

基于服务质量的动态服务组合研究

吴海波^{1,2} 邓木生² 陈新喜²

(中南大学信息工程学院 长沙 410083)¹ (湖南铁道职业技术学院电气工程系 株洲 412001)

摘要 Web 服务技术近年来发展迅速,由于 Web 服务的质量问题,采用率一直很低。在 Web 服务动态组合的项目中分析了 Web 服务的 QoS 问题,提出了一种 QoS 计算模型和 QoS 优化计算方法,该方案具有支持 Web 服务动态绑定、自动化及半自动化执行和基于 QoS 的 Web 服务最优化选择等特点,最后给出了实现原型。

关键词 Web 服务,动态组合,QoS 模型

中图分类号 TP31 **文献标识码** A

Research on Dynamic Service Composition Based on QoS

WU Hai-bo^{1,2} DENG Mu-sheng² CHEN Xin-xi²

(College of Information Science and Engineer, Central-South University, Changsha 410083, China)¹

(Department of Electrical Engineering, Hunan Railway Professional Technology College, Zhuzhou 412001, China)²

Abstract Web Services technology has been developing rapidly in the recent years, but the adopt of rate is very low all along because of the problem of QoS in Web Services. So, this paper analyzed problem of QoS of Web Services in the project of Dynamic Composition of Web Services, presented a computation model of QoS and a computing method of QoS Optimization, the scheme supports dynamic binding of Web Services, automatic and semiautomatic executions and optimal choice of Web Services based on QoS and so on. Lastly, the prototype and implementation were introduced.

Keywords Web services, Dynamic composition, QoS model

1 引言

Web 服务的努力目标是通过使用 Web 标准实现应用程序之间通用的互操作性。Web 服务的基本规范(SOAP, WSDL, UDDI)使应用程序能够遵循一个松散耦合、与平台无关的模型来找到对方并进行交互。因此,Web 服务技术不受系统架构、地域和技术方案的限制和约束,可以非常快捷、灵活和方便地集成现有 IT 资源;通过对 Web 服务领域的研究,作者认为 Web 服务的组合是关键。目前以工作流方式对业务流程建模,实现业务流程处理的自动化,是较为成熟和合理的方式。因此,本文认为采用工作流方式组合 Web 服务来实现开放的、清晰的、松散耦合的业务流程,是下一代商业软件的模式,为企业业务集成提供了一种新的解决方案。在以工作流方式组合 Web 服务的过程中,流程执行阶段中的任务可能有许多候选的 Web 服务能满足流程建模阶段所提供的功能性需求,但这些候选 Web 服务又有不同的非功能性属性,比如价格等。为了在动态选择服务时区分这些 Web 服务,必须考虑这些 Web 服务的非功能性属性。为此,本文首先给出一套基于不同质量标准(如非功能性属性)的 Web 服务 QoS 模型;然后给出 QoS 模型的计算模型和最优化计算方法,并给出实现原型;最后进行总结并对下一步工作进行

展望。

2 Web 服务 QoS 模型

组合 Web 服务最大的特色就是能在运行时有效地动态选择服务。组合服务时由 Engine 依据一套 QoS 机制来动态选择 Web 服务。为此,首先要对 Web 服务的非功能性属性进行分析,并提出 Web 服务 QoS 的模型。在具体的应用领域,Web 服务的 QoS 模型中可能只有有限的几个 QoS 标准,但 QoS 模型必须是可扩展的,并且新 QoS 标准的加入不能对服务选择技术进行基础的改动。

针对如此之多的服务质量标准,有些服务质量带有很大的主观性,比如环境的 QoS,它可能依赖于 ISP 提供商的网络环境。为此,本文主要针对系统应用级的 QoS 标准,建议对组合 Web 服务采用一个可扩展的、多空间的 Web 服务 QoS 模型。模型建议采用了非功能性的属性的 QoS 标准是^[1-3]: (1) 执行价格 execution price; (2) 执行周期 execution duration; (3) 声誉 reputation; (4) 补偿率 compensation; (5) 惩罚率 punishment。

执行价格: Web 服务提供者给定的一个服务操作 op 的执行价格 $q_{price}(s, t)$ 作为服务请求者为完成流程中的任务 t 所执行该服务操作 op 要付出的费用。 t 表示组合服务中的一

到稿日期:2010-11-03 返修日期:2011-03-11 本文受湖南省自然科学基金(09JJ6193)资助。

吴海波(1979—),男,硕士,讲师,主要研究方向为网络安全、计算机应用技术、嵌入式系统;邓木生(1967—),男,副教授,主要研究方向为电气自动化、电子信息技术;陈新喜(1969—),男,讲师,主要研究方向为嵌入式系统、电子信息技术。

个任务, s 表示满足该任务 t 的 Web 服务。

执行期限: 对于执行任务 t 的每个服务 s 的执行期限 $Q_{du}(s, t)$ 等于请求发送到结果被收到的这段时间, 以秒为单位。执行期限的计算表达式: $Q_{du}(s, t) = T_{process}(s, t) + T_{trans}(s, t)$; $T_{process}(s, t)$ 是服务处理时间, $T_{trans}(s, t)$ 是传送时间, 它可从以前执行的服务操作中估计出来。

声誉: 一个服务 s 的声誉是对服务可信性衡量。它主要依靠终端用户使用服务的经验。不同的终端用户可能对同一个服务评价不同, 因此声誉标准也叫做环境 QoS 标准。为此, 采用终端用户给出的平均等级来衡量声誉 $q_{rep}(s_i)$ 的值, 其计算公式: $q_{rep}(s) = \sum_{i=0}^n i * p_i(s)$; $p_i(s)$ 是终端用户给服务 s 评价等级为 i 的概率, n 是等级范围。一般都会给终端用户一个评价等级范围, 就如同网络上给书评价等级一样, 一般是 $[0, 5]$ 。

补偿率: $q_{comp}(s)$ 表示 Web 服务的补偿率, 即服务提供者不能按其所承诺的标准来完成服务时, 应在顾客原先所付的执行价格中退款的比率。

惩罚率: $q_{pen}(s)$ 表示 Web 服务的惩罚率, 即当服务的请求者在已经超时, 不可回滚撤销的情况下要求取消已提交的服务时, 应付给服务提供者的赔偿与原执行价格的比率。

综合到上面给出的质量标准定义, 一个 Web 服务 s 的综合的质量向量是:

$$Q(s) = (q_{price}(s); q_{du}(s); q_{rep}(s); q_{comp}(s); q_{pen}(s)) \quad (1)$$

对上述这些单个质量标准值的计算方法并不是唯一的, 为了使质量模型有更好的扩展性, 并且质量标准的添加与删除并不影响 QoS 综合计算技术, 本文还须提出一套计算方法, 将这些多空间的质量标准统一在方法中, 并能通过方法选择出最优的 QoS 质量的 Web 服务。

3 QoS 计算模型及 QoS 最优化计算方法

在流程定义中的节点上可能有多个满足条件的候选 Web 服务, 根据上面提出的 Web 服务 QoS 模型, 为了找到最优的 Web 服务, 需要考虑多空间的质量标准。式(1)提供了单个候选 Web 服务的质量向量, 若有 n 个满足条件的候选 Web 服务 $\{s_1, s_2, \dots, s_n\}$, 则该任务 t 的质量 Q_t 表示如下:

$$Q_t = \left\{ \begin{array}{ccccc} Q_{price}(s_1) & Q_{du}(s_1) & Q_{rep}(s_1) & Q_{comp}(s_1) & Q_{pen}(s_1) \\ Q_{price}(s_2) & Q_{du}(s_2) & Q_{rep}(s_2) & Q_{comp}(s_2) & Q_{pen}(s_2) \\ \dots & \dots & \dots & \dots & \dots \\ Q_{price}(s_n) & Q_{du}(s_n) & Q_{rep}(s_n) & Q_{comp}(s_n) & Q_{pen}(s_n) \end{array} \right\} = \left\{ \begin{array}{ccccc} Q_{1,1} & Q_{1,2} & Q_{1,3} & Q_{1,4} & Q_{1,5} \\ Q_{2,1} & Q_{2,2} & Q_{2,3} & Q_{2,4} & Q_{2,5} \\ \dots & \dots & \dots & \dots & \dots \\ Q_{n,1} & Q_{n,2} & Q_{n,3} & Q_{n,4} & Q_{n,5} \end{array} \right\} \quad (2)$$

由于各种类型的质量标准的刻画度数和形式均不一样, 为了统一衡量服务质量, 本文采用一个简单的权重技术来计算 Web 服务的 QoS。本文归纳出所有的质量标准有两种可

能的特性: 有些质量标准是消极的, 值越高, 质量越低(比如执行时间、执行价格、补偿率和惩罚率); 而有些质量标准是积极的, 值越高, 质量越高(比如声誉)。

1. 针对消极的质量标准, 给出其权值的计算公式:

$$V_{ij} = \begin{cases} \frac{Q_j^{max} - Q_{ij}}{Q_j^{max}}, & Q_j^{max} \neq 0 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

Q_j^{max} 是式(2)中一种质量标准的最大值, 即 $Q_j^{max} = \text{Max}(Q_{i,j}), 1 \leq i \leq n$ 。同理, Q_j^{min} 是式(2)中一种质量标准的最小值, 即 $Q_j^{min} = \text{Min}(Q_{i,j}), 1 \leq i \leq n$ 。 V_{ij} 是式(2)中 $Q_{i,j}$ 所对应的权值, $V_{ij} \in [0, 1], j=1, 2, 4, 5$ (在这里分别表示执行价格、执行期限、补偿率和惩罚率); 式(3)表达的意思是: $Q_{i,j}$ 的值越大, 权值 V_{ij} 就越小, 其质量也越低, 反之同理。例如价格质量标准, 即 $Q_{1,j}$, 当执行价格越高, 即 $Q_{1,j}$ 越高, 则 $V_{1,j}$ 的值就越小, 质量也就越低, 反之亦然。

2. 针对积极的质量标准, 给出其权值的计算公式:

$$V_{ij} = \begin{cases} \frac{Q_{ij}}{Q_j^{max}}, & Q_j^{max} \neq 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

相关符号解释同上, 式(3)表达的意思是: $Q_{i,j}$ 的值越大, 权值 V_{ij} 就越大, 其质量也越高, 反之亦然。例如声誉标准, 即 $Q_{5,j}$, 当声誉越高, 即 $Q_{5,j}$ 越高, 则 $V_{5,j}$ 的值就越大, 质量也就越高, 反之亦然。

但是, 质量标准类型是可扩展的, 也是多空间的, 因此为了计算出每个服务综合的质量标度, 本文为每类质量标准提供了权值机制, 这样就统一了不同服务质量的质量标度。用户可按需求加入新的质量标准。对于不同质量标准, 在不同的应用需求中所占的比重是不同的, 为此本文引入权重 W_j 。Web 服务 QoS 计算公式如下:

$$Score(s_i) = \sum_{j=1}^n (V_{ij} * W_j) \quad (5)$$

式中, $Score(s_i)$ 表示服务 s_i 的 QoS 最终综合值, $W_j \in W_j [0, 1]$, 并且 $\sum_{j=1}^n W_j = 1$ 。 W_j 表示每个质量标准的权重。

综上所述, 从 n 个满足条件的候选 Web 服务中选择最优的 Web 服务的计算公式是:

$$S_{excellence}(i) = \text{MAX}_{i=1}^n (Score(s_i)) \quad (6)$$

4 Web 服务 QoS 计算模型实现原型

4.1 Web 服务对 QoS 的支持

当前 Web 服务体系对 QoS 的支持很有限, 其中 WSDL 对于 Web 服务的 QoS 描述能力非常有限, 它没有对服务的 QoS 度量进行具体的定义和描述, UDDI 服务发现机制也没有根据 QoS 指标查找服务的功能。为此, 本文对现有 WSDL 服务描述和 UDDI 的数据结构进行扩展, 在保证与现有规范兼容的同时支持 QoS。

由于同一 Web 服务功能性描述文档可能被多个不同 QoS 性能的 Web 服务实现文档所引用, 为了区分不同的 Web 服务实现, 在 Web 服务接口实现描述文档中加入 QoS 描述

机制:扩展 Web 服务接口实现描述文档中的 service 元素,在其中加入 QoS 子元素,同时,对相应的 UDDI 规范中的数据结构也进行扩展,增加一个新的数据结构,构成 UDDIe(Universal Description, Discovery and Integration Extension)。扩展的数据结构为 QoSBag,由 QoSName, QoSType 和 QoSValue 3 部分组成。由于 Web 服务的 QoS 模型是可扩展的,针对这种扩展性本文利用 UDDI 中的 tModel 来实现扩展机制,通过使用 tModel 注册一个 QoS 质量模型实例,一旦通过这种方法注册后,就可以方便地在 categoryBag 中增加一个相应的 tModel 标识符(称为 tModelKey)的引用来表示提供了符合某种 Web 服务的 QoS 的质量模型。因此, tModel 可以成为一种很好的描述 QoS 模型的方法。采用 tModel 作为 QoS 模型的载体,其最大的优点就是它具有很好的扩展性。

4.2 基于 QoS 的服务选择

基于 QoS 的服务选择^[4,5]如图 1 所示,首先需要服务提供者提供服务的功能性信息和 service 的质量信息,并将这些信息注册在 UDDIe 注册中心中。在 Web 服务动态组合的项目中,采用 BPEL4WS 规范来描述 workflow 方式的 Web 服务组合, Web 服务组合引擎本身也是一个 Web 服务。QoS 管理器用来实现 Web 服务的 QoS 机制。它可以实现为一个单独的 Web 服务或者实现为组合 Engine 的功能模块。其与引擎主要的交互过程有:

- 1) 服务请求者发送带有 QoS 需求的请求;
- 2) 引擎根据功能性请求从 UDDIe 中查找服务提供者;
- 3) 引擎调用 QoS 管理器获取满足功能性需求的服务提供者的 QoS 信息;
- 4) QoS 管理器根据 QoS 计算模型,计算出满足服务请求者的 QoS 需求的 Web 服务,并更新 Web 服务的 QoS 信息;
- 5) 引擎与 QoS 管理器返回的 Web 服务绑定并实施调用。

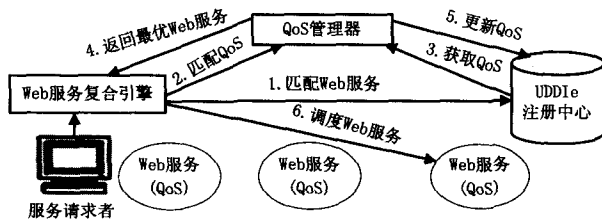


图 1 基于 QoS 选择

4.3 QoS 管理器的实现

Web 服务的 QoS 计算模型是可扩展的^[6-9],因此 QoS 管理器的设计也必须是可扩展的。在 Web 服务动态组合的项目中,并未将 QoS 管理器实现为单独的 Web 服务,而是将其实现为功能模块。QoS 管理器由 3 个重要部分组成: QoS 采集器、QoS 计算器和 QoS 更新器。QoS 采集器负责从候选的 Web 服务中获取 QoS 信息,其返回所有候选 Web 服务的质量 vector。QoS 计算器根据上面的 QoS 计算模型和最优计算方法,获取最优的候选 Web 服务,并将 Web 服务返回给引擎执行调度。QoS 更新器负责对 Web 服务的 QoS 值进行更新,比如执行时间等。其具体实现如图 2 所示。

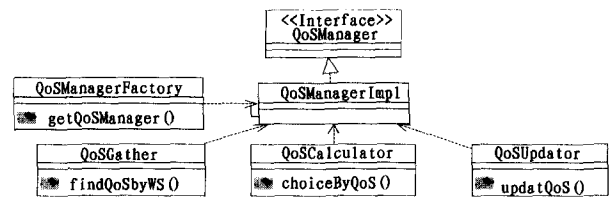


图 2 QoS 管理器类图

结束语 基于 QoS 选择的 Web 服务动态组合是个很活跃的研究课题。本文给出了一个可扩展的 QoS 模型,并给出了 QoS 模型的计算模型和最优化计算方法。结合目前的 Web 服务规范体系,扩展了相应的规范来使 Web 服务支持 QoS 机制。在动态组合 Web 服务项目中用 QoS 管理器实现了 QoS 模型及其计算模型。QoS 问题在 Web 服务领域是一个关键的问题。今后,我们将主要在 Web 服务领域中 QoS 的协商和 QoS 的主动性方面做进一步工作。

参考文献

- [1] Zeng L, Benatallah B, Dumas M, et al. Quality Driven Web Services Composition[C]// Proceedings of the 12th international conference on World Wide Web (WWW). Budapest, Hungary: ACM Press, May 2003
- [2] Sheth A, Cardoso J, Miller J, et al. QoS for service-oriented middleware[C]// Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI02). July 2002; 528-534
- [3] Fedosseev P. Composition of Web Services and QoS aspects[C]// Seminar: Data Communication and Distributed System WS2004
- [4] Al-Ali R, Rana O, Walker D, et al. G-QoSM: Grid service discovery using QoS properties [J]. Computing and Informatics Journal, Special Issue on Grid Computing, 2003, 21(5)
- [5] ShaikhAli A, Rana O F, Al-Ali R, et al. UDDIe: An Extended Registry for Web Services[C]// Workshop on Service Oriented Computing, Models, Architectures and Applications at SAINT Conference. Florida, US. IEEE Computer Society Press, January 2003
- [6] Al-Ali R J, Rana O F, Walker D W. G-QoSM: A Framework for Grid Quality of Service Management[C]// Proceedings of the UK e-Science Programme All Hands Meeting 2003. Nottingham, UK, September 2003
- [7] Liu Y, Ngu A H H, Zeng L Z. QoS Computation and Policing in Dynamic Web Service Selection[C]// WWW2004. New York, USA, 2004
- [8] Xia Yun-ni, Zhu Qing-sheng, Huang Yu, et al. A novel reduction approach to analyzing QoS of workflow processes[J]. Concurrency and Computation, Practice and Experience, 2009, 21(2): 205-223
- [9] Tian M, Gramm A, Ritter H, et al. Efficient Selection and Monitoring of QoS-aware Web services with the WS-QoS Framework [C]// 2004 IEEE/ WIC/ ACM International Conference on Web Intelligence (WI'04). Beijing, China, Sep. 2004,