

常用 Web 服务技术研究

单剑锋, 马德锦

(南京邮电大学 电子科学与工程学院, 江苏 南京 210003)

摘 要:随着 Web 服务的发展,如何从大量的 Web 服务实现方式中选择出一种适合自己的 Web 服务已经成为一种难题。文中从三种常用的 Web 服务:远程过程式调用(RPC)、面向服务的架构(SOA)以及表征状态转换(REST)出发,首先简要阐述了各自的特点以及 Web 服务的区别。然后从扩展性、灵活性和开销上对传统的 RPC 风格和 REST 风格进行了对比分析,发现 REST 更加简单而且有效,对服务的生产者可以降低创建服务的成本,对服务的消费者可以降低学习的曲线等诸多优势。最后认为 REST 采用的是一种新的思维,更加充分地利用了现行 Web 的特性,更加符合 Web 的设计初衷,并探讨了 REST 服务的未来发展,为 Web 服务架构的发展提供了一些方向。

关键词:Web 服务;表征性状态转换;远程过程调用

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2013)06-0253-03

doi:10.3969/j.issn.1673-629X.2013.06.065

Research on Common Web Services Technology

SHAN Jian-feng, MA De-jin

(College of Electronic Science and Engineering, Nanjing University of
Posts & Telecommunications, Nanjing 210003, China)

Abstract: With the development of Web services, how to choose an appropriate Web service from a large number of Web services implementation has become a problem. Starting from three commonly used Web services: remote procedure call (RPC), the service-oriented architecture (SOA) as well as representational state transfer (REST), briefly describe their respective characteristics and Web service difference. Then from the scalability, flexibility and spending on traditional style of RPC and REST are compared and analyzed, found that REST is more simple and effective, the service producers can reduce the service cost, service consumers can reduce the learning curve and many other advantages. Finally think the REST is a kind of new thinking, make full use of the current Web characteristics, more in line with the Web design original intention, and discuss the future development of REST services, for the development of Web service architecture provide some direction.

Key words: Web services; representational state transfer; remote procedure call

0 引 言

Web 服务(Web service)是最近十年网络技术发展的一个热点,而其能够得到很好发展的一个重要原因在于网络和电商以及社交平台的迅速崛起;原因在于 Web 服务具有良好的数据互操作性、扩展性和松散耦合性,这些特点简化了复杂的软件应用方式,为资源共享与协同工作提供了很好的支持,因此企业对 Web 服务的应用越来越广泛,使得 Web 服务成为新一代电子商务的框架^[1]。Web 服务可以理解成为一个软件系统,为用来实现不同客户端和服务器或客户端之间进

行互动操作而生,通过 API 和网络来实现交互,例如 Internet 的远程服务器端,执行客户所提交服务的请求,Web 服务是基于网络的、分布式的模块化组件,它执行特定的任务,遵守具体的技术规范,这些规范使得 Web 服务能与其它兼容的组件进行互操作。它可以使用标准的互联网协议,像超文本传输协议 HTTP 和 XML,将功能体现在互联网和企业内部网上。Web 服务的目的就在于保证不同机器之间可以进行操作交互^[2]。

收稿日期:2012-08-20

修回日期:2012-11-23

网络出版时间:2013-05-14

基金项目:南京邮电大学引进人才启动项目(NY207024)

作者简介:单剑锋(1967-),男,博士,副教授,硕士生导师,研究方向为无线通信系统中的信号处理技术、智能信息处理、目标识别等;马德锦(1986-),男,硕士研究生,研究方向为 JAVA 网络分布式设计、J2EE 企业应用开发。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130514.1711.010.html>

1 三种常用 Web 服务简介

Web 服务实际上可以看成一套工具,其实现的方法有很多种,而在实际开发中最常用的三种手段为:远程过程调用(RPC)、面向服务架构(SOA)以及表征性状态转移(REST)。

1.1 远程过程调用

远程过程调用是一种计算机通信协议,它是通过网络从远程计算机上去请求服务的,采用客户机/服务器模式。请求程序的一端是客户机,提供程序的一端就是服务器。首先,客户端会发送一个信息到服务端,然后会等待服务器端作出响应;而服务器端最初会处于睡眠状态,等待调用信息的到来,当客户端的调用信息抵达服务器端后,服务器端会进行相应的处理,然后将处理的结果返回给客户机,然后再次进入睡眠,等待下一个调用信息的到来;最后客户端会收到服务器端返回的答复信息,进行相应的客户端处理,然后调用执行继续进行下去。

RPC 基本操作流程见图 1,过程为:

- 1) 客户应用向客户端程序(Stub)发出常规的函数调用,客户端程序接收有关参数或返回有关参数(参数可以是输入参数、输出参数或者既是输入也是输出的参数);
- 2) 客户端程序把输入参数的数据转换成一种标准的公共数据表达形式,组成一个包含该输入参数的数据块,调用客户运行时(runtime)程序,这些运行时程序往往以函数形式供使用;
- 3) 客户运行时程序是把有关数据块发送给服务器端的运行时程序;
- 4) 服务器运行时程序向服务器端程序发出一个调用,传递有关参数,服务器端程序则把有关参数转换成本地的数据表达形式,最后服务器端程序向服务器应用发出调用,随后,服务器就可以按照有关约定来处理接收到的信息;

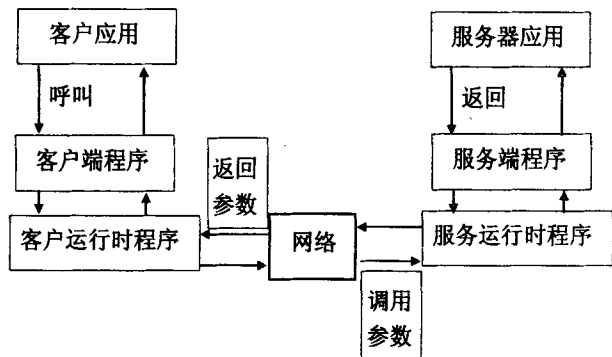


图 1 远程过程调用基本操作流程图

5) 当服务器处理完成后,它向调用方返回有关出口结果数,整个过程和调用过程相反,即服务器->服

务器端程序->服务器运行时程序->客户端运行时程序->客户端程序->客户应用程序。

1.2 面向服务的体系架构

面向服务的体系架构(SOA, Service Oriented Architecture)是为了解决在 Internet 环境下业务集成的需要,通过连接能完成特定任务的独立功能实体实现的一种软件系统架构,将应用程序的不同功能单元(服务)通过这些服务之间定义良好的接口和契约联系起来^[3],图 2 为 SOA 应用程序流程。

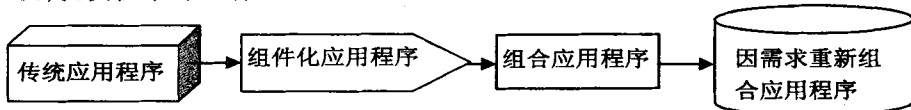


图 2 SOA 应用程序流程

指导原则定义了 SOA 的开发、维护和使用的一些基础规则。这些原则具有以下特性,它们能为设计样式提供固有特征,并应该包括项目的以下几个方面:

1) 指导原则^[4]。

* 重用(Reuse)、粒度(granularity)、模块性(modularity)、可组合性(composability)和组件化(componentization);

* 与标准(一般的或是特定于行业的)一致;

* 服务的识别和归类、提供和传递、监控和跟踪。

2) SOA 架构样式。

* 基于服务的设计,这些服务代表的是组成企业业务流程的业务活动;

* 服务提供上下文描述(业务流程、目标、规则、策略、服务接口和服务组件),用服务编排来实现服务;

* 对技术设施提出要求,建议采用开放标准,以便更好地实现可交互性和透明性;

* 实现是基于特定环境的,约束和使能都必须在上文中进行描述;

* 服务描述和实现需要很好的治理;

* 需要进行验证式的测试之后才能决定这个服务是否是好的服务。

SOA 将业务应用拆分为动态的和可重用的服务:

* 将应用分解为模块和可重用的函数以及服务;

* 组合服务和模块以符合业务需求;

* 重用现有的服务和模块以满足变化的业务需求。

3) 实现 SOA 的相关技术。

其中涉及主要技术包括以下几个:XML、SOAP、WSDL 和 UDDI,如图 3 所示。

* XML(eXtensible Markup Language):可扩展的标记语言,可扩展意味着相对于 HTML 而言,在编辑网页文件时,所有能够在网页文件中使用的 HTML 标签都

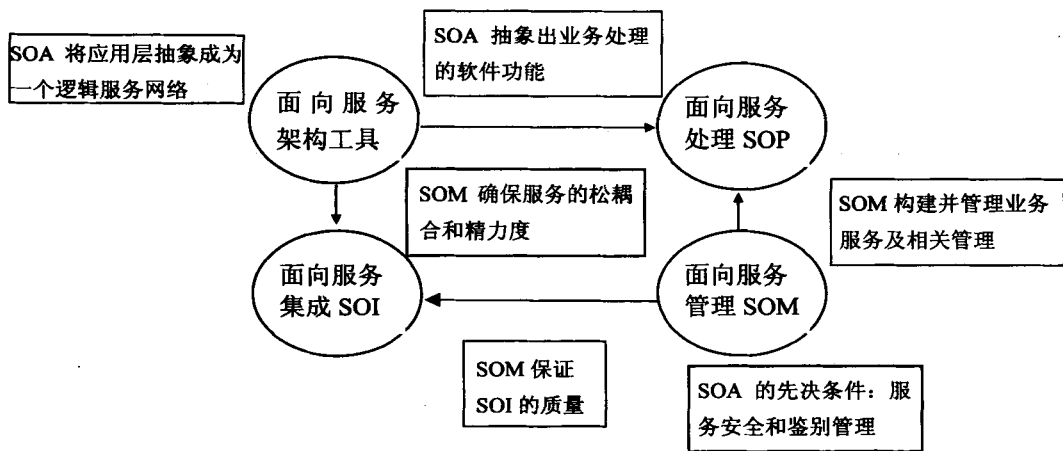


图 3 SOA 技术实施的示意图

是预定义好的,不能够根据自己意愿添加规定外的标签,而 XML 可以根据自己意愿定义标签来完成。

* SOAP(Simple Object Access Protocol):简单对象访问协议。SOAP 是在松散的、分布的环境中使用 XML 交换结构和类型化的信息的一种简单协议。SOAP 本身并不定义任何应用语义,如编程模型或特定语义实现,只定义了一种简单的以模块化的方式包装数据的机制^[5]。

* WSDL(Web Services Description Language):Web 服务描述语言。WSDL 是一种用来描述 Web Service 的功能特征的语言,把服务访问点和消息的抽象化描述与具体的服务部署和数据格式的绑定分离,从而使对服务的抽象定义可以方便地重用。

* UDDI(Universal Description, Discovery, and Integration):统一描述发现和集成,提供一种发布和查找服务描述的方法^[6]。

1.3 表征性状态转移

Representational State Transfer(表征性状态转移,简称 REST)是由 RoyFielding 博士在其博士论文中首次提出,作为分布式系统设计的一种架构风格,在 REST 的概念中,整个 Web 被看作一组资源的集合,资源由 URI 标识,对资源进行的操作由客户端指定的 URI 和 HTTP 协议动词的组合来实施,同时将资源和资源的表示分开,为构建可扩展、简单性、可移植和松耦合的 Web 程序提供了一个架构上的准则,REST 这种基于资源的设计改变了传统的基于服务的设计思想。目前在三种主流的 Web 服务实现方案中,因为 REST 模式的 Web 服务与复杂的 SOA 和 RPC 相比更加灵活简易,因此越来越多的企业开始使用 REST 风格。可以说:REST 是一种针对网络应用的设计和开发方式,可以降低开发的复杂性,提高系统的可伸缩性。表现在:

- 1)网络上的所有事物都被抽象为资源;
- 2)每个资源对应一个唯一的资源标识;

- 3)通过通用的连接器接口对资源进行操作;
- 4)对资源的各种操作不会改变资源标识;
- 5)所有的操作都是无状态的^[7]。

规划 REST 服务:

当要规划一个 REST 服务的时候,首先要求开发人员显示地使用 HTTP 方法,并且使用方式与协议定义一致。这个基本 REST 设计原则建立了创建、读取、更新和删除(Create、Read、Update and Delete,CRUD)操作与 HTTP 方法之间的一对一映射。根据此映射:若要在服务器上创建资源,应该使用 POST 方法;若要去检索某个资源,应该使用 GET 方法;若要更改资源状态或对其进行更新,应该使用 PUT 方法;若要删除某个资源,应该使用 DELETE 方法。这样,客户端通过一个 URL 来请求资源,返回的并不是资源,而是资源的一种特定的表现形式,而这种表现形式常用的有:HTML、XML、JSON 等等。具体选择某种表现形式则需要根据实际情况作出选择。个人推荐 JSON,原因为:在编码上,虽然 XML 和 JSON 都有各自的编码工具,但是 JSON 编码要比 XML 简单,且其也具有可读性;主观上,JSON 更为清晰且冗余更少些,JSON 网站提供了对 JSON 语法的严格描述,不过,最重要的还是资源的利用率最大,可以让同样的资源去服务到更多的人。

图 4 为 REST 围绕资源的三角关系。

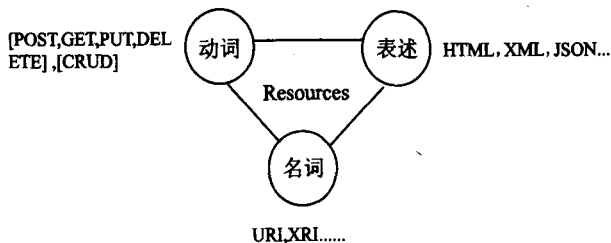


图 4 REST 围绕资源的三角关系

即:REST 是以资源为中心,名词即资源的地址,动词即施加于名词上的一些有限操作,表述是对各种资源形态的抽象。

2 三者的区别及对比分析

2.1 核心区别

由于 SOAP 有两种风格的基本消息类型^[8]:一种是远程过程调用[RPC]方式,它符合 JAX-PRC 编程规范,另一种是通过 Message 消息方式,它符合 JAXM 编程规范。

其中 RPC 即远程过程调用,很简单的概念,像调用本地服务(方法)一样调用服务器的服务(方法)。通常的实现有 XML-RPC,JSON-RPC,通信方式基本相同,所不同的只是传输数据的格式。RPC 的重点是落在调用,即是以动词为中心的。

比如一个简单的 RPC 通信过程常为:

Request:

```
<? xml version="1.0"? >
<methodCall>
  <methodName>member.get_username_by_id</methodName>
  <params>
    <param>
      <value><i4>2</i4></value>
    </param>
  </params>
</methodCall>
```

Response:

```
<? xml version="1.0"? >
<methodResponse>
  <params>
    <param>
      <value><string>ma dejin</string></value>
    </param>
  </params>
</methodResponse>
```

即向服务器发送一个过程调用的方法及其参数,得到服务器返回的方法执行的结果。而在 XML-RPC 之后又有了更加强大的 SOAP,从本质上说,SOAP 是一种无状态(stateless)协议,它提供复合的单向消息交换框架,以便在称之为 SOAP 节点的 SOAP 应用程序之间传输 XML。因为 XML-RPC 只能使用有限的数据类型种类和一些简单的数据结构,而 SOAP 是通过利用 XML Schema 的不断发展来定义数据类型,即是 RPC 的一种升级。

REST 强调资源(名词)有统一的接口对它们寻址,换句话说,也即是 REST 是以名词为中心的。通过有限的操作或者是“动词”(POST, GET, PUT, DELETE)来实现客户端和服务端之间的通信。REST 架构体现在灵活,会为每一个资源都分配一个唯一的 URL,而这个 URL 即对应着资源客户,通过 HTTP 协议提供的接口,去实现客户和服务之间的通信,这样就可

以通过上述有限的操作或者是“动词”去访问任意的一个 URL,这样,无论提供多少个 URL,REST 都能轻易快速地去获取到用户想要的资源,而不需要向 SOAP 那样通过 XML 来冗余传输。

举个例子:假如需要通过 Web 服务去获取某个图书馆中科学类的一本书,那么用户就需要去通过一个 GET 请求,类似的 URL: http://localhost/library/science/12345。在处理这个请求的时候,“/library”将映射到一个服务中,通过对应的服务对“science/12345”进行解析,来定位到后台对应的某个 ID,找到对应的资源,从而获得指定的书籍。同样在服务器端作出回复的时候,服务器端会根据你头文件传递过来的内容进行解析,指定返回的格式,格式可以为 HTML 或 XML,但无论返回什么格式,在客户端都可以去解析出相应的界面。

综上核心区别为:RPC 是以动词为中心,REST 是以名词为中心;以动词为中心,意味着,当你需要加入新功能时,你必须添加更多的动词,这时服务器端需要实现相应的动词(方法),客户端需要知道这个新的动词并进行调用。而以名词为中心,假使请求的是 library/science/, 无论这个 URI 对应的服务怎么变化,客户端是无需关注和更新的,而这种变化对客户端也是透明的。通过这个根本区别来进行如下对比分析。

2.2 对比分析

1) 可扩展性。

RPC 即远程过程调用,在这种风格中,服务是由细粒度的自定义操作组成,不同的服务具有不同的专有接口,而每个接口都具有自己的语义和操作参数,客户端若想正确地访问服务,那么就必须正确知道每个服务的接口定义和描述^[9]。但是会带来一个问题,因为如今的 Web 服务都是大规模的,如果目前网络中 Web 页面都去定义了自己的专有接口,要求浏览器必须下载或编写相适应的接口,否则就无法正确的访问和交互,那么客户端浏览器将不得不安装数百万的不同插件,另外,RPC 类型在使用上虽然简单,但当数据很大的时候,它的性能却会急剧下降^[10],这些情况恐怕是用户不能接受的。

REST 采用的是统一接口约束,即要求 REST 架构要为所有的资源提供统一的操作接口,统一接口就使得客户端和服务端解耦,可以独立化进行,只要接口不变就不会影响到对象,REST 的统一接口约束,将接口的变化转移到了资源表示上,这种变化的转移,使数据成为了最终关注的重点和目标。

2) 灵活性。

REST 的意义在于通过将数据操作的资源化,给客户端的使用相对 RPC 和 SOAP 而言,提供足够的灵活

性。如通常开发一个业务系统,对数据会做两件事:

(1)采用某种技术去实现一个通用的 DAO,或通过模板工具,将所有数据表映射到对象并创建对应的 CRUD 操作甚至关联操作。

(2)对应非 CRUD 的操作,则重新编写专门的代码来实现。

但对于 REST 而言:

(1)将数据的 CRUD 操作资源化发布,前面也提到,REST 主张使用 HTTP 标准的 GET、POST、PUT 以及 DELETE 来进行请求和响应。

(2)对于复杂的操作需求,编写专门的服务实现。

应该来说,REST 的架构,某种意义上应该理解为客户端应用导向的框架,更加关心的是能够给予客户端更多的灵活性而不是更多的限制。

3)构建和开销。

首先,SOAP 是基于 XML 的协议,必然会在 XML 中去定义一些操作的编码及处理的方法,SOAP 是基于 XML 语言和 XSD 标准,编码的规则由四部分组成:SOAP 信封,序列化机制,SOAP 绑定,SOAP RPC,可以说一条 SOAP 消息就是一个普通的 XML 文档,而 SOAP 应用程序之间传输的也是 XML。因此,SOAP 需要一层 XML 包装器包装所有的请求和响应消息,这会增加消息的大小;另外,SOAP 是基于 XML 消息传输,而 XML 使用冗余的编码,且解析 XML 文件是一个很耗时间和内存资源的过程,产生一种有效的载荷^[11]。

REST,在这方面,提供了更轻量级的消息传输替代方案——JSON;而且,易于构建,无需工具支持,因为 REST 是基于 HTTP 的,可以使用熟练的技术,比如 JavaServlet API 和 Java HTTP 支持编写 REST 服务的实现端和客户端,而无需任何特定工具的帮助。

3 REST 的未来发展

云计算技术逐渐成为了未来发展的趋势,而且云计算也确实带来了许多的好处,比如:数据的集中储存和事件的快速反应等。展望未来,应该会有越来越多的 Web 服务将为云计算建立并且运用云计算的优势。而企业开发人员需要一个框架,使他们能够建立可以通过互联网或者云计算访问安全的 Web 服务,那么就需要外部人员能够通过某种身份识别提供接入,现在一些 WS-* 协议的目标就是为此提供一个处理方案,同样为了保证 REST 资源的安全,并且随着开发人员建立一些 Web 服务向日益扩大的统一的用户提供数据访问,身份识别管理是非常重要的。当用户出示证书以访问一个 Web 服务的时候,大型系统中必须有一项验证功能,以便允许用户访问他们要求的内容,就如同你出示信用卡进商店之前提取现金一样,采用这种

Web 服务的分布式模式能够让开发人员通过强调这个服务而不是验证安全证书来创造更大的价值。而且随着云计算的普及和发展,相信通过云计算访问 Web 服务安全性方面会越来越重要,那么身份识别将会成为 REST 发展的一个方向,而且安全性和保密性已经成为 Web 服务广泛应用的瓶颈,要构建大型的、松散的、健壮的 Web 服务应用系统还需要长期的努力。


4 结束语

文中描述了三种常用 Web 服务的设计原则和特点,以及这三种常见 Web 服务的区别,并通过从扩展性、灵活性、构建和开销方面进行了对比分析,认为 REST 必然会对 Web 服务的体系结构产生重要的影响;同时提出伴随着云计算的普及,身份识别将会成为 REST 安全性方面的一个发展方向。且基于 REST 的 Web 服务必将推进当前 Web 服务技术的大规模应用和进一步发展,使 Web 服务成为真正基于 Web 的服务。

参考文献:

- [1] 李程程,张永胜,李 静,等.一种简单的 Web 服务安全通信模型研究[J]. 计算机技术与发展,2010,20(9):157-158.
- [2] W3C Web Service Architecture [EB/OL]. 2004. <http://www.w3.org/2002/ws/arch/>.
- [3] Koch C. A New Blueprint for the Enterprise[M]//CIO Magazine. [s.l.]:[s.n.],2005.
- [4] Balzer Y. 改进您的 SOA 专案规划[M]. [s.l.]:IBM,2004.
- [5] Martin G, Marc H, Noah M, et al. SOAP version 1.2 part 0: primer (second edition) [EB/OL]. [2008-05-08]. <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/>.
- [6] 柴晓路,梁宇路. Web Services 技术、架构和应用[M]. 北京:电子工业出版社,2003.
- [7] 徐 涵,李红军,胡 伟. Restful Web Services 中文版[M]. 北京:电子工业出版社,2008.
- [8] Martin G, Marc H, Noah M, et al. SOAP version 1.2 part1: messaging framework (second edition) [EB/OL]. [2008-05-08]. <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>.
- [8] 冯新扬,沈建京. REST 和 RPC:两种 Web 服务架构风格比较分析[J]. 小型微型计算机系统,2010(7):1394-1395.
- [10] 李海峰,杨小虎. 基于 RPC 风格与 Message 风格 SOAP 消息的 Web 服务性能比较[J]. 计算机应用与软件,2007(1):79-80.
- [11] Vorobiev A, Han Jun. Security Attack Ontology for Web Services[C]//Proceedings of the IEEE Second International Conference on Semantics, Knowledge and Grid. [s.l.]:[s.n.], 2006.

常用Web服务技术研究

作者: [单剑锋](#), [马德锦](#), [SHAN Jian-feng](#), [MA De-jin](#)
作者单位: [南京邮电大学电子科学与工程学院, 江苏南京, 210003](#)
刊名: [计算机技术与发展](#) 
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2013, 23 (6)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjfz201306065.aspx