

Web 服务运行支撑环境的研究与实现^{*}

刘必欣 王玉峰 周 斌 贾 焰

(国防科大计算机学院网络与信息安全研究所 长沙410073)

摘 要 Web 服务作为一种新型的应用集成技术在近几年中已成为分布式计算领域的热点。然而 Web 服务的开发、部署和管理问题没得到很好的解决。本文讨论了 Web 服务运行时环境的构造问题并提出了一种“总线+容器+服务”体系结构。着重讨论了基于分布对象的实现适配的关键问题。自主研制了相应的 Web 服务支撑环境产品 StarWebService 来验证本文的思想。

关键词 Web 服务运行时环境, SOAP 总线, Web 服务容器, 自适应, 分布式对象

The Research and Implementation of Web Service Runtime Environment

LIU Bi-Xin WANG Yu-Feng ZHOU Bin JIA Yan

(National University of Defense Technology, Changsha 410073)

Abstract Web Services have been enjoying great popularity in recent years as an emerging application integration technology. However the key issues of Web Services development, deployment and management are not well addressed. This paper addresses the problem of Web Service runtime environment construction and proposes a “bus-container-service” architecture. Specially, we tackle the problems of adaptation between Web Service and distributed object technologies in Web Service runtime environment. StarWebService has been implemented as a Web Service runtime environment product that realizes the architecture and technologies proposed in this paper.

Keywords Web service runtime environment, SOAP bus, Web Service container, Adaptation, Distributed objects

1 引言

Web 服务作为一种基于面向服务的体系结构的新型应用集成框架在近几年中成为分布计算领域的热点技术。尽管 Web 服务的规范层出不穷,但它们主要旨在规范服务的提供者与服务消费者之间各个层次和各个维度的交互合约,对于 Web 服务开发、部署和管理的关键技术和基本模式,特别是如何为 Web 服务的开发、部署和管理提供良好的运行支撑环境缺乏深入的探讨。虽然主流的应用集成厂商已相继推出 Web 服务产品,但它们通常与特定厂商的应用服务器紧密耦合,其体系结构和实现方法是私有的。另一类产品采用封装和适配的策略,支持自底向上的基于已有企业应用开发 Web 服务,代表产品有 CapeClear 和 XMLBUS。我们认为,基于适配模式的 Web 服务运行支撑环境应对了 Web 服务作为系统边界上的技术的初衷,有利于实现 Web 服务与成熟的企业计算技术(特别是分布对象技术)相结合,发挥 Web 服务的长处,保护已有投资,有着广阔的应用前景。

为此本文着重研究了基于适配模式的 Web 服务的运行支撑环境的构造方法,提出“总线+容器+服务”的体系结构,深入讨论了支持分布对象的服务适配技术,并自主研制了相应的 Web 服务支撑环境产品 StarWebService 以验证本文的思想。

2 Web 服务运行支撑环境体系结构

根据 W3C 的定义^[1], Web 服务是由具体软件 Agent 实

现的一组抽象的功能。对于不同的宿主环境,实现 Web 服务的功能的软件实体可以是简单的 Java 对象、脚本,或者是处于复杂企业计算系统中的 CORBA 对象、EJB 构件等。为了将表征功能的服务与实现具体操作的软件实体关联起来,需要为不同类型的软件实体提供一致的部署方法、标准的协议框架和统一的 Web 服务发布与访问接口。Web 服务运行支撑环境为实现上述目标提供了基本的运行框架和管理手段。我们借鉴传统应用服务器技术^[4,5]中容器和构件的思想,采用“总线+容器+服务”的体系结构构建 Web 服务运行支撑环境,如图1所示。

2.1 SOAP 消息总线

SOAP 消息总线是 Web 服务运行支撑环境的通信基础架构,提供独立于底层协议(如 HTTP, JMS)的 SOAP 消息传输功能。

SOAP 消息总线由抽象消息接口和具体消息传输引擎组成。抽象消息接口提供了统一的 SOAP 消息表示及公共的 SOAP 消息操纵方法;消息传输引擎则实现基于特定协议的 SOAP 消息打包与解析功能。消息引擎可以利用其它服务器中高性能的协议处理部件实现,例如利用 Web 服务器中的 Web 容器技术实现 HTTP 协议引擎。

2.2 Web 服务容器

Web 服务容器是 Web 服务实例的运行环境,负责服务实例的创建、执行与管理。Web 服务容器为服务实例的执行提供公共的支持服务和管理服务,其主要功能包括:

- 服务实例生命期管理:依据服务元信息创建服务实例,

^{*} 本课题受国家自然科学基金项目“新一代网络中间件的体系结构、协议及实现机制”(No. 90104020)和国家863课题“基于 Web Services 的应用集成关键技术研究”(No. 2002AA116040)的资助。

并管理其生命期。

· 服务实现适配: 依据服务元信息确定服务实现的类型, 并激活相应的实现适配器, 定位服务实现, 将 SOAP 请求转换化服务实现所能理解的协议形式激发服务实现的相应操作, 并将操作结果转换为 SOAP 应答。

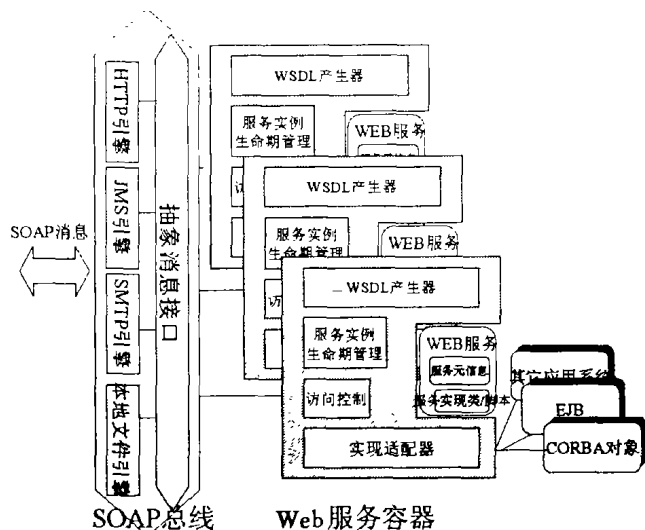


图1 Web 服务运行支撑环境体系结构

· 服务描述生成: 支持运行时刻依据服务的元信息和服务实现的相关信息动态地产生服务的 WSDL 描述。

· 服务访问控制与 QoS 支持: 实现基于角色的访问控制, 保证服务仅被授权用户消费。

每个 Web 服务容器与特定的消息引擎相关, 负责处理该消息引擎收到的全部或部分 SOAP 请求, 因此每个服务容器有其唯一的位置标识。

2.3 Web 服务

Web 服务是部署在 Web 服务容器内的支持特定操作的功能实体。一个 Web 服务在物理上表现为一组服务元信息描述以及相应的服务实现。服务元信息包含服务 ID、实例生命期、服务实现类型、服务实现位置、访问权限等服务部署信息及应用相关的属性描述, 以 XML 文档的形式保存在 Web 容器内。服务实现是具体执行服务操作的软件实体, 它可以与服务容器位于同一宿主环境 (如本地 Java 类), 或是位于 Web 服务运行支撑环境之外独立运行的后端应用组件 (如 CORBA 对象或 EJB 构件)。后一种情况服务容器根据服务实现类型激活相应的服务实现适配器来访问后端应用。

3 支持分布对象的服务适配技术

Web 服务容器最重要的功能是将 Web 服务与实现 Web 服务的具体软件实体关联起来。分布对象技术作为实现大型应用系统的主流技术, 支持分布对象实现 Web 服务有着重要的现实意义。本节讨论支持分布对象的服务适配的关键技术。

3.1 基于 FACADE 模式的服务模型

Web 服务与分布对象在模型上具有相似性, 但是它们在抽象粒度上存在不同, 为此我们认为, 简单地将服务理解为分布对象的 Wrapper, 在它们之间实现一对一的适配会将分布对象之间的紧耦关系同构地映射到 Web 服务之间, 暴露了不必要的交互细节。因此, 我们将 Web 服务定义为一组分布对象的功能集合, 在 Web 服务与实现服务的分布对象之间建立如下关系:

设 S 为容器内已部署的 Web 服务的集合, O 为实现 S 的

分布对象的集合。令函数 $\text{Target}: S \rightarrow \{O | O \subseteq O\}$, 表示由 Web 服务到实现该服务的分布对象集合的函数, 则函数 Target 满足如下条件:

① 对于任意 $ws \in S$, $|\text{Target}(ws)| \geq 1$ 。即, 一个 Web 服务由一或多个分布对象实现;

② $\bigcap_{ws \in S} \text{Target}(ws)$ 可以非 Φ 。即, 一个分布对象可以为多个 Web 服务提供实现。

上述 Web 服务与分布对象的关系可用图 2 形象地说明。Web 服务与实现服务的分布对象之间的关系呈现出 FACADE 模式^[12]: 一个 Web 服务与一组分布对象相关联 (如 service1), 它作为这组分布对象的 facade 为客户提供更高层的功能抽象, 并将对 Web 服务的调用将转发给后端的分布对象。同时, 一个分布对象也可以为多个 Web 服务提供实现 (如 ObjectC)。Web 服务的运行支撑环境负责提供请求分派的机制, 以保证调用由正确的后端对象执行。

这种模型不同于当前大多数 Web 服务厂商所实现的一个 Web 服务封装一个后端对象的模型。由于服务屏蔽了分布对象复杂的接口语义和交互, 因此可以简化客户编程并有助于进一步实现客户与应用实现的松耦合, 只要服务接口不变, 实现的后端系统或对象结构都可以独立地进行修改。

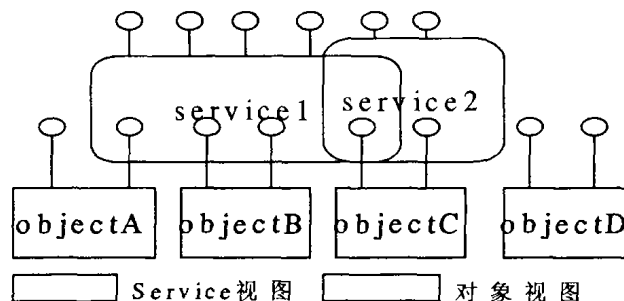


图2 Web 服务与分布对象的关系示意图

3.2 接口/服务描述映射

基于上述服务模型, 根据下列映射关系将分布对象接口中的主要描述项转换为 Web 服务 WSDL 文档中相应的描述:

分布对象描述	Web 服务描述
接口	端口类型, 表征服务的功能刻画
操作	操作
属性	get (只读属性) 或 get/set 操作 (读写属性)
接口继承关系	通过重复定义而平板化

3.3 目标端点映射

目标端点映射是指运行时刻实现适配器根据 SOAP 请求确定后端分布对象的引用。为了实现目标端点的映射, 一方面需要 Web 容器能够以恰当的形式生成 WSDL 中服务的访问端点, 另一方面需要实现适配器对 Web 服务访问端点进行正确的理解并进行相应的转换。服务的访问端点信息包含服务容器的位置信息、服务 ID 和端口类型标识, 从而唯一地标识当前请求所访问在服务及端口。由于服务元信息中记录了服务的每个端口类型对应的分布对象接口及其引用, 因此实现适配器将利用元信息完成目标端点的转换。

3.4 数据类型映射

SOAP 是一种基于 XML 的消息协议, 其数据类型以 XMLSchema 为基础, 比较简单。而分布对象系统通常提供了丰富的数据类型, 例如 CORBA IDL 中不仅简单类型分类详

细(如表达整形的数据类型就包括6种),而且还定义了特有的复杂类型,如序列、值类型、对象引用等。由于XML本身具有良好的可扩展性,因此通过扩展XMLSchema和SOAP数据类型,使得每个IDL类型唯一地映射为一种XML数据类型。产生WSDL文档时,分布对象接口中的IDL类型声明映射为WSDL文档中相应的XML类型Schema,使得SOAP客户能够以XML数据类型构造SOAP请求。目前,OMG已经制定了这一映射的相关标准^[2]。

4 StarWebService 系统实现

StarWebService是我们实现的基于适配模式的Web服务运行支撑环境,它基于J2EE体系结构,采用可插拨的灵活设计支持多种后端服务实现。目前,StarWebService提供了SOAP消息的HTTP绑定,支持多线程并发地处理HTTP请求。其服务容器运行在符合Servlet2.3规范的Web容器之中,主要包括服务管理模块、数据类型转换模块和实现适配模块。StarWebService全部采用动态机制设计实现适配模块,从而支持在运行时刻动态地部署、修改、去部署各种类型的服务。StarWebService对当前主流的分布对象技术的Web服务化提供了全面的支持,允许使用CORBA对象、EJB等多种后端系统提供Web服务的功能。特别地,对于后端系统为CORBA

的服务而言,StarWebService遵循了OMG的最新标准^[2],支持IDL语言的全部简单类型及主要复杂类型的编码转换,与国外同类产品功能相当。StarWebService还包括友好的图形化部署工具及应用开发API。

总结 本文针对Web服务运行支撑环境的构造问题,提出“总线+容器+服务”的体系结构,并着重讨论了基于分布对象的实现适配的关键问题和自主研制的支持分布对象适配的Web服务运行支撑环境StarWebService。下一步工作将重点围绕可管理的Web服务容器展开,并在此基础上展开支持网格计算^[3]的Web服务运行支撑环境的研究。

参考文献

- 1 W3C Working Draft. Web Services Architecture. <http://www.w3.org/>, May 2003
- 2 OMG. CORBA to WSDL/SOAP Interworking Specification. <http://www.omg.org>, 2003,1
- 3 Foster I, Kesselman C, Nick J M, Tuecke S. Grid Services for Distributed Systems Integration. IEEE Computer, 2002, 35(6)
- 4 Sun Microsystems Inc. JavTM2 Platform Enterprise JavaBeansTM Specification, v2.1, Final Draft, 2002
- 5 OMG. CORBA Component Model Specification, 2002, 6
- 6 Gamma E, Helm R, Johnson R, Vlissides J. Design Patterns: Elements of Resusable Object-Oriented software, 1994, 10

(上接第97页)

空间元数据(GeoMetadata),是关于地理相关数据和信息资源的描述性信息。它通过对地理空间数据的内容、质量、条件、位置和其他特征进行描述与说明,帮助和促进人们有效地定位、评价、比较、获取和使用地理相关数据。对空间数据某一特征的描述,称为一个空间元数据项。空间元数据是一个由若干复杂或简单的元数据项组成的集合。如果说地理空间数据是对地理空间实体的一个抽象映射,那么,可以认为,空间元数据是对地理空间数据的一个抽象映射。从这个意义上来说,空间元数据和地理空间数据是对地理空间实体不同抽象层次的描述,是对地理信息的不同深度的表达,它们统一于它们所反映的客观内容,同时说明数据集的主要内容、属性的统计特性和属性在数据挖掘算法中的角色,是空间数据挖掘的基础。空间元数据管理包括元数据的产生和维护等。

4 系统实现

Agent之间的通信实际上是通过Faciliator完成的,Faciliator概念来自对中间件技术的推广与发展。Agent之间的通信实际上首先是与同一结点上的Faciliator通信。Faciliator负责将消息分解,然后发往不同结点上的Faciliator,再由本地的Faciliator传送给相应的Agent。

在本系统中,数据挖掘代理DMA负责存取空间数据和从空间数据中挖掘高级别的信息。DMA以并行方式工作,DMA之间通过适配器来通信和共享信息。适配器协同代理,提供信息给用户,并反馈用户信息给代理。系统的基本工作原理如下:

- (1)用户(或远程数据挖掘系统)发出挖掘请求;
- (2)UIA接受挖掘请求,并把挖掘请求按照预定格式打包后转发给Faciliator;
- (3)Faciliator对挖掘请求进行分析,确定需要涉及到的DMA;
- (4)Faciliator检查DMA状态,如果DMA没有运行,就创建DMA;

(5)Faciliator把挖掘请求广播给DMA;

(6)DMA根据挖掘请求自动挖掘出相应的信息;

(7)Faciliator从各个DMA中收集相应的信息,然后进行综合分析,得出最终的结果信息;

(8)把结果信息提交给用户(或远程数据挖掘系统)。

在空间数据挖掘的过程中,用户可以通过消息通信机制对Agent进行监控或终止。如果用户的挖掘请求不能在本地完成,则可以根据远程空间数据库的信息,由Faciliator把挖掘请求转给远程数据挖掘系统的用户接口代理,直到远程数据挖掘系统挖掘出结果信息后再把结果信息传回。

结束语 网络的开放性、动态性与异构性以及信息不断变化更新的特点,是空间数据挖掘所面临的一个挑战。本文就此问题提出了基于Agent的解决方案并进行了实现。进一步的工作中需要实现更多的挖掘算法并集成到该系统以及选择更多的数据进行实验。

参考文献

- 1 Kotz D, Gray R S. Mobile Agents and the Future of the Internet. ACM Operating Systems Review, 1999, 33(3): 7~13
- 2 Lange D B, Oshima M. Seven Good Reasons for Mobile Agents. Communications of the ACM, 1999, 42(3): 88~89
- 3 周海燕,王家耀,等. 空间数据挖掘技术及其应用[J]. 测绘通报, 2002, 37(2): 11~13
- 4 邱凯昌,李德仁,等. 空间数据挖掘和知识发现的框架[J]. 武汉测绘科技大学学报, 1997, 22(4): 328~333
- 5 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]. Processing of ACM SIGMOD, May 1993. 207~216
- 6 王大玲,于戈,鲍玉斌,等. 基于概念层次树的数据挖掘算法的研究与实现[J]. 计算机科学, 2001, 28(6): 88~91
- 7 Han Jiawei, Micheline K. Data Mining: Concepts and Techniques [M]. Morgan Kaufmann Publishers, 2000
- 8 Han J, Koperski K, Stefanovic N. GeoMiner: A system prototype for spatial data mining[M]. In: Proc. ACM SIGMOD Int. Conf. on Management of Data, Tucson, Arizona, 1997. 560~563
- 9 Koperski K, Adhary J, Han J. Spatial Data Mining: Progress and Challenges. SIGMOD'96 Workshop on Research Issues on data Mining and knowledge Discovery (DMKD'96). Canada: Montreal, 1996