

基于 QoS 生命周期管理的服务组合在线优化方法

崔曙光^{1,2}, 高春鸣¹, 万里平¹

CUI Shu-guang^{1,2}, GAO Chun-ming¹, WAN Li-ping¹

1. 湖南师范大学 数学与计算机科学学院, 长沙 410081

2. 株洲职业技术学院, 湖南 株洲 412001

1. College of Mathematics and Computer Science, Hunan Normal University, Changsha 410081, China

2. Zhuzhou Professional Technology College, Zhuzhou, Hunan 412001, China

E-mail: csg0628@163.com

CUI Shu-guang, GAO Chun-ming, WAN Li-ping. Web service composition online optimization method based on QoS lifecycle management frame. Computer Engineering and Applications, 2007, 43(19): 152-156.

Abstract: Currently, most researches only focus on some phase of QoS lifecycle management, this paper proposes a frame that supports the management of whole QoS lifecycle. Based on this frame, an AHP approach with automatically modifying judgement matrix is proposed to implement online strategy of local QoS optimization. Traditionally, consumer can only provide relative importance of QoS properties, but cannot describe the global QoS requirement. Aiming at this scenario, this approach provides an online local optimization solution.

Key words: QoS lifecycle; management frame; online optimization; Analytical Hierarchy Process(AHP); fault-tolerant

摘 要: 针对当前基于 QoS 的 Web 服务组合研究仅限于 QoS 生命周期的某一环节, 提出了一个支持 QoS 生命周期管理的 Web 服务框架, 基于该框架研究了服务组合在线优化的方法, 提出了自动修正判断矩阵的层次分析法来实现基于 QoS 的在线局部优化策略。该方法针对因各种原因用户只能提供各 QoS 属性之间的相对重要性, 而不能提供全局的 QoS 需求的应用场景, 提供了一种在线局部优化解决方案。

关键词: QoS 生命周期; 管理架构; 在线优化; 层次分析法; 容错

文章编号: 1002-8331(2007)19-0152-05 **文献标识码:** A **中图分类号:** TP311; TP393

1 前言

在分布、异构和自治的分布式应用的场景下, 多个 Web 服务按照 workflow 模型实现服务组合以满足用户需求, Web 服务质量的保证已是服务组合方法优劣的重要影响因素。有效的服务组合需要针对用户需求, 对工作流程中的每个业务服务动态绑定既满足功能需求又满足 QoS 需求的 Web 服务。随着相似 Web 服务数量的爆炸性增长, 具有相同或相近服务功能的服务数量巨大, 利用 QoS 属性选择成分服务是 Web 服务组合实用化的关键技术。

目前存在组合服务 QoS 属性的生命周期管理与管理架构研究不够。如文献[1]侧重于解决服务可信度的问题, 提出一个由第三方实体代理执行的两阶段 QoS 属性认证技术。文献[2]提出要对单个 Web 服务进行测试和监控, 以保证 Web 服务 QoS 的一致性, 强调通过统计分析用户反馈而得到服务执行的历史数据, 并以此更新服务的 QoS 值。以上提到的关于 QoS 属性的研究基本上只涉及组合服务 QoS 属性的生命周期的某一阶段, 缺乏 QoS 属性生命周期全面的管理; 另外支持 QoS 的

Web 服务的框架研究只考虑单个服务的情况, 而没有涉及到商业流程组合服务对 QoS 的支持。

服务组合优化策略可以分为局部优化策略与全局优化策略。大多数全局优化策略的研究在算法的通用性或求解效率尚不够理想, 例如文献[3]使用整数规划进行求解, 其求解效率与服务组合的规模和服务候选集合的大小直接相关。模拟退火或遗传算法等通用的智能搜索算法可以用于求解任意目标的组合服务 QoS 优化问题, 但是一般而言它们的收敛速度较慢^[4], 因此对于那些要求在运行时刻快速搜索组合方案的应用而言并不太适用。从上述方法所适用的组合服务场景分析, 许多工作提出的方法主要为脱机离线算法。

针对以上 Web 服务组合优化研究中出现的 QoS 属性的生命周期管理研究不足、算法求解效率较低而不能应用于运行时的优化和容错处理, 本文实现了 Web 服务 QoS 属性的生命周期管理框架, 研究的服务组合在线优化与容错算法效率较高, 可用于服务组合运行时的优化与容错处理。本文的 QoS 属性的生命周期管理框架支持服务组合的局部优化与全局优化, 但

基金项目: 湖南省重点科技攻关项目(the Key Technologies Program of Hunan Province of China under Grant No.05GK2002); 湖南省自然科学基金(the Natural Science Foundation of Hunan Province of China under Grant No.03JJY6023)。

作者简介: 崔曙光(1976-), 男, 硕士研究生, 主要研究方向为 Web 服务与中间件; 高春鸣(1957-), 通讯作者, 男, 教授, 硕士生导师, 主要研究方向为软件体系结构; 万里平(1981-), 男, 硕士研究生, 主要研究方向为分布式计算。

本文研究的是针对因各种原因用户只能提供各 QoS 属性之间的相对重要性,而不能提供全局的 QoS 需求的应用场景,如何实现在线局部优化的问题。

本文第 2 章给出了支持 QoS 生命周期管理的 Web 服务组合架构 WebJet;第 3 章介绍基于 QoS 的 Web 服务组合在线优化算法;第 4 章模拟与仿真;第 5 章总结。

2 支持 QoS 生命周期管理的 Web 服务组合架构 WebJet

WebJet 是一个服务组合组装与运行平台,遵循 Web Services 技术框架,为基于 Web 服务组合开发及应用提供核心功能,包括服务描述存储和服务组合平台两部分,服务组合平台在运行时产生的与服务发现、优化和容错临时信息由相应的代理管理。WebJet 实现了服务的发布、管理、组合、发现和匹配、优化和容错以及服务组合的运行。平台在服务装配的原理上类似于文献[5],但与之不同的是 WebJet 实现了整个组合服务流程中 QoS 生命周期管理。

2.1 支持 QoS 生命周期管理的 Web 服务组合架构

WebJet 支持 QoS 属性的生命周期管理,从服务发布开始,到服务组合的成分服务的发现与选择、服务组合的优化与容错处理环节中,QoS 属性作为重要因素出现。在服务发布阶段,将软件业务功能和 QoS 属性值发布到服务存储平台,服务存储测试验证用户提交的服务 QoS 属性;在服务的发现与选择阶段,综合考虑用户对单个服务的 QoS 需求和领域专家挖掘历史信息所得的经验值,对每个服务的 QoS 属性设定一个阈值,从每个候选广告服务集合中淘汰不满足阈值的服务;在服务组合优化阶段,基于 QoS 使用自动修正判断矩阵的层次分析法对候选广告服务集合中的服务进行排序,为每个活动从候选服务集合中选择最优的成分服务;在服务执行时,基于服务 QoS 监测的机制,在实际服务较广告服务的 QoS 下降或调用某个服务失败时会启动容错机制;流程执行完毕后,对反馈的相关 QoS 数据进行统计分析,结果写入 QoS 属性数据统计表,该统计结果实时同步到服务存储之中。QoS 属性的生命周期管理原理见图 1。

2.2 Web 服务组合架构 WebJet 的功能模块模块设计

WebJet 平台由以下功能模块构成:

(1) 服务发布模块

包括一个可扩充的发布框架,发布框架读入一份带注解的 Java 源文件(利用 J2SE 5.0 的 Annotation 程序注释功能),服务提供者对 QoS 属性测试并将测试数据用注解语言在 Java 类中描述,根据不同的 java 元数据定义,调用不同的注解解析模块及不同的发布接口发布到 SOAP 平台使客户端能够远程调用该服务,以及不同的服务描述存储方便客户发现服务,包括 UDDI 和 Web 服务语义描述存储仓库。Web 服务元数据^[7]可用于编辑 Java 文件来创建 Web 服务,自动地从 Java 文件生成 WSDL,或从给定的 WSDL 文档生成 Java Web 服务文件的框架。采用 java 元数据与不同服务描述存储的映射关系^[8],服务提供者提供的实测数据可存储在扩展的 UDDI 和 OWL-S 语义存储中作为 QoS 初始值。

为了提高服务提供者所提供的服务的可信度,在发布前需要服务测试代理对服务进行测试,包括功能与 QoS 属性两方面测试:提供者发布一个经过 XML 编码的测试套件,服务测试代理根据测试套件生成的测试用例对服务的功能进行测试;并用 QoS 断言对服务的 QoS 进行测试^[9]。只有通过测试后平台才将服务发布到服务存储。在测试时要考虑测试环境 with 用户实际运行环境的差异性,并由此对测试结果进行相应的转换。

(2) 服务描述存储模块

广告服务的存储采取语义 Web 服务 OWL-S 与 UDDI 结合的方式。UDDI 作为一种 Web 服务的分类目录,目前仍是 Web 服务发布和发现的首选;为了描述服务的语义,在 UDDI 之外建立 Web 服务语义存储,并建立语义 Web 服务与具体 Web 服务的一一对应^[8]。采用 OWL-S 的 profile 的可扩展属性定义 QoS 属性。当前的大多数 Web 服务描述模型用“名-值对”来表达 QoS 信息,针对这种数据结构的表达能力有限的弊端本文建立一个 QoS 本体来描述 QoS,使用 OWL Web 本体语言来实现,可以作为 OWL-S 的一个扩展来使用。

为了存储服务和 QoS 的信息,本文设计了服务表及 QoS 属性统计表,将 QoS 属性分为设置型和统计型两类,设置型是服务提供者预先定义好的并且在重新定义之前保持不变,如服务的价格,统计型表示 QoS 的值在服务调用结束时需重新计

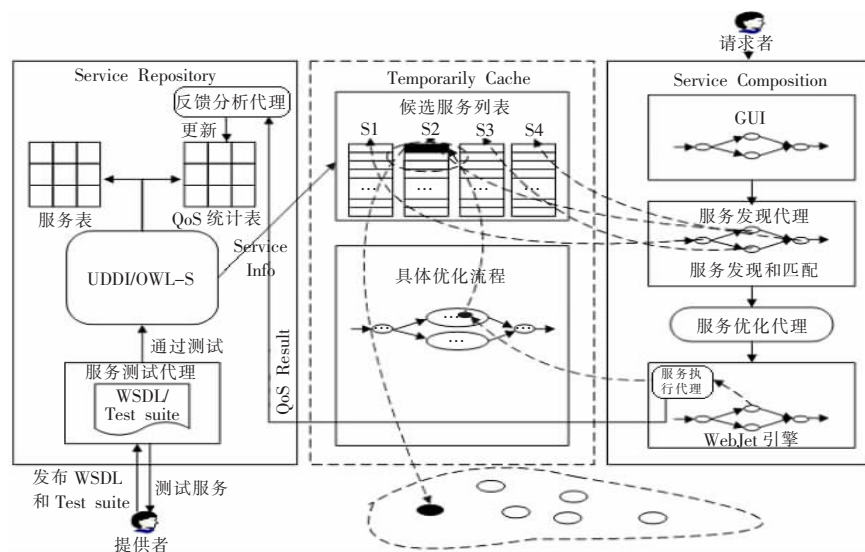


图 1 支持 QoS 生命周期管理的 Web 服务组合架构 WebJet

算,如服务的可用性。提供者只有修改设置型的属性值的权限,而统计型的值由反馈分析代理负责更新。这样充分体现了 QoS 值的公正性。

(3) 服务需求模块

在服务装配 GUI 界面上,用户或领域专家基于领域本体概念库将用户需求表达成 BPEL 描述的抽象业务服务流程。本文使用本体语言 OWL-S 来描述用户需求,具体包括基本信息、功能信息以及非功能属性的约束要求。用户通过操作导航界面输入需求信息,需求编辑器将用户输入的信息转换成 OWL-S 描述。基本信息包括服务名字约束,服务提供者名字的约束,服务所在地区的约束。功能信息主要从功能分类和输入、输出、前置条件和效果来描述对所请求服务的约束。QoS 属性约束要求可以用本文提出的 QoS 本体来描述,用户可以灵活地、动态地增加或者删减服务非功能属性约束条目。同时,用户通过设置各属性间的相对重要性来表达自己的 QoS 偏好。在这个过程中,用户提供的信息有可能不完整,可以在导航界面的帮助下尽可能精确地描述用户需求。

(4) 服务发现与匹配模块

Web 服务发现代理解析该 BPEL 程序的每个抽象服务请求描述,服务请求描述分为功能需求和 QoS 需求两部分。服务发现代理到广告服务的语义描述存储(或 UDDI 注册中心)中查询,首先进行功能的匹配,选出一个满足功能需求的候选服务集合。为了提高查全率和查准率,本文利用语义匹配来发现服务。服务发现代理首先通过 UDDI 的分类查找获取结果集合 T ,然后对集合 T 中的每一个服务进行语义及 IOPEs 匹配,得出排好序的结果集合 S ,再对 S 集合中的每一个广告服务与业务服务需求进行除 OWL-S 外的匹配,如 QoS 匹配。在 QoS 匹配时,综合考虑用户对单个服务的 QoS 需求和领域专家通过挖掘历史信息得到的经验值这两方面的主客观因素,对每个成分服务 QoS 属性设定一个阈值,从每个候选广告服务集合中淘汰不满足阈值的服务,缩小候选服务的数量,以减少后继在线优化算法的计算复杂性。

(5) 服务部署模块

由服务部署代理向 Web 服务支撑平台以及 BPEL 引擎部署抽象服务流程。

(6) 服务执行时优化与容错模块

QoS 属性的生命周期管理框架支持运行时服务组合的优化与容错处理。服务执行代理实现多重角色:实现从抽象服务到广告服务(即服务描述)的间接绑定,广告服务再与其服务描述所对应的服务进行交互;服务优化代理依据在线优化的原则,优化一步,执行一步。对优化的每个广告服务,在其对应候选广告服务列表中选择一个子集,该子集中的服务均满足替换对应广告服务所需的约束条件,该子集中的服务可以互为替换,故可称之为替换子集。容错策略采取用替换子集中的替换服务替代失效的服务。服务执行代理负责对服务的执行情况进行监控,在遇到服务失效或 QoS 值降低的情况下,执行代理会捕获到这个异常,然后使用轮叫调度算法,将用户请求转发到替换子集中的最佳服务来继续流程的执行,同时仍可保证局部流程的优化。在这个过程中用户并未感知到发生了错误,即执行代理提供了容错机制;服务执行代理收集调用服务的可用性、响应时间、吞吐量和可靠性等方面的值,并把这些值发送给反馈分析代理进行统计分析后,作为服务调用的历史记录存入 QoS 统计表中。

3 基于 QoS 的 Web 服务组合在线优化算法

在线算法是在确定流程的控制结构后对每个活动选择最优服务,避免了全局组合算法由于对组合服务的 QoS 估算不准而引起故障恢复的风险^[10],并且算法的时间开销小,有利于实时计算。本文的在线优化局部策略是以活动为粒度选择成分服务,分别搜索各个活动的候选服务集,从中选择最佳服务作为执行该活动的成分服务。目前,局部优化策略通常采用直接加权法^[9]方法对候选服务实行打分机制,本文采用自动修正判断矩阵的层次分析法是对直接加权法的改进,既详细地反应了用户的偏好又体现服务质量属性值的客观性。

3.1 服务质量属性与 QoS 模型

文献[4]描述了通用的 Web 服务 QoS 属性,在此基础上,本文定义四维通用服务质量属性:

(1) 执行价格:服务 s 的执行价格 $q_{pr}(s, op)$ 是用户调用方法 op 需支付的费用。

(2) 执行时间:服务 s 的执行时间 $q_{du}(s, op)$ 等于请求发出和该请求结果收到之间的时间延迟。执行时间使用表达式计算: $q_{du}(s, op) = T_{process}(s, op) + T_{trans}(s, op)$ 。传送时间根据过去服务方法的实际执行的基础上估算,更新函数如下:

$$[T_{trans}(s, op)]_{n+1} = \frac{(n-1)[T_{trans}(s, op)]_n + T_n(s, op)}{n}$$

这里 $T_n(s, op)$ 是第 n 次服务传送时间, $[T_{trans}(s, op)]_{n+1}$ 是第 n 次后服务传送的期望时间, n 是过去执行次数的数量。

(3) 执行的成功率: $q_{rat}(s) = N_c(s)/K$, 其中 $N_c(s)$ 是服务 s 在最大期望时间内成功完成的次数, K 是调用的总次数。

(4) 可靠率: $q_{aw}(s)$ 是服务 s 一定时间范围内可用的概率,其表达式为: $q_{aw}(s) = \text{服务可用时间} / \text{考察的总时间}$ 。

根据以上定义,服务 s 的方法 op 的服务质量向量定义为:

$$q(s, op) = (q_{pr}(s, op), q_{du}(s, op), q_{rat}(s), q_{aw}(s))$$

此 QoS 模型可以根据用户的需求任意扩展。

3.2 在线优化的局部计算模型

3.2.1 单个广告服务的各属性的标准化

在实际应用中,用户对服务的选择有多个 QoS 属性的要求,提供商提供的服务也具有多个 QoS 属性,这些属性具有不同的量纲(单位),在进行服务选择时难以对各属性的相对重要性作比较。为便于对具有不同量纲的多目标问题作决策分析,需要将各 QoS 属性都无量纲化和归一化,这就是所谓的规范化。归一化是指把各属性的值都映射到 $[0, 1]$ 区间。

设在流程中任一任务 t_k 对应一个候选 Web 服务集 $S_k = \{S_k^1, S_k^2, \dots, S_k^m\}$ 能执行这个任务。将所有候选服务的 QoS 质量向量合并,就可建立一个矩阵 $Q(S_k) = (Q_{ij}(S_k)); 1 \leq i \leq m, 1 \leq j \leq 4$, 矩阵的每行 $Q_i(S_k)$ 对应于一个可选服务 S_k^i 的所有属性值,列 $Q_j(S_k)$ 对应于一个质量维各候选服务的属性值。

本文采用非比例变换法来做规范化,此法则将属性之差按一定比例进行归一化和无量纲化。用 $V_{ij}(S_k)$ 表示 $Q_{ij}(S_k)$ 经过规范化后的值,在这里只考虑两种情况:成本型(愈小愈好型)、效益型(愈大愈好型):

对于成本型目标,按下式进行变换:

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

对于效益型目标,则

$$V_{ij}(S_k) = \begin{cases} \frac{Q_{ij}(S_k) - Q_j^{\min}(S_k)}{Q_j^{\max}(S_k) - Q_j^{\min}(S_k)} & \text{if } Q_j^{\max}(S_k) - Q_j^{\min}(S_k) \neq 0 \\ 1 & \text{if } Q_j^{\max}(S_k) - Q_j^{\min}(S_k) = 0 \end{cases} \quad (2)$$

$Q_j^{\max}(S_k)$ 为 m 个可选服务中第 j 个质量维的最大值, $Q_j^{\min}(S_k)$ 为 m 个可选服务中第 j 个质量维的最小值。用上两式进行归一化的变换是非比例的, $V_{ij}(S_k)$ 只有相对值意义。标准化后得到的矩阵为:

$$V(S_k) = (V_{ij}(S_k); 1 \leq i \leq m, 1 \leq j \leq 4) =$$

$$\begin{bmatrix} V_{11}(S_k) & \cdots & V_{14}(S_k) \\ V_{21}(S_k) & \cdots & V_{24}(S_k) \\ \vdots & & \vdots \\ V_{m1}(S_k) & \cdots & V_{m4}(S_k) \end{bmatrix}$$

3.2.2 候选广告服务集合中服务的排序

候选广告服务集合经过在服务发现与匹配阶段的缩小处理后,每个任务的候选服务集合中的服务需要排序,以便进行局部优化处理。要计算每个候选服务 QoS 的评分,需确定每个 QoS 属性的权值。文献[4]中属性的权值由系统直接给定的,不能客观地反映客户需求。本文采用层次分析法^[6],通过与用户的交互界面,获取用户 QoS 属性之间的相对重要性偏好,并采用自动修正判断矩阵的方法对多个 QoS 属性进行加权处理,计算 QoS 属性的权重,将多目标的问题转化为单目标多约束问题。算法如下:

步骤 1 构造判断矩阵。

在服务组合运行时的交互界面上,由用户提供各 QoS 属性的相对重要性,由此得到一个判断矩阵:

	B_1	B_2	B_3	B_4
B_1	b_{11}	b_{12}	b_{13}	b_{14}
B_2	b_{21}	b_{22}	b_{23}	b_{24}
B_3	b_{31}	b_{32}	b_{33}	b_{34}
B_4	b_{41}	b_{42}	b_{43}	b_{44}

在这里 B_1 代表价格, B_2 代表执行时间, B_3 代表成功率, B_4 代表可用性。 b_{ij} 表示属性 B_i 对属性 B_j 的相对重要性,通常 b_{ij} 取 1, 2, ..., 9 及它们的倒数,其含义为: 1 表示属性 B_i 与属性 B_j 相比,两者重要性相同; 3 表示属性 B_i 比属性 B_j 稍重要; 5 表示属性 B_i 比属性 B_j 重要; 7 表示属性 B_i 比属性 B_j 强烈重要; 9 表示属性 B_i 比属性 B_j 极端重要。 2, 4, 6, 8 表示上述相邻判断的中间值。各数的倒数有下列意义: 若属性 i 与属性 j 的重要性之比为 b_{ij} , 那么属性 j 与属性 i 重要性之比为 $b_{ji} = \frac{1}{b_{ij}}$, 判断矩阵具有下面的性质:

(1) 判断矩阵是互反矩阵,因此仅需给出 $n(n-1)/2$ 个元素数值。

(2) 判断矩阵必须满足一致性。

违反一致性的判断是违反常识的,混乱而经不起推敲的判

断会导致决策的失误。而用户需求给出的 QoS 属性相对重要性的部分数据不合理会导致违反一致性的错误发生。因此根据判断矩阵计算 QoS 属性的权重时,要对判断矩阵进行一致性检验。当找到错误时,采用自动修正判断矩阵的方式,修改 QoS 属性的相对重要性。

步骤 2 采用近似方根法求 λ_{\max} 和 W 。

由判断矩阵 B 满足等式 $BW = \lambda_{\max} W$, λ_{\max} 是最大特征根, W 是对应的特征向量,即各属性的权重,则有

$$W_i = \frac{\sqrt[n]{\prod_{j=1}^n b_{ij}}}{\sum_{i=1}^n \sqrt[n]{\prod_{j=1}^n b_{ij}}} \quad \lambda_{\max} = \sum_{i=1}^n \frac{(BW)_i}{nW_i}$$

式中 $(BW)_i$ 表示向量 BW 的第 i 个分量。

步骤 3 判断矩阵的一致性检验。

随机一致性比率 CR 是判断一致性的决定性指标, $CR = \frac{\lambda_{\max}}{RI(n) * (n-1)}$ 。其中平均随机一致性指标 RI 是已知常量,其 1-9 阶矩阵的数据^[6]由表 1 可得:

表 1 n 阶矩阵的随机指标 RI

阶数 n	1	2	3	4	5	6	7	8	9
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45

当 $CR < 0.10$ 时认为判断矩阵具有满意的一致性,否则判断矩阵不满足一致性,这时可以通过下面算法系统对其进行自动调整,直到它满足一致性为止。该算法如下所述:

(1) 输入已构造出来的判断矩阵 B 的阶数 n 及其值 b_{ij} 。

(2) 计算判断矩阵 B 的一致性指标 CR , 如果 CR 小于 0.1, 则结束调整, 转(7); 否则, 转(3)。

(3) 对判断矩阵 B 作归一化处理, 设归一化后的矩阵为 $B' = (b'_{ij})_{n \times n}$, 其中 $b'_{ij} = \frac{b_{ij}}{\sum_{i=1}^n b_{ij}}$ ($j=1, 2, \dots, n$)。

(4) 任取 B' 中的一列向量 (不妨取第一列) 的各分量, 去除矩阵 B' 的每一列向量中的对应分量, 得矩阵 $B'' = (b''_{ij})_{n \times n}$, 其中 $b''_{ij} = \frac{b'_{ij}}{b'_{i1}}$, $i, j=1, 2, \dots, n$ 。

(5) 取最大的 b''_{ij} ($i=1, 2, \dots, n; j=2, 3, \dots, n; i \neq j$), 当 b_{ij} 为整数时, 新的 $b_{ij} = b_{ij} - 1$, 其对应的 $b_{ji} = 1/(b_{ij} - 1)$, 当 b_{ij} 为整数的倒数时, 新的 $b_{ij} = 1/(1/b_{ij} + 1)$, 其对应 $b_{ji} = 1/b_{ji} + 1$ 。

(6) 重新计算调整后的判断矩阵的一致性指标 CR , 如果 $CR < 0.1$, 转(7); 否则, 转(3)。

(7) 调整结束, 得到的判断矩阵满足一致性要求。

步骤 4 方案排序。

得到随机一致性比率 CR 的权重 W 后, 就可计算出每个候选服务集合的评分向量 E_k :

$$E_k = (e_k^1, e_k^2, \dots, e_k^m)^T = V(S_k) \times W = \left(\sum_{j=1}^4 V_{1j}(S_k) W_j, \sum_{j=1}^4 V_{2j}(S_k) W_j, \dots, \sum_{j=1}^4 V_{mj}(S_k) W_j \right)^T$$

对向量 E_k 进行排序就可得到评分值排序。将排序第一位的广告服务作为优化服务, 其他的满足用户需求的广告服务则作为优化服务的替换子集, 在优化服务发生故障时, 使用轮叫

调度算法调用次优服务来响应用户的请求。

4 仿真与模拟

本章对提出的算法进行了仿真,为简单起见,只考虑了顺序组合模型的流程计划,对于并发、选择和循环模型都可以约为顺序模型^[11]。流程中任务的数量在 5 到 20 之间随机产生,每个任务的候选集合在 10 到 50 之间随机产生。现随机产生服务质量矩阵 $Q(8,15,4)$,即流程中一共有 8 个任务,每个任务有 15 个可选服务,四维服务质量;用户给出各 QoS 属性之间的相对重要性的判断矩阵:

$$A=[\begin{matrix} \frac{1}{9} & 3 & \frac{1}{5} & 9 & 1 & 5 & 2; & \frac{1}{3} & \frac{1}{5} & 1 & \frac{1}{2}; & 5 & \frac{1}{2} & 2 & 1 \end{matrix}]$$

由于此时随机一致性指标 $CR=0.1520$,显然 $CR \geq 0.10$,判断矩阵不满足一致性,经过自动修正算法进行一次自动调整后,得

$$A=\begin{pmatrix} 1.0000 & 0.1111 & 3.0000 & 0.2000 \\ 9.0000 & 1.0000 & 5.0000 & 2.0000 \\ 0.3333 & 0.2000 & 1.0000 & 0.3333 \\ 5.0000 & 0.5000 & 3.0000 & 1.0000 \end{pmatrix}$$

此时 $CR=0.1314$,判断矩阵还不具有一致性,再用自动修正算法来调整,第二次调整后:

$$A=\begin{pmatrix} 1.0000 & 0.1111 & 3.0000 & 0.2000 \\ 9.0000 & 1.0000 & 6.0000 & 2.0000 \\ 0.3333 & 0.1667 & 1.0000 & 0.3333 \\ 5.0000 & 0.5000 & 3.0000 & 1.0000 \end{pmatrix}$$

此时 $CR=0.1162$,判断矩阵仍不具有一致性,需要通过自动修正算法再次调整,第三次调整后:

$$A=\begin{pmatrix} 1.0000 & 0.1111 & 3.0000 & 0.2000 \\ 9.0000 & 1.0000 & 6.0000 & 2.0000 \\ 0.3333 & 0.1667 & 1.0000 & 0.2500 \\ 5.0000 & 0.5000 & 4.0000 & 1.0000 \end{pmatrix}$$

算得 $CR=0.0975$,这时 $CR < 0.10$,判断矩阵已具有满意的一致性,这时计算得到四维服务质量的相应权值为 0.0868,0.5507,0.0586,0.3038,然后将每个候选服务的 4 个 QoS 属性乘以对应权值再求和,得到每个候选服务的 QoS 评分,限于篇幅,这里仅给出第 1 个任务和第 2 个任务候选服务的 QoS 评分:

表 2 第 1 和第 2 个任务对应的候选服务评分表

任务编号	候选服务评分							
1	0.6606	0.4436	0.5159	0.3880	0.8656	0.9020	0.6822	0.1247
	0.2288	0.7869	0.3035	0.3920	0.8461	0.4165	0.3922	
2	0.4445	0.3668	0.7517	0.2920	0.7985	0.6322	0.6890	0.7676
	0.4587	0.3000	0.2187	0.6386	0.6720	0.5308	0.3364	

根据评分的高低得到每个任务对应的候选服务的排序列表如表 3。

从表 3 中可以看出,对任务 1,最优的候选服务是第 6 号服务,选择其后的 5 个服务作为替换子集来支持容错,即在第 6 号服务失效时,执行代理会转而调用替换子集中最优的第 5

表 3 每个任务对应的候选服务排序表

任务编号	候选服务排序
1	6,5,13,10,7,1,3,2,14,15,12,4,11,9,8
2	5,8,3,7,13,12,6,14,9,1,2,15,10,4,11
3	12,2,7,13,11,8,9,14,10,3,5,4,1,6,15
4	13,14,15,3,1,10,12,6,8,9,4,7,5,11,2
5	8,2,13,11,1,9,10,14,3,15,4,6,12,5,7
6	10,9,14,1,3,12,7,13,5,15,4,11,2,6,8
7	6,2,14,3,13,10,5,15,7,4,11,12,1,9,8
8	13,12,10,15,2,9,7,8,3,11,6,4,5,1,14

号服务来响应用户的请求,依次类推。对任务 2-8,处理流程和任务 1 相同。

5 结论

本文实现了 Web 服务 QoS 属性的生命周期管理框架,提出了自动修正判断矩阵的层次分析法来实现基于 QoS 的在线局部优化策略,既详细地反应了用户 QoS 属性之间的相对重要性需求又体现服务质量属性值的客观性,在线优化效率较高,可用于服务组合运行时的容错处理。

(收稿日期:2006 年 11 月)

参考文献:

[1] Serhani M A,Dssouli R,Hafid A,et al.A QoS broker based architecture for efficient web services selection[C]//ICWS,2005;1(5): 113-120.

[2] Lee Eunjo,Jung Woosung.A framework to support QoS-aware usage of web services[C]//ICWE,2005,3:318-327.

[3] Zeng Liang-zhao,Benattallah B,Ngu A H H,et al.QoS-aware middleware for web services composition[J]IEEE Transactions on Software Engineering,2004,30(5):311-327.

[4] Jin Hai,Chen Han-hua,Lu Zhi-peng,et al.Q-SAC:toward QoS optimized service automatic composition[C]//5th IEEE/ACM International Symposium on Cluster,Computer and the Grid(CCGRID), 2005,2:623-630.

[5] Agarwal V,Dasgupta K.A service creation environment based on end to end composition of web Services.WWW,2005:128-137.

[6] 岳超源.决策理论与方法[M].北京:科学出版社,2003:204-209.

[7] Introduction to web services metadata[EB/OL].http://dev2dev.bea.com/pub/a/Anil_WServices.html.

[8] 刘燕.语义 Web 服务发布与发现的关键技术研究[D].湖南师范大学,2006.

[9] Bruno M,Canfora G.Using test cases as contract to ensure service compliance across releases[C]//ICSOC,2005:87-100.

[10] Canfora G,Di Penta M,Esposito R,et al.QoS-aware replanning of composite web services[C]//ICWS,2005,1(5):121-129.

[11] Cardoso J,Sheth A,Miller J,et al.Quality of service for workflows and web service processes[J].Web Semantics Journal:Science,Services and Agents on the World Wide Web Journal,2004,1(3): 281-308.