

基于 FAHP 与规划图融合的 Web 服务组合方法

范国栋 祝 铭 李 静 崔晓柳

山东理工大学计算机科学与技术学院 山东 淄博 255000

(fgdsdut@163.com)



摘 要 近年来,随着云计算的发展,越来越多的服务被发布在网上。如何将不同的 Web 服务组合在一起并使其满足功能性需求和非功能性需求成为了一个研究难点。Web 服务质量 (Quality of Service, QoS) 感知的 Web 服务组合问题属于 NP 难问题。为了解决这个问题,文中提出一种融合 FAHP 与改进 Graphplan 算法的方法 (FAHP and Improved Graphplan, FIGP)。首先,根据用户偏好使用模糊分析层生成服务的综合 QoS;其次,在 Graphplan 向前扩展中,使用动态阈值对竞争力较差的服务进行剪枝,在保留关键服务的同时降低了时间复杂度;最后,在 Graphplan 向后搜索阶段,在满足功能性需求的前提下选择综合 QoS 最好的服务加入到组合中。实例分析和实验结果表明,与普通的 Graphplan, Skyline 及其他方法相比, FIGP 不仅较好地提高了服务组合的质量,而且显著缩短了程序的执行时间。

关键词: Web 服务组合, QoS, FAHP, Graphplan, 自动组合

中图法分类号 TP393

Web Service Composition by Combining FAHP and Graphplan

FAN Guo-dong, ZHU Ming, LI Jing and CUI Xiao-liu

College of Computer Science and Technology, Shandong University of Technology, Zibo, Shandong 255000, China

Abstract In recent years, with the advance of cloud computing, more and more services have been published online. How to search an optimal composition with both functional and non-functional requirements has become a challenging problem. QoS-aware web service composition is an NP-hard problem. To solve this problem, a system combining FAHP with improved Graphplan algorithm was proposed. Firstly, the overall QoS of service is generated by using FAHP according to user preferences. Secondly, in the forward expand stage of Graphplan, dynamic threshold is used to prune less competitive services, which reduces time complexity while ensuring that critical services are retained. Finally, in the backward searching stage of Graphplan, service with best overall QoS is selected into the composition, under the premise of meeting the functional requirements. Experimental results show that the proposed algorithm not only improves the quality of service composition, but also reduces the program running time significantly compared with the ordinary Graphplan, Skyline and other methods.

Keywords Web service composition, Quality of service, Fuzzy analytical hierarchy process, Graphplan, Automatic composition

1 引言

近年来,随着云计算的发展,网络服务越来越丰富,单个 Web 服务已不能满足用户提出的复杂需求。自动化服务组合选择多个已有 Web 服务并将其组合在一起,可解决单个 Web 服务无法解决的复杂任务。如何在 Web 服务组合过程中选择合适的服务并满足非功能性需求,是目前的一个研究难点。为了提高用户的满意度,许多 Web 服务优化方法应运而生,如 Skyline 算法^[1]、规划图 (Graphplan) 算法^[2]、模糊控制^[3]及其他算法^[4-6]。但是,上述方法也存在一些局限性。例如, Skyline 无法找到能够更好地满足用户偏好的 Top-k Web 服务;模糊控制将用户偏好建模为 IF-THEN 规则,但是随着

QoS 属性的增多,用户必须定义更多的规则。为了突破这些局限性,作为文献[1,3,7]研究的延续,本文提出了一种将模糊层次分析 (Fuzzy Analytic Hierarchy Process, FAHP) 与改进的规划图 (Improved Graphplan) 相融合的 QoS 感知的服务自动组合方法 (简称 FIGP)。该方法能够匹配出符合用户功能性需求的服务组合,也能准确反映候选服务综合 QoS 的水平,同时为 Web 服务的自动组合中基于偏好权重的模糊表达问题提供了新思路。

2 背景知识

2.1 QoS 感知的 Web 服务组合

定义 1 (服务质量, QoS) 网络在传输业务流时,业务流

到稿日期:2018-11-30 返修日期:2019-04-01 本文已加入开放科学计划 (OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金项目 (61473179);淄博市校城融合发展计划项目 (2018ZBXC295);山东理工大学科技项目 (4041-417010)

This work was supported by the National Natural Science Foundation of China (61473179), Zibo City and University Integration Development Projects (2018ZBXC295), Science and Technology Projects of Shandong University of Technology (4041-417010).

通信作者:祝铭 (zhu_ming@sdut.edu.cn)

对网络服务的需求的集合。其中,业务流是指与特定 QoS 相关的、从源到目的的分组流^[8]。也就是说,QoS 是应用业务对 Web 服务提出的一组可度量的非功能性要求,包括服务响应时间、吞吐量及费用等。

定义 2 Web 服务 w 可以用三元组 $(w_{in}, w_{out}, w_{QoS})$ 来表示。其中, w_{in} 是 w 的输入参数有限集; w_{out} 是 w 的输出参数有限集; w_{QoS} 是 w 的服务质量有限集。

定义 3 Web 服务组合问题可以用四元组 (S, C_{in}, C_{out}, Q) 来表示。其中, S 是服务有限集; C_{in} 是服务的输入参数有限集; C_{out} 是服务的输出参数有限集; Q 是服务质量有限集。

Web 服务可以按照顺序方式连接,也可以按照流方式连接。顺序连接的服务是逐个被唤醒的,即 $(w_1; w_2; \dots; w_n)$;按流连接的服务是被并行唤醒的,即 $(w_1 \parallel w_2 \parallel \dots \parallel w_n)$ 。常用的 Web 服务质量标准如下。

1)响应时间(R):从最开始收到查询消息到发送响应消息的时间间隔(单位:ms)。

$$R(w_1; w_2; \dots; w_n) = \sum R(w_i) \quad (1)$$

$$R(w_1 \parallel w_2 \parallel w_3 \parallel \dots \parallel w_n) = \max R(w_i) \quad (2)$$

2)吞吐量(T):经由通信通道成功传递信息的平均速率。

$$T(w_1; w_2; \dots; w_n) = \min T(w_i) \quad (3)$$

$$T(w_1 \parallel w_2 \parallel w_3 \parallel \dots \parallel w_n) = \min T(w_i) \quad (4)$$

3)花费(C):用户为使用 Web 服务资源所付出的代价(单位:元)。

$$C(w_1; w_2; \dots; w_n) = \sum C(w_i) \quad (5)$$

$$C(w_1 \parallel w_2 \parallel w_3 \parallel \dots \parallel w_n) = \sum C(w_i) \quad (6)$$

2.2 规划图算法

规划图是解决 Web 服务组合问题的方法,其包含两个阶段:向前扩展和向后搜索。向前扩展阶段从初始状态开始构建规划图,通过循环 Web 服务存储库,将符合已知输入的 Web 服务选出来加入规划图,规划图中的 Web 服务又可以作为其他 Web 服务的输入,当规划图不再添加新的服务时结束循环。向后搜索阶段是从最后一层到第一层的反向搜索循环,根据要找的目标,向前搜索路径,检索解决方案。QoS 感知的 Web 服务组合是 NP 难问题,本文通过两阶段的规划图算法降低时间复杂度,得到了满足用户需求的服务组合。

定义 4 规划图是一种有向分层图,其中顶点层形成文字和操作符交替的序列: $(P_0; A_1; P_1; A_2; P_2; A_3; P_3; \dots; A_k; P_k)$ 。

在规划图中, P 表示一组状态; A 表示一组动作。初始层中 P_0 表示规划问题的初始状态, A_i 的每个节点都有 P_{i-1} 层的输入弧与 P_i 层的输出弧。假设计划有多种方案能够实现相同的功能,但服务组合拥有不同的 QoS 值,那么规划图方法可以提供更多的选择。

2.3 归一化

QoS 属性的描述方式、测量单位、取值范围等可能存在差异,导致无法对各个 QoS 属性进行横向对比,因此需要将 QoS 属性值进行归一化。归一化在不改变 QoS 原始分布的情况下,将 QoS 属性转换为 $[0,1]$ 区间上的相对值。如果已知匹配程序产生的分数界限,即 Web 服务在 QoS 标准下的最大值 $Q_i^{\max}$ 与最小值 $Q_i^{\min}$,则可以进行 min-max 归一化处

理。对于积极的 QoS 属性,例如吞吐量、可靠性,使用式(7)计算 QoS 的归一化值。对于消极的 QoS 属性,例如响应时间、花费,则用式(8)计算 QoS 的归一化值。

$$U_i(w_j) = \begin{cases} \frac{Q_i(w_j) - Q_i^{\min}}{Q_i^{\max} - Q_i^{\min}}, & \text{if } Q_i^{\max} - Q_i^{\min} \neq 0 \\ 1, & \text{if } Q_i^{\max} - Q_i^{\min} = 0 \end{cases} \quad (7)$$

$$U_i(w_j) = \begin{cases} \frac{Q_i^{\max} - Q_i(w_j)}{Q_i^{\max} - Q_i^{\min}}, & \text{if } Q_i^{\max} - Q_i^{\min} \neq 0 \\ 1, & \text{if } Q_i^{\max} - Q_i^{\min} = 0 \end{cases} \quad (8)$$

2.4 模糊理论

用户选择的 QoS 属性往往具有不确定性和模糊性,需要用模糊推理量化。本节介绍了一些模糊理论的概念,包括模糊集合、模糊数以及 FAHP 算法。

2.4.1 模糊集合和模糊数

模糊集合理论最早由 Zadeh^[9] 提出,它将数学的研究范围扩大到模糊现象的领域。模糊集合由定义的隶属函数表示,该隶属函数将值映射到 $[0,1]$ 区间。

定义 5 设 $\tilde{A}$ 是论域 X 到封闭区间 $[0,1]$ 上的一个映射,即 $\tilde{A}: X \rightarrow [0,1], x \rightarrow \tilde{A}(x) \in [0,1]$,则 $\tilde{A}$ 是 X 上的一个模糊集合, $\tilde{A}(x)$ 是模糊集合 $\tilde{A}$ 的隶属函数, $\tilde{A}(x)$ 的值称为 x 对模糊集合的隶属度。

模糊数是一个归一化模糊集,隶属函数至少是局部连续的,函数值 $\mu_A(x) = 1$ 。三角模糊数可以用一个三元组 (a, b, c) 定义,其隶属函数的定义如下。

定义 6 三角模糊数由数集 $\tilde{A} = (a, b, c)$ 表示,其中 $a < b < c$,且 a, b, c 均属于实数集。其隶属函数为:

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & a < x \leq b \\ \frac{x-c}{b-c}, & b < x \leq c \\ 0, & \text{其他} \end{cases}$$

2.4.2 模糊层次分析法(FAHP)

许多实际问题很难甚至不可能建立精确的数学模型来进行定量分析。针对这种情况,可以应用模糊层次分析法^[10-12] 将问题折射为权重来表示。

根据 FAHP 算法的要求,对两两属性进行比较得出决策矩阵,重要程度使用表 1 中的模糊量化语言得出。在这个过程中,还要考虑不确定性水平标注参数 β ,当描述最大可能的不确定性时, $\beta=0$;当描述完全确定的情况时, $\beta=1$ 。参数值 β 是以 $[0,1]$ 为区间的任意数,本文所用 FAHP 算法的 β 值都取 0.5。

表 1 偏好程度的语言描述

Table 1 Language description of preference degree

重要性程度	含义	模糊数
1	同等重要	自身相比 $(1,1,1)$ 其他 $(1\beta, 1, (2-\beta)1)$
3	稍微重要	$(3\beta, 3, (2-\beta)3)$
5	明显重要	$(5\beta, 5, (2-\beta)5)$
7	重要的多	$(7\beta, 7, (2-\beta)7)$
9	极端重要	$(9\beta, 9, (2-\beta)9)$
$x=2,4,6,8$	中间值	$(x\beta, x, (2-\beta)x)$

注:如果 $x\beta < 1$,则 $x\beta = 1$;如果 $(2-\beta)x > 9$,则 $(2-\beta)x = 9$

根据表 1 中提供的模糊量化语言,对两两属性进行比较得到相应的三角模糊数和决策矩阵。

$$\tilde{A} = \begin{bmatrix} (1,1,1) & (l_{12},m_{12},u_{12}) & \cdots & (l_{1n},m_{1n},u_{1n}) \\ (l_{21},m_{21},u_{21}) & (1,1,1) & \cdots & (l_{2n},m_{2n},u_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ (l_{n1},m_{n1},u_{n1}) & (l_{n2},m_{n2},u_{n2}) & \cdots & (1,1,1) \end{bmatrix}$$

$\tilde{a}_{ij}$ 表示两个元素相对强度的三角模糊数, $\tilde{A}=(\tilde{a}_{ij})_{nn}$, 决策矩阵是对称的。

$$\tilde{a}_{ij}=[\tilde{a}_{ij}]^{-1}=(l_{ij},m_{ij},u_{ij})^{-1}=(\frac{1}{u_{ij}},\frac{1}{m_{ij}},\frac{1}{l_{ij}})$$

利用模糊算术运算定义第 i 个对象的模糊综合程度值为:

$$\tilde{S}_i=\sum_{j=1}^n\tilde{a}_{ij}\otimes[\sum_{i=1}^n\sum_{j=1}^n\tilde{a}_{ij}]^{-1}$$

$\tilde{S}_1\geq\tilde{S}_2$ 的可能性公式为:

$$V(\tilde{S}_1\geq\tilde{S}_2)=\begin{cases} 1, & m_1\geq m_2 \\ 0, & l_2\geq u_1 \\ \frac{l_2-u_1}{(m_1-u_1)-(m_2-l_2)}, & \text{其他} \end{cases} \quad (9)$$

当有 k 个模糊数时, $\tilde{S}_i(i=1,2,3,\cdots,k), \tilde{S}\geq\tilde{S}_k$ 的可能性公式为:

$$\begin{aligned} V(\tilde{S}\geq\tilde{S}_1,\tilde{S}_2,\cdots,\tilde{S}_k) &= V[\tilde{S}\geq\tilde{S}_1 \text{ and } (\tilde{S}\geq\tilde{S}_2) \text{ and } \cdots \\ &\quad \text{and } (\tilde{S}\geq\tilde{S}_k)] \\ &= \min V(\tilde{S}\geq\tilde{S}_i), i=1,2,3,\cdots,k \end{aligned} \quad (10)$$

假设 $d'=\min V(\tilde{S}_i\geq\tilde{S}_k)$, 权向量被定义为 $\mathbf{W}'=(d_1',d_2',\cdots,d_n')^T$ 。通过归一化,得出权向量为:

$$\mathbf{W}=(d_1,d_2,\cdots,d_n)^T$$

3 架构和算法

系统的架构如图 1 所示,分为如下 3 个主要模块:服务信息模块、改进的规划图模块和 FAHP 模块。

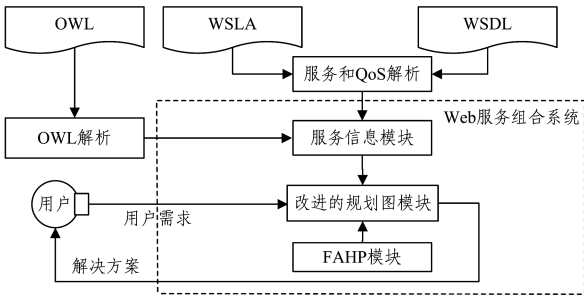


图 1 系统架构

Fig. 1 System architecture

服务信息模块:该模块用于解析 WSDL 中的 Web 服务的基本信息、WSLA 中的 QoS 信息以及 OWL 中的语义信息。关联这些信息得到 Web 服务的所有信息,将其存储后用于响应用户需求的服务组合中。

改进的规划图模块:该模块是基于 Graphplan 算法的改进,是 QoS 感知的 Web 服务组合的关键模块。首先,将 Web 服务的排名融入其中,在响应用户功能需求的同时满足其满意度;其次,在 Graphplan 向前扩展的过程中进行服务剪枝,

构造规划图,并根据功能需求和综合 QoS 进行反向搜索,获取解决方案。

FAHP 模块:FAHP 模块考虑用户偏好,生成服务的综合 QoS。该模块包括 3 个步骤:1)归一化 QoS,将服务的每个 QoS 归一化在 $[0,1]$ 区间;2)计算 QoS 属性的权重,根据用户的喜好,采用 FAHP 算法计算得到权重;3)计算综合 QoS,即各个归一化的 QoS 乘以其权重得到综合 QoS。

本文从两个方面对 Graphplan 算法进行改进:1)在向前扩展中,通过综合 QoS 对服务进行剪枝,为了保证能够找到解决方案,引入动态阈值法;2)根据综合 QoS,在向后搜索阶段的每层中,选择最佳候选服务进行组合。

算法 1 DynamicThreshold 函数

输入:服务的集合 SR,请求参数 request,目标结果 goal,阈值 threshold,步长 step

输出:规划图 pg

1. $pg \leftarrow \text{forwardExpand}(SR, request, goal, threshold)$
2. while $(goal \not\subseteq pg) \ \&\& \ (threshold > 0)$ do
3. $step = step * 2$
4. $Threshold = step$
5. end while
6. return pg

规划图的向前扩展阶段:通过引入阈值,一方面可以剪枝竞争力较差的 Web 服务;另一方面可以加快向前扩展的速度。为了保证找到解决方案,采用算法 1 所示的动态阈值法,其结合了规划图的向前扩展,当向前扩展找不到解决方案时,会减小阈值,再次进行向前扩展,直到在向前扩展阶段找到结果为止。考虑到查询速度的问题,算法 1 使用加速减小阈值的方法,即用当前阈值减去上一次步长的两倍。

规划图的向后搜索阶段:在规划图向后搜索阶段的每一层,始终选择满足最多功能性需求并且综合 QoS 值最高的候选服务,具体如算法 2 第 6-12 行所示。上述改进可以保证每次的选择都是当前状态下最优的。

算法 2 ServiceSelect 函数

输入:当前层服务的集合 layerSrv,当前层目标结果 goal

输出:当前层选择的服务 selectedSrv

1. while $layerSrv = \varnothing \ \&\& \ goal \neq \varnothing$ do
2. $maxSrv \leftarrow \varnothing$
3. $maxSize, maxScore \leftarrow 0$
4. for each service srv in layerSrv do
5. $size = count(srv.out \cap goal)$
6. if $(size > maxSize) \ ||$
7. $(size == maxSize \ \&\& \ srv.getScore > maxScore)$ then
8. $maxSrv \leftarrow srv$
9. $maxSize \leftarrow size$
10. $maxScore \leftarrow srv.getScore$
11. end if
12. end for
13. if $maxSize == 0$ then
14. break
15. end if
16. $selectedSrv \leftarrow selectedSrv \cup maxSrv$
17. $layerSrv \leftarrow layerSrv \setminus maxSrv$

```
18. goal←goal\maxSrvout
19. end while
20. return selectedSrv
```

4 实例分析

本节通过实例分析来演示所提算法。全部服务的信息如表 2 所列,其中包括服务名称、输入、输出、响应时间(*Resp*)、吞吐量(*Tpt*)和成本(*Cost*)。假设用户给定输入 *A,B,C* 及目标 *F*,求一组 Web 服务组合。

表 2 一组可用的 Web 服务

Table 2 Set of available services

w_i	输入	输出	<i>Resp</i>	<i>Tpt</i>	<i>Cost</i>
w_1	A,B	D	450	16 000	50
w_2	A,C	E	480	2 500	20
w_3	C,D	E	80	12 000	65
w_4	E	F	150	13 000	55
w_5	E	F	200	19 000	80

4.1 计算每个服务的综合 QoS

第一步 构建决策矩阵。按照 *Resp,Tpt,Cost* 的顺序,通过两两比较 Web 服务 QoS 属性的重要程度,得到决策矩阵。假设用户认为 *Resp* 稍微重要于 *Tpt*,*Tpt* 稍微重要于 *Cost*,*Resp* 较 *Cost* 介于稍微重要与明显重要之间。由于决策矩阵是对称的,因此用户只需要比较 3 次即可。

$$\tilde{A} = \begin{pmatrix} (1,1,1) & (1,5,3,4,5) & (2,4,6) \\ (1,5,3,4,5)^{-1} & (1,1,1) & (1,5,3,4,5) \\ (2,4,6)^{-1} & (1,5,3,4,5)^{-1} & (1,1,1) \end{pmatrix}$$

第二步 计算每个标准的模糊数。

以 *Resp* 为例,模糊数为:

$$S_{Resp} = \frac{\tilde{1} + \tilde{3} + \tilde{4}}{\tilde{1} + \tilde{3} + \tilde{4} + \tilde{3}^{-1} + \tilde{1} + \tilde{3} + \tilde{4}^{-1} + \tilde{3}^{-1} + \tilde{1}}$$

分别求得 *Resp,Tpt,Cost* 的模糊数为:

$$\begin{aligned} S_{Resp} &= (0.2269, 0.5749, 1.3355) \\ S_{Tpt} &= (0.1373, 0.3114, 0.7161) \\ S_{Cost} &= (0.0700, 0.1138, 0.2516) \end{aligned}$$

第三步 计算每个 QoS 的权重。根据式(9),分别得到 *Resp,Tpt,Cost* 两两之间的重要程度:

$$\begin{aligned} V(S_{Resp} \geq S_{Tpt}) &= 1 \\ V(S_{Resp} \geq S_{Cost}) &= 1 \\ V(S_{Tpt} \geq S_{Resp}) &= 0.6500 \\ V(S_{Tpt} \geq S_{Cost}) &= 1 \\ V(S_{Cost} \geq S_{Resp}) &= 0.0509 \\ V(S_{Cost} \geq S_{Tpt}) &= 0.3666 \end{aligned}$$

权重由式(10)计算,归一化后得到 *Resp,Tpt* 和 *Cost* 的权重 **W** = (0.5879, 0.3822, 0.0299)。

第四步 计算服务归一化的 QoS。假设响应时间的范围是[0,500];吞吐量的范围是[0,20 000];花费的范围是[0,100]。根据式(7)、式(8),以服务 w_1 的 *Cost* 属性为例,归一化的结果为 $\frac{100-50}{100-0}=0.5$ 。归一化后的 Web 服务的 QoS 如表 3 所列。

表 3 归一化后的 Web 服务的 QoS

Table 3 Normalized QoS of services

w_i	<i>Resp</i>	<i>Tpt</i>	<i>Cost</i>
w_1	0.10	0.80	0.50
w_2	0.04	0.16	0.80
w_3	0.84	0.60	0.35
w_4	0.70	0.65	0.45
w_5	0.60	0.95	0.20

第五步 计算每个服务的综合 QoS。如,服务 w_1 的综合 $QoS(w_1)=0.1 \times 0.5879+0.80 \times 0.3822+0.5 \times 0.0299=0.3795$ 。得到每个 Web 服务的综合 QoS 值后,根据综合 QoS 值的大小对服务进行排序,结果如表 4 所列。

表 4 按综合 QoS 排名的服务

Table 4 Services ranked by overall QoS

编号	w_i	综合 QoS
1	w_3	0.7336
2	w_5	0.7218
3	w_4	0.6734
4	w_1	0.3795
5	w_2	0.1086

4.2 使用改进后的 Graphplan 获得解决方案

在计算综合 QoS 之后,使用图 2 所示的改进 Graphplan 获得解决方案。

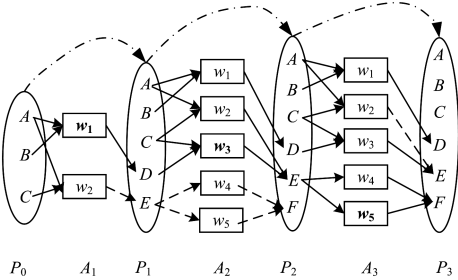


图 2 改进的规划图

Fig. 2 Improved Graphplan

在前向扩展阶段,改进的规划图算法会根据综合 QoS 对服务进行剪枝操作。例如,由于 w_2 的综合 QoS 比较低,对其进行剪枝,后续过程中 w_2 (图 2 中用虚线表示) 实际上不再可用。

在向后搜索阶段,从 A 层中选择最佳服务进行组合。最后一层 A_3 中的目标结果 F 可由服务 w_4 或 w_5 输出,比较 w_4 和 w_5 的综合 QoS 值,选择综合 QoS 较高的 w_5 ,其需要的输入为 E;在 A_2 层,E 是由 w_2 或 w_3 输出的,由于 w_2 已被剪枝掉,选取 w_3 ,又因 C 已知,其需要的输入为 D;最后,在 A_1 层,只有 w_1 输出 D,其输入已知为 A,B。此时,得到最佳的服务组合为($w_1;w_3;w_5$)。根据式(1)一式(6),组合($w_1;w_3;w_5$)的相应时间 *Resp*、吞吐量 *Tpt* 和成本 *Cost* 分别为:

$$\begin{aligned} Resp &= Resp(w_1) + Resp(w_3) + Resp(w_5) = 730 \\ Tpt &= \min(Tpt(w_1), Tpt(w_3), Tpt(w_5)) = 12 000 \\ Cost &= Cost(w_1) + Cost(w_3) + Cost(w_5) = 195 \end{aligned}$$

5 实验结果

5.1 实验环境

实验在以下配置的计算机上进行:1)CPU: Intel(R) Core (TM) i5-7200U 2.50 GHz 2.71 GHz; 2) RAM 为 8.00 GB DDR4-2400; 3) 硬盘为 LITEON T11 256 GB, TOSHIBA

MQ01ACF050 500GB;4)操作系统为 Windows 10 专业版 64 位。算法用 Java 实现。

5.2 数据集

使用 TestsetGenerator2009^[13]生成 5 个数据集。每个数据集都由一个 WSDL 文件来描述 Web 服务的基本信息,一个 OWL 文件列出服务之间的语义关系,一个 WSLA 文件记录服务的 QoS 属性。Dataset1 具有 1 000 个服务、3000 个概念和 398 行 BPEL 解决方案。Dataset2 具有 5 000 个服务、15000 个概念和 955 行 BPEL 解决方案。Dataset3 具有 10000 个服务、30000 个概念和 1231 行 BPEL 解决方案。Dataset4 拥有 15000 个服务、45000 个概念和 1741 行 BPEL 解决方案。Dataset5 拥有 20000 个服务、60000 个概念和 1876 行 BPEL 解决方案。

5.3 性能分析

为了说明算法的有效性,将 FIGP 的结果与普通 Graphp-

lan^[2], Skyline^[1] 以及没有剪枝的 FIGP 进行比较。其中, FIGP 表示属性之间的重要性相同;FIGP Cost 表示 Cost 比响应时间和吞吐量稍微重要,响应时间和吞吐量同样重要;响应时间和吞吐量的占优方法也是如此。

服务数量的实验结果如表 5 所列,用斜杠分隔的数值依次表示服务数量、响应时间、吞吐量、成本。使用 Dataset4 时, FIGP 比普通规划图算法多 1 个服务。使用 Dataset5 时, FIGP 比普通 Graphplan 少 7 个服务,比 Skyline 算法少 13 个服务。实验表明,FIGP 方法不会明显增加服务的数量,更多时候会减少服务的数量使结果更优。可以看出,与普通 Graphplan 和 Skyline 相比,FIGP 获得了具有更好的响应时间、吞吐量和成本的解决方案。FIGP 与没有剪枝的 FIGP 的结果基本相同。对于较大的数据集 Dataset5,FIGP 获得了更好的结果。因此,剪枝的 FIGP 算法可以获得质量较好的服务组合。

表 5 WSC09 数据集上的实验结果
Table 5 Results on WSC09 datasets

	Ordinary	Skyline	FIGP	FIGP without pruning	FIGP Cost	FIGP Resp	FIGP Tpt
Dataset1	11/2650/2000/492	9/1990/5000/395	11/2220/4000/500	11/2220/4000/500	11/2600/4000/386	11/1880/3000/509	11/2460/4000/516
Dataset2	20/3180/1000/892	20/2890/2000/1017	20/2880/2000/1051	20/2960/2000/1047	20/3900/1000/779	20/2760/2000/1013	20/3830/2000/921
Dataset3	30/6920/3000/1449	27/4080/2000/1473	27/3620/2000/1454	27/3620/2000/1454	30/4910/2000/1347	27/3490/2000/1463	28/4450/2000/1485
Dataset4	43/5710/1000/2094	44/5490/3000/2296	44/5370/3000/2246	44/5370/3000/2246	47/5920/1000/1829	51/5050/1000/2554	44/6030/3000/2249
Dataset5	52/7810/1000/2727	58/7730/1000/3239	45/4510/1000/2416	50/6920/2000/2716	45/4570/1000/1949	45/4050/1000/2418	50/6950/2000/2670

用户偏好的实验结果如表 5 的最后 3 列所示,解决方案随着用户偏好的不同而改变。比如:具有成本优先的返回了成本较低的解决方案,具有响应时间优先的返回了时间较少的解决方案,具有吞吐量优先的返回了更高吞吐量的解决方案。以上表明 FIGP 算法在用户偏好方面是有效的。

执行时间的结果如图 3 所示。在相同条件下,使用 FIGP 的搜索时间最短, Skyline 算法在向前扩展阶段需要很长的时间。如图 4 所示,对服务进行剪枝可以减少向后搜索阶段的执行时间。

选择最优服务,并且可以在前向扩展阶段对竞争力差的服务进行剪枝。普通 Graphplan 和 Skyline 方法只能从特定集合中随机选择服务。实验表明,相比其他算法,FIGP 的 QoS 感知的 Web 服务组合中的组合结果以及算法效率更优。然而, FIGP 的局限性在于局部最优组合的最终解决方案不一定是全局最优的。

6 相关工作

UDDI(通用描述、发现和集成)规范已用于发现 Web 服务。但是,它仅限于关键字匹配,并且返回的结果准确性不高。为了克服这些缺点,语义网被提出。随后,几种语义语言被提出,用以表示 Web 中的信息。资源描述格式(RDF)是一种语义标记语言,它根据命名属性和值来描述资源^[14]。Web Ontology Language(OWL)是一种语义 Web 语言,旨在表示术语和它们之间关系的含义。OWL 通过使用本体将 Web 服务的前提和效果表达为“概念”^[15],该领域的更多研究可以参见文献[16-17]。文献[18]设计了基于实际案例的移动 Web 应用程序。首先,其根据上下文信息,使用通用搜索引擎发现相关服务;然后,通过分支定界算法实现服务组合。

模糊逻辑是一种用于处理不确定性和支持决策的近似推理技术。模糊逻辑已被用作服务选择、发现和组合领域的支持工具。文献[19]提出了一种 FQ(Fuzzy-QoS)架构,它处理模糊用户偏好和服务描述的不确定性。Avile 等提出了一种适应方法,其采用模糊逻辑来优化服务选择^[20]。具体地说,他们提出了两种模糊支持系统,一个用于评估每个步骤中组合物的 QoS 值,另一个用于确定服务选择过程中 QoS 标准的权重。Zhao 等首先利用加权 Tchebycheff 距离对多目标优化问题进行建模,然后提出一个模糊偏好模型来表示多个 QoS 准则的优先顺序,最后给出两种服务组合算法^[21]。Ma 等提

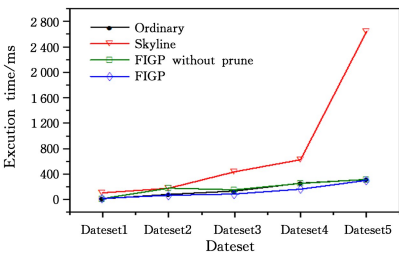


图 3 向前扩展阶段的执行时间

Fig. 3 Execute time in forward expand phase

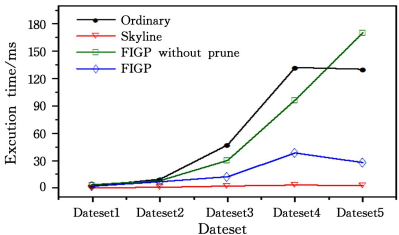


图 4 向后搜索阶段的执行时间

Fig. 4 Execute time in backward search phase

综上,FIGP 可以根据综合 QoS 和功能要求在每一层中

出一种对用户偏好的模糊性进行准确量化,进而计算综合权重的方法^[22]。Almulla 等提出一种基于客观加权技术的排序算法,该技术利用了 QoS 属性之间的距离相关度^[23]。文献^[24]提出一种针对 Web 服务选择中服务请求偏好权重表达的模糊性及 QoS 属性之间存在相互制约关系的组合赋权法。

结束语 本文提出了一种将 FAHP 与改进 Graphplan 相融合的算法,来解决 QoS 感知的 Web 服务组合的问题。首先,使用归一化与 FAHP 算法取得综合 QoS 并得到服务的排名;然后,在 Graphplan 中融合服务的综合 QoS,对服务进行筛选。实验证明,FIGP 在功能需求、非功能需求、用户偏好及执行时间方面都有较好的性能。下一步将拟解决微服务架构下的服务组合问题。

参 考 文 献

- [1] LI J, YAN Y H, LEMIRE D. Scaling up web service composition with the skyline operator[C]// IEEE International Conference on Web Services. USA: ICWS, 2016: 147-154.
- [2] YAN Y H, CHEN M. Anytime QoS-aware service composition over the graphplan[J]. Service Oriented Computing and Applications, 2015, 9(1): 1-19.
- [3] ZHU M, FAN G D, LI J, et al. An Approach for QoS-aware Service Composition with GraphPlan and Fuzzy Logic[C]// International Conference on Emerging Ubiquitous Systems and Pervasive Networks. Procedia Computer Science, 2018: 56-63.
- [4] LI J, YAN Y H, LEMIRE D. Full Solution Indexing for top-K Web Service Composition[J]. IEEE Transactions on Services Computing, 2016, 11(3): 521-533.
- [5] LI J, YAN Y H, LEMIRE D. A Web Service Composition Method Based on Compact K2-Trees[C]// IEEE International Conference on Services Computing. IEEE Computer Society, 2015.
- [6] HUANG L T, DENG S G, DAI K, et al. Automatic Service Composition in Parallel with MapReduce[J]. Chinese Journal of Electronics, 2012, 40(7): 1397-1403.
- [7] ZHU M, LI J, FAN G D, et al. Modeling and Verification of Response Time of QoS-aware Web Service Composition by Timed CSP[C]// International Conference on Emerging Ubiquitous Systems and Pervasive Networks. Procedia Computer Science, 2018: 48-55.
- [8] CUI Y, WU J P, XU K, et al. A Survey of Research on Internet Service Quality Routing Algorithms[J]. Journal of Software, 2002(11): 2065-2075.
- [9] ZADEH L A. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning—III[J]. Information Sciences, 1975, 9(1): 43-80.
- [10] DARKO B, DRAGAN P, BOBAN D. Modification of analytic hierarchy process (AHP) method and its application in the defense decision-making[J]. Scientific Research and Essays, 2012, 7(1): 24-37.
- [11] BOZANIC D, DRAGAN P, DRAGAN B. Modification of the analytic hierarchy process (AHP) method using fuzzy logic; Fuzzy AHP approach as a support to the decision making process concerning engagement of the group for additional hindering[J]. Serbian Journal of Management, 2015, 10(2): 151-171.
- [12] CHARILAS D, MARKAKI O, PSARRAS J, et al. Application of Fuzzy AHP and ELECTRE to Network Selection[M]// Mobile Lightweight Wireless Systems. Springer Berlin Heidelberg, 2009: 63-73.
- [13] WS-Challenge. Testsetgenerator2009[OL]. <https://code.google.com/p/wsc-pku-tcs/downloads/list>.
- [14] MANOLA F, MILLER E. Rdf primer [OL]. <http://www.w3.org/tr/rdfschema/>.
- [15] BECHHOFFER S, HARMELEN F, HENDLER J, et al. OWL Web Ontology Language Reference [OL]. <https://www.w3.org/TR/owl-ref/>.
- [16] DOU D, MCDERMOTT D, QI P. Ontology Translation on the Semantic Web[M]// Journal on Data Semantics II. Springer-Verlag, 2004: 35-57.
- [17] LEE J K, SOHN M. eXtensible Rule Markup Language[J]. Communications of the Acm, 2003, 46(5): 59-64.
- [18] ZHANG Y F, WANG J N, YAN Y H. Context-Aware Generic Service Discovery and Service Composition[C]// IEEE International Conference on Mobile Services. IEEE Computer Society, 2014.
- [19] SORA I, TODINCA D. Dealing with fuzzy QoS properties in service composition[C]// IEEE Jubilee International Symposium on Applied Computational Intelligence & Informatics. Institute of Electrical and Electronics Engineers (IEEE), 2015: 197-202.
- [20] AVILA S D, DJEMAME K. Fuzzy Logic Based QoS Optimization Mechanism for Service Composition[C]// IEEE International Symposium on Service Oriented System Engineering. Institute of Electrical and Electronics Engineers (IEEE), 2013: 182-191.
- [21] ZHAO X, SHEN L, PENG X, et al. Toward SLA-constrained service composition; An approach based on a fuzzy linguistic preference model and an evolutionary algorithm[J]. Information Sciences, 2015, 316: 370-396.
- [22] MA Y, WANG S G, SUN Q B, et al. Web Service Quality Metric Algorithm Employing Objective and Subjective Weight[J]. Journal of Software, 2014(11): 2473-2485.
- [23] ALMULLA M, ALMATORI K, YAHYA OUI H. A QoS-Based Fuzzy Model for Ranking Real World Web Services[C]// IEEE International Conference on Web Services. IEEE Computer Society, 2011: 203-210.
- [24] SUN X T, NIU J, GONG Q B, et al. services selection strategy based on combination weighting approach[J]. Application Research of Computer, 2017, 34(8): 2408-2411.



FAN Guo-dong, born in 1990, master student. He is currently working on automated Web service composition. His main research interests are Web service composition, micro-service architecture and machine learning.



ZHU Ming, born in 1983, Ph.D., is member of China Computer Federation (CCF). His main interests are related to process-oriented programming, Web service composition, event modeling, and concurrent systems.