

分布式对象中间件技术的分析与研究^{*}

李 芬

(武汉理工大学 信息工程学院, 湖北 武汉 430070)

摘 要: 介绍分布式对象中间件技术, 对 CORBA、DCOM 以及 EJB 三大主流的分布式对象中间件规范以及实现技术进行了深入的分析与比较, 在此基础上给出了中间件技术的发展趋势以及有待解决的问题。

关键词: 分布式对象; 中间件; CORBA; DCOM; EJB

中图分类号: TP311

文献标识码: A

文章编号: 1672- 9846(2007) 03- 0081- 04

1 引言

随着计算机技术的不断发展, 企业的信息系统和以往相比, 有了巨大的进步。企业级的应用不再满足于单机系统和简单的客户机服务器系统, 而是向着更加合理的三层或多层体系结构的分布式环境不断迈进。在分布式环境中, 无论是硬件平台还是软件平台都不可能做到统一。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, 为了更好的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, 中间件技术就应运而生。

中间件是一种独立的系统软件或服务程序, 分布式应用借助这种软件可以在不同技术之间共享资源。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讯。中间件的目标是要解决系统的互连和互操作, 他的核心是解决名字服务、安全控制、并发控制、可靠性保证、效率保证等方面的问题, 要实现互联还要解决支持各种通讯协议、各种通讯服务模式、传播各种数据内容、数据格式翻译、流量控制、数据加密、数据压缩等。通俗地说, 中间件就是处于中间的一个软件层, 提供一些 API 供应用程序调用。它负责处理不兼容的操作系统或者文件结构, 调节客户与服务器或服务器与服务器之间的

通信, 互通互联应用软件中的各个构件。

按照通信模型的不同, 中间件可以分为 3 类, 即基于远程过程调用的中间件、基于消息传递的中间件和基于分布式对象的中间件。由于面向对象的技术一直是软件界努力追求的目标, 所以分布式对象中间件就受到了特别的重视, 本文的研究就是基于该类中间件的。

2 三种主流分布式对象中间件技术

2.1 CORBA

CORBA 是为解决分布式处理环境中硬件和软件系统的互联而提出的一种解决方案, 它具有不依赖于编程语言、软硬件平台、网络协议的特点, 已逐渐成为分布计算技术的标准。CORBA 主要分为 3 个层次: 对象请求代理、公共对象服务和公共设施。最底层是对象请求代理 ORB, 位于所有的 CORBA 进程中, 规定了分布对象的定义 (接口) 和语言映射, 实现对象间的通讯和互操作, 是分布对象系统中的软总线。在 ORB 之上定义了很多公共服务, 可以提供诸如并发服务、名字服务、事务 (交易) 服务、安全服务等多种服务。最上层的公共设施则定义了组件框架, 提供可直接为业务对象使用的服务, 规定业务对象有效协作所需的协定规则。CORBA 规范目前的发展增加了面向 Internet 的特性, 服务质量控制和 CORBA 构件模型。CORBA 技术规范利用了当

^{*}收稿日期: 2007- 07- 20

作者简介: 李 芬 (1983-), 女, 湖南岳阳人, 武汉理工大学信息工程学院硕士研究生。

今软件技术发展的最新成果,在基于网络的分布式应用环境下实现应用软件的集成,使得面向对象的软件在分布、异构环境下实现重用、可移植和互操作。

2.2 EJB

EJB是 Sun推出的基于 Java的服务器端构件规范 J2EE的一部分,自从 J2EE推出之后,得到了广泛的发展,已经成为应用服务器端的标准技术。Sun EJB技术是在 Java Bean本地构件基础上发展的面向服务器端分布应用构件技术,它基于 Java语言,提供了基于 Java二进制字节代码的重用方式。EJB给出了系统的服务器端分布构件规范,这包括了构件、构件容器的接口规范以及构件打包、构件配置等标准规范内容。EJB技术的推出,使得用 Java基于构件方法开发服务器端分布式应用成为可能。从企业应用多层结构的角度来看,EJB是业务逻辑层的中间件技术,与 Java Bean不同,它提供了事务处理的能力,自从三层结构提出以后,中间层,也就是业务逻辑层,是处理事务的核心,从数据存储层分离,取代了存储层的大部分地位。从分布式计算的角度来看,EJB像 CORBA一样,提供了分布式技术的基础,提供了对象之间的通讯手段。从 Internet技术应用的角度,EJB Servlet和 JSP一起成为新一代应用服务器的技术标准,EJB中的 Bean可以分为会话 Bean和实体 Bean,前者处理会话,后者处理事务。Servlet负责与客户端通信,访问 EJB,并把结果通过 JSP产生页面传回客户端。

2.3 COM /DCOM

DCOM (D istributed Component Ob ject M odel)是微软公司与其他厂商合作提出的一种分布组件对象模型,是微软的组件对象模型 COM (Component Ob ject M odel)的一个分布式的扩充。它主要是为不同网络环境中的分布对象提供交互的标准。

组件对象模型 COM 是一种以组件为发布单元的对象模型,这种模型使各软件组件可以用一种统一的方式进行交互。COM 组件是遵循 COM 规范编写的,以 W IN 32 动态链接库 (DLL)或可执行文件 (EXE)形式发布的可执行二进制代码。COM 组件能够满足对组件架构的所有需求,可以为应用程序和操作系统以及其它的组件提供服务。在 COM 应用程序中,客户应用程序与组件间的工作方式是采用客户服务方式的。客户端通过获得的 COM 对象界面指针,与 COM 对象通讯,调用 COM 对象所提供

的服务。

DCOM 是 COM 发展的新阶段,它扩展了组件对象模型技术,使其能够支持在局域网、广域网甚至 Internet上不同计算机的对象之间的通讯。使用 DCOM,应用程序就可以在位置上达到分布性,从而满足客户和应用的需求。COM 运行库向客户和组件提供了面向对象的服务。并且使用 RPC 和安全机制产生符合 DCOM 线路协议标准的标准网络包,对于客户和组件分别运行在不同机器上的情形,DCOM 只是简单地把本地跨进程通信用一个网络协议传输来替代,客户代码和组件对象都感觉不到。

3 三种分布式对象技术的比较

CORBA 采用的接口定义语言是 OMG IDL,它采用类似于 C++ 的语法,并提供到多种语言的映射,包括 Java C++、C等,可以在使用不同语言的客户机和服务器之间实现异构通信。DCOM 相对来说更适合与 C++ 紧密集成,因为 DCOM 对象之间需要有类似于 C风格的指针传递,而对于根本不支持指针的 CORBA 和 JAVA 语言而言,开发 DCOM 分布式应用相对困难。EJB 是基于 JAVA 技术体系的组件,因此它只能使用 Java语言和 Java RMI API进行接口定义和应用开发。

在这三种技术中,CORBA 提供了一致的多功能的对象模型,具有高度的透明性,客户不用区分本地和远程对象,对象拥有自己的状态,可以被全局标识。DCOM 的组件对象都是暂时对象,没有全局标识,违反了很多分布式技术的原则。EJB 组件将面向对象的思想引入了一个更高的层次,使模块的可重用性更高,适用范围更广。另外,由于 DCOM 目前不管是在操作系统、WEB服务器和浏览器方面,都只是基于 Microsoft平台,因此只适合基于 Microsoft Window平台的分布式系统的构建。EJB所能适应的平台相对较多,几乎支持所有的操作系统和WEB服务器、浏览器产品。EJB使用 Java远程方法调用接口 RMI它是以 Java为中心的分布式对象通信模型。基于 RMI的 Java客户可以利用 Java远程方法协议 JRMP (Java RemoteMethod Protocol)作为通信传递协议。CORBA 组件模型提供了一种以异构语言实现并且运行在异构的硬件和操作系统平台之上的客户和服务端都可以进行交互的标准模型。CORBA 使用 IDP和 GIOP作为通信层协议,两个协议从本质上讲都十分简单,但提供了建立可扩展的 CORBA 服务器的能力。

DCOM 没有提供自动的容错和负载平衡服务,而是由微软的事务服务 MTS(Microsoft Transaction Service)提供事务处理回滚和负载的平衡。如果客户不愿使用 MTS,微软还提供了一个让客户检验是否和服务器还存在连接的迂回协议,如果探测到服务器已死或者连接已被中断,周密设计的客户程序可以在必要的时候重新连接服务器,并处理丢失或被破坏的数据。但是在实践中,回溯一个程序或者手工恢复数据是很繁琐的工作,可能会花费很长时间,而且让大量对象在网络中迂回将是一个很巨大的开销。CORBA 规范中的对象事务服务 OTS(Object Transaction Service)为分布式 CORBA 对象实现事务行为定义了接口,它支持平面事务和嵌套式事务。加上 OTS 倚赖于 X/Open DTP 标准,所以不是基于 CORBA 的应用程序也可以与 OTS 互操作。EJB 使用 Java 事务服务 JTS(Java Transaction Service)提供分布式事务服务,基于 Java 的应用程序和应用程序服务器通过 Java 事务接口 JTA(Java Transaction API)访问事务管理功能,JTA 通过 JTS(JAVA Transaction Service)与事务管理器实现交互,JTS 可以说是 OTS(OMG Object Transaction Service)的 Java 映射。

4 对象中间件的发展趋势

中间件,从本质上是对分布式应用的抽象,因而抛开了与应用相关的业务逻辑的细节,保留了典型的分布交互模式的关键特征。经过抽象,将纷繁复杂的分布式系统经过提炼和必要的隔离后,以统一的层面形式呈现给应用。从中看出中间件具备构件的一些基本属性,中间件本身作为软件产品,构件化的软件开发对中间件同样适用。基于构件技术的中间件的实现包括构造可灵活组合的构件集,维护这些构件在不同的约束条件下如何产生有意义配置的知识以及通过构件状态管理功能保证系统状态在配置变化时的完整性和连续性,这些都是基于构件技术的中间件研究的关键。

尽管对象中间件技术发展到现在比较成功,而且相应的一些标准也比较完善,然而当前的中间件技术,无论是 CORBA、DCOM 还是 Java RMI 基本上都采用了黑箱抽象的原则,它们在灵活性和适应性方面存在先天不足。随着新的应用形式的出现以及传统应用环境的改变,中间件系统不断面临着一些新的挑战,例如群件、多媒体、实时以及移动系统所提出的需求。这些应用系统都具有一个共同的特

点,它们都需要根据运行环境的改变而动态地改变。例如,移动系统需要监测和适应连接的变化;移动环境中的分布式多媒体应用必须根据用户连接方式的改变而提供相应的服务,比如改变流量等。为了适应运行环境的改变,支持动态的配置,传统的中间件平台必须引入新的反射机制来实现具有开放性的中间件系统;也就促使了像 Open ORB 这样的反射中间件平台的出现和发展。反射中间平台具有可配置性、动态重配置性和扩展性特征,能满足特定应用的需求。随着移动计算、网络计算技术、多媒体技术的进一步发展,处于开放环境中的反射中间件系统将面临更多的挑战,并由此而产生一系列的技术问题,如系统资源的动态管理、反射过程的一致性维护、对自主性服务实体的反射等,有待进一步研究。

分布式对象的三大主流技术都有一个缺陷——属于 RPC 模型,这种模型要求客户端和服务端是同质的而且必须紧密耦合在一起,否则就不能实现互操作,这使得它们十分脆弱且扩展性差。在此情况下,一种能够方便地实现 Internet 上跨平台、联接松散、应用程序不受端口变化的分布式计算模式应运而生,这就是面向服务的 Web Services(Web 服务)。Web Services 采用的是面向服务的模型,通过通用的网络协议和数据格式来访问,是基于 Internet 的,它通过 URL 来标识目标服务,使用 XML 作为数据描述方式,请求方与服务方之间是非常松散的耦合关系,整个系统是开放的模式,扩展性好且易于集成,用户只要能够创建并使用为 Web Services 接口定义的消息,就可以用任何语言在任何平台上调用它。虽然目前 Web Services 还处于发展的初期,其技术和实际应用还面临着不少挑战,但 Web Services 仍然以其在分布式应用方面所展现出的强大生命力而成为一个研究热点。OGC 认为它提供了一个可互操作的、开放的、动态链接的空间信息服务网络体系平台,具有广阔的应用前景。

5 总结

分布式对象技术已经成为建立服务应用框架和软件构件的核心技术。本文主要介绍了分布式对象技术的基本思想及关键技术,并从几个方面对比介绍了 DCOM、J2EE 和 CORBA 三种主流的分布式技术。就这三种主流技术来说,DCOM 对基于微软的系统有较好的支持,但不适合应用于异构的系统。CORBA 则具有较强的通用性,可以方便的应用于异构系统中,从某方面也增加了其复杂性。EJB 具有

良好的跨平台性, 但只适合 Java 开发的系统。

国内外软件产商均在其主流产品中支持 CORBA、EJB 和 DCOM 技术以支持企业级应用和 Web 应用, 这些产品在企业 ERP 和电子商务中均得到大量的应用。分布对象技术在比较大型的网络系统中应用较多, 比如像医学方面的医学信息系统、银行业、交通、电信, 几乎各行各业都有成功的例子。目前, Java 和 CORBA 的融合成为解决分布式计算问题较流行的技术, 在 Web 上应用也有了很解决方案。微软公司的 .NET 技术的应用也使我们感受和体验到分布式对象计算应用的广阔前景和美好未来。但分布对象技术应用起来比较复杂, 涉及到网络、对象技术等多种技术, 另外在大型软件中采用分布对象技术, 在项目开发过程的控制方面与 traditional 方法比经验不足, 所以在一定程度上限制了分布对象技术的应用。因此, 进一步对分布对象成熟的技术进行研究和应用实践, 需要有一系列方法论、软件工程的方法和工具来支持。分布式对象技术将向

着智能化、小型化和标准融合的方向发展, 并在移动计算和电子商务的应用中发挥巨大的作用。

参考文献:

- [1] Carl- Fredrik Sørensen. A Comparison of Distributed Object Technologies[DB/OL]. <http://www.idi.ntnu.no/~carlfrs/publications/essay2001-calle.pdf>, 2004-01-25.
- [2] Douglas C. Schmidt. Distributed Object Computing and CORBA Documents[DB/OL]. <http://siesta.cs.wustl.edu/~sdmilt/corba.html>, 2004-03-05.
- [3] Morgenthal J.P. The real impact of XML eReal impact on B2B integration[J]. eAI Journal 2001(7): 42-44.
- [4] 李文军, 周晓聪, 李师贤. 分布式对象技术[M]. 北京: 机械工业出版社, 2004.
- [5] 赵靖, 李石君, 刘海青. 基于对象的分布式系统比较[J]. 武汉大学学报(理学版), 2005, 51(S2): 151-154.
- [6] 李纪云, 董小社, 童端. 分布式对象中间件技术研究[J]. 计算机工程与设计, 2004 25(2): 170-173.

(上接第 80 页)

④(X, Pos) = ReadMatrix(A₂, Pos); //读取当前路径上的下一个字符

⑤i = 2 //循环中实际上从第二个字符开始比较

⑥While(X = A_i && i < n) { //依次比较当前路径上的字符与查询串上相应的字符

⑦Path = Path + (X, Pos)

⑧(X, Pos) = ReadMatrix(X, Pos); //读取当前路径上的下一个字符

⑨}

⑩if(i = n) Result = Result + Path //如果当前查询路径与查询串完全匹配, 则把当前查询路径加入结果集合

⑪return(Result);

模型性能分析:

索引建立的时间复杂性: O(N);

查询时间复杂性: 查询字符串 A₁, A₂, ..., A_n 时, 需要遍历所有与此查询串具有相同前缀的字符串, 直到该字符串与查询串不符或者完全相

符。设全文中 A₁ 出现 f₁ 次, A₁, A₂ 出现 f₂ 次, A₁, A₂, ..., A_n 出现 f_n 次, 则算法需读取磁盘 f = $\sum_{i=1}^n f_i$ 次。若所有与查询串 A₁, A₂, ..., A_n 具有相同前缀的字符串的平均长度为 r, 则算法需要读取磁盘 O(rf) 次, 模型的检索效率高于倒排表;

空间复杂性: O(N); 由于可以通过互关联后继树还原原文, 不需要保留原文档, 因此其膨胀比为 1。

参考文献:

- [1] ZHOU Shuigeng, HU Yunfa, GUAN Jhong. Adjacency Matrix Based Full-Text Indexing Models 软件学报[J]. 2002, 13(10): 1933-1942.
- [2] 刘兴宇. 基于倒排索引的全文检索技术研究[D]. 华中科技大学, 2004.
- [3] 陶晓鹏. 面向(中文)全文数据库的全文索引的研究[D]. 复旦大学, 2000.
- [4] 申展, 王建会, 吴爱华, 胡运发. 互关联后继树模型——一种新颖的全文检索模型[J]. 计算机科学, 2003 30(10): 351-354.