

中间件技术综述

周园春 李 森 张 建 李晓欧 张 飞
(中国科学院合肥智能机械研究所,合肥 230031)
E-mail: yczhou@mail.iim.ac.cn

摘 要 计算机技术和网络技术的发展产生了中间件技术。现在中间件技术已是计算机中发展最快的技术之一。该文介绍了中间件技术的背景、概念、基本内容、主要产品,提出了中间件未来的发展方向。

关键词 中间件 COM CORBA RPC MOM TPM

文章编号 1002-8331-(2002)15-0080-03 文献标识码 A 中图分类号 TP319

Overview on Middleware Technology

Zhou Yuanchun Li Miao Zhang Jian Li Xiao'ou Zhang Fei

(Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031)

Abstract: The development of computer and network technology makes the generation of middleware technology. Now, it develops most rapidly in computer technology. This paper discusses about the background, the concept, the basic content and the main products of the middleware technology. At last, it introduces the prospect prediction of the middleware development.

Keywords: Middleware, COM, CORBA, Remote Procedure Call (RPC), Message Oriented Middleware (MOM), Transaction Processing Monitor (TPM)

1 引言

计算机技术和网络技术的不断发展使得客户机/服务器体系结构得到蓬勃发展,但是随着应用水平的不断提高以及企业应用的不断扩大,构建在两层客户机/服务器之上的计算机应用系统的局限性就愈发地暴露出来。而且在分布式计算模式环境中,无论是硬件平台还是软件平台都不可能做到统一。而大规模的应用软件通常要求在软硬件各不相同的分布式网络上运行。所以为了克服这种局限性、更好地开发和应用能够运行在这种异构平台上的软件,迫切需要一种基于标准的、独立于计算机硬件以及操作系统的开发和运行环境,中间件技术就此应运而生。

2 中间件的概念

中间件是介于应用系统和系统软件之间的一类软件,它使用系统软件所提供的基础服务(功能),衔接网络上应用系统的各个部分或不同的应用,能够达到资源共享、功能共享的目的。目前,它并没有很严格的定义,但是普遍接受 IDC 的定义:中间件是一种独立的系统软件或服务程序,分布式应用软件借助这种软件在不同的技术之间共享资源,中间件位于客户机服务器的操作系统之上,管理计算资源和网络通信。从这个意义上可以用一个等式来表示中间件:中间件=平台+通信,这也就限定了只有用于分布式系统中才能叫中间件,同时也把它与支撑软件和实用软件区分开来。一般认为,中间件必须具有以下特点:

(1)标准的协议和接口。

(2)分布计算,提供网络、硬件、操作系统透明性。

(3)满足大量应用的需要。

(4)能运行于多种硬件和操作系统平台。

其中具有标准的接口和协议非常重要,因为它可以实现不同硬件和操作系统平台上的数据共享和应用互操作。

从理论上讲,中间件有以下的工作机制:在客户端上的应用程序需要从网络中的某个地方获取一定的数据或服务,这些数据或服务可能处于一个运行着不同操作系统和特定查询语言数据库的服务器中。客户/服务器应用程序中负责寻找数据的部分只需访问一个中间件系统,由中间件完成到网络中找到数据源或服务,进而传输客户请求、重组答复信息,最后将结果送回应用程序的任务。

在具体实现上,中间件是一个用 API 定义的软件层,具有强大的通信能力和良好的可扩展性的分布式软件管理框架。

3 中间件的作用和分类

中间件的作用简单来说就是试图通过屏蔽各种复杂的技术细节使技术问题简单化。具体地说,中间件屏蔽了底层操作系统的复杂性,使程序开发人员面对一个简单而统一的开发环境,减少程序设计的复杂性,将注意力集中在自己的业务上,不必再为程序在不同系统软件上的移植而重复工作,从而大大减少了技术上的负担。所以说中间件带给应用系统的,不只是开发的简便、开发周期的缩短,同时也减少了系统的维护、运行和管理的工作量,还减少了计算机总体费用的投入。其次,中间件作为新层次的基础软件,其重要作用是将不同时期、在不同操

基金项目:国家 863 高技术研究发展计划信息技术项目基金(编号:863-306-ZD05-01-4)

作者简介:周园春,硕士研究生,研究方向为专家系统与中间件技术。李森,研究员,研究方向为专家系统与中间件技术。张建,副研究员、李晓欧,硕士。

作系统上开发的应用软件集成起来,彼此像一个天衣无缝的整体协调工作,这是操作系统、数据库管理系统本身做不了的。

目前,针对不同的应用涌现出各具特色的中间件产品。从不同的角度和层次对中间件有不同的分类。根据中间件在系统中所起的作用和采用的技术不同,可以把中间件大致划分为以下几种:

3.1 数据访问中间件(Data Access Middleware)

在分布式系统中,重要的数据都集中存放在数据服务器中,它们可以是关系型、复合文档型、具有各种存放格式的多媒体型,或者是经过加密或压缩存放的,数据访问中间件是在这种系统中建立数据应用资源互操作的模式,实现异构环境下的数据库联接或文件系统联接的中间件,从而为在网络上虚拟缓冲存取、格式转换、解压等带来方便。数据访问中间件在所有的中间件中是应用最广泛、技术最成熟的一种。一个最典型的例子就是 ODBC,ODBC 是一种基于数据库的中间件标准,它允许应用程序和本地或者异地的数据库进行通信,并提供了一系列的应用程序接口 API,当然,在多数情况下这些 API 都是隐藏在开发工具中,不被程序员直接使用。不过在数据访问中间件处理模型中,数据库是信息存储的核心单元,中间件完成通信的功能,这种方式虽然是灵活的,但是并不适合于一些要求高性能处理的场合,因为它需要大量的数据通信,而且当网络发生故障时,系统将不能正常工作。

3.2 远程过程调用中间件(RPC)

远程过程调用是另外一种形式的中间件,它在客户/服务器计算方面,比数据库中间件又迈进了一步。通过这种远程过程调用机制,程序员编写客户方的应用,需要时可以调用位于远端服务器上的过程。它的工作方式如下:当一个应用程序 A 需要与远程的另一个应用程序 B 交换信息或要求 B 提供协助时,A 在本地产生一个请求,通过通讯链路通知 B 接收信息或提供相应的服务,B 完成相关处理后将信息或结果返回给 A。RPC 的灵活特性使得它有比数据库中间件更广泛的应用,它可以应用在更复杂的客户/服务器计算环境中。远程过程调用的灵活性还体现在它的跨平台性方面,它不仅可以调用远端的子程序,而且这种调用是可以跨不同操作系统平台的,而程序员在编程时并不需要考虑这些细节。

RPC 也有一些缺点,主要是因为 RPC 一般用于应用程序之间的通信,而且采用的是同步通信方式,因此对于比较小型的简单应用还是比较适合的,因为这些应用通常不要求异步通信方式。但是对于一些大型的应用,这种方式就不是很适合了,因为此时程序员需要考虑网络或者系统故障、处理并发操作、缓冲、流量控制以及进程同步等一系列复杂问题。

3.3 面向消息中间件(MOM)

消息中间件能在不同平台之间通信,实现分布式系统中可靠的、高效的、实时的跨平台数据传输,它常被用来屏蔽掉各种平台及协议之间的特性,实现应用程序之间的协同;其优点在于能够在客户和服务器之间提供同步和异步的连接,并且在任何时刻都可以将消息进行传送或者存储转发,这也是它比远程过程调用更进一步的原因。另外消息中间件不会占用大量的网络带宽,可以跟踪事务,并且通过将事务存储到磁盘上实现网络故障时系统的恢复。当然和远程过程调用相比,消息中间件不支持程序控制的传递。

消息中间件适用于需要在多个进程之间进行可靠的数据

传送的分布式环境。它是中间件中唯一不可缺少的,也是销售额最大的中间件产品,目前在 Windows 2000 操作系统中已包含了其部分功能。

3.4 面向对象的中间件(Object Oriented Middleware)

当前开发大型应用软件通常采用基于组件技术,在分布系统中,还需要集成各节点上的不同系统平台上的组件或新老版本的组件;组件的含义通常指的是一组对象的集成,其种类有数百万种,但这些组件面临着缺乏标准而不能相互操作,各厂家的组件只能在各自的平台上运行。为此,连接这些组件环境的面向对象的中间件便应运而生。面向对象的中间件是对象技术和分布式计算发展的产物,它提供一种通讯机制,透明地在异构的分布计算环境中传递对象请求,而这些对象可以位于本地或者远程机器。在这些面向对象的中间件中,功能最强的是 CORBA,可以跨任意平台,但是太庞大;JavaBeans 较灵活简单,很适合于作浏览器,但运行效率差;DCOM 模型主要适合 Windows 平台,已广泛使用。但是 DCOM 和 CORBA 这两种标准相互竞争,而且两者之间有很大的区别,这在一定程度上阻碍了面向对象中间件的标准化进程。当前国内新建系统实际上主要是 UNIX(包括 LINUX)和 Windows,因此针对这两个平台建立标准的面向对象中间件是很有必要的。

3.5 事务处理中间件(TPM)

事务处理中间件是在分布、异构环境下提供保证交易完整性和数据完整性的一种环境平台;它是针对复杂环境下分布式应用的速度和可靠性要求而实现的。它给程序员提供了一个事务处理的 API,程序员可以使用这个程序接口编写高速而且可靠的分布式应用程序——基于事务处理的应用程序。

事务处理中间件向用户提供一系列的服务,如应用管理、管理控制、已经应用于程序间的消息传递等。常见的功能包括全局事务协调、事务的分布式两段提交(准备阶段和完成阶段)、资源管理器支持、故障恢复、高可靠性、网络负载平衡等等。

3.6 网络中间件

它包括网管、接入、网络测试、虚拟社区、虚拟缓冲等,也是当前研究的热点。

3.7 终端仿真/屏幕转换中间件

它的作用在于实现客户机图形用户接口与已有的字符接口方式的服务器应用程序之间的互操作。

4 中间件主要产品介绍

中间件是软件技术发展的一种潮流,被誉为发展最快的软件品种,目前国内外有不少厂商都是中间件技术的开发商和产品提供商。全球最大的独立中间件厂商 BEA 公司已开发出许多中间件产品,如面向消息中间件 BEA eLink,事务处理的中间件 BEA Tuxedo 等。除了 BEA 之外,像 IBM、Oracle 和 Microsoft 及 Sun 等大厂商也早已注重中间件的开发。IBM 在 90 年代及时调整了其产品策略,将中间件产品作为其今后开发的重中之重,并推出了面向消息的中间件产品-MQ Series;另外 IBM 则正在改进其 WebSphere 网站开发工具,以期将之改造成一个完整的电子商务平台。Oracle 公司也正式发布了 Oracle9i 应用服务器无线版,这是一个专门针对通信公司、消费类门户、应用服务提供商(ASP)及相关公司开发和部署无线 Internet 内容及应用服务的全套中间件。而微软则也在准备着自己消息中间件的无线版本。

在国内,也有好多中间件产品问世,东方通科技有限公司率先推出国内第一个企业级的网络商务中间件平台 TongWeb,而后又推出了消息中间件 TongLINK/Q、交易中间件 TongEASY 等中间件产品。2000 年 7 月底,金蝶公司又宣布投入巨资成立以中间件产品研发及销售为主营业务的 Apusic 软件公司,进军中间件市场。目前其首推产品是基于 J2EE 技术的 Web-Enabled 应用服务器,它实现了 EJB、HTTP SERVLET、SSL 加密及 XML 解析器等 J2EE 核心技术,是第一个具有自主知识产权、完全用 Java 开发的 J2EE 中间件产品,适用于独立软件开发商、系统集成商和电子商务公司。此外,国防科技大学和中科院软件所也在做中间件方面的工作。

5 小结和展望

中间件是属于计算机软件中比较底层的内容,它和计算机操作系统的关系是相当密切的,操作系统的一部分功能可以由中间件来实现,一些中间件的功能也可以由操作系统来实现。因此,操作系统和中间件会进一步融合,从而推动计算机软件体系结构的变革。其次,人们希望实现不同业务逻辑、不同功能的中间件能够集成在一起,像一个完整的系统那样协调一致地工作;这些集成的中间件服务应该封装在一个框架中。另外,中间件要涉及软件的所有标准、规范和技术,它含有更多的内涵,

(上接 73 页)

第三段则是一个缩放型的虚拟存储,它的一个块等于下层的三个块大小。虚拟存储层次图中不允许出现环路,更进一步,如果限制一个虚拟存储至多只能被封装一次,则虚拟存储层次图退化为文中例子所示的虚拟存储层次树。在虚拟存储层次图的每一层,可以有自己的存取策略和存取粒度,并负责实现上下层之间的数据转换。

在虚拟存储的基础上,较容易解决系统级的动态扩展问题。如前所述,传统的存储系统在系统级之所以难于动态扩充,其原因在于控制数据区中对数据区的数据块的描述使用的是绝对地址。笔者认为,只有将元数据区的数据块和数据区的数据块分别独立编址,即他们有不同的寻址基准点,才能较好地解决这个问题。这样,可以分别追加数据区和元数据区的空间,而不会产生相互干扰或对原有信息的破坏。而在虚拟存储的基础之上,则很容易实现这样的想法,从而做到系统的动态扩展。图 3 给出了一个简单的说明性的例子。

数据区数据块的寻址起点

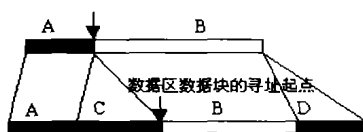


图 3 系统级的扩展

假定原来的系统由虚拟存储 A 和虚拟存储 B 连接而成,其中虚拟存储 A 存放的是控制数据,虚拟存储 B 是数据区;其中,A 中记录的关于 B 中的数据块的信息都是以虚拟存储 B 的起点为基准的,如图中的箭头所示。当为 B 追加空间-虚拟存储 D 时,如果需要增加控制数据且虚拟存储 A 的空间已经用满,则可以给控制数据区追加空间-虚拟存储 C,此时只须将数据区数据块的寻址起点后移。由于 A 中记录的关于 B 中的数据块的信息都是以虚拟存储 B 为起点的,这样,增加虚拟存

因为它包括平台功能,自身具有自治性、自主性、隔离性、社会化、激发性、主动性、并发性、认识能力等特性,是近似于 Agent (代理)的结构。最后,计算机网络的发展非常迅速,中间件技术要得到发展,就离不开网络。

因此采用 Agent 的概念和技术来建立中间件模型和规范是促进中间件技术发展的核心措施。而且中间件技术的发展,将有机地结合操作系统、系统集成以及网络技术的发展,成为推动软件技术发展的重要力量。(收稿日期:2001 年 12 月)

参考文献

1. 仲豪. 构筑分布式应用的关键技术[J]. 网络世界[EB/OL], <http://www.cnw.com.cn/cnw/2000>
2. 应用中坚-IBM 谈中间件[J]. 网络世界[EB/OL], <http://www.cnw.com.cn/cnw/2000>
3. Roger S Pressman. Software Engineering: A Practitioner's Approach[M]. 3rd Edition, McGraw Hill, 1992
4. 周之英. 现代软件工程(下)新技术篇[M]. 科学出版社, 2000
5. Ashish Singhai, Aamod Sane, Roy H. Campbell. Quarterware for Middleware[C]. In: Proceedings of the 18th IEEE International Conference on Distributed Computing System, Amsterdam, 1998: 192~201
6. Dale Rogerson 著. 杨秀章译. COM 技术内幕[M]. 清华大学出版社, 1999

储 C 对 A 中原有的元数据没有任何影响,从而可以做到系统动态扩充容量。

6 小结

该文探讨了构造可扩展存储系统,以满足不断增长的存储容量的需求的一种解决方式。通过定义这几种基本的虚拟存储的类型,利用它们的组合可以较好地实现系统的扩展性;任何存储设备都可以通过物理存储和虚拟存储层的转换后,供系统使用。但是,在海量存储系统方面,仍然有很多重要的问题需要探讨,如在提高系统的性能方面,有数据的放置和迁移策略,直至数据管理系统的设计等,这些都需要人们做更深入的研究。(收稿日期:2001 年 8 月)

参考文献

1. Gray Jim, Shenoy Prashant. Rules of Thumb in Data Engineering[C]. In: Proc of IEEE 16th International Conference on Data Engineering, San Diego, California, 2000-03: 3~12
2. R W Watson, R A Coyne. The Parallel I/O Architecture of the High-Performance Storage System (HPSS)[C]. In: Proc IEEE Symposium on Mass Storage Systems, Monterey, CA, 1995-09
3. R Grossman, D Hanley, X Qin. Caching and Migration for Multilevel Persistent Object Stores[C]. In: Proc IEEE Symposium on Mass Storage Systems, Monterey, CA, 1995-09
4. P Triantafillou, T Papadakis. On-demand data elevation in a hierarchical multimedia storage server[C]. In: Proc 23rd Very Large Database Conf, 1997: 226~235
5. IEEE Storage System Standards Working Group. Reference Model for Open Storage Systems Interconnection. Available at <http://www.sss-wg.org>, 1994-09
6. Storage Networking Industry Association. <http://www.snia.org>
7. VERITAS Volume Manager(tm) User's Guide, Release 2.5. VERITAS Software Corp., 1997