维普资讯 http://www.cqvip.com

基于Ajax和Web services企业信息资源整合研究

沈记全,矫吉祥
(河南理工大学 计算机科学与技术学院,河南 焦作 454000)

摘 要:在企业的信息化发展中,不同应用系统、数据的信息交流和无缝连接成为日益迫切的要求。分析了企业信息的实际需求,在原有的基于Web Services整合框架的基础上,提出了Ajax和Web Services相结合的一种资源整合框架。该框架可以充分发挥Ajax和Web Services各自的长处,使资源整合系统得到更好的人机交互界面和响应速度。

关键词:Ajax;Web Services;虚拟服务;信息资源整合
中图分类号:TP393.02 **文献标识码**:A **文章编号**:1673-629X(2008)04-0092-03

Research on Enterprise Information Resources Integration Based on Ajax and Web Services

SHEN Ji-quan, JIAO Ji-xiang
(College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454000, China)

Abstract: The information exchanging and seamless connection between deferent application systems and datas have become increasingly insistent demands in the process of developing enterprise informationization. Analyzed the enterprise information actual demand, a resource integration framework combinated Ajax and Web services was provided, which based on the original integration framework that based on Web services. The framework not only can give full play to Ajax and Web services respective strengths but also enable the resources conformity system to obtain the better man-machine interactive contact surface and the speed of response.

Key words: Ajax; Web services; virtual service; information resourse integration

0 引 言

人们面临的是一个信息爆炸的时代,其特点是人们生产信息、获取信息、使用信息的手段和方式都多种多样,而利用计算机技术的信息系统主要就是想辅助解决人们在生产、获取和使用信息中遇到的问题。然而大多数的企业都拥有各种不同的遗留系统、应用程序、商业流程和数据资源。它们在设计之初主要是为了满足特定的业务需求,并未考虑到数据、信息、应用的共享问题,而且随着企业业务的发展,不断要有新的应用加入到企业系统中,这样形成了企业内部出现多个"信息孤岛"的问题。随着应用系统越来越多,系统之间迫切需要数据与应用的共享与互操作以提高工作效率,对应用的集成成为一项极其重要的工作;然而,由于遗留应用的异构性,使得集成工作变得非常困难。

目前较为传统的企业/行业间系统集成方案是采用COM/DCOM, CORBA, RMI等技术来实现的,由于这些技术、方法或多或少存在缺陷,导致采用它们而建立集成系统存在着严重的资源浪费现象,成本过高,缺乏统一的接口机制等问题。

Web服务[1]基于一组开放的技术标准,采用XML[2]标准进行应用程序间的通信,从而有利于企业间不同应用系统的信息交流和无缝整合。Ajax(Asynchronous JavaScript and XML)[3]是在客户端加入一个沟通用户界面与服务器的中间层,实现页面呈现与应用的分离以及用户操作与服务器响应的异步化。采用Ajax后不仅可以利用客户端闲置的处理能力承担一部分服务器的工作,减轻带宽和服务器的负担而且降低了页面重载的频率,给予用户更好的使用环境。

1 Web services和Ajax技术简介

1.1 Web services技术

Web服务是新一代的Web应用[1],它是自包含、模块化的应用程序,它可以在网络(通常为Web)中被描述、发布、查找以及调用。Web服务是一种部署在Web的对象,它具有对象技术所承诺的所有优点;而

收稿日期:2007-07-09
基金项目:河南省骨干青年教师资助计划项目(649038)
作者简介:沈记全(1969-),男,湖北黄梅人,副教授,硕士生导师,研究方向为智能网络与网格计算。

维普资讯 http://www.cqvip.com

且 Web 服务是基于 XML 技术的、开放的 Web 规范技术,因此它具有比对象技术更好的开放性。Web 服务体系结构[4]如图 1 所示。Web 服务作为一种 Web 对象[4],具有以下几个显著的特点:封装性;松耦合性;使用标准协议和规范;互操作性;高度的集成能力。

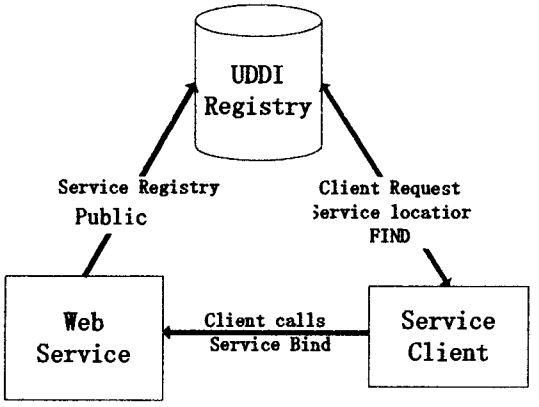

图 1　Web Services 体系结构

Web 服务是建立在 HTTP[1]、SOAP[1]、WSDL[4]和 UDDI[4]等标准以及 XML 技术之上的,其最大优势是允许在不同平台上以不同语言编写的各种程序以基于标准的方式相互通信。

1) XML:可扩展标记语言。XML 是 W3C[5]制定的可扩展的文本标记语言。其优越性在于简单地利用文本标记语言表现、描述数据和定义复杂类型数据的结构,并且可以利用目前标准的网络协议进行传输。XML 不依赖特定的平台,有进行分布式计算的先天优势。它消除了在不同平台和不同语言编写的系统之间的数据结构和消息交换模式的差异,使得数据交换方式在 XML 技术的支持下统一了起来。XML 是 Web 服务实现的技术基础。

2) SOAP:简单对象访问协议。SOAP 是一基于 XML 的,用于分布式计算环境下数据交换简单、轻量级协议。它一般绑定在 TCP/IP[1]的应层协议 HTTP 上传输,因此可以充分利用现有网协议的优势。SOAP 提供了一种将消息打包的标机制,可以使客户端和远程服务器之间以 RPC[1]形式通信。SOAP 是同类协议中第一个被世界主要软件公司都接受的协议。

3) WSDL:Web 服务描述语言。WSDL 是一基于 XML 的语言。它提供了客户端与 Web 服务行交互所必需的信息。WSDL 文档的根元素是 definition,该元素包含 types,message,portType,binding,service 5 个子元素。

4) UDDI:通用描述、发现和集成。UDDI 是一种发布有关 Web 服务技术信息的集中的目录服务。UDDI 规范跟前面的几种技术标准一样,也是由微软、IBM 和 Ariba 等公司提出和支持的。UDDI 的基础结构由一组注册表和注册器组成。注册表用于保存 UDDI 目录的完整副本,注册器用于为客户提供 UDDI 注册服务。

1.2　Ajax 技术

Ajax(即异步 JavaScript 和 XML)是一种 Web 应用程序开发的手段,它采用客户端脚本与 Web 服务器交换数据。所以,不必采用会中断交互的完整页面刷新,就可以动态地更新 Web 页面。使用 Ajax,可以创建更加丰富、更加动态的 Web 应用程序用户界面,其即时性与可用性甚至能够接近本机桌面应用程序。

Ajax(Asynchronous JavaScript and XML)其实是多种技术的综合,包括 JavaScript、XHTML 和 CSS、DOM、XML 和 XSTL、XMLHttpRequest。其中:使用 XHTML 和 CSS 标准化呈现,使用 DOM 实现动态显示和交互,使用 XML 和 XSTL 进行数据交换与处理,使用 XMLHttpRequest 对象进行异步数据读取,使用 JavaScript 绑定和处理所有数据。

在 Web 应用的体系结构中,Ajax 是介于用户和服务器之间的一个中间层,它使用户操作与服务器响应异步化。并自主处理如数据验证和数据处理等业务逻辑,只有当需要从服务器读取新数据时,Ajax 引擎才代表用户向服务器发出服务请求。Ajax 使 Web 中的界面与应用分离(即数据与呈现分离),无需刷新更新页面,减少用户实际和心理等待时间,达到更好的用户体验;且把以前的一些服务器负担的工作转嫁到客户端,利用客户端闲置的处理能力来处理,减轻服务器和带宽的负担,节约空间和带宽租用成本。

2　系统总体框架设计

基于 Web Services 的资源整合是基于 Web Services 技术,集成引擎通过各类接口将不同的应用封装成 Web Services 组件后发布到 UDDI 注册中心。它是连接各类应用的桥梁,采用的是松散的耦合方式,即任何应用都可以调用对应的接口连接到系统中来,方式灵活、简单快速,真正实现了"即插即用"。

为了更好地使企业资源无缝组合,这里采用了物理资源层、虚拟服务层和用户应用层的三层系统框架,如图 2 所示[6]。为了便于企业之间和企业内部快速无差别地访问数据,在网络上实现资源共享,通过 Web 服务提供的一套异构的分布式框架,通过 XML 协议和信息格式来封装信息,采用 SOAP 协议在网络和系统各个组件间相互传送信息,这样可以穿过企业内部的防火墙,获得用户所需的信息。

现在企业中存在许多应用系统、数据库系统和文件系统,随着企业规模的不断扩大和完善,还有可能加

维普资讯 http://www.cqvip.com

入新的应用系统，它们存在于企业的不同部门，甚至在不同的地区，在新的框架中全部把它们归为物理资源节点。

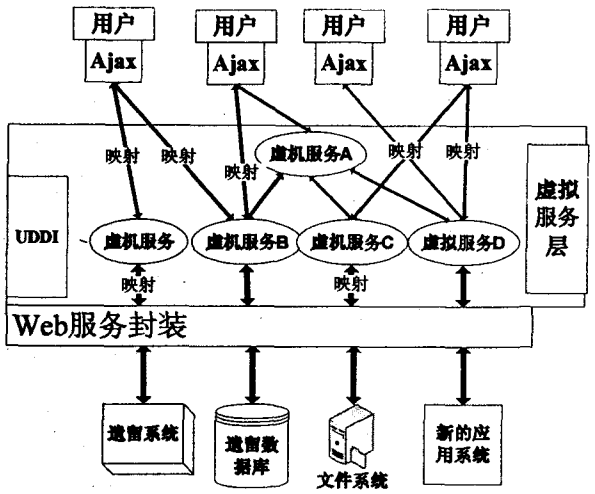

图 2 系统总体框架

对这些应用系统的主要函数，数据库进行数据抽象。通过不同的封装器，封装成 Web 服务，然后通过 UDDI 注册到虚拟服务层。这些 Web 服务在虚拟服务层都有统一的结构，在虚拟服务层是无差别的，统称为虚拟服务。虚拟服务层中的虚拟服务是相对应的物理节点映射[7]，虚拟服务也可能不都是来自物理节点，它也可能是两个甚至几个虚拟服务的再次抽象封装，例如图 2 中的虚拟服务 A 它便是虚拟服务 B、C 和 D 的再次组合。现在企业中单一的信息查询或者检索已不能满足用户的实际需要，组合信息检索以及有效信息整合已是大势所趋，这也是资源整合的主要目标。

用户应用层是用户根据自己的需求对已注册到虚拟服务层数据进行检索的工具。传统 Web 服务都是在服务端对用户的请求做出处理，这无疑对服务端提出了很高的要求，服务端负载很大，特别是在用户数众多的情况下，服务端响应可能会非常慢甚至导致服务端死掉，而客户端可能长时间等待服务端响应造成服务端的资源浪费，在客户端引入 Ajax 机制可很好地避免这种情况。

在用户和服务端之间引入 Ajax 处理引擎，该引擎成为用户和服务端的中间处理机构，帮助服务端处理用户请求。用户根据自己需要自己需求检索信息，将请求发往 Ajax 引擎，Ajax 处理引擎获得用户上层请求，分析用户需求，再将请求发送到服务端，服务端接收到请求，对请求进行分析处理，将客户端可以自己处理的数据发回给 Ajax 处理引擎，服务端转而处理其他用户请求，Ajax 引擎处理服务端返回的数据并把最终的处理结果返回给上层用户，用户最终获得所需的信息。这样 Ajax 引擎分担了部分服务端的工作，减少了服务端的压力从而提高上层用户的响应速度减轻服务节点的负荷。

下面是引入 Ajax 引擎的用户和服务端的简单工作流程。假如用户向虚拟服务 A 发出数据请求，虚拟服务 A 响应后发现请求数据分别在虚拟服务 B 和 C 上，然后响应用户请求，用户接到响应后由 Ajax 引擎处理转而向虚拟服务 B 和 C 分别发送请求，虚拟服务 B 和 C 响应请求返回数据，Ajax 引擎处理后得到结果，呈现给用户。而虚拟服务 A 在处理完用户响应后处理其他用户请求，不必等待虚拟服务 B 和 C 的处理结果，提高了处理效率，使每一个请求在更短的时间内得到响应。时序图如图 3 所示。

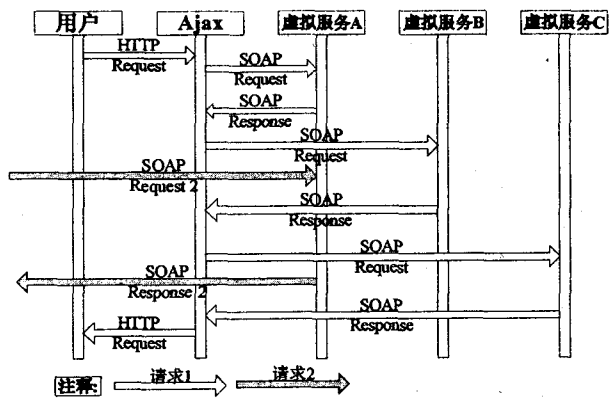

图 3 请求响应时序图

3 结 语

Ajax 的“按需取数据”的理念能够尽最大可能减少冗余请求和减轻服务器的负担。Web 服务技术采取简单的、易理解的标准 Web 协议作为组件界面描述和协同描述规范，采用了屏蔽系统环境差异的 SOAP 作为通信协议，它松散耦合的应用特性，解决了企业应用集成的诸多问题，方便地实现了现有系统的集成与新的应用的部署。基于 Ajax 和 Web 服务的企业资源信息整合框架，具有更好的可扩展性、易于管理底层数据、响应时间短的特点。随着这两种技术的不断发展，它们可能对其它领域产生影响。

参考文献：

[1] 柴晓路，梁宇奇. Web Services 技术、架构和应用[M]. 北京：电子工业出版社，2003.

[2] Harold H R. Java 语言与 XML 处理教程[M]. 刘文红译. 北京：电子工业出版社，2004.

[3] Asleson R, Schutta N T. AJAX 基础教程[M]. 金 灵等译. 北京：人民邮电出版社，2006.

[4] Foster I, Kesselman C. 网格计算[M]. 第 2 版. 金 海，袁平鹏，石 柯译. 北京：电子工业出版社，2004.

（下转第 97 页）

维普资讯 http://www.cqvip.com


```
RETURNS @BasicData table --遍历结果表,结构等同与MyTreeDATA表
( [NodeCode] [varchar] (9) , --结点编码
  [NodeName] [varchar] (20) , --结点名称
  [PNodeCode] [varchar] (9) --此结点的父结点编码
) AS
BEGIN
declare @TempData table(.....) --数据临时存放表,结构等同与MyTreeDATA表
declare @MTemple table (.....) --中间临时表,结构等同与MyTreeDATA表
declare @MExchange table (.....) --中间交换临时表,结构等同与MyTreeDATA表
delete @TempData ,@MTemple,@MExchange ,@BasicData
insert into @TempData select * from MyTreeDATA
insert into @MTemple select * from @TempData where PNodeCode =@NodeCode
insert into @BasicData select * from @TempData --结果临时表
while (exists(select * from @MTemple where NodeCode in (select PNodeCode from @TempData ))) --当某层的结点全为叶子时,停止循环
begin
  insert into @MExchange select * from @MTemple --中间交换临时表
  delete @MTemple
  insert into @MTemple select * from @TempData where PNodeCode in (select NodeCode from @MExchange )
  delete @MExchange
  insert into @BasicData select * from @MTemple
 end
 insert into @BasicData select * from @TempData where NodeCode =@NodeCode
return
END
```

*存储过程实现:

```
CREATE TABLE #TempData(.....) --数据临时存放表,结构等同与MyTreeDATA表
CREATE TABLE #MTemple (.....) --中间临时表,结构等同与MyTreeDATA表
CREATE TABLE #MExchange (.....) --中间交换临时表,结构等同与MyTreeDATA表
CREATE TABLE #BasicData (.....) --遍历结果表,结构等同与MyTreeDATA表
CREATE PROCEDURE FindSubTree (@NodeCode varchar(9)) --求某结点下子树所有结点
AS
truncate table #TempData, #MTemple, #MExchange, #BasicData
insert into #TempData select * from MyTreeDATA --数据临时存放表
insert into #MTemple select * from #TempData where PNodeCode =@NodeCode --中间临时表
insert into #BasicData select * from #MTemple --结果临时表
while (exists(select * from #MTemple where NodeCode in (select PNodeCode from #TempData ))) --当某层结点全为叶子时停止循环
begin
  insert into #MExchange select * from #MTemple --中间交换临时表
  truncate table MTemple
  insert into #MTemple select * from #TempData where PNodeCode in (select NodeCode from #MExchange)
  truncate table #MExchange
  insert into #BasicData select * from #MTemple
 end
 insert into #BasicData select * from #TempData where NodeCode =@ NodeCode
GO
```

3　结　语

通过存储过程和自定义函数的遍历实现,并在河流相随机游走储层建模开发中加以实际应用,发现数据提取速度得到明显的改善,特别在多个客户端进行不同数据节点遍历请求时,网络速度快且拥塞现象不明显。

参考文献:

[1] 陆　鑫.存储过程及其应用方法[J]. 计算机应用,1999,19(2):12-14.

[2] 万　波,周顺平.SQL SERVER扩展存储过程实现机制及应用方法初探[J].武汉科技大学学报,2001(3):294-297.

[3] 郑　刚,彭　宏,郑启伦.存储过程在嵌入式多功能数据挖掘器中的应用[J].计算机应用,2006,26(6):102-104.

[4] 王道学.使用SQL Server存储过程递归遍历层次结构[J].计算机应用,2003,23(6):126-128.

[5] 檀　明,钟伯成,胡学友,等.一种基于存储过程的BOM遍历算法[J].合肥学院学报,2007,17(1):40-45.

(上接第94页)

[5] W3C-The World Wide Web Consortium[EB/OL].1994-2007.http://www.w3.org.

[6] 李　伟,徐志伟.一种网格资源空间模型及其应用[J].计算机研究与发展,2003,40(12):1756-1762.

[7] 徐志伟,李晓林,游赣梅.织女星信息网格的体系结构研究[J].计算机研究与发展,2002,39(8):948-951.