

网络服务及其推送访问模式^{*}

Web Services and its Push Access Model

张培坤 康建初

(北京航空航天大学计算机科学与工程系 软件开发环境国家重点实验室 北京100083)

Abstract With the development of Internet technologies, concept of Web services has come up. The introduction of computing model of Web services simplifies distributed computing and resource sharing based on Internet. This paper gives a research to Web services and its core protocol: Simple Object Access Protocol, and points out the limitation of Web services access model - PULL model which is used currently. Based on Push technology, this paper presents a new kind of Web services access model - PUSH model, which allows transport of real-time data between providers and providers, or providers and consumers, of Web services.

Keywords SOAP, Web services, PULL, PUSH

一、引言

目前,应用程序开发技术正经历着一场革命性的变革。

过去,人们开发应用程序时,通常是通过调用操作系统的 API 函数来集成本地的服务。在这种情况下,开发人员只可以访问本地的资源,但能够严格地控制应用程序的执行过程。

随着网络技术的迅速发展,尤其是分布式计算技术的发展,开发人员已经不再局限于使用本地资源,而是越来越多地将程序的计算任务分布到网络上,于是出现了 CORBA 和 DCOM 等分布式计算模型。这些计算模型虽然能够较好地解决一些分布式计算问题,但在应用规模和应用实践上仍存在一些缺陷,难于在 Internet 上发挥作用。因此,网络服务(Web Service)的概念应运而生。网络服务计算模式的出现,能够有效地解决基于 Internet 的分布式计算和资源共享等问题,有助于人们进行简便快速的应用开发。

本文讨论网络服务及其核心协议“简单对象存取协议(SOAP)”,指出目前常用的网络服务访问模式(“拉”模式)的局限性,提出基于“推送”(PUSH)技术的网络服务访问模式及与其相关的两种实现方案。

二、网络服务及其相关技术

2.1 网络服务

从应用角度而言,网络服务是指“由企业发布的、能够完成某些特定商务需求的在线应用服务”。其他公司或应用软件可以通过 Internet 访问和使用这些在线服务^[5]。

从技术角度而言,网络服务是“分布于 Internet 之上、具有松散耦合特性的可重用软件组件,用户可以通过 Internet 标准协议对其进行访问和使用”^[6]。此定义可以进一步简述如下:

首先,网络服务是一种可重用的软件组件。它延伸了面向对象程序设计的概念,从语义上封装一些离散的软件功能,软件开发人员可以直接将这些组件的功能集成到自身的应用程序中去。为了方便起见,网络服务以独特的方式描述其的输入输出和网络访问模式。应用程序通过这种描述,可以获取网络服务提供的有关功能、调用方式以及其返回结果的形式等信息。

息。

其次,网络服务具有基于 Internet 的松散耦合特性。网络服务利用 XML 描述其数据以及其接口,使用 HTTP 发送请求和接收响应。XML 的自描述性和可扩展性以及 HTTP 使用的广泛性和对防火墙的透明性,使得网络服务可以在 Internet 这一完全异构的环境下得到应用。只要网络服务的调用接口不变,其他的任何改变对于服务的调用者来说都是透明的。

第三,对网络服务的访问和使用具有可编程性。与网站和桌面程序不同,网络服务面向的是软件开发人员,而不是最终用户,因此它对外提供的是调用接口,而不是图形用户界面。软件开发人员通过这些接口调用网络服务,获取相关数据,并将这些数据集成到自己的应用中。当网络服务的类型和数量足够多时,基于网络服务的应用程序开发实质上可视为针对网络服务的编程。

网络服务协议栈如图1所示,分为核心协议和上层协议两个部分。其中,核心协议层用于定义网络服务的通讯方式,上层协议用于定义业务流程的相关内容。

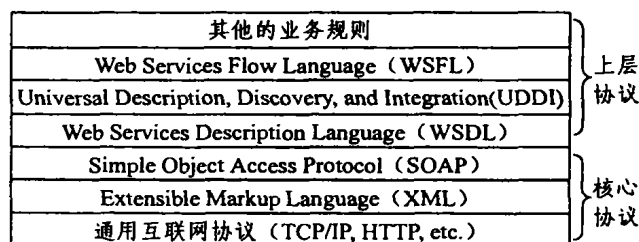


图1 网络服务协议栈

协议栈中的核心协议可分别叙述如下:

- 通用互联网传输协议。网络服务利用 Web 服务器和浏览器采用的互联网协议 HTTP 作为其消息交换协议,以便获取最广泛的应用。
- 可扩展标记语言(XML)。XML 是一种被广泛接受的、用于描述数据、数据交换接口以及数据相关语义信息格式的描述语言。它是协议栈其它各协议层的应用基础。
- 简单对象存取协议(SOAP)。SOAP 是在应用程序和网络

^{*} 本文得到国家重点基础研究973项目的资助(G1999032709)。张培坤 硕士生,研究方向:电信网络管理、网络服务。

络服务之间传输消息和进行 RPC 通讯的协议。

协议栈中的上层协议可分别叙述如下:

- 网络服务描述语言(WSDL)。WSDL 将某特定网络服务的互联和通讯方式抽象为一个 XML 文件,其内容记述与本身服务进行连接的方式。它是统一描述、发现和集成协议(UD-DI)的关键元素。

- 统一描述、发现和集成协议(UDDI)。UDDI 是一套基于 Web 的、分布式的、为网络服务提供信息注册的标准规范,它还包含企业自定义的网络服务注册的实现标准^[5],它使其它企业能够很方便地发现本企业提供的网络服务并共享其服务。通俗地说,UDDI 可视为网络服务目录。

- 网络服务 workflow 描述语言(WSFL)。WSFL 是当前网络服务协议栈中开发最少的一部分。WSFL 工作小组在 IBM 的资助下,希望定义一个框架,使得网络服务的开发者可以利用此框架方便地对网络服务组装过程中所涉及的业务逻辑进行描述。

在网络服务协议栈中,由于简单对象存取协议(SOAP)是其核心协议,是网络服务在 Internet 上实现松散耦合和开放互联的应用基础,本文下面将着重介绍简单对象存取协议。

2.2 简单对象存取协议

简单对象存取协议(SOAP)的前身是 XML-RPC^[3],它是一种应用 XML 进行远程过程调用(RPC)的技术。目前,简单对象存取协议已被提交到 W3C(World Wide Web Consortium)进行标准化。

SOAP 作为一个分布式环境下支持信息交换的轻量级协议^[1],主要由以下四部分组成:①信封:一个描述消息内容及其处理方式的框架;②编码规则:相关程序(包括应用程序和网络服务)之间交换自定义数据类型实例的一系列机制;③RPC 调用规则:SOAP RPC 的请求和响应规则;④传输协议绑定规则:关于交换消息的底层协议的绑定方式。

SOAP 的设计目标是注重简单化和可扩展性,因此 SOAP 规范中并不包含传统消息系统和分布对象系统的某些性质,如分布式垃圾回收、成批消息传送、对象引用和激活等。

与传统的分布式对象模型(如 CORBA 和 DCOM)相比,SOAP 具有以下优点:

- SOAP 采用 HTTP 作为其传输协议,因此它可以任意跨越各种防火墙和路由器;而 CORBA 和 DCOM 传输的消息却被阻隔于防火墙之外。

- SOAP 不被绑定于任何对象模型。它利用 XML 的自描述特性,可实现客户端和服务端的松散耦合;而 CORBA 和 DCOM,无论是客户端还是服务器,都需要与其相关的复杂的运行环境的支持。

- SOAP 可以用任意语言,在任何操作系统平台上实现,只要客户端能保证将正确书写的 SOAP 请求发送给远端的服务器即可;而即使在相同的平台、相同的操作系统上,不同厂家实现的 CORBA 产品之间的互联都存在问题。

下面图2和图3给出了一个基于 SOAP 的应用实例。其中,图2是用户向网络服务发送的 SOAP 请求包;图3是网络服务向用户返回的 SOAP 响应包。此例中,用户通过图2的 SOAP 请求包向远程的某个提供时间和气温信息的网络服务发出请求,该请求包中包含了请求参数:Dallas。远程网络服务收到请求包之后,经过解析和相关计算,以图3所示的形式向用户返回相应的 SOAP 响应包,包中包含了有关 Dallas 的气温资料信息:108。

迄今,虽然 SOAP 规范中尚未涉及到双向通讯问题,其安全性还有待于进一步开发,但是在业界各大公司的支持下,SOAP 必将有更大的发展。目前,除了 IBM 和微软对 SOAP 的大力支持外,Iona 将在 Orbix 2000 中支持 SOAP,Rogue Wave 将在 Nouveau 对象请求代理中使用 SOAP,Develop-Mentor 已经完成了 SOAP 的 Perl、Java 和 COM 的参考实现版本^[2]。有迹象表明,在下一代 Internet 的体系结构中,SOAP 将会在网络服务方面起举足轻重的作用。

```
POST/temperature HTTP/1.1
Host: www.timeandtemperature.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:GetCurrentTemperature xmlns:m="Some-URI">
<city>Dallas</city>
</m:GetCurrentTemperature>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

图2 SOAP 请求包

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
```

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/"
SOAP-ENV:encodingStyle=
"http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<m:GetCurrentTemperatureResponse
xmlns:m="Some-URI">
<temperature>108</temperature>
</m:GetCurrentTemperatureResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

图3 SOAP 响应包

三、网络服务访问模式

3.1 网络服务访问的 PULL 模式

在 Internet 上,通常采用拉(PULL)模式访问网络服务读取信息,即由用户首先发出请求,服务器根据请求的内容向用户发送信息。其具体访问过程可归纳如下:

1. 客户根据网络服务描述,构造 SOAP 请求数据包;
2. 并通过 HTTP 将此数据包发送到网络服务提供端;
3. 网络服务提供端解析 SOAP 数据包,分析出客户请求方法名和请求参数,进行相关计算。这里的计算可以是基于本机的计算,也可以是基于 CORBA 或 DCOM 的分布式计算。
4. 网络服务提供端将计算结果封装到 SOAP 响应数据包中,返回给客户。
5. 客户端解析返回的结果,并利用此结果完成自己的业务逻辑。

在该类访问过程中,网络服务提供端完全处于被动地位,只有当客户发出请求时,它才被激活,返回客户所需的结果。在此模式中,网络服务端无法提供主动服务,而主动服务往往是客户所需要的。比如说,客户在自己的应用中集成了股票行情的网络服务,他希望当某种股票的振幅超过一定的门限值时,股票网络服务能够主动地向他报告,以避免自己不断查询所带来的麻烦。对于这种需求,网络服务访问的 PULL 模式

就显得无能为力。

本文提出的网络服务访问的推送模式,即 PUSH 模式,可以支持网络服务端向客户提供主动服务。

3.2 网络服务访问的 PUSH 模式

PUSH 是指“由网络发起的信息传送”^[4]。在 PUSH 模式中,网络可以主动地向用户发起呼叫,并向用户发送信息。网络向发起呼叫的条件可以由用户预先指定。PUSH 模式为用户开办一系列,如提醒服务、信息定制服务等业务提供了方便。PUSH 技术目前在 WAP 中已得到广泛的应用。本文通过对 PUSH 技术和网络服务的特点进行研究,首次提出了将 PUSH 技术应用于网络服务的两种实现方案。下面以客户端对网络中设备进行故障实时监测为例,给出 PUSH 模式的这两种实现方案。

假设在被监测的网络中,有一台数据采集机,用于获取被监测设备故障的实时数据;有一台服务器,对外提供网络服务;有若干台客户机,通过图形界面实时监视设备的故障状况。

第一种网络服务访问 PUSH 模式的实现方案如图 4 所示,它需要客户端和服务器端同时对外提供网络服务。

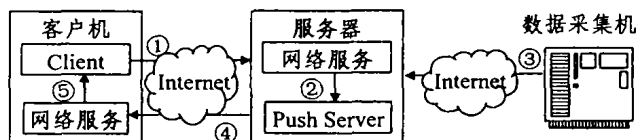


图4 网络服务访问 PUSH 模式的实现方案一

图4所示的客户端不仅需要运行监视设备告警信息的用户图形界面,还需要对外提供一个简单的网络服务。该服务的唯一的调用者是服务器,其主要功能是接收由服务器端发送来的实时告警数据。

服务器端也由两部分组成:完成业务逻辑所需的网络服务的集合和网络服务访问推送器(Push Server)。服务器端提供的网络服务集合中至少应包含两种服务:“客户登记服务”和“实时数据接收服务”,前者用于记录所有有关需要实时数据的客户的信息,后者用于接收来自数据采集机的实时告警信息。这两种服务分别由客户机和数据采集机调用。“告警数据推送器(Push Server)”用于向客户端推送实时告警信息。

数据采集机与被监测的设备直接相连,负责采集实时告警数据。

在此方案中,数据推送过程如下:客户在需要服务器提供“实时告警信息监测”的网络服务时,他通过 UDDI 的“服务发现”找到该服务以及其服务描述文件。然后,客户调用服务器端的“客户登记网络服务”,将自己的“实时告警信息接收”网络服务的 URL 发送到该服务器(①)。服务器将此信息传递给“Push Server”保存(Push Server 可以同时保存多个客户信息)(②)。数据采集机采集到告警信息后,调用服务器的“实时数据接收”网络服务,将采集到的告警信息发送到服务器(③)。服务器的“Push Server”收到实时告警后,根据已注册客户的 URL,逐个地调用各客户端的“实时告警信息接收”网络服务,推送相关的告警信息(④)。最后,客户端将接收到的告警信息显示在图形界面上(⑤)。

下面图5是服务器端“客户登记”网络服务和“实时数据接收”网络服务的算法流程;图6是客户端获取网络服务的算法流程。

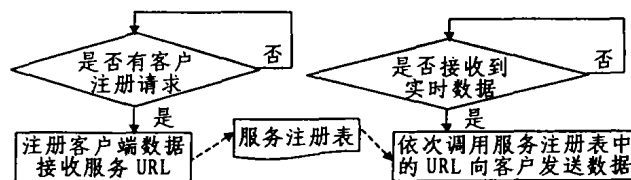


图5 服务器端提供网络服务流程

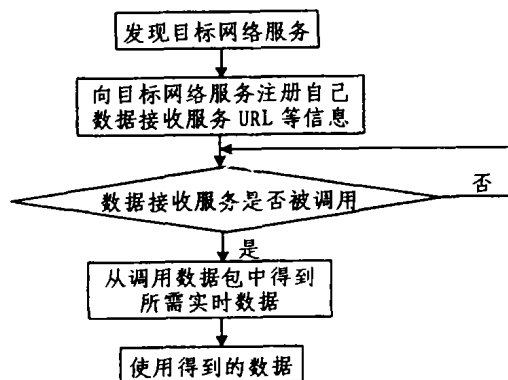


图6 客户端获取网络服务流程

在上述数据推送方案中,客户端是通过对外提供网络服务来接收告警数据的。也就是说,这里的客户端必须安装 Web 服务器,开通 WWW 服务,这对于某些追求瘦客户端的客户来说是无法接收的。因此,这种方案更适合于服务器到服务器的实时数据推送。

本文提出网络服务访问 PUSH 模式的第二种实现方案如图7所示,其中,客户端不需要向外提供数据接收网络服务,因此可以满足用户关于瘦客户端的需要。

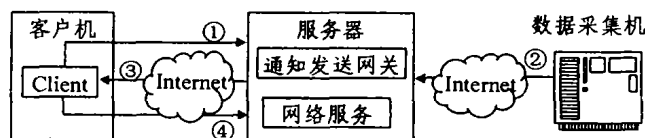


图7 网络服务访问 PUSH 模式的实现方案二

在方案二中,数据采集机的工作方式与方案一相同,仅客户端和服务器端的配置发生了变化。客户端不再配置“数据接收”网络服务,只提供与用户交互的输出显示控制 Client。服务器端也不再配置“告警数据推送器”(Push Server),而是增加了一个“通知发送网关”,此网关用于在服务器和 Client 之间建立一条用于发送通知的持久连接。

服务器端至少对外提供如下两种网络服务,即,“实时数据接收”服务和“实时数据发送”服务。前者被数据采集机调用,接收来自数据采集机的实时告警数据;后者由 Client 调用,用于向 Client 推送实时告警数据。

方案二的数据推送过程如下:当客户端需要利用网络服务的数据推送模式监测实时告警数据时,首先访问服务器端的“通知发送网关”(①)。网关在服务器端和客户端建立起一条持久的 TCP 连接。当数据采集机采集到实时告警数据后,调用服务器端的“实时数据接收”网络服务,将数据传送到服务器(②)。服务器一旦收到实时告警数据,它的“通知发送网关”便通过已建立起来的连接,向 Client 发送一个通知,告知

(下转第93页)

要的反映文本之间的差异的因素,用公式(3)同样可以计算出它们的差异程度。具体可以这样来做,对于不同文本中相应的特征,每出现一次就用一位的码字表示,再比较它们的位数,按位数长的码字为基准,位数短的补齐到与位数长的一样,相差的部分按照该位未出现该文本特征而对应的码字补齐,然后再用公式(3)计算文本之间的相似度,同样可以得到文本之间的定量的差异描述。

4 具体的步骤

根据以上分析,给出基于汉明距离的文本相似度计算的具体步骤:

- 1) 给出文本集与查询式的信息特征,排列成 n 位的序列;
- 2) 根据文本特征信息,得到与每一文本 1-1 对应的码字,由此构成码字集合;
- 3) 根据公式(3)计算相似度;
- 4) 根据文本之间相似度的大小,进行具体的文本分类等后续工作。

5 实验

我们做了有关的实验,进行比较测试。给定一个含有 50 篇文本的集合,这些文本分为计算机应用、图像信号处理、分布式系统等 5 个类别,各类的文本数见下表:

网络与智能	分布式系统	计算机应用	图像信号处理	计算机辅助
5篇	7篇	18篇	17篇	3篇

首先对文本进行分词、索引、文摘等处理,然后建立索引库。选定智能技术、神经网络、专家系统、信息处理、计算机应用、计算机辅助等 10 个信息特征,排成一个序列,集合中的每

一文本按照这 10 个分量的有或无与 10 位的码字建立 1-1 对应关系,构成码字集合。用户给定的查询式也用同样的序列得到码字。根据(3)式计算查询式与文本之间的相似度,然后进行文本分类,得到查全率和查准率分别为 83% 和 87%。而采用传统的向量空间模型方法(用公式(1)进行相似度计算),得到的查全率和查准率分别为 79% 和 81%。因此基于汉明距离的文本相似度计算方法优于传统方法。

本实验用的文本集相对较小,所以两种方法速度上的差别不是很明显,若文本集很大,相信该方法在速度上的优势一定更显著。

分析与结论 本文提出了文本相似度计算的一种新方法,与其它文本相似度计算公式相比较,该方法跳出传统的思想,而是采用码字的方法来表征文本信息的特征,它不仅限于关键字等孤立的信息的表征,还为联合地描述文本信息提供了可能。其次该方法只是利用模 2 加等运算,其方便性是不言而喻的,它完全避免了诸如在欧氏空间中求相似度的大量乘法运算,可较大地提高计算速度。实验表明,该方法是有优越性的。目前,我们正着手做进一步的后继工作,完善该方法并做进一步的实验。

参考文献

- 1 Salton G. Automatic text processing: the transformation analysis and retrieval of information by computer. Reading, Pennsylvania: Aoldison-wesley, 1989
- 2 Sheridan P, Ballerini J P. Experiments in multilingual information retrieval using the SPIDER system. In: Proc the 8th Annual ACM SIGIR Intl. Conf. on Research and Development in Information Reterieval, Zurich, 1996. 58~65
- 3 潘谦红,王炬. 基于属性论的文本相似度计算. 计算机学报, 1999, 22(6): 651~655
- 4 周荫清. 信息理论基础. 北京航空航天大学出版社, 1993

(上接第 68 页)

客户端服务器端已收到实时告警数据(③)。客户端收到通知后,调用服务器端的“实时数据发送”服务,获取服务器推送过来的数据(④)。

如前所述,网络服务的优势之一是可以穿越任何防火墙和路由器进行工作。然而,这一特征使网络服务的实现受到了限制:网络服务只能从 80 端口对外提供服务。所以“通知发送网关”与 Client 建立连接时,也必须使用 80 端口。本文利用传统 CGI 程序处理 HTTP 请求和响应的特性,解决了这一端口使用上的矛盾,建立了“通知发送网关”与 Client 之间的持久连接。“通知发送网关”的工作流程与客户端工作流程分别如图 8 和图 9 所示。

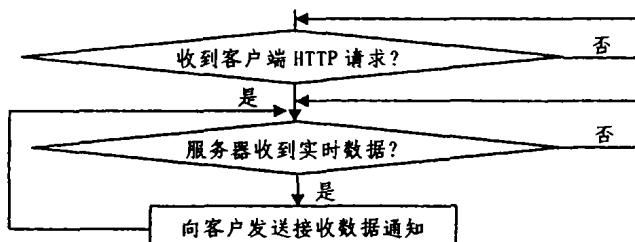


图 8 持久连接的通知发送网关流程

相比较而言,第二种方案实现简单,对客户端没有额外的配置要求,比较适合于最终用户对网络服务的使用。其缺点是当大量客户访问同一服务时,客户端与服务器的“通知发送网关”之间建立持久连接的数目也会增加,这样可能导致服务器性能下降。因此,这一部分还需进一步优化。

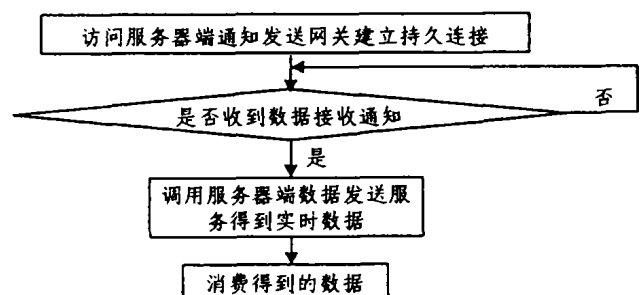


图 9 持久连接的客户端流程

结论 针对拉模式的不足,本文提出了在网络服务中服务器推送数据的两个模式,一个模式适用于网络服务之间的实时交互,另一个模式适用于网络服务与普通客户的交互。这两种模式基本上弥补了拉模式访问网络服务的局限性,成为拉模式访问网络服务的一个很好的补充。

参考文献

- 1 Gudgin M. et al. SOAP Version 1.2. W3C Working Draft 9 July 2001. <http://www.w3.org/TR/soap12/>
- 2 Bosworth. Developing Web Services. Data Engineering 2001 Proceedings 17th International Conference on, 2001(s): 477~481
- 3 Jepsen. SOAP cleans up interoperability problems on the web. IT Professional, 2001, 3(1): 52~55
- 4 WAP Forum. Wireless Application Protocol Push Architectural Overview. Nov. 1999. <http://www.wapforum.org>
- 5 Ariba, IBM, Microsoft. UDDI Technical White Paper. Sep. 6 2000. <http://www.uddi.org/>
- 6 Sleeper B, Robins B. Defining Web Services. <http://www.stencilgroup.com/>