

异构服务和分布式数据的动态集成

彭树青 陈德运

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

摘 要 对于企业而言,充分利用内、外部资源来提高效率是极为重要的。给出了以企业服务总线 ESB 为基础的服务-数据动态集成框架,在此框架中将业务操作分解为多个服务-数据流,同时提出了一种新的服务-数据流评价策略以衡量分解的合理性。利用动态路由机制完成抽象服务与服务提供者、抽象数据与数据源之间的映射。服务-数据路由对抽象服务描述表 ASDT、抽象服务映射表 ASMT、抽象数据描述表 ADDT 和抽象数据映射表 ADMT 进行更新来消除其中的冲突。最后将研究应用于“黑龙江省运政系统”的设计中,证明了该方法的可行性。

关键词 企业服务总线,服务-数据流,动态路由机制,服务-数据路由

中图分类号 TP311.5 **文献标识码** A

Dynamic Integration of Disparate Services and Distributed Data

PENG Shu-qing CHEN De-yun

(College of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract It is very important for enterprises to utilize resources efficiently inside and outside in order to improve the efficiency. A new framework that services and data are composed dynamically based on enterprise service bus(ESB) was proposed, in which business process was analyzed into service-data flows and the weight computation method was used for evaluating the separating rationality. Dynamic routing mechanism was applied to map abstract service to service providers and map abstract data to datasources. Abstract service description table (ASDT), abstract service mapping table (ASMT), abstract data description table (ADDT), and abstract data mapping table (ADMT) were modified by service-data router for solving the conflicts. Finally, the method has also been applied in Yunzheng System of Heilongjiang Province, in order to prove it's feasibility.

Keywords Enterprise service bus, Service-data flows, Dynamic routing mechanism, Service-data router

1 引言

目前,面向服务的体系结构 SOA(Service Oriented Architecture)已成为企业应用集成的主导方向,该方法可以在不修改客户端应用的前提下,将互联网上的资源进行动态集成^[1],以减小开发成本提高开发效率。Fujii 实现了基于语义的服务动态合成系统,该系统主要包括:语义功能评价函数、中间件和基于语义的服务合成机制^[2]。Penta 依据功能性和非功能性约束实现了一个服务动态绑定框架,该框架支持 3 种形式的绑定,即运行前绑定、运行时绑定和运行时再绑定^[3]。Laliwala 提出了基于语义和规则的服务动态集成机制,其业务处理过程采用了事件-条件-行为规则、前向链算法和后向链算法^[4]。Chafle 给出了生成和执行服务的解决方案,该方案能够适应集成、部署和运行阶段的系统环境变化^[5]。

ESB 的出现是消息中间件发展的必然结果^[6],它采用了“总线”这种模式来管理和简化应用之间的集成拓扑结构,以

开放标准为基础支持应用之间在消息、事件和服务级别上的动态互联。因此,ESB 是一种在松散耦合服务和应用之间的标准集成方式^[7]。在 SOA 中,ESB 经常被视为服务连接和信息转换的基础,这种结构的显著特点是服务定义与调用相分离。服务和数据经常被部署在互联网和企业内部网上,使用 Web 服务描述语言(WSDL)进行定义,有自己的接口和方法,任何结点中的数据库都可以作为数据源。同时,ESB 也是一种支持消息接受、网络服务、数据转换和智能路由的标准集成平台,能实现多种应用的可靠连接和协调^[8]。ESB 主要包括 4 种元素:基于消息导向的中间件(MOM)、网络服务、基于内容的智能路由和基于 XML 的数据转换机制。智能路由的核心功能是支持分布式应用和异构性服务,所采用的协议通常包括:HTTP, SOAP, JMS, MQ, SMTP 和 FTP^[9]。但目前 ESB 的路由技术仅局限于静态路由表^[10],使用服务提供者和数据源的 URI 地址来描述路由路径,一旦 ESB 初始化之后,路由路径将不再发生变化。

本文给出了一种异构服务和分布式数据的动态集成框

到稿日期:2009-07-20 返修日期:2009-09-28 本文受国家自然科学基金(60572153),国家教育部重点科技项目(204043),黑龙江省重点科技攻关项目(GC05A510),哈尔滨市高新技术产业基金(2008FG2CG021)资助。

彭树青(1973—),男,博士生,主要研究方向为软件工程和软件体系结构,E-mail: peng04510451@yahoo. com. cn;陈德运(1962—),男,博士,教授,博士生导师,主要研究方向为软件工程和软件体系结构。

架,在此框架中将业务操作分解为多个服务-数据流,同时提出了一种新的服务流评价策略以衡量分解的合理性。利用动态路由机制完成抽象服务与服务提供者、抽象数据与数据源之间的映射。同时将该方法应用于“黑龙江省运政系统”的设计中,实现了省运管局各部门和单位之间异构服务和分布式数据的动态共享与集成。

2 服务-数据动态集成框架

业务操作可以通过集成若干服务,并在一定的数据源上执行来实现。当服务按照顺序执行结束时,整个业务操作也就完成了。服务提供者分布在网络的各个结点上,为了集成异构服务与数据源,本文使用 ESB 来连接服务请求者和服务提供者。动态集成和执行服务的框架如图 1 所示。

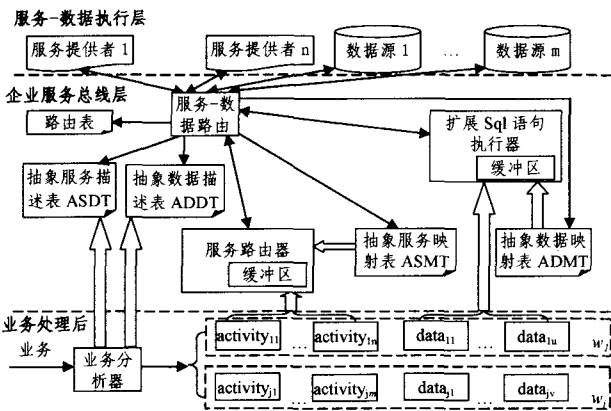


图 1 业务操作中的服务-数据动态集成

该框架主要包括:业务处理层、企业服务总线层和服务-数据执行层。业务处理层负责将业务操作分解为若干服务-数据流,并提交企业服务总线层进行处理;企业服务总线层负责集成异构服务和分布式数据;服务-数据执行层给出了服务提供者和数据源。

2.1 业务处理层

在业务处理层中,业务分析器依据抽象服务描述表 ASDT 和抽象数据描述表 ADDT 将业务分解为若干服务流和数据流。对服务流和数据流分别进行考虑的原因是:数据访问通常可以使用扩展的 Sql 语句在指定的时间内完成,而不同服务实现模块之间粒度的差别却很大。对二者分别进行考虑,可以便于度量业务分解的合理性。从逻辑上讲,一项业务可以分解为若干个服务流。按照某种观点可将其业务操作分解为 9 个服务,同时按照另外一种观点又可以将其分解为 6 个服务。但无论按照哪种方式进行分解,都可以完成这项业务操作。业务分解的原则如下:

(1)业务应该依据其逻辑工作顺序进行分解,分解的每个步骤都可以视为一项服务。对于每个步骤而言,如果不能在 ASDT 中找到相应的服务,则应该继续对其进行分解。

(2)业务分析器通过查询 ASDT 寻找更可靠的服务来实现业务操作,同时应使所用到的服务的数量最小化。尽管这种分解方式未必符合逻辑工作顺序,但可以提高工作效率,降低风险代价。

在业务分解完成之后,某些服务提供者会变得不稳定或不能访问。为了使整个业务操作得以顺利执行,应采取一定的评价策略来度量服务-数据流的执行效率与可靠程度。对

于每种分解方式,给出一个权值以衡量分解的合理性。例如,业务操作分解为服务-数据流 $activity_{y1}, \dots, activity_{yi}, \dots, activity_{yn}, data_{11}, \dots, data_{1n}$, 其权值设定为 w_1 ; 同时该业务操作也可以分解为 $activity_{y1}, \dots, activity_{yj}, \dots, activity_{ym}, data_{11}, \dots, data_{1n}$, 其权值设定为 w_j 。对每个服务-数据流进行评价,并依据其评分进行排序。得分最高的服务-数据流将首先被执行,如果它不能正常工作,再依次尝试执行得分低的。通常评价权值是根据分解的步骤、服务实现模块的粒度和服务属性来进行计算的。对于服务-数据流 $activity_{y1}, \dots, activity_{yj}, \dots, activity_{ym}, data_{11}, \dots, data_{1n}$, 其评价权值 w 的计算如式(1)所示:

$$w = \sum_{i=1}^m (c(activity_{yi}) + \frac{1}{g(activity_{yi})}) \tag{1}$$

从逻辑上讲,如果服务 $activity_{yi}$ 符合工作顺序流程,则将 $c(activity_{yi})$ 的值设置为 0.1; 否则其值设置为 0.2。其含义是:如果分解过程不符合逻辑工作顺序,代价将加倍。 $g(activity_{yi})$ 表示服务 $activity_{yi}$ 实现模块的粒度,其值按照实现模块的规模来进行计算。式(1)的含义是:如果分解过程符合逻辑工作顺序且使用的服务数目更少,则得到的服务-数据流应该首先被考虑。

2.2 企业服务总线层

企业服务总线层主要包括服务-数据路由、服务路由器和扩展 Sql 语句执行器 3 个部件。抽象服务描述表 ASDT、抽象数据描述表 ADDT、抽象服务映射表 ASMT、抽象数据映射表 ADMT 和路由表为服务和数据派发提供依据。ASDT 描述了通过 ESB 所能获得的所有服务,这仅仅是服务的抽象描述,不是真正的实现模块。在 ASDT 中,主要包括:服务名及对应实现模块的函数名和规模等信息,如表 1 所列。

表 1 抽象服务描述表 ASDT

服务名	函数名	规模	...
As ₁	Find_Account	90k	...
...	...	...	...
As _i	Compute_interest	10k	...
...	...	...	...

ADDT 描述了通过 ESB 所能访问的数据源,这仅仅是数据源的抽象描述,不是真正的实现模块。在 ADDT 中,主要包括:表名、数据库及表描述等信息,如表 2 所列。

表 2 抽象数据描述表 ADDT

表名	数据库	表描述	...
Table ₁	DataBase ₁	TableDec ₁	...
...	...	...	...
Table _i	DataBase _i	TableDec _i	...
...	...	...	...

ASMT 用于完成抽象服务与服务提供者之间的映射,某个服务可以对应几个不同的服务实现模块,如表 3 所列。

表 3 抽象服务映射表 ASMT

服务名	服务提供者的 URI	服务实现模块描述	...
As ₁	202.114.202.808	ServiceDec ₁	...
...	...	...	...
As _i	202.114.201.808	ServiceDec _i	...
...	...	...	...

ADMT 用于完成抽象数据与数据源之间的映射,如表 4 所列。

表 4 抽象数据映射表 ADMT

表名	数据源的 URI	数据源描述	...
Table _i	202.114.101.808	DataSourceDec _i	...
...	...	...	...
Table _j	202.114.102.808	DataSourceDec _j	...
...	...	...	...

服务提供者给出了具有不同功能的服务实现模块,因而某项服务可由不同的提供者来实现。系统运行一段时间之后,提供者会发现自身的实现模块中存在着一些问题,因而必须对模块进行修改或发布新的模块。这种变化必须通知服务请求者,以解决抽象服务与其实现模块之间的冲突。提供者发送消息通知服务-数据路由将要增加新服务、删除服务或更新某项服务,消息的主要内容包括服务变化说明 service_change_specification。当删除现有数据源或增加新数据源时,数据提供者应发送消息通知服务-数据路由,消息的主要内容包括数据变化说明 data_change_specification。服务-数据路由将修改 ASDT,ADDT,ASMT,ADMT 和路由表,其目的是使抽象服务和提供者、抽象数据与数据源之间保持一致。ASDT,ADDT,ASMT,ADMT 和路由表的更新步骤如下:

(1)服务-数据路由发送消息通知服务路由器和扩展 Sql 语句执行器 ASDT,ADDT,ASMT,ADMT 将被修改(接收到此消息后,服务路由器和扩展 Sql 语句执行器将停止工作)。

(2)if(update_type == “service”)服务-数据路由依据 service_change_specification 修改 ASDT 和 ASMT 中的对应项,同时根据 service_change_specification 的 URI 地址路径修改路由表。

(3)if(update_type == “data”)服务-数据路由依据 data_change_specification 修改 ADDT 和 ADMT 中的对应项,同时根据 data_change_specification 的 URI 地址路径修改路由表。

(4)服务-数据路由发送消息通知服务路由器和扩展 Sql 语句执行器更新过程结束(接收到此消息后,服务路由器重新处理缓冲区中的服务流,扩展 Sql 语句执行器重新处理缓冲区中的数据流)。

服务路由器从缓冲区中提取服务流,通过查寻 ASMT 得到每个服务提供者的 URI 地址及其描述信息。由于不同的服务提供者可以实现同一服务功能,因而需要从多个候选提供者中选择一个最佳的。服务路由器将依据用户预先定义的规则(例如,用户的喜好)、服务-数据流的评价权值和服务实现模块的属性来确定最佳提供者。

最佳服务提供者和数据源的 URI 地址及其相关信息被发送到服务-数据路由中,以等待进一步的处理。服务-数据路由依次执行每项服务请求和数据操作请求,通过查询路由表,根据 URI 地址确定其路由路径。当服务实现模块发生变化时,服务提供者和数据源将发送消息通知服务-数据路由停止当前的处理工作,完成 ASDT,ADDT,ASMT,ADMT 和路由表的更新任务。

服务-数据路由可以完成具有不同格式和内容的消息之间的相互转换,同时它还能屏蔽消息和协议中的差异。这一特性可以使服务-数据路由透明地发送和接收消息。

2.3 服务-数据执行层

在执行层中存在着大量的服务提供者和数据源,每个提供者给出了多种服务功能,包括:计算和通讯等。在任何时刻,提供者都可以添加、删除和修改服务实现模块和数据。但

在更新之后,必须发送消息通知服务-数据路由。通过使用动态集成框架,多种服务可以被透明地集成在一起来完成某项业务操作处理。

3 服务路由和扩展 SQL 语句的执行

依据评分,对服务-数据流进行排序。首先尝试得分最高的,将其服务流送入服务路由器的缓冲区中,将其数据流送入扩展 Sql 语句执行器的缓冲区中,并分别进行处理。如果失败,则将依次尝试得分低的服务-数据流。服务-数据路由通过查询 ASDT 和 ASMT 将服务流转化为一系列的服务提供者的 URI 地址;扩展 Sql 语句执行器通过查询 ADDT 和 ADMT 将数据流转化为一系列的<sql command,数据源 URI 地址>。服务-数据路由查询路由表获取服务提供者和数据源的路由路径,并分别向其发送服务请求和数据访问请求。

服务和数据路由的时序图如图 2 所示。

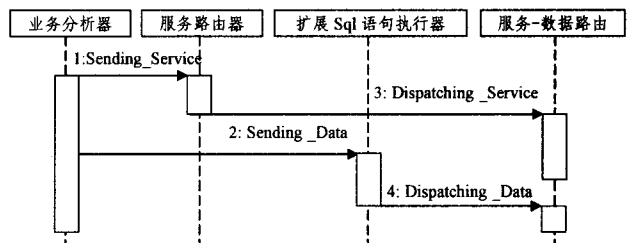


图 2 服务和数据路由的时序图

当业务分析器调用 Sending_service 方法时,服务流将被传送到服务路由器中。服务路由器从候选服务中选择最佳服务,并调用 Dispatching_service 方法将其结果传送到服务-数据路由中。当业务分析器调用 Sending_data 方法时,数据流将被传送到扩展 Sql 语句执行器中。扩展 Sql 语句执行器确定对应的数据源,同时调用 Dispatching_data 方法将其结果传送到服务-数据路由中。

4 系统实现

本文提出的服务-数据动态集成框架已应用在“黑龙江省运政系统”业务处理模块的设计中。系统运行结果如图 3 所示。在黑龙江省运管局各部门及各地下属单位部署该系统,实现了部门、单位间异构服务和分布式数据的动态共享和集成。该系统覆盖全省 13 个地市,采用统一管理、分散应用的方式来组织和管理全省的道路运输政务业务。省运管局通过“黑龙江省运政系统”,可以从全省的角度,统一管理和监视全省的运营业务发展情况和变化,从而为行业监管行为提供可靠的支持。通过该系统近两年的应用运行,使省运管局的工作效率和业务质量有了较大程度的提高。

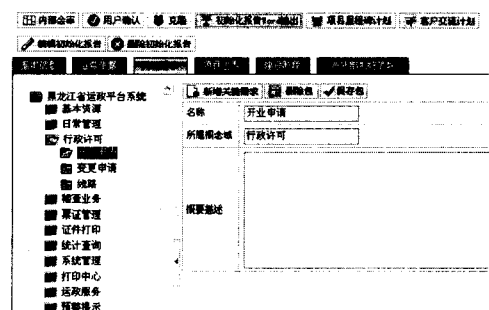


图 3 “黑龙江省运政系统”运行图

平均值由强制性失效的值评估冲突失效。

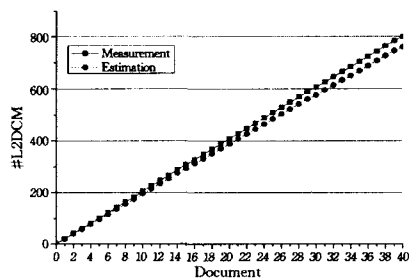


图3 圈内模型的实验数据图

5.2 跨圈模型的实验数据

利用第2个文档集来实验跨圈过滤,得到的实验结果如图4所示。我们还是使用在圈内过滤时的冲突失效和强制性失效的平均比率来评估冲突失效值。

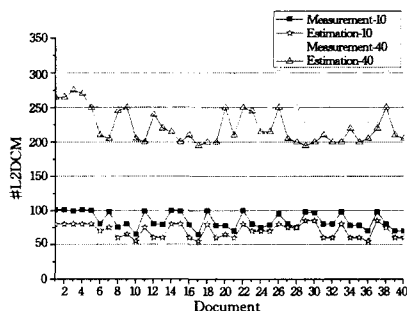


图4 跨圈模型的实验数据图

结束语 本文主要提出在通过自动机对XML数据流进行过滤的情况下如何过滤的结果进行精确的评估。通过对缓

存的分解,两段式地评估过滤的过程,从而得到与实际结果较为符合的评估结果。本文提出的圈内模型和跨圈模型经过实验证明存在较高的精确度。下一阶段的研究重点可以考虑,对于优化自动机的过滤方式,如何利用本文所建立的模型验证优化过的自动机能够既保证过滤效率的提高又能确保缓存查询的丢失。

参考文献

- [1] Altinel M, Franklin M J. Efficient filtering of XML documents for selective dissemination of information[C]//VLDB. 2000
- [2] Chan C-Y, Felber P, Garofalakis M, et al. Efficient filtering of XML documents with XPath expressions[J]. VLDB Journal (Special Issue on XML), 2002, 11(4)
- [3] Diao Y, Altinel M, Franklin M J, et al. Path sharing and predicate evaluation for high-performance XML filtering [C] // TODS. December 2003
- [4] Diao Y, Fischer P, Franklin M J, et al. Yfilter: Efficient and scalable filtering of XML documents[C]//ICDE. 2002
- [5] Green T J, Miklau G, Onizuka M, et al. Processing XML streams with deterministic automata[C]//ICDT. 2002
- [6] Organization T S P. Sax: Simple API for XML[EB/OL]. <http://www.saxproject.org>
- [7] 吴劲, 卢显良, 任立勇, 等. 缓存失效策略的性能分析数学模型[J]. 电子科技大学学报, 2005, 34(4): 225-228
- [8] 吴劲, 卢显良, 任立勇, 等. 移动计算环境中缓存失效策略的归类研究法[J]. 计算机科学, 2004, 31(1): 39-41

(上接第170页)

结束语 本文给出了一种基于ESB的服务动态集成框架,业务操作在此框架中被分解为若干服务-数据流。同时提出了一种新的评价策略以衡量分解的有效性,给出了ASDT, ASMT, ADDT, ADMT和路由表的更新算法,以解决服务提供者和数据源更新时所引发的抽象描述与其实现模块之间的冲突。将该方法应用于“黑龙江省运政系统”的设计中,实现了省运管局各部门和单位之间异构服务和分布式数据的动态共享与集成。

参考文献

- [1] Kuppuraju S, Kumar A, Kumari G P. Case Study to Verify the Interoperability of a Service Oriented Architecture Stack[C]//Proc. of International Conference on Services Computing. 2007: 678-679
- [2] Fujii K, Suda T. Semantics-Based Dynamic Service Composition [J]. IEEE Journal on Selected Areas in Communications, 2005, 23(12): 2361-2372
- [3] Penta M, Esposito R, Villani M. WS Binder: A Framework to Enable Dynamic Binding of Composite Web Services[C]//Proc. of the 2006 International Workshop in Service-Oriented Software Engineering. 2006: 74-80

- [4] Laliwala Z, Majumdar P, Chaudhary S. Semantic and Rules Based Event-Driven Dynamic Web Services Composition for Automation of Business Process[C]//Proc. of IEEE Services Computing Workshops. 2006: 175-182
- [5] Chafle G, Dasgupta K, Kumar A. Adaptation in Web Service Composition and Execution[C]//Proc. of International Conference in Web Services. 2006: 549-557
- [6] 李晓东, 杨扬, 郭文彩. 基于企业服务总线的数据共享与交换台[J]. 计算机工程, 2006, 32(21): 217-219
- [7] 邵欢庆, 康建初. 企业服务总线的应用[J]. 计算机工程, 2007, 33(2): 220-222
- [8] Maréchaux J-L. Combining Service - Oriented Architecture and Event-Driven Architecture Using an Enterprise Service Bus [M]. IBM Developworks, 2006
- [9] Luo Min, Goldshlager B, Zhang Liang-jie. Designing and Implementing Enterprise Service Bus(ESB) and SOA Solutions[C]//Proc. of the International Conference on Services Computing. 2005: 14
- [10] Bai Xiao-ying, Xie Ji-hui. DRESR: Dynamic Routing in Enterprise Service Bus[C]//Proc. of International Conference on E-Business Engineering. 2007: 528-531