾客/服务员模型。

在这样的模型中,下列问题日趋严重:1)调用必须是异步的但多个顾客可能同时请求同一个服务;2)并发是分布式系统的规范,但程序的运行却是真正的重叠;3)调用可能因顾客与服务员之间的距离或者因暂时的通信问题而延迟,服务器也可能因超载使得响应迟滞;4)不能被透明性所隐蔽的灾难性故障也可能发生,例如,计算机瘫痪、网络连接中断等。

尽管透明性机制可以减缓上述问题的发生,但是它并不完善,应用程序的设计者不得不考虑在应用层次隐蔽并发、延迟和故障。因此,在开放分布式处理系统中,必须提供应用设计者必要的工具来建立分布式应用的模型、管理和程序,并且定义一个针对开放分布式处理的应用程序设计观点。

1.3 异构性

分布式系统的另一个问题是异构性,通常包括:

设备异构性(例如不同厂商的设备或者因技术的更新使设备换代等);操作系统异构性(例如在不同的工厂、办公室等运行的系统);计算异构性(例如使用不同的程序设计语言,数据库语言等)。

在开放分布式处理系统中,一个顾客不能预知远程数据的格式或者网络协议所支持的数据类型,因此必须通过抽象数据类型来克服这种异构性并建立自动沟通不同技术的桥梁。

1.4 配置动态性

在分布式系统中,构成系统的部件将由于系统扩张、改良、排障或负载平衡等原因而不断动态地变化,这要求系统必须能够:

1)动态连接。即由于系统的动态更新,一个顾客应用必须在运行时刻动态地连接到所需的服务上;

2)动态类型检查。为了达到最大的安全性,所有的访问必须在连接建立时进行类型检查;

3)部件平等性。为了达到最大的灵活性,所有组成的部件,无论其大小类别,必须平等地组织和管理。

系统资源的动态变化要求强类型检查和动态连接。在一个开放分布式系统中,顾客必须能发现哪一个服务员可提供所需的服务,即满足顾客所要求的服务类型和特性,并动态连接到该服务所在的程序上。

1.5 互操作性

在开放分布式处理系统中,构成系统的各组织必须能自主地管理、设计属于他们自己的系统。另一方面,在这些相对独立的系统之间必须具有互操作性、有意义地交换信息和互相使用系统功能。

因此,在各组织之间必须有一个转换和调节机制来实现互操作。这样的机制包括动态资源发现(Resource Discovery)、动态服务连接、动态服务选择、动态服务类型检查。

2. 开放分布式处理参考模型 RM-ODP

开放分布式处理参考模型 RM-ODP 给出了一组标准,其目标是定义一个框架,使得尽管构成分布式系统的各部件具有异构性和自治性,但应用程序在全局范围内仍可以合作和集成。

2.1 RM-ODP 的结构

RM-ODP 标准将包括四个部分:

第一部分:概述及使用指南(ISO 10746-1, CCITT X.901);

第二部分:描述模型(ISO 10746-2, CCITT X.902);

第三部分:规范模型(ISO 10746-3, CCITT X.903);

第四部分:结构语义(ISO 10746-4, CCITT X.904)。

第二、三部分文本是目前标准化工作的重点,这两部分特别是第三部分定义了 RM-ODP 的框

架,为了解决ODP系统的复杂性,RM-ODP从不同的观点来考虑该系统,每一个观点代表了对原始系统的一个不同的抽象。为此,第三部分给出了五个不同观点。文本给出了每一个观点的一组定义和规则,并提供一个语言来对该观点标准化。下面我们简要叙述这些观点。

2.2 企业观点(Enterprise Viewpoint)

在开放分布式系统中,企业观点用来表示目的、策略和界限(即对“开放”的限制)。在企业观点中,社会和组织的策略可以用下面的术语来定义:

代理(agents):“主动的”对象,可以开始一个动作;例如,学生、图书管理员等。

受体(artifacts):“被动的”对象,不能开始一个动作,例如,图书,期刊,借书帐号等。

团体(communities):一组代理或受体,例如,一组学生等;

许可(permission):规定什么可以做,例如,学生可以借书;

禁止(prohibition):规定什么不可以做,例如,学生不能借期刊;

责任(obligations):规定什么必须做,例如,学生必须如期归还所借的图书。

2.3 信息观点(Information Viewpoint)

信息观点用来表示分布式系统中信息及信息间关系的信息模型和信息流等,使用下面的术语来定义:

模式(schemas):描述了一个对象的状态和结构,例如一个学生的借书帐号包括所借图书和最多借书量;

关系(relation):描述了对象间的联系。例如,学生和借书帐号间的关系;

完整性规则(integrity rule):是关于模式的断言,例如,学生必须至少拥有一个借书帐号,并且每个借书帐号必须至少被一个学生所拥有。

2.4 计算观点(Computational Viewpoint)

计算观点用来表示一个ODP系统的功能性分解、互操作性和可移植性。RM-ODP的计算观点是基于对象的,即对象包括数据和处理(即行为);对象提供与其它对象交互的接口界面;对象可以提供多个接口界面。

一个计算规约说明定义了ODP系统中的对象、它们的活动及它们之间的交互。对象间的交互可以是面向操作的(远程过程调用)或面向流的(生产者/消费者)。

面向操作的接口提供了一个顾客/服务员模型,实现了一个远程过程调用模式。面向流的接口提供了在生产者对象和消费者对象之间的一个逻辑上连续的数据流,一个接口可以包含多条流。RM-ODP中包含流的交互是为了满足多媒体和通信应用的需要。

2.5 工程观点(Engineering Viewpoint)

工程观点用来表示支持分布式处理所要求的结构。工程观点不涉及到ODP应用的语义,但它决定了对分布和分布透明性的要求。

在工程观点中描述的基本实体是对象和通道。对象在该观点下可分为两类:基本工程对象和结构对象,前者对应于计算观点中所定义的对象,后者包括所有其它对象,例如,通信协议对象。

一个通道提供通信机制,包含并控制基本工程对象间交换所需的透明性功能,即顾客透明性对象、服务透明性对象和多个支持透明性对象。

2.6 技术观点(Technology Viewpoint)

技术观点用来表示支持ODP的技术的合适性。它为ODP系统选择合适的实现,维护和测试技术,选择操作系统,计算机网络和硬件部件。RM-ODP几乎没有给出用于技术规约说明的规则。

3. ODP 支撑环境实例

本节从ODP的角度出发,介绍了目前现有的、能支持或部分支持ODP目标的分布计算环境ANSAware、OSF/DCE、OMG/OMA。

3.1 ANSA 体系结构

ANSAware(Advanced Network Systems Architecture)是一个支持开放分布式处理的分式式环境。目前是由英国电话公司、DEC、HP等多家公司支持的一个软件包。ANSA体系结构基于对象模型,支持顾客/服务员模式。一个对象表示一个资源或一个服务,每一个对象上的操作可以分为较小的组,称为接口,每一个接口上的操作通常提供某些特定的服务。提供服务的对象称为服务员,使用服

务的对象称为顾客。一个顾客通过 RPC 调用服务员接口的操作。ANSAware 目前提供两种透明性:存取透明性和位置透明性。

ANSAware 提供三个开发分布式应用的语言:接口定义语言 IDL、嵌入 PREPC 的 C 语言和分布式程序设计语言 DPL。

ANSAware 最重要的运行时刻工具是服务交易器(Trader)工厂,结点管理和布告服务,服务交易器支持动态连接,它允许在运行时刻发现所需服务。工厂提供动态对象的例化和终止。结点管理提供对结点上的服务的建立、简单管理和撤消。布告服务将对象的终止通告有关部分。

3.1.1 对 ODP 目标的支持 ANSAware 目前支持多种平台,例如 SUNOS、HP/UX、VMS 和 Ultrix、DRS/NX 6000 和 DOS 等。ANSAware 的应用可以通过 RPC 在任何平台之间互相操作。

ANSAware 允许对象和接口动态建立和撤消,通过动态连接实现运行时刻重构,通过子类型关系动态替换接口,从而达到系统动态更新的目标。

ANSAware 提供一个称为交易器联邦(Trader Federation)的机制来实现多个服务中介的协调,从而保证系统的可扩展性。

3.2 OSF/DCE

DCE (Distributed Computing Environment) 是 OSF (Open System Foundation) 开发的产品,目前在很多种硬件结构、操作系统平台上都有相应的 DCE 版本。DCE 为用户提供通信机制,目录服务,安全性和分布式时间服务,利用顾客/服务员模式来构造分布式应用,通过 RPC 实现通信,DCE 将所有计算机分成若干个组,称为单元(CELL),每个单元被分别管理,并且至少包含一个单元目录服务、一个安全性服务和一个分布式时间服务。

3.2.1 数据类型管理 DCE 系统是一个运行在现有操作系统(例如 OSF/1 或 Ultrix 等)之上的一组运行时的库和服务。DCE 提供足以构造顾客和服务的库和应用,并且提供一个描述和表示数据类型的与机器无关的机制来支持异构型环境下的分布式应用,使得与服务连接和通信有关的程序代码通过一个由 DCE 接口描述语言(IDL)描述的接口规约说明生成,并且应用程序代码的其它部分与局

部过程相同。

3.2.2 资源发现和后期联编 当一个服务程序开始运行时,首先将它所支持的接口通知 DCE 的单元目录服务,单元目录服务维护一个服务所在的结点地址,当一个顾客程序希望调用一个已知接口的一个过程时,首先根据接口名,查询单元目录服务,得到包含服务程序所在的结点地址,然后再根据该地址进行远程过程访问。这种资源发现的过程由 DCE 运行时库完成,而在其它系统中通常必须由程序员来决定。

当一个服务程序在单元目录服务中注册一个接口时,还可能给出接口的属性,这些属性将在顾客程序查询时随结点地址一起返送给顾客程序,以使顾客程序可以动态连接到服务程序上。

3.2.3 单元目录服务 DCE 提供单元目录服务的方法,实现 ODP 系统所要求的资源定位和资源发现,单元目录服务提供了一个资源的分布式目录接口,DCE 单元目录维护所有 DCE 单元中对象的名,例如,用户,服务接口、打印机资源和文件系统资源等。命名方法可以使用 CCITT X.500 协议,所有单元的单元目录构成 DCE 的全局目录,因此,每一个资源都有一个全局名,它由单元名和局部名构成。

3.3 OMG/CORBA

OMG(Object Management Group)的对象管理结构 OMA(Object Management Architecture)由一组工具和协议构成,提供一个支持“可互操作的软件部件标准接口”的体系结构。

该系统可以一种分布式对象管理的概念支持分布式应用,即一个对象管理结构 OMA 系统包含各种分布在整个计算机网络中的对象,每一个对象提供一个可使其它对象通过 RPC 方式与之通信的接口。

3.3.1 对象管理结构 OMA

OMA 的基本软件构件是对象,一个对象作为一个类型的实例存在,对象类型以一个继承层次结构来组织,其根是一般对象类型 Object,每一个类型有一个接口,定义了该类型对象的数据类型和操作,在 OMA 中,一个类型的实例作为对象而不是作为类型或操作来管理。

对象管理结构由以下主要部分组成:

1) 对象请求中介 ORB (Object Request Broker): 允许对象与其它对象建立通信联系, 其方法是提供一个对象接口注册机构, 一个用以查询和维护注册机构的在编译时刻就被所有对象所知的标准接口以及一种与从注册机构得到的接口交互的方法:

2) 对象服务 (Object Service): 其基本服务可被所有对象类型继承, 对象服务着重于提供一组通用的功能, 例如建立或者撤消一个对象的实例, 存贮对象的状态, 维护安全性和完整性, 提供一个版本管理等。

3) 公共设施 (Common Facilities): 一些可能被大多对象共用的服务, 例如编目和浏览、文本打印和电子邮件等。

4) 应用对象 (Application Objects): 针对某一应用的对象, 对应于应用程序这一传统的表示方法。

3.3.2 类型和数据表示管理 OMA 的一个真正的包含所有特点的对象系统的实现是公共对象请求中介结构 CORBA (Common Object Request Broker Architecture), 它定义了一组标准数据类型、新类型的构造方法、转换语法、接口描述语言 IDL 以及一个接口库。

CORBA IDL 是 C++ 的一个扩充, 一个接口可以用 IDL 来定义成类似于一个 C++ 的类, 因此, 一个接口可以把一个或多个其它接口指名为超级类型 (Supertypes), 同时, 也可以对一个接口指定若干属性。

对象关系服务提供一个建立、删除和抽取系统中对象间的关系信息。该服务利用一个查询对象库 (如关系库和接口库等) 的标准语法提供查询对象服务。

3.3.3 对象连接和资源发现 CORBA 提供两种方法从顾客程序来访问对象的实现: 一是通过一个由 IDL 产生的根 (STUB) 进程, 二是通过 CORBA 接口库和动态调用接口 (Dynamic Invocation Interface)。接口库允许在运行时刻搜索接口, 动态调用接口允许同一个没有顾客根进程的对象接口交互, 即该接口在编译时刻是未知的。一个 IDL

说明和一个接口库中的接口定义有相同说明的能力, 但是, 接口库中的信息可以随时改变, 而 IDL 根是一个对象源代码的一部分, 它们在编译时刻必须确定并且被连接到执行代码中。从服务对象的角度来看, 它不必区分操作调用是通过 IDL 根还是动态调用接口, 无论服务在编译时刻还是运行时刻被选定, 对象系统的能力都不受影响。

4. 结束语

开放分布处理 ODP 提出了一个构造分布式系统的框架以解决软件接口问题, 实现应用程序在全局范围内可以合作与集成。开放分布处理参考模型 RM-ODP 即将成为国际标准, 但它不是一个实际标准。它的目标是可移植性、互操作性和分布透明性。在开放分布处理领域, 仍有很多问题值得研究与探讨。随着分布式应用向着实用化、商业化和国际化的方向发展, 开放分布处理将成为分布式系统的主要研究方向之一。

参考文献

- [1] ISO/IEC JTC/SC21, Basic Reference Model & Open Distributed Processing, Part 1 to 4, 1993
- [2] P. F. Linington, Introduction to Open Distributed Processing Basic Reference Model, Intl IFIP Workshop on Open Distributed Processing, Berlin, Oct. 1991
- [3] A. Herbert, The Challenge of ODP, same to [2]
- [4] M. Bearman and K. Raymond, Federating Traders; An ODP adventure, same to [2]
- [5] Object Management Architecture Guide 2.0, OMG TC Document 92.11
- [6] The Common Object Request Broker Architecture and Specification, OMG Document Number 93.XX.YY, Revision 2.0 Draft 29 Dec. 1993
- [7] ORB 2.0 RFP Submission, Sept. 28, 1994, OMG TC Document 94.9