

基于功能划分图的 Web 服务组合规划和最优选择

吴 芳 朱尚明*

(华东理工大学计算机科学与工程系 上海 200237)

摘 要 扩展 Web 服务类型的组合方式、实现服务的无缝组合和提高服务组合的可靠性是当今 Web 服务组合的研究热点。针对 Web 服务类型组合方式多样性和无缝服务组合问题,根据请求服务的功能划分图来计算可用于服务组合的候选服务类型,动态规划各种服务类型的组合方式,并提出第一级服务类型的装配算法。针对服务组合的可靠性问题,将 Web 服务自身对运行环境的要求和自身的优先条件表示为上下文,并提出相应的局部最优选择算法和全局最优选择算法,以找到真实的、具有高可靠性的服务组合。最后,通过仿真实验验证了第一级服务类型装配算法、局部最优和全局最优选择算法的性能。

关键词 Web 服务组合 功能划分图 第一级服务装配算法 局部最优选择算法 全局最优选择算法

中图分类号 TP393 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2016.09.003

WEB SERVICE COMPOSITION PLANNING AND OPTIMAL SELECTION BASED ON FUNCTION PARTITION MAP

Wu Fang Zhu Shangming*

(Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

Abstract To expand the composition mode of Web service types, implement seamless Web service composition and improve the reliability of service composition are the focuses of current Web service composition research. Considering the diversity of Web service composition modes and the seamless service composition issue, we calculate the candidate service types applicable to the service composition based on the function partition map of service, dynamically plan the composition mode of various service types, and propose an assembly algorithm of the first stage service type. Aiming at the reliability issue of service composition, we express the operating environment required by Web service itself and the preferential conditions of its own as the context, and propose the correlated local optimum and global optimum selection algorithms to find the real service composition with high reliability. Finally, we verify the performances of the assembly algorithm of first stage service type, the local optimum and the global optimum selection algorithms through simulation experiments.

Keywords Web service composition Function partition map Assembly algorithm of first stage service type Local optimum selection algorithm Global optimum selection algorithm

0 引 言

随着互联网以及云计算技术的发展,越来越多的 Web 服务出现在互联网以及云计算平台上。Web 服务是可以跨越整个 Web 被发布、定位以及调用的自包含、自描述的模块化应用。如何有效地组合分布于 Internet 中的各类 Web 服务,实现服务之间的无缝集成,形成功能丰富的企业级服务流程,已经成为 Web 服务发展过程中的一个重要步骤。Web 服务组合的研究正是在这种背景下被提出来,并吸引了学术界和工业界的广泛关注。Web 服务组合作为一种应用模式,可以在保证 Web 服务的质量和服务独立性以及满足用户约束的前提下,发现一组服务并将其组合成一个新的、功能更加复杂的服务^[1]。

Web 服务组合是针对 Web 服务进行资源管理与整合的重要技术和手段。然而,随着 Web 服务的迅速增长,通过逐一匹配请求服务的单个抽象功能来进行服务组合的方式对 Web 服务的利用率变得越来越低。互联网环境千差万别,不同的 Web

服务正常运行的环境也不一样,在服务选择阶段,仅仅通过运行成功率这一统计性标量已不足以选择到真实环境下高可靠性的服务。实现 Web 服务的无缝组合和提高服务的可靠性、扩展性已成为当今 Web 服务组合研究领域的热点问题^[2,3]。

本文在 Web 服务组合方式规划上将针对组合方式的多样性和无缝性问题,提出一种基于请求服务功能划分图的动态 Web 服务组合方法。在组合服务的选择上将针对服务的可靠性问题,提出带约束上下文的局部最优选择算法和全局最优选择算法。目前云平台上的 Web 服务既有独立的集成软件,也有适用于工业流程的组件,本文提出的算法对这两类应用场景都适用。

1 相关工作

目前对 Web 服务组合技术的研究可以分为两类:基于工作

收稿日期:2015-05-22。计算机与法律复合应用型人才培养基础案例库建设项目(A-3101-14-17-172103)。吴芳,硕士生,主研领域:计算机网络与应用。朱尚明,教授。

流的服务组合和基于人工智能规划的服务组合^[4]。例如,文献[5-9]介绍了基于 workflows 的动态服务组合和选择算法,其思想是根据流程图逐步选择最优的服务,将最优服务组合和选择问题转换为求最优路径问题。文献[1]介绍了基于人工智能的服务组合优化方法,通过统计特征来把握候选服务的全局特性,使用遗传算法求出最优解。随着互联网以及云计算技术的发展,越来越多的 Web 服务出现在互联网上,只逐个匹配请求服务的各个抽象功能来确定候选服务,并从中选择和组合出 Web 服务已经不能满足当前的需要了。为扩展 Web 服务组合的方式,很多学者都把目标放在扩大候选服务类型集上,文献[2]提出了服务内扩展和服务外扩展方法,文献[10]采用聚类技术和结构化服务发现相结合的方法为 Web 服务归类,文献[11]采用计算流程相似度来查找所需候选服务。服务类型集的扩大必然引起组合方式的多样性,在最优服务组合选取上,文献[3]采用嵌套组合的方法,文献[12]则提出了基于相似度的模糊预测的方法,文献[13]和文献[14]也各提出一种模型来力求最大限度找到能组合出满足用户需求的候选服务并将它们装配起来。但是,这些方法都不能完整表现出服务间的关系。因此,对于包含多个请求服务的抽象功能的候选服务,其服务组合方式也就不确定,从而产生了无缝组合问题。Web 服务的无缝组合就是如何完整和正确地将候选服务装配成满足请求服务要求的组合。

Web 服务千差万别,其对运行环境的要求也各有不同,Web 服务组合面临多样性的问题。文献[2]提出了服务约束-感知机制来确保 Web 服务在正确的环境中运行,但是文献[2]在初始化及预包装上没有提出相应的算法,而且深度优先的图算法的复杂度很高。文献[15]提出了用上下文来表述服务特殊要求的方法,其思想是后退一步来查验当前选择的服务是否是最优的,是则继续到下一步,否则在退后一步的前提下选择最优。然而这个算法只适用于功能呈顺序关系,不能满足请求服务的功能间呈非顺序关系的情况。

本文提出的 Web 服务组合和选择方法分为两个阶段:(1)基于请求服务的功能划分图来计算出所有可用于服务组合的候选服务类型,同时通过精确匹配来确定出各个类型的候选服务集,并规划出各种服务类型组合方式;(2)查找各个组合方式下的最优服务组合,比较产生最优的服务组合。考虑到各个 Web 服务会受环境的影响,本文采用上下文结构来表述每个具体 Web 服务或服务组合自身对运行环境的要求和自身具有的优先条件,并提出基于这种上下文的局部最优选择算法和全局最优选择算法来找出高可执行性的最优服务组合。

2 Web 服务组合方式规划

目前,在 Web 服务的识别上有基于输入和输出参数的^[4],还有基于功能语义描述的^[2],本文采用解析请求服务的输入和输出参数来确定单个服务的类别。对于用户的请求服务,如何抽象出其包含的各个功能和它们之间的关系,以及每个功能对应的参数,目前已有许多研究。本文将在确定出请求服务的各个抽象功能、各个抽象功能的输入和输出参数以及请求服务的功能划分图的基础上,进行 Web 服务组合的研究。

服务不仅具有功能特性,还有输入、输出参数的差别^[9]。本文采用分级思想,对于在功能及相应功能的输入、输出参数上都完全或部分符合请求的服务采用第一级服务组合;对于在功

能上满足请求服务的某个功能,但是在输入、输出参数上却不满足相应要求的服务采用第二级服务组合。本文重点研究第一级服务组合。

2.1 基本概念

为便于理解 Web 服务的组合方式,首先介绍本文定义的几个基本概念。

定义 1 请求服务的功能关系:指请求服务的各个功能间的行为关系和数据关系。

一般,功能间的行为关系分为顺序关系和并行关系两种,且呈顺序关系的两个功能才可能有数据关系。本文通过匹配输入参数和输出参数来抽象各个功能,而各个参数又反映了功能间的数据关系。因此,也可以通过匹配输入和输出参数来确定出整个功能间的关系。

定义 2 请求服务的功能划分图:指反映请求服务的各个抽象功能及各个抽象功能间关系的图,图有两个虚拟节点表示开始与结束,其他节点表示请求服务的抽象功能,抽象功能节点间的边表示抽象功能间的功能关系。

定义 3 第一级服务:指一个 Web 服务能满足一个或多个 Web 请求服务的抽象功能,这些抽象功能间的关系必须与请求服务的功能关系一致。同时其输入参数和输出参数也满足请求服务在这些功能上的参数约束。

定义 4 第二级服务:指一个 Web 服务能满足某一个请求服务的抽象功能,但其输入参数和输出参数无法满足请求服务在这个功能上的参数约束。第二级服务只涉及一个请求服务的抽象功能,所以没有功能关系的限制。

匹配技术、分类技术和装配技术是形成 Web 服务组合的三项关键技术^[16]。匹配 Web 服务的过程也是搜索候选 Web 服务的过程,为了匹配出上述所提到的第一级服务,可以在匹配 Web 服务的过程中,将 Web 服务分为完全匹配服务、部分匹配服务、约束不匹配服务和完全不匹配服务。

定义 5 完全匹配服务:指一个服务的输入参数类型满足请求服务的所有输入参数类型,且服务的输出参数类型也满足请求服务的所有输出参数类型。同时,这个服务的输入参数和输出参数在取值上满足请求服务的约束。

定义 6 部分匹配服务:指一个服务的输入参数和输出参数在类型上分别部分匹配请求服务的输入参数和输出参数,且这些匹配的输入参数与输出参数在取值上也与请求服务的相应参数一致。

定义 7 约束不匹配服务:指一个服务的输入参数类型和输出参数类型都完全或部分出现在请求服务的输入参数和输出参数中,但是它们不都满足其对应的参数约束。

定义 8 完全不匹配服务:指不满足上述三类中任意一类的服务。

完全匹配服务和部分匹配服务归为第一级服务,用于后续的服务组合和选择技术,至于完全不匹配服务则直接舍弃。

2.2 服务组合规划

规划服务组合分为 3 个阶段:

- 构造请求服务的功能划分图;
- 匹配、分类第一级服务,产生第一级服务类型;
- 装配第一级服务类型,产生抽象服务类型组合。

1) 构造功能划分图

将整个请求服务中的各个功能看作一个节点,先设定两个

虚拟节点,即开始节点和终结节点,之后将这些节点初始化为一个有向图,这里称其为请求服务的功能划分图。

定义深度为功能划分图的最大路径长度,宽度为功能划分图中所有功能节点的最大直接后继服务节点数量。功能划分图的构造方法如下:

(1) 功能节点依照功能间的行为关系进行连接,若两个功能之间的行为关系是并行或不相邻的先后关系,则它们不需要连接;若它们的行为关系为顺序且相邻,则依照它们的先后顺序用有向边连接。

(2) 开始节点用有向边指向所有无前驱的节点,所有无后继节点指向终结节点。

2) 匹配、分类第一级服务

这一阶段的目的是搜索符合要求的第一级服务并将它们分类,由此产生各个第一级服务类型。为便于理解,对第一级服务类型作如下定义:

定义 9 第一级服务类型:第一级服务经第一级组合的分类阶段产生的抽象服务类型。

根据初始化后的功能划分图,标记图中的各分支和各抽象服务相对于各个分支所在的位置,并确定各分支间的隶属关系。之后,通过将各个服务与这个有向图对比来对服务进行分类。

已知第一级服务都能部分产生目标输出,且它们对应的节点之间有关联表示后者节点的输入依赖前者节点的输出,可得出以下推论:

推论 1 对于一个抽象服务功能,当其输入不能完全由一个第三方提供的服务决定时,则这个第三方提供的服务一定不包括这个抽象服务功能。

推论 2 一个无输入依赖的抽象服务功能,其前驱节点一定是开始节点。

推论 3 一个第三方提供的服务如果包含两个或多个在不同分支但它们的分支却隶属于同一分支 A 的抽象服务功能,记这两个或多个抽象服务功能所在的分支为 B 类分支,则这个第三方提供的服务必然包含分支 A 上的服务功能及其在 B 类分支上的直接后继抽象服务功能。

推论 4 一个第三方提供的服务所涉及到的抽象服务功能中,如果有其所在分支隶属于同一个分支的,则它们在请求服务功能划分图中的位置必是连续的。

将请求服务的功能划分图的开始节点留下,终结节点删去。根据上述 4 个推论可知,每个第一级服务类型对应到请求服务功能划分图中就是一个树,因此,每个第一级服务类型都可以由一个树结构来表示。

3) 第一级服务类型装配算法

第一级服务类型对应的装配算法简称第一级装配算法。其具体过程分为两阶段:分层阶段和装配阶段。

已知第一级服务类型的表示是一个树结构,本文根据各个第一级服务类型表示树的根节点来分层,即第一级服务类型所属的层数就是其表示树的根节点在请求服务功能划分图中的层数。

计算请求服务功能划分图中,求各个抽象服务节点的层数的方法如下:先求出请求服务功能划分图中从开始节点到终结节点的所有路径;接着,计算各个路径上的抽象服务在这个路径上的位置,设定开始节点的位置为 0,其直接后继节点的位置为 1,就这样依次加 1 得到各个路径上的各个抽象服务的位置;最后,对于每个抽象服务,取在经过自己的所有路径上的最大位置数作为自己的层数。

装配阶段就是依次对第 0 层的各个服务类型进行装配来得到它们的所有服务类型组合方式,进而得到请求服务所对应的所有服务类型组合方式,具体算法流程见图 1 所示。

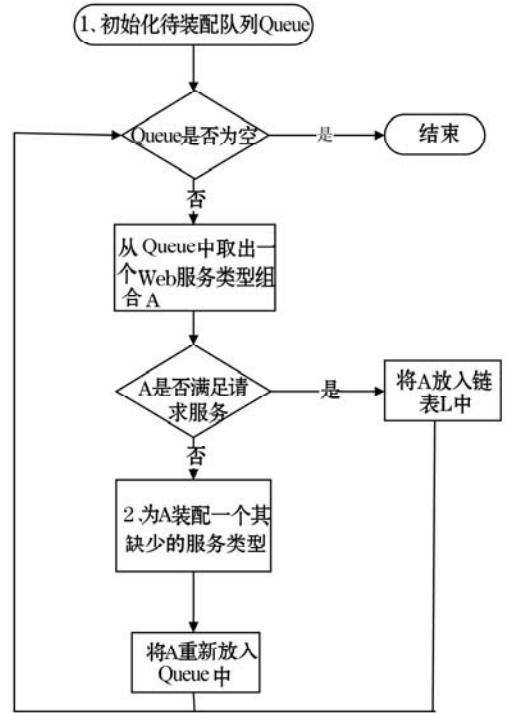


图 1 第一级装配算法流程图

(1) 初始化队列 Queue: 设 Queue 为一个先进先出队列,将属于 0 层的所有 Web 服务类型都封装为 Web 服务类型组合并依次放入 Queue 中。

(2) 装配一个缺少的服务类型:按层次数从小到大来找到服务类型组合 A 所需的第一个服务类型,并将这个服务类型装配到 A 上。

3 最优 Web 服务组合选择算法

Web 服务组合选择阶段则是根据请求服务的服务质量、客户偏好等非功能属性,选择出最优的、真实的服务组合。文献 [17] 提出了服务质量标准和相应的计算公式,探讨了对于确定的 Web 服务类型组合方式,求其最优服务组合解的方法。本文将 Web 服务组合的非功能属性分为完全依赖属性和部分依赖属性,用多领域决策来产生集成函数,通过集成函数的权值来表现请求服务的非功能性属性。最优 Web 服务组合选择算法就是选择出分数最高的服务组合的算法。

定义 10 完全依赖属性:由 Web 服务组合中的所有 Web 服务计算得到的属性,即服务组合在此属性上的值是所有 Web 服务在此属性上的值运算得到的。

定义 11 部分依赖属性:由 Web 服务组合中的部分 Web 服务计算得到的属性,即服务组合在此属性上的值是通过某种规则选出部分 Web 服务,由它们在这个属性上的值经过运算得到的。

从定义 10 和定义 11 可知,例如花费就是完全依赖属性,消耗的时间就为部分依赖属性。本文采用整数规划的思想,将所有非功能性属性的计算公式都转化为和的形式。这样,整个服务组合的分数就是整个服务组合在各个非功能属性项上的值乘

以相应的权值之后将这些项进行累加。当 Web 服务或 Web 服务组合在这些属性上的值是相互独立时,对于完全依赖属性,其在整个 Web 服务组合上的值为这个 Web 服务组合所包含的所有 Web 服务在这个属性上的值的和;对于部分依赖属性,求解其在整个 Web 服务组合上的值实质上是一个关键路径问题,只不过具体问题依属性而不同。

由定义 10、定义 11 知,完全依赖属性和部分依赖属性是没有逻辑联系的,而在部分依赖属性上选择出最优服务须得形成所有服务组合,这样问题就转换成图的最优路径选择问题。由文献[5]知,这是一个 NP-hard 问题,因此本文采取只根据完全依赖属性选择最优服务组合,在计算分数时会考虑整体的属性分数。

对于完全依赖属性,求解其在整个 Web 服务组合上的最优解问题实际上是个贪心问题,用贪心算法得到的最优完全依赖属性解在现实环境中并不保证一定能运行。另外,不是所有 Web 服务或服务组合在这些属性上的取值都是独立的,Web 服务或服务组合间可能有优先条件,所以本文采用上下文来表示 Web 服务或服务组合在这些属性上的关联和它们自身对运行环境的要求。在 3.1 节和 3.2 节将介绍结合上下文来求解最优完全依赖属性服务组合的局部最优算法和全局最优算法。

3.1 带约束上下文的局部最优选择算法

一般来说,一个 Web 服务的自身运行限制是与其前驱服务相关的,而其与其相邻服务的优先条件也可以表示为受其前驱服务的限制。因此,每个 Web 服务的约束上下文都可以表示为对其后继服务有限制。

带约束上下文的局部最优算法的实质就是逐层地找出满足上一层选出服务的约束条件的本层最优服务组合。考虑到会有没有后继服务的服务被选中而造成“无解”的情况,这里引入失败回退机制,具体算法过程如下:

1) 初始化:初始化这个服务类型组合方式中的各个服务类型的层次关系,设第 0 层的前驱层的最优子结构为 null 且没有对后继的约束。最后,设当前层为第 1 层。

2) 选择出满足前驱层的当前层最优服务组合,如果没有,则更新服务状态并回退至前驱层;如果存在则进入第 3) 步。

3) 判断当前层是否为最后一层,是则退出算法;否则设下一层为当前层,接着执行第 2) 步。

3.2 带约束上下文的全局最优选择算法

带约束上下文的全局最优算法结合了动态规划思想和上述的局部最优算法,因此也带有失败回退的机制,具体算法过程如下:

1) 初始化:与局部最优算法的初始化过程一样。

2) 计算当前层的局部最优解。

3) 将上步得出的结果与第 0 层到当前层的前驱层筛选出的全局最优服务作对比,若完全一样则进入第 4) 步,否则进入第 5) 步。

4) 筛选出只在当前层上,比第 2) 步选出的当前层服务组合更优秀的服务组合,并依次找出包含这些服务组合的从 0 层到当前层的最优子结构;之后通过对比这些解找出 0 层到当前层的最优子结构,接着进入第 6) 步。

5) 找到开始出现变化的层数,更新服务状态,设开始出现变化的层为当前层,进入第 2) 步。

6) 判断当前层是否为最后一层,是则退出算法;否则记下一层为当前层,重新进入第 2) 步。

4 实验及性能分析

4.1 实验环境

为验证上述算法的性能,本文进行了仿真实验。仿真实验采用的开发环境如下:操作系统为 Windows7 32 位操作系统,算法开发语言为 C++,采用 Microsoft Visual C++ 6.0 为开发工具。用链表结构存储请求服务功能划分图,用树结构表示第一级服务类别,并为每个服务类别添加一个唯一的标识符。建立 Composition 类来模拟符合请求服务的单个服务组合,在 Composition 类中使用一个 vector 容器类型的成员变量来存储组成这个服务组合的所有服务类型的标识。

对第一级装配算法,分别采用单线程和多线程两种计算方式,以功能划分图的深度和宽度作为标量,以功能数、服务类型数、组合方式数、装配时间等作为服务类型装配算法的性能指标,进行了仿真试验。对选择算法,以最优服务组合的分数和花费时间作为服务组合选择算法的性能指标,进行了仿真实验。

4.2 第一级服务类型装配算法性能分析

为验证第一级装配技术受请求服务功能划分图的影响,深度和宽度是反映请求服务功能划分图的复杂度的两个重要标量。表 1 给出了宽度为 2 时功能数、第一级服务类型数、组合方式数、装配时间等指标随深度的变化情况。从表 1 中可以看出,当深度从 1 增加到 3 时,第一级服务类型数量的变化比第一级服务类型组合方式数量的变化明显要大,采用多线程计算方式比单线程计算方式耗费的时间明显要少。

表 1 深度对性能指标的影响(宽度为 2)

| 深度 | 1 | 2 | 3 |
|----------------|---|----|------|
| 最大功能数 | 2 | 6 | 14 |
| 第一级服务类型数 | 3 | 28 | 699 |
| 第一级服务类型组合方式数 | 2 | 32 | 5657 |
| 单线程装配时间(单位:毫秒) | 0 | 10 | 1002 |
| 多线程装配时间(单位:毫秒) | 0 | 5 | 250 |

表 2 给出了深度为 2 时功能数、第一级服务类型数、组合方式数、装配时间等指标随宽度的变化情况。从表 2 中可以看出,当宽度从 1 增加到 3 时,第一级服务类型数量的变化比第一级服务类型组合方式的数量变化要小,采用多线程计算方式比单线程计算方式耗费的时间要少,但差值不大。

表 2 宽度对性能指标的的影响(深度为 2)

| 宽度 | 1 | 2 | 3 |
|----------------|---|----|------|
| 最大功能数 | 2 | 6 | 12 |
| 第一级服务类型数 | 3 | 28 | 737 |
| 第一级服务类型组合方式数 | 2 | 32 | 2560 |
| 单线程装配时间(单位:毫秒) | 0 | 10 | 532 |
| 多线程装配时间(单位:毫秒) | 0 | 5 | 273 |

对比表 1 和表 2 可以得出,服务类型数量更易受宽度的影响,而服务类型组合方式数量更易受深度的影响。另外,多线程计算方式更适合装配深度较大的请求服务。

图 2 是深度为 1,宽度从 1 到 10 变化时,分别用单线程和多线程计算方式所得的处理时间对比图。从图 2 中可以看出,宽度达到一定值时,用多线程产生的开销已经远大于装配的开销,使得多线程计算方式在处理效果上还不如单线程。

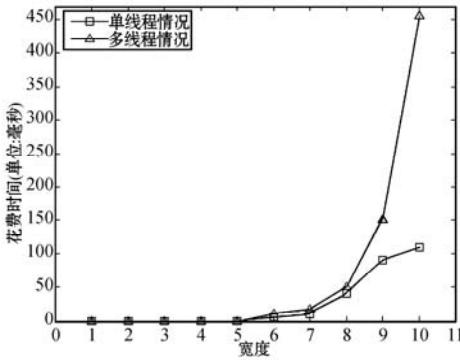


图2 宽度对计算时间影响图

4.3 局部最优和全局最优选择算法性能分析

这部分实验采用单线程、单机计算方式。为验证 Web 服务组合选择算法在约束上下文条件下的性能,实验用后继匹配率来表示满足服务约束的后继服务占有情况,即后继匹配率 = 满足服务的后继服务数量/后继服务的整体数量。设定整个服务组合的满分为 100,选择算法负责选择分数最高的服务组合。由表 1 知,当请求服务功能划分图的深度(路径长度)为 2(不包含初始节点)、宽度为 2 时,最大请求服务的功能数为 6,服务类型数为 28,服务类型组合方式数为 32。图 3 给出了这种情况下后继匹配率为 0.5,每种服务类型的候选服务数量取 10~100 间的 10 倍数离散点时,分别用局部最优选择算法和全局最优选择算法所得的服务组合的分数情况。图 4 给出了对应的局部最优选择算法和全局最优算法所花费的时间情况。从图 3 和图 4 可以看出,全局最优算法在效果上略优于局部最优算法,但是其花费的时间要比局部最优选择算法大得多,而且易受候选服务数量的影响。

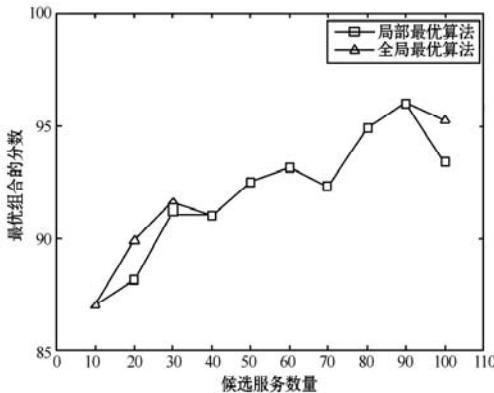


图3 局部最优和全局最优选择算法的分数对比

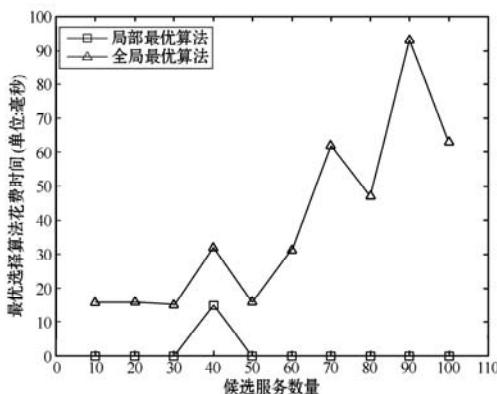


图4 局部最优和全局最优选择算法的花费时间对比

5 结 语

本文通过构造请求服务的功能划分图,来计算用于 Web 服务组合的服务类型并规划出相应的服务类型组合方式。结合后退一步上下文算法以及动态规划思想,将非功能属性分类求解,并通过计算局部最优来缩小计算范围的方法,来简化局部最优和全局最优选择问题的复杂度。最后通过仿真实验,分析了功能划分图对第一级装配算法的影响,以及局部最优选择算法和全局最优选择算法的性能。

参 考 文 献

- [1] 刘恒,张公让,吴曼. 基于分布估计算法的 Web 服务组合优化[J]. 计算机技术与发展,2014,24(6):10-14.
- [2] PengWei Wang, ZhiJun Ding, ChangJun Jiang, et al. Constraint-Aware Approach to Web Service Composition[J]. IEEE Transactions on Systems, Man, and Cybernetics; Systems, 2014, 44(6):770-784.
- [3] Incheon Park, Wuhui Chen, Michael N. Huhns. A Scalable Architecture for Automatic Service Composition[J]. IEEE Transactions on Services Computing, 2014, 7(1):82-95.
- [4] 赵明雪,赵文栋,彭来献,等. 基于业务抽象规划的分布式动态服务组合算法[J]. 计算机工程,2014,40(4):37-41,47.
- [5] Liangzhao Zeng, Boualam Benatallah, Anne H H Ngu, et al. QoS-Aware Middleware for Web Services Composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5):311-327.
- [6] Dongnei Liu, Zhiqing Shao, Caizhu Yu, et al. A Heuristic QoS-Aware Service Selection Approach to Web Service Composition[C]//Eighth IEEE/ACIS International Conference on Computer and Information Science; IEEE Computer Society, 2009:1184-1189.
- [7] Nina Turajlić, Ivana Dragović. A Hybrid Metaheuristic Based on Variable Neighborhood Search and Tabu Search for the Web Service Selection Problem[J]. Electronic Notes in Discrete Mathematics, 2012, 39(1):145-152.
- [8] Shuiguang Deng, Longtao Huang, Wei Tan, et al. Top-k Automatic Service Composition; A Parallel Framework for Large-Scale Service Sets [J]. IEEE Transactions on Automation Science and Engineering, 2014, 11(3):891-905.
- [9] Cristina Bianca Pop, Viorica Rozina Chifu, Ioan Salomie, et al. Ant-inspired Technique for Automatic Web Service Composition and Selection [C]//2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (2010); IEEE Computer Society, 2010:449-455.
- [10] 赵文栋,田畅,彭来献. CBSD:一种基于 Chord 的模糊服务发现方法[J]. 计算机科学,2014,41(1):172-177.
- [11] 贺兴亚,王海艳. 基于 Petri 网的组件服务发现方法[J]. 计算机技术与发展,2014,24(7):136-140.
- [12] Jinjun Cheng, Cong Liu, MengChu Zhou, et al. Automatic Composition of Semantic Web Services Based on Fuzzy Predicate Petri Nets[J]. Automation Science and Engineering, 2015, 12(2):680-689.
- [13] Yang Jie, Zhou Xianzhong, Wang Jiacun, et al. A Novel Method for Web Service Composition Based on Extended BDI[C]//2014 IEEE 11th International Conference on Networking, Sensing and Control, 2014:310-315.
- [14] Xuanzhe Liu, Gang Huang, Junfeng Zhao, et al. Data-Driven Composition for Service-Oriented Situational Web Applications [J]. IEEE Transactions on Services Computing, 2014, 8(1):2-16.

3) 分析基于需求偏好的不同赋权模式的服务选择。当 $\alpha = 0$ 时,即在计算 QoS 属性权重时,完全依赖于用户对 QoS 的需求偏好,称其为需求偏好的主观赋权模式;当 $\alpha = 1$ 时,即在计算 QoS 属性权重时,忽略用户的需求偏好,仅考虑 QoS 属性值客观数据,称其为需求偏好的客观赋权模式;当 $\alpha = 0.5$ 时,即在计算 QoS 属性权重时,既考虑用户的需求偏好又考虑 QoS 属性值,称为需求偏好的主客观赋权模式。候选服务的总优势度如表 3 所示。

表 3 候选服务的总优势度

| | 主观模式 | 客观模式 | 主客观模式 |
|------|---------|---------|---------|
| Ws1 | 1.7501 | 4.7111 | 3.2275 |
| Ws2 | 0.3655 | 2.4949 | 1.4289 |
| Ws3 | 1.6704 | 4.7111 | 3.1912 |
| Ws4 | 0.9817 | -1.2991 | -0.1599 |
| Ws5 | -5.1660 | -7.1100 | -6.1159 |
| Ws6 | -3.3107 | -3.7467 | -3.5246 |
| Ws7 | 2.3336 | 5.0652 | 3.6993 |
| Ws8 | -0.1014 | 1.3730 | 0.6451 |
| Ws9 | -4.6336 | -5.7277 | -5.1869 |
| Ws10 | 1.9450 | -0.5091 | 0.7154 |
| Ws11 | 2.8892 | 3.1235 | 3.0044 |
| Ws12 | 2.4358 | 2.8418 | 2.6374 |
| Ws13 | 0.0802 | 3.4294 | 1.7596 |
| Ws14 | 0.2458 | -5.5920 | -2.6783 |
| Ws15 | -1.9831 | -3.7665 | -2.8836 |

选取排序前四的 4 个候选服务,计算它们在各个 QoS 属性上的均值(在计算响应时间时,取区间左右两端值的平均值)。为了能在一个图中直观显示与比较基于不同赋权模式的服务选择的排序结果,将可用性、声誉的属性值放大 100 倍,费用和响应时间的属性值缩小 10 倍,如图 4 所示。

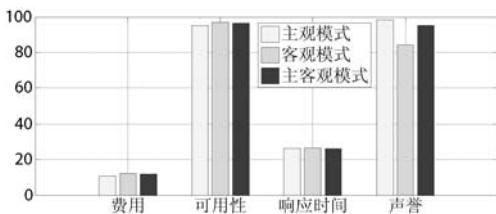


图 4 不同赋权模式的服务选择结果对比

由图 4 可知,主观赋权模式分别在费用、声誉达到最大,但可用性偏低,响应时间偏高,用户偏好使得服务 QoS 属性信息的实际效用不一致;客观赋权模式在可用性、响应时间达到最优,但声誉、费用不是最佳;而主客观赋权模式在各个 QoS 属性信息上表现了很好的中间性,既能表达用户偏好又能反映服务 QoS 的客观特征,显示出更强的灵活特性。

6 结 语

在开放的 Internet 环境中,描述服务 QoS 属性的方式具有多样性。本文使用精确数值型、区间数值型、三角模糊数值型对 QoS 属性值分别进行描述,利用序关系向量表示用户的主观偏

好,同时利用熵权表示 QoS 属性的客观权重。在此基础上,采用相对优势度算法给出混合型 QoS 属性的服务选择过程。

通过模拟仿真和实验验证,相对优势度算法能够有效地对功能相似的服务优劣作出选择,筛选出既能满足用户 QoS 需求偏好又具有较高质量的服务;并且该算法与现有算法相比具有较高的执行效率。本文采用需求偏好的主客观赋权模式在各个 QoS 属性值上能够表现出很好的中间性,进一步证明了主客观赋权模式的合理性。如何把相对优势度算法用于大批量的混合型 QoS 属性的服务选择是下一步需要研究的问题。

参 考 文 献

- [1] 王伟锋,刘亚军. 基于 QoS 的语义 Web 服务选择框架[J]. 计算机与数字工程,2009,37(9):15-19.
- [2] Tian M, Gramm A, Ritter H, et al. Efficient selection and monitoring of QoS-aware web services with the WS-QoS framework[C]//Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, Beijing, China,2004:152-158.
- [3] 王飞,邹仕洪,陈山枝,等. 基于模糊数学的 Web 服务 QoS 建模[J]. 计算机应用研究,2007,24(4):214-216.
- [4] 李蜀瑜,周娟. 基于 WSMO 服务质量的语义 Web 服务发现框架[J]. 计算机应用,2009,29(8):2299-2302.
- [5] 王胜利. 数据挖掘技术在 Web 服务分类中的应用研究[D]. 广东:暨南大学,2010.
- [6] 马玉倩,付晓东,代志华,等. 一种混合 QoS 感知的 Web 服务选择方法[J]. 计算机应用与软件,2012,29(6):39-42.
- [7] Klir G J, Yuan B. Fuzzy Sets and Fuzzy Logic: Theory and Applications[M]. New Jersey: Prentice Hall PTR, 1995.
- [8] 杨放春,苏森,李祯. 混合 QoS 模型感知的语义 Web 服务组合策略[J]. 中国科学 E 辑:信息科学,2008,38(10):1697-1716.
- [9] 马友,王尚广,孙其博,等. 一种综合考虑主客观权重的 Web 服务 QoS 度量算法[J]. 软件学报,2014,25(11):2473-2485.
- [10] 辛乐,范玉顺,李想,等. 基于服务信誉评价的偏好分析与推荐模型[J]. 计算机集成制造系统,2014,20(12):3170-3181.
- [11] 程冬,董才林,喻莹. 一种基于模糊理论的 Web 服务信任评估模型[J]. 计算机应用与软件,2012,29(10):82-84,179.
- [12] 郭秀英. 区间数指标权重确定的熵值法改进[J]. 统计与决策,2012(17):32-34.
- [13] 李小林,张力娜. 基于混合 QoS 模型的服务选择策略[J]. 科学技术与工程,2012,12(22):5653-5657,5660.
- [14] Qian Y H, Liang J Y, Song P, et al. On dominance relations in disjunctive set-valued ordered information systems[J]. International Journal of Information Technology and Decision Making, 2010,9(1):9-33.

(上接第 14 页)

- [15] Hong Qing Yu, Stephan Reiff-Marganiec. A Backwards Composition Context Based Service Selection Approach for Service Composition [C]//2009 IEEE International Conference on Services Computing (SCC): IEEE Computer Society, 2009:419-426.
- [16] Rajesh Karunamurthy, FerhatKhendek, Roch H Glitho. A Novel Architecture for Web Service Composition[J]. Journal of Network and Computer Applications, 2012,35(2):787-802.
- [17] Rajeswari M, Sambasivam G, Balaji N, et al. Appraisal and Analysis on Various Web Service Composition Approaches Based on QoS Factors [J]. Journal of King Saud University-Computer and Information Sciences, 2014,26(1):143-152.