

从分布式对象到 Web 服务^{*})

From Distributed Object to Web Service

吴敏强 张 潇 杨书慧 陈道蓄 谢 立

(南京大学软件新技术国家重点实验室 南京210093) (南京大学计算机科学与技术系 南京210093)

Abstract It is distributed object technology that combined by network and object oriented technology in 90's last century, and it is difficult to extend across Internet. Contributed by HTTP and XML, Web service is becoming the component called through the Web. In this article, we describe and analyze some current distributed object technologies, and then introduce the emerging Web Service technology. We introduce and analyze the two software strategies: Microsoft .NET and Sun ONE. Web Services will transform the current isolated Internet Web into interconnected services.

Keywords Distributed objects, Web services, Microsoft .NET, Sun ONE

1 引言

分布式对象技术是上个世纪90年代,针对RPC这种比较原始的C/S模式的不足而提出来的。它的一个基本思想是构件。OMG的CORBA是其中最早的也是比较成熟的代表,目前,CORBA、COM/DCOM和Java/RMI是分布式对象技术的三个种主要实现。

Web服务(Web Services)是近几年才提出来的一个概念,它的基本思想则是软件是一种服务。开发和提供Web服务,是Microsoft.NET战略和Sun ONE架构的主要功能。目前,.NET风头正盛,而Sun、IBM等也纷纷提出支持Web服务。

Web服务是分布式对象技术针对应用需求的进一步扩展。这可以从两个方面来看。

在企业计算中,原始的C/S模式下,开发人员通过集成本地系统服务构建应用程序;分布式对象及构件技术的出现,n层计算的流行,使开发人员可以专注于应用逻辑,企业网内部的分布式应用的开发变得相对容易。但是,如果企业计算需要扩展到Internet,分布式对象在跨越企业防火墙时就会遇到困难。把分布式计算的高生产率、紧耦合的特点和Web面向消息、松耦合的特点有机地糅合在一起就形成了一种新的计算模式——Web服务。Web服务使用标准的Internet协议,像HTTP和XML,将功能展示在互联网和企业内部网上。Web服务可以看作Web上的构件。

从Web技术的发展来看,最早主要用静态的HTML作页面显示,用http来传送HTML,用CGI进行业务处理。随着应用和技术的发展,Web技术的客户端和服务端越来越灵活。客户端有了Applet、ActiveX等动态技术,服务端的一个趋势则是Web和事务处理分开,成为三层结构。用CORBA、JAVA/RMI构筑多层结构,增加服务端的灵活性和处理能力。Web技术的进一步发展,不仅是所有网站,而是所有的计算机群、相关设备和服务协同工作,提供更加广泛和多样的服

务,而不是像现在这样的信息孤岛。包括XML和SOAP在内的新标准将把信息从服务器中解放出来,使它们能够被重新组织、调整和编程,然后以任何可能的方式、在任何设备和系统上显示出来。开放的Web服务是其基础设施。

本文的主要内容是分析比较目前比较流行的几种分布式对象技术,介绍新兴的Web Services概念,并着重业界的两个主要代表:Microsoft.NET战略和Sun ONE战略。

2 几种分布式对象技术的比较与展望

分布式对象技术是随着网络和面向对象技术的发展而兴起的。与传统的Client/Server模式相比,它通常表现为多层结构,能够适应更大规模的应用,并且能够通过中间件机制来屏蔽网络硬件平台的差异性和操作系统与网络协议的异构性,从而适应异构的网络计算环境。分布式对象技术的基本概念之一是构件(component)。构件是一些独立代码的封装体,用来在分布计算的环境下提供一定的服务,可以跨越平台、网络、语言、应用程序、工具、硬件而运行。在构件的基础上,又产生了软件总线的概念,符合软件总线标准的构件,可以无缝连接,即插即用。

目前,主流的分布式对象技术主要有OMG的CORBA,Microsoft的COM/DCOM,以及JAVA/RMI。这三种实现方式中,CORBA最受学术界推崇,是分布式对象技术研究的典型,目前还是一个研究的热点。CORBA的内容大而全,具有良好的跨平台性、强壮的编程语言支持和安全性能,是发展最早也是最完善的。但是,缺少方便的开发工具,缺少强有力的厂商支持,是它目前遇到的两个问题。COM/DCOM技术经过微软的力推,在实践中已经相当成熟。它有许多优秀的开发工具支持,有许多现成的构件可以使用,能够帮助开发者在微软的平台下快速高效地完成开发工作,这使得它成为目前应用最广的构件技术。COM/DCOM技术的一个致命弱点就是其跨平台性比较差,如何实现与第三方厂商的互操作性始终是它的一个大问题。Sun公司也提供了一整套的Java分布式计算解决方案,即RMI(Remoter Method Invocation)和EJB

^{*})本文写作得到863项目(编号2001AA113050)的资助。吴敏强 硕士生,主要研究方向为分布计算,网络多媒体。张 潇 硕士生,主要研究方向为分布式计算。杨书慧 硕士生,主要研究方向为并行计算。陈道蓄 教授,博导,主要研究方向为分布式计算,网络。谢 立 教授,博导,主要研究方向为并行、分布计算,网络安全。

(Enterprise JavaBean),其方案由于出现较晚而在技术上比较先进。目前,在企业级的服务器端,Java EJB 架构已经拥有很大的市场。

CORBA 目前主要发展的方向在于进一步适应技术发展和应用的需求,如实时应用、安全的 CORBA、嵌入式应用和移动计算中的应用等。COM/DCOM 后,微软又推出了 COM+,在后面将要介绍的 Microsoft.NET 中,虽然已经没有 COM 这个名词,但是实际上.NET Framework 采用了很多 COM/DCOM 的思想。Java 吸收 CORBA 技术的优点,或者说正与 CORBA 融合,取得了长足的进步。RMI 已经建立在 CORBA 的 ORB 通信机制 IIOP 之上。通过 IIOP 协议,EJB 可以和其他的非 Java 客户机进行通信,如 EJB 可以存取 CORBA 服务器,CORBA 客户机也可以存取 EJB 服务器。J2EE 已经成为企业级计算的一大标准,同时也是 Sun ONE 战略核心技术之一。

3 Web Services 简介

Web Services 是最近一两年才热起来的,但是服务(services)本身并不是新的概念。像 RPC,COM,CORBA,RMI 和 JINI 等技术都可以称之为面向服务(Service-oriented)的系统。随着应用的发展,从跨越 Internet 的需要来说,新的利用通用互联网协议的面向服务的系统——Web Services 就成为必要。从商业应用角度看,太过于紧耦合的分布式对象技术也不适用于 B2B 这样的电子商务应用。IBM 认为未来的应用必然是动态发现和配置的服务的组合,特别是在未来基于 Web 的电子市场中^[8],Web Services 是这种理念下的下一代面向 Web 的电子商务构架。Web Services 也是 Microsoft 将基于 COM/DCOM 的 n 层计算模型扩展到 Web 的结果,以实现其.NET 战略的宏伟构想。

IBM 认为是她最早提出了“Web Services 革命”^[7],2000 年 5 月,IBM 和微软等公司合作推出了 SOAP 1.1 规范,并首先推出了基于 Java 的开发工具包,而 SOAP 则是 Web Services 的一个关键技术。随后,2000 年 6 月微软公布其宏伟的.NET 战略。而 Sun 直到 2001 年 3 月才正式公布其 Sun ONE 战略。

IBM 认为 Web Services 是能够被描述、发布、定位和通过网络(一般是 WWW)调用的自包含的模块化的应用^[8]。Sun ONE 中的 Web Services 是表示一种能够通过 Web 存取的事务(business)、应用或系统功能^[2]。微软认为 Web Services 是一个通过标准的 Internet 协议访问的可编程的应用逻辑^[10]。

从上面三个定义可以看到,Web Services 的概念还是比较统一的,都是可以通过 Web 调用的应用逻辑或功能,Web Services 实际上就是构件技术和 Web 技术的融合。

考虑实现 Web 上可调用的构件,构件本身的功能实现不是问题,问题在于怎样对服务进行描述,怎样对服务进行注册,怎样发现服务,怎样定义调用服务的消息格式等等。这里主要用到 XML 技术以及其他的相关协议,如 SOAP,UDDI,WSDL 等。各个公司的技术可能各有特点,但是为了达到能够互相协作的目的,最终的标准必须是兼容的。

下面我们介绍一下实现 Web Services 共有的一些标准和协议。

XML 在这里是一个核心技术,由它衍生出来的相关协议,如 SOAP,WSDL 等是 Web Services 的基础协议。甚至有

人这样简单地说:微软的.NET 就是 C# 加上 XML,而 Sun ONE 则是 Java 加上 XML。XML (Extensible Markup Language)是 W3C 的一套标准,是用来定义语言的语言,即它是一种元语言。在不同的应用中,XML 可以定义不同的 Schema,这样就可以有不同的语言,如 HTML 就可以由它定义,SOAP 的一部分就是 XML 的一个 Schema。WSDL (Web Services Description Language)是一种由 IBM、Microsoft 和 Ariba 推出的技术,它实质上是一个通用的描述 Web 服务的 XML 框架。XML 的传输协议用的是 HTTP,这就很容易穿过企业防火墙。XML 是一个与平台无关的中性的数据表示协议,XML 把数据序列化成一个可以传递的形式,使得它能够容易地在任何平台上被解码,对于简化企业之间的商业数据交换起到很大作用。

SOAP (Simple Object Access Protocol)是一个基于 XML 的简单的轻量级协议,用于在 Web 上交换结构化的有类型的信息。SOAP 主要由 IBM、Microsoft 和 DevelopMentor 开发推出。作为用于 XML 消息传递的一种普遍的协议,SOAP 正在迅速崛起。SOAP 规范目标有两个方面:一是提供一种建立在 Internet 标准之上的标准对象调用协议,使用 HTTP 来承载消息,使用 XML 来格式化消息;二是创建一种可扩展的协议和编码格式,以便能够随着时间的推移而进化。SOAP 能与互联网上已有的各种协议和规格一起使用,包括 HTTP、SMTP 及 MIME,可以支持从消息系统到 RPC 的广泛的应用。SOAP 是“一个没有发明任何新技术的技术”,是一种用 HTTP 和 XML 在客户和服务端之间传递命令和参数的机制,可以跨语言、跨操作系统进行远程过程调用。SOAP 协议包括三个部分:封装结构(envelope)、编码规则和 RPC 表示。封装结构定义了一个整体框架,描述消息中包含什么内容及如何处理;编码规则定义了应用程序定义的数据类型的实例;RPC 表示则是一个用来表示远程过程调用和应答的约定。所有的 SOAP 消息用 XML 编码。

UDDI (Universal Description, Discovery and Integration)标准定义了 Web 服务的发布与发现的方法。UDDI 提供了一种基于分布式的商业注册中心的方法,该商业注册中心维护了一个企业和企业提供的 Web 服务的全球目录,其中的信息描述格式是基于通用的 XML 格式的。商业注册是 UDDI 的核心组件。UDDI 规范在 XML 和 SOAP 的基础之上定义了新的一层,在这一层次,不同企业可以用相同的方法描述自己所能提供的服务和查询对方所能提供的服务。UDDI 规范包含了对基于 Web 的 UDDI 商业注册中心可以实施的整套共享操作。从商业层次上来说,UDDI 商业注册中心可以被用于核查某个合作伙伴是否拥有特定的 Web 服务的调用界面,或是去找出在某一个行业中能提供某类服务的公司。

由上面的技术介绍可以看到,关于 Web Services 的大多数技术是由 Microsoft 和 IBM 共同推动和开发的。微软在 XML、SOAP 和开发工具方面处于领先地位,IBM 则没有推出明确的软件策略,Sun 则是刚刚起步而已。

4 Microsoft.NET 与 Sun ONE 战略简介与评述

2000 年 6 月,微软公司正式向全世界宣布了其宏伟的 Microsoft.NET 战略。之后,2001 年 3 月,Sun 推出了一个和微软竞争的 Sun ONE 战略。在这之前,微软已经推出不少用于.NET 的 SDK,其下一代开发工具 Visual Studio .NET 的试用版已经推出,Web Service 主力开发语言 C# 及 CLI (Com-

mon Language Infrastructure)已经成为欧洲计算机制造商协会(ECMA)的工业标准^[13]。由于微软的强大影响力,以及它在世界各地的宣传,掀起了一股学习.NET技术的潮流。反观Sun,好像就没有这么大的反响了。但是Sun ONE是目前唯一和MS.NET对抗的战略。下面,我们将分别介绍这两大战略。

1. Microsoft.NET 战略

关于Microsoft.NET,给人的感觉是一个极其全面、庞大的系统。微软中文网站的白皮书^[1]是这样描述它的:“微软公司正在创造新一代的高级电脑软件,它革命性地综合了计算与通信技术,为开发商们提供了工具,致力于网络和其他计算技术的变革。我们把这项创新称为Microsoft®.NET,它第一次使开发商、企业和消费者能够按照自己的意愿来运用科学技术。Microsoft®.NET将创造出真正的分布式网络服务,它能够整合并协调为数众多的辅助程序,为消费者提供独到的服务——而这正是今天的网站所梦寐以求的。Microsoft®.NET将推动新一代因特网的发展。它将真正地让人们可以在任何时间、任何地点、通过任何设备得到信息。”

Microsoft®.NET背后的基本理念就是:不再关注单个的网站和单个设备,而是要让所有的计算机群、相关设备和服务商协同工作,提供更加广泛和丰富的解决方案。人们将能够控制何种信息、在何时、以何种方式传送给自己。

要达到这样一个目标,微软认为XML技术是关键,而Web Service则是运行的形式。微软的.NET英文主页^[11]上的FAQ中,直接把Microsoft.NET定义为微软的XML Web Service平台。这个平台包括一个建立.NET体验(experiences)的软件平台(.NET Platform);一套建立和整合XML Web Service的编程模式和工具(.NET Framework);一套可编程的XML Web Services(第三方Web Service)。

在微软的Web Service体系中,包括了三个基本部分:SOAP、SOAP Discovery和WSDL。SOAP规定了数据及调用命令在XML中的表示,SOAP Discovery则规定了如何在互联网上发现Web服务,而WSDL则描绘了Web Service本身的格式,即它可以接受的数据和可以理解的方法和命令。

Microsoft.NET的开发工具Visual Studio.NET遵循的是.NET Framework。在这个体系里,它提供了一个通用语言运行时CLR(Common Language Runtime),这是给所有可以在.NET框架里使用的语言共享使用的。所有的语言都必须经过修改,遵循CLR规范,才能在.NET中使用,并且都只能编译成中间语言(IL)代码(C++可以除外),运行时再进行即时编译(Just-In-Time compile)。在这个框架下,理论上,程序员可以使用任何它所熟悉的语言开发Web Service,因为所有语言调用的是同一套类库。微软还推出了一种类似于C++和Java的语言C#(C sharp),专门用来开发Web Service。在这个.NET Framework的上层,微软推出了ASP.NET,它提供外观上类似于Win Form(windows窗体)的Web Form。这种漂亮的控件可以自动转化为标准的HTML,方便了开发人员。

微软在Microsoft.NET战略上下了重注,它以后的所有产品都将围绕这个战略进行。微软认为这个战略将对消费者、企业和独立软件开发者产生重大的、有益的影响。对于消费者,他将得到简单化的整体服务,随时随地的个性化和绝对自由;对于企业,将得到得力的信息管理和电子商务工具,它可以在基于内部网和因特网的服务程序之间灵活地切换,为

动态商务伙伴关系的时代提供支持;对于独立软件开发商,意味着他们将得到更多的机会,为因特网时代创造更多的新型高级服务。

2. Sun ONE 战略

2001年3月,Sun微系统公司首席执行官Scott McNealy在旧金山发布了其新的软件战略——Sun ONE。Sun ONE是“Open Network Environment”英文单词的缩写,是为网络服务提供创建、聚集、配置的基础建设,是软件开发工具、网络基础软件和操作系统三种不同领域软件的大融合。

Sun ONE包括Sun公司的J2EE和EJBs以及其他为i-Planet电子商务解决方案开发的应用服务器产品。Sun ONE的关键部分是一系列软件开发工具,它可以帮助程序员们利用Java语言开发基于Web的各种应用,使软件开发人员能够将大型电子商务软件拆分为他们可随意定制模块。对此,Scott McNealy称“智能网络服务对于信息时代而言就象可互换的零部件对工业时代一样重要”。Sun目的就是希望利用成套的工具包吸引程序开发人员,使他们远离竞争对手,尤其是微软公司。

Sun ONE白皮书^[2]提出了两种Web Services,一种就是通常意义上的Web Services;另一种是进一层的Smart Web Services。前一个Web Services,定义为能够通过Web存取的业务、应用或系统功能的单元,而Smart Web Services则是一个能够理解情形上下文(situational context)及和其他Services共享上下文的Web Services。它们主要的不同有:(1)上下文(Context);(2)多网(Multi-net);(3)服务质量(QoS)。这里的所谓上下文,是指能够利用一些相关的信息,如身份、角色、位置、时间、profile、限制,以及与请求相关的其他信息。这就是说,Smart Web Services能够知道“who, what, when, where, and why”等问题。所谓多网,指Smart Web Services能够部署在不同的网络,如IP网,电话网,GPS等。QoS是指它能针对不同的业务,提供不同的QoS,举例来说,一个订购\$12书的请求,就不需要和一个\$12,000,000的汽车订单同样的确认。这个Smart Web Services带来的结果就是更丰富的用户体验(experience)。

Sun认为,Web Services是一种全新的计算模式,将会很快取代目前的单机维护模式。而Web Services将会很快进化到Smart Web Services,从而提供更大的价值和更多的用户体验,它的定制化(customization)和个人化(personalization)将会超越目前通用的Web Services。Sun ONE就是这个思想下的产物,Sun给它的定位是它是一个开放的框架,能够支持今天的Web Services,并为明天的Smart Web Services打下基础。

在技术和标准方面,Sun采用的也都是现成的技术,这些技术分为3个方面:

1)XML标准及其衍生技术。包括核心W3C XML建议,XML parser,表示格式(XHTML, Voice XML等),XML Schema系统,XML消息系统(如SOAP, ebXML消息服务及W3C的标准),XML注册和仓储(UDDI, ebXML服务),服务描述语言(WSDL),XML管理信息交换框架,B2B XML框架和XML安全系统等。

2)Java技术。包括Java平台(J2EE, J2SE和J2ME,以及移动信息设备轮廓(Mobile Information Device Profile)和Java Card API等),Java serverlet, JSP, JDBC,以及Java API for XML(包括用于处理,数据绑定,注册和传送消息的

API)。

3) 基础架构的标准,如 HTTP,SSL 和 LDAP 等。

Sun 公司认为,企业部署整合 Smart Web Services 可以给他们顾客更多的服务,如透明的外部采购,自动审核和管理;而公司也可以从中获利,因为内部网络和 Services 的互连可以提供很多的服务和便利。服务提供者可以提供和竞争者完全不同、针对不同客户的 Smart Web Services,给用户以改进的个性化的网络体验。开发人员也将获利,因为他们开发 Service 的过程可以协同、整合、个性化其他的 Services。随着更多的设备和服务被部署,同一信息可以用不同的形式在适当的设备上展现。

除了 Microsoft 与 Sun,其他公司也纷纷支持 Web Services。IBM 虽然没有明确的软件战略,但其电子商务应用框架就是基于 Web Services 的,它在标准和开发工具方面走在很前面。它的 Web Services Toolkit 已经出到 1.2 版本,包括了对 SOAP,WSDL,UDDI 等的 API。IBM 宣称其开发工具将支持一切开放的和主流的技术。其他一些大公司像 HP 的 e-services 方案和 Oracle 的动态服务框架等都是支持 Web Services 的。由此可见,Web Services 确实是业界认同的潮流,是下一代 Web 的方向。

微软和 Sun 是业界软件和硬件的两大巨头,微软是桌面的霸主,目前正向网络转型;Sun 是一个以硬件为主的公司,也是“网络就是计算机”的倡导者,它的 Java 取得了巨大成功。Sun 在 8 个月之后,匆忙推出了这个 Sun ONE 战略,明显是感受到了来自微软巨人的压力。下面将从多个角度对比两大战略:

1) 战略的内容,两者基本相同。两者都力图建立一个超越目前的 Web 形式的新一代网络计算模式,其思想都是要实现自由联结所有的网站和设备。Sun 提出了 Smart Web Services 这个新概念,应该说,在微软的 .NET 战略里,同样的功能也是能够实现的。这一点从两个战略中描绘的未来网络应用场景看起来,是极其相似的。

2) 两者在服务描述、交互方面是一致的。在服务本身开发工具上,两者是不同的,但在服务的描述、注册、定位和调用方面,两者采用的都是类似的基于 XML 的相关协议和技术。这是因为双方在实现时必须考虑和其他系统的交互,至少在名义上必须如此。

3) 开发运行框架上,两者也开始接近。在微软的 .NET 中,出现了通用语言运行时 CLR 这个类似于 Java VM 的东西。所有语言都必须先编译成 CLR 上的中间代码,由 CLR 运行时及时编译。CLR 负责语言底层的细节,如对象的创建和释放、安全性管理等。这个 CLR 明显是借鉴了 Java VM 的思想。CLR 的存在,主要是为了实现语言无关性,各种语言可以方便地互相调用,即使要实现跨平台也是可能的。而在 Java 语言中,所有源代码都先编译成虚拟机上的代码,运行时解释执行,可以比较容易地跨平台运行。

4) 开发语言方面,两者极其不同。在 .NET 中,理论上可以用所有语言开发,C# 是其主力开发语言,而在 Sun ONE 中则是 Java。在 .NET 中,所有语言(除了 Java)都可以在 CLR 上运行,可以自由互相调用。微软由于许可证的问题,已经放弃 Java 作为其开发语言之一,这对于微软和广大程序员都是一大损失。

5) 台移植性问题。.Net 核心只工作在 Windows 环境下,但从理论上讲可以支持多种语言开发。也就是说:SOAP 的能

力允许在其它平台上的构件和 .Net 构件进行数据交换。而 .Net 中的一些元素,作为公共部分提供构架的核心仍由微软掌握。微软扮演的是整个 .Net 开发环境和运行时环境提供者的角色,微软的前提是“世界上只有 Windows 平台”,因此它不会将这一套移植到其他平台的。另一方面,Java 构件只要遵循 Java VM 和一组平台需要的服务就可以在任何平台上工作,所有这些定义都已经公开发表。但是 Java 是单语言的,若用其它语言调用或访问对象,可能需要通过 CORBA。

从上面的比较可以看出,Sun ONE 与 Microsoft .Net 几乎正在做同样的事情,两者在体系结构上不存在竞争,但是彼此用了不同的方式和工具之后,客户面临的选择则是两者取其一:Java 还是 C#;ONE Webtop 还是 Office .Net;Forte 开发工具还是 Visual Studio .net。在各个层面,Sun 与 Microsoft 都可谓针锋相对。由于微软在保证 Java 兼容性和终止使用 Java 之间选择了后者,两公司之间的竞争从 Java 以及它对 Windows 操作系统的威胁,升级到了企业将选择 Unix 还是 Windows 作为网络服务平台,也就是说,是选择 Java 语言,还是微软的开发工具。

从目前来看,Sun ONE 还很成熟,而 Microsoft .NET 已经走在了前面。虽然,从某种意义上讲,Java 实际上已经成为企业和网络应用的标准语言,非常深入人心;但是 Sun ONE 的其他部分,如 iPlanet、Forte 开发工具及 Sun ONE WebTop 组件,或者不能用于 Windows 平台,或者应用远不如微软的相应产品。对微软而言,桌面计算的优势是压倒性的。在 XML、SOAP 方面,微软已经远远领先于 Sun,在 Web 设计方面,ASP.NET、Web Form 和 ADO.NET 已经做得比 JSP 好了。目前,业界和微软自己都认为,Microsoft .NET 最大的困难是怎样使广大的 VB 程序员转移到 VB.NET 上来,因为符合 CLR 规范的 VB 已经和原来的 VB 大不一样了。不过,C# 的不成熟,Windows 的安全性,也是其不利的一面;同时,过多地依赖微软的产品而导致微软在互联网时代的再一次垄断,也是大多数人不愿意看到的。

对微软而言,在桌面计算上的统治地位无疑是手中的王牌,有庞大用户群的支持,微软的任何举动都不会无人跟随。微软已经意识到,尽快让企业认识到自己在高端计算上的改进是目前的主要任务。微软已经将前途压在了 .NET 上,以微软的实力、到目前为止的工作以及业界的反映来看,.NET 成为新一代的标准是有可能性的。

Java 语言为 Sun 带来了巨大的荣誉,但他们似乎从来就没有能够把这样有价值的品牌转换成实际的软件销售,直到现在,软件开发和销售在 Sun 的业务群中只占很小的部分。甚至在 Java 开发工具上,也没有多少程序员用 JDK。因此,Sun 给人的印象还是硬件中心主义。这次,Sun ONE 软件战略的提出,可能是 Sun 的一次战略转变,但也可能只是为了刺激其硬件产品的销售。但是,凭着这几年来 Java 的深入人心,以及其一贯的开放性,加上人们不愿看到微软的垄断,Sun ONE 应该还是有一定的市场的。

对于竞争的双方来说,速度至关重要,相信双方推出产品和服务的时间表将牵动这场 IT 业标准大赛的每一根神经。

总结 分布式对象技术是上个世纪 90 年代,网络技术和对象技术的结合。它最初的目标是在局域网络环境下,不同系统之间的服务调用问题。随着 Internet 的发展,这种调用应该扩展到 Internet 环境,而这用分布式对象技术是比较难以实

(下转第 21 页)

对于一个进程来说,它访问共享内存的方式和访问常规内存是一样的,都需要经过页目录表和页表,将虚拟地址翻译成物理地址后,这时才可读写内存。管理内存的方法也是一样的, Linux 为进程的每一段共享内存指派一个 `vm_area_struct` 结构,这些 `vm_area_struct` 结构和常规内存的 `vm_area_struct` 结构组成一链表,链头就是进程 `task_struct` 的 `mm` 成员项指向的 `mm_struct` 结构的 `mmap` 项。进程初次访问共享内存时,必须先申请连接到该进程的 `vma` 链表中。

Linux 查找共享内存从 `shm-segs` 数组入口, `shm-segs` 数组的元素均是指向 `shmid_ds` 结构的指针。每个 `shmid_ds` (私有的部分存储在 `shmid_kernel` 结构中) 结构描述一个共享内存段的长度,以及它占用的物理页帧,共享该块内存的所有进程,以及各个进程分别用什么虚存地址对应该块共享内存段等等,例如, `struct shmid_kernel` 结构中的 `shm_npages` 为表示共享内存区域容量的页面数, `shm_pages` 用于跟踪这块共享内存区域分配的页表,而 `struct vm_area_struct` 结构类型的 `attaches` 链表则代表各自进程对这块共享内存区域进行映射的 `vma` 链表。

进程第一次访问一块共享内存段前,必须通过系统调用 `shmat` 申请连接,如果不存在的话,由系统调用 `shmget` 创建,连接成功的话,会生成一个新的 `vm_area_struct` 结构,一方面插入到该进程 PCB 的 `mmap` 指示的 `vma` 虚存段链表,另一方面将插入到由 `shmid_kernel` 指示的链表,当进程不再需要共享内存时,通过系统调用 `shmdt` 申请脱离连接,当所有共享该内存段的进程都申请脱离连接,该共享内存段的 `shmid_ds` 结构及相应的物理帧被释放。

7.2 核心2.4所做的改进

Linux kernel 2.4 在共享内存方面也做了很大的改进,在数据结构方面所做的改进同消息机制,功能方面所做的改进主要是当申请一段共享内存时, Linux kernel 2.2 使用的方法是利用 `vmalloc` 分配页表来获得共享内存空间,而 2.4 核心则使用了文件系统的 `i` 节点结构(管道也是这样的)来管理和分

配共享内存空间。这样做的好处是显而易见的,当申请一段共享内存区域时,不再涉及到分配页表这样一些底层实现细节,代码更易读,可移植性得到了提高,如果底层页表分配函数有什么改动的话,共享内存这部分就不需要改动了。

7.3 共享内存机制的局限性

从共享内存的实现就可以看出它是最快、效率最高的一种通信方式,但是也有其不足之处,主要就是这种机制不提供同步,应用程序在使用共享内存通信机制时,需要自己考虑同步问题。

小结 以上所述各种进程间通信方式各有优缺点,信号和管道是出现较早也是我们最经常使用的通信方式,但是它们在功能上都有一些局限性,信号的实现方式虽然简单,但是传递的信息量有限,管道传递数据流的方向是单向的,后来 System V 推出的消息机制,信号量,共享内存虽然功能强大,但是也有其内在的不足, Richard Stevens 建议我们尽可能避免使用消息队列和信号量,因为管道等与 Unix 核心的集成要好得多。当然, Linux 也存在同样的问题,当你看这部分源代码时,就会发现为了保持与 System V 设计相一致的兼容性, Linux 这部分源代码的有些实现就显得比较松散。

参考文献

- 1 Bach M J. The Design of the Unix Operation System. Prentice-Hall, 1990
- 2 Maxwell S. Linux Core Kernel Commentary. Coriolis Group, 2000
- 3 Vahalia U. Unix Internals-The New Frontiers. Prentice-Hall, 1996
- 4 Tanenbaum A S, Woodhull A S. Operation Systems Design and Implementation Second Edition. Prentice-Hall, 1997
- 5 Stevens W R. Advanced Programming in the Unix Environment. Addison-Wesley, 1992
- 6 Tanenbaum A S. Modern Operating Systems. Prentice-Hall, 1999
- 7 李善平, 郑扣根. 操作系统及试验及教程. 机械工业出版社, 1999
- 8 樊建平, 等. Unix System V 操作系统内核代码剖析. 海洋出版社, 1992
- 3 Siegel J. What's Coming in CORBA 3. <http://www.omg.org/technology/corba/corba3releaseinfo.htm>
- 4 Saleh K, Khanafer H. CORBA-based telecom application of distributed object computing. In: Proc. of the 17th IASTED Intl. Conf. on Applied Informatics. (AI'99), Innsbruck, Austria
- 5 Platt D S. Understanding COM +, 1999, Microsoft Press, Washington, U. S. A. 10~26
- 6 Farley J. Java distributed computing. Journal of Information and Software Technology, 1999, 41(4), 235~241
- 7 Snell J. Web 服务: Web 服务内幕, 第 1 部分——我们已走了多远? IBM developerWorks 中文网站, IBM 2001
- 8 IBM Web Services Architecture Team. Web Services architecture overview: The next stage of evolution for e-business, IBM developerworks, Sep. 2000
- 9 Dave Fisco, IBM's Web Services architecture debuts: An architecture overview and interview with Rod Smith, IBM developerworks, Internet consultant, Sept. 2000
- 10 Kirtland M. A Platform for Web Services, Microsoft Developer Network, Jan. 2001
- 11 <http://www.microsoft.com/net> 微软.NET 英文主页
- 12 <http://www.sun.com/software/sunone/> sun ONE 英文主页
- 13 <http://www.ecma.ch> 欧洲计算机制造商协会(ECMA)主页

(上接第15页)

现的。Http 协议在 Internet 上的通用性, XML 在数据表示上的一致性, 这两者优点的集合就成为了 Web Services 技术的基础。Web Services 技术, 使得 Internet 上的服务不再是孤立的, 而是可以相互调用、互相联系的。

本文首先分析了当前流行的几种分布式对象技术, 然后介绍了正在出现的 Web Services 技术, 并且着重介绍了 Web Services 中的两个代表 Microsoft .NET 和 Sun ONE 两大战略。由于涉及到标准问题, 关于 Web Services 的竞争是相当激烈的。目前看来是微软走在了前面, 但是鹿死谁手还很难预料。

Web Services 技术将是计算模式的一个重大改变, Internet 将会因为 Web Services 而改变, 我们的生活方式也将因此而改变。

参考文献

- 1 Microsoft .NET 让新一代因特网变成现实. 微软公司白皮书, 2000
- 2 Sun™ Open Net Environment (Sun ONE) Software Architecture, An Open Architecture for Interoperable, Smart Web Service. Sun co. 2001, 2