

# 一种自治愈的QoS感知的Web服务组合方法

叶恒舟<sup>1,2</sup>, 李陶深<sup>3</sup>, 关云慧<sup>2</sup>

(1. 广西大学电气工程学院 南宁 530004; 2. 桂林理工大学信息科学与工程学院 广西 桂林 541000;

3. 广西大学计算机与电子信息学院 南宁 530004)

**【摘要】**基于工作流的服务质量(QoS)感知的Web服务组合是Web服务领域的一个研究热点,其目标是选择一个满足QoS约束且QoS效用最大的组合服务。该文给出了常见工作流的一种形式化定义方法及虚拟任务的合成规则,从而提出了一种基于top- $k$ 优选策略的Web服务组合方法,其核心思想是递归运用虚拟任务的合成规则将原工作流转换成一个虚拟任务,并在每次合成虚拟任务时,仅为其保留 $k$ 个优选的服务或组合服务实例。该文还提出了一种新的寻找替换服务的方法以支持自治愈性。仿真实验分析了 $k$ 值对效用的影响、算法的效用与时间开销以及可替代性能。

**关键词** QoS感知; 自治愈; 服务组合; top- $k$ 优选

**中图分类号** TP393 **文献标志码** A **doi**:10.3969/j.issn.1001-0548.2017.03.023

## A Self-Healing QoS-Aware Web Service Composition Method

YE Heng-zhou<sup>1,2</sup>, LI Tao-shen<sup>3</sup>, and GUAN Yun-hui<sup>2</sup>

(1. College of Electrical Engineering, Guangxi University Nanning 530004;

2. College of Information Science and Engineering, Guilin University of Technology Guilin Guangxi 541000;

3. School of Computer, Electronics and Information, Guangxi University Nanning 530004)

**Abstract** The quality of service (QoS)-aware web service composition problem is a research hotspot. Its objective is to select a composite service which can maximize the QoS utility while preserving QoS constraints. By giving the definition of usual workflow and the combination rule of virtual tasks, a top- $k$  based web service composition approach is proposed. Its core idea is to transform the original workflow into a virtual task by recursively calling the combination rule of virtual tasks. Moreover, a new scheme is presented to select suitable replaceable services to support the self-healing property. The effectiveness of the  $k$  value and the performances of algorithms are analyzed through simulation experiments.

**Key words** QoS-aware; self-healing; service composition; top- $k$  optimization

随着Web服务应用领域的持续延伸,单个简单的Web服务已经无法满足实际应用需求,人们常常需要将多个Web服务组合起来以完成一个较复杂的任务。用户在关注组合服务功能的同时,往往更加注重服务的QoS(响应时间、信誉度、可靠性、价格等)是否满足自身的需求。因此,如何为用户提供高效且满足QoS需求的Web服务组合是一个关键问题<sup>[1]</sup>。QoS感知的服务组合优化问题(optimizing the QoS-aware service composition, O-QSC)应运而生。

若不考虑Web语义<sup>[2-5]</sup>, O-QSC的问题可以建模为一个多维多选择背包问题(multidimensional multiple choice knapsack problem, MMKP)<sup>[6]</sup>,这是众所周知的NP难问题。所以,如何为用户提供一个可信赖且

可靠的服务组合是一个重大的挑战。目前,基于工作流的QSC问题已经获得了很多的研究,大体可分为三类:局部优化方法(local optimization, LO)<sup>[6-10]</sup>,全局优化方法(global optimization, GO)<sup>[11-24]</sup>以及质量约束分解方法(quality constraint decomposition, QCD)<sup>[25-28]</sup>。

当为工作流中的每个任务选择候选服务时,LO方法通常会对满足组合流程的一组服务的QoS指标进行加权排序,并以此作为选择各个服务实例的依据。由于每个服务都是被独立选择的<sup>[6-7]</sup>,因此该方法表现出较好的性能。文献[8]提出了一种中间件平台,该平台解决了如何选择Web服务的问题,同时描述了一种基于局部选择服务的方法。文献[6]给出

收稿日期: 2015-02-24; 修回日期: 2015-06-19

基金项目: 国家自然科学基金(51365010); 广西自然科学基金(2014GXNSFBA118269)

作者简介: 叶恒舟(1980-),男,副教授,主要从事服务组合、云计算及优化设计方面的研究。

了一种启发式算法, 它从局部视角寻找接近最优解的解决方案。虽然LO方法具有很好的时间复杂度, 但不容易满足全局约束, 且不容易获得最优解。

GO方法是以服务组合的整体质量满足全局QoS约束为目标的, 因此在对各个服务进行选择时, 需要综合考虑各服务组合后的总体质量, 从而生成一个最佳组合方案。当该最佳服务组合服从全局约束时, 能使聚合效用达到最大值。为了选择出最合适的候选组合, 文献[11]讨论了如何确定可考虑不同QoS类别的标准, 并开发出一些特定的启发式算法来解决服务组合问题。文献[12]提出了一种采用路径模板编码机制的遗传算法来解决多路径全局优化问题, 在算法的收敛性与复杂度方面有所提升。在文献[13]中, O-QSC问题被建模为具有一个多项式数变量和约束的混合整数型线性规划, 但它要求目标函数是线性的, 且时间复杂度较高。GO方法适应性强, 可以获得最优或次优解, 容易满足全局约束, 但时间复杂度较高。

为了综合LO方法和GO方法的优点, 文献[25-26]采用了QCD方法。QCD过程旨在把全局约束分解为与组合服务任务相关联的局部约束集, 但其主要思想是全局约束和用户喜好能引导着丢弃那些质量比预期的最小质量还要小的Web服务。文献[25]仅考虑了顺序模式的工作流以及和类型的QoS属性。

可见, 当前O-QSC方法存在以下一个或多个不足: 1) 算法效率不高或者效用不佳; 2) 仅针对顺序工作流或某个具体的工作流; 3) 未考虑组合服务运行异常时的自愈愈能力。为此, 本文提出了一种带有一定自愈愈能力的基于Top-k优选的服务组合方法, 主要贡献包括: 1) 给出了一种正则工作流的形式定义, 为自动处理工作流奠定了基础; 2) 给出了一种高效的QoS感知的Web服务组合方法, 可以以较小的时间复杂度获得较优的组合方案; 3) 可以为所选组合方案所涉及的每个服务寻找备用服务集, 具备一定的自愈愈能力。

# 1 工作流模型

## 1.1 工作流的基本模式

工作流描述了服务功能的结合方式以满足用户需求<sup>[29]</sup>。在工作流中, 一个任务代表功能相同的一系列Web服务, 一种模式代表不同任务之间的依赖关系<sup>[13]</sup>。与文献[13]一样, 本文将考虑顺序模式(SEQ)、并发模式(AND)以及选择模式(XOR)等3种常见模式(如图1), 其中 $A_i$ 代表某个任务, AND-split、

AND-join分别表示并发模式的起点与终点, XOR-split、XOR-join分别表示选择模式的起点与终点。

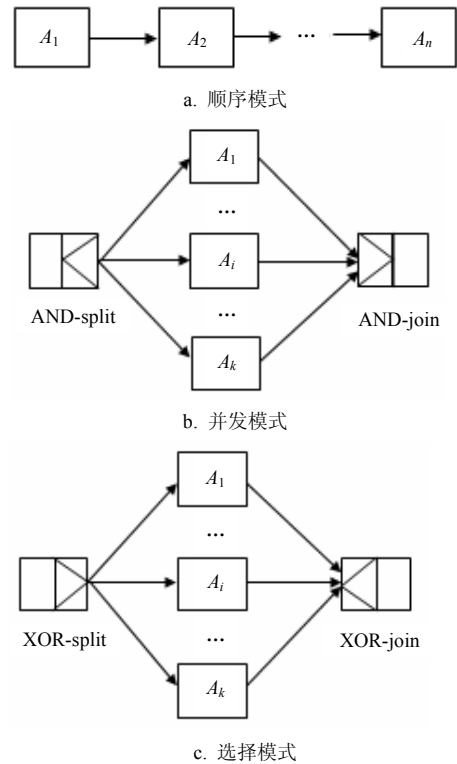


图1 工作流基本模式

## 1.2 复杂工作流

常见的复杂工作流由以上3种模式(顺序、并发、选择)经过有限次的连接嵌套而成。如果分别用 $\odot$ ,  $\oplus$ 和 $\otimes$ 来表示SEQ、AND和XOR模式, 那么常见的复杂工作流可递归定义为:

- 1) 当A和B是任务时,  $(A \odot B)$ 、 $(A \oplus B)$ 和 $(A \otimes B)$ 是工作流;
- 2) 当P和Q是工作流时,  $(P \odot Q)$ 、 $(P \oplus Q)$ 和 $(P \otimes Q)$ 也是工作流;
- 3) 当A是一个任务, P是一个工作流时,  $(A \odot P)$ 、 $(A \oplus P)$ 和 $(A \otimes P)$ 是工作流。

由此, 一个复杂工作流可以表示为一个字符串。以图2所描述的旅行规划工作流<sup>[13]</sup>为例, 它可以表示为 $(A_1 \odot ((A_2 \odot (A_3 \oplus A_4)) \otimes A_5 \otimes (A_6 \odot A_7)))$ 。

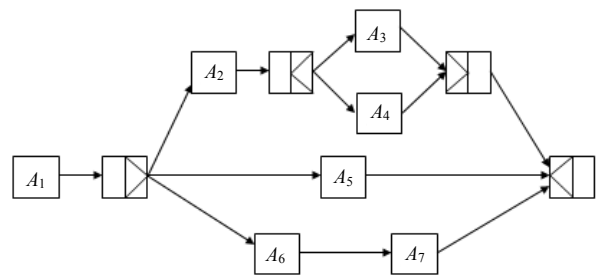


图2 复杂工作流的例子

### 1.3 虚拟任务

为了描述方便, 将一个任务看作一个二元组  $A \langle ID, SG \rangle$ ,  $ID$  是一个用来唯一确定  $A$  的标识符,  $SG$  是执行任务  $A$  的所有 Web 服务的集合。当用一个符号串来表示一个 workflow 时, 串中任何一对小括号可描述该 workflow 的一个子 workflow, 可将其看成一个虚拟任务, 并描述为一个三元组  $VA \langle ID, SG, UA \rangle$ ,  $ID$  是用来唯一确定  $VA$  的标识符,  $SG$  表示其所有实例的集合,  $UA$  表示其所涉及的任务的集合。

用  $Q(s) = (q_1(s), q_2(s), \dots, q_r(s))$  表示服务  $s$  的 QoS 属性, 其中  $q_i(s)$  决定了  $s$  的第  $i$  个属性的值。设  $A_1, A_2$  为两个任务,  $@$  表示某种模式, 则  $(A_1 @ A_2)$  可合并成一个虚拟任务  $VA \langle ID, SG, UA \rangle$ , 其中  $VA.ID = A_1.ID + A_2.ID$ ;  $VA.SG = \{cs_1 @ cs_2 | cs_1 \in A_1.SG \text{ and } cs_2 \in A_2.SG\}$ ,  $q_i(cs_1 @ cs_2) = f_k(q_i(cs_1), q_i(cs_2))$ ,  $f_k(x, y)$  是一个聚合函数(见表1);  $VA.UA = \{A, B\}$ 。该合并过程也可以用于两个虚拟任务或一个虚拟任务与一个任务之间。

表1 聚合函数的定义

QoS属性	顺序	并发	选择
响应时间	$q(x)+q(y)$	$\max\{q(x), q(y)\}$	$(q(x)+q(y))/2$
吞吐量	$\min\{q(x), q(y)\}$	$\max\{q(x), q(y)\}$	$(q(x)+q(y))/2$
可靠性	$q(x)q(y)$	$q(x)q(y)$	$(q(x)+q(y))/2$

$VA.UA$  中的每个元素代表了可完成  $VA$  的一个虚拟任务  $vs$ , 其效用函数可用式(1)定义:

$$U(vs) = \sum_{i=1}^r w_i \times sq_i(vs) \quad (1)$$

式中,  $w_i \geq 0$  且  $\sum_{i=1}^r w_i = 1$ ,  $sq_i(vs)$  为  $vs$  的第  $i$  个 QoS 属性的归一化值, 由式(2)定义:

$$sq_i(vs) = \begin{cases} \frac{q \max(VA, i) - q_i(vs)}{q \max(VA, i) - q \min(VA, i)} & q_i \text{ 为消极属性} \\ \frac{q_i(vs) - q \min(VA, i)}{q \max(VA, i) - q \min(VA, i)} & q_i \text{ 为积极属性} \end{cases} \quad (2)$$

式中,  $q \max(VA, i)$ 、 $q \min(VA, i)$  分别为  $VA.UA$  中第  $i$  个 QoS 属性的最大值与最小值。当  $q \max(VA, i) = q \min(VA, i)$  时,  $sq_i(vs)$  取值为 1。

## 2 Top-k 服务组合方法

根据 workflow 及虚拟任务的描述, O-QSC 问题可以看成是从里到外的一系列虚拟任务的合并过程。以串  $((A \odot B) \oplus (C \otimes D))$  为例, 可将  $(A \odot B)$  合并为  $V_1$ ,  $(C \otimes D)$  合并为  $V_2$ , 再将  $(V_1 \oplus V_2)$  合并为  $V_3$ , 最后从

$V_3.SG$  中优选一个组合实例。若某 workflow 涉及  $n$  个任务, 每个任务有  $m$  个候选服务, 则最后生成的虚拟任务将有  $m^n$  个候选服务, 仍为指数形式的时空复杂度。本文采用的 Top- $k$  方法则是在每次合并虚拟任务时, 仅为其保留  $k$  个最优的实例。该过程可由算法 1 描述。

若用合并虚拟任务的次数来度量算法 1 的时间复杂度, 显然为  $O(n)$ 。考虑到每次合成虚拟任务时, 需要从  $m^2$  或  $m \times k$  或  $k^2$  个组合实例中优选  $k$  个, 若采用先排序后选择的策略, 且  $k < m$ , 则算法 1 的时间复杂度可达到  $O(nm \log_2 m)$ 。合并过程中, 算法 1 将产生的所有虚拟任务依次保存于虚拟任务列表 (VAL) 中, 它将作为算法 2 的输入之一。

算法 1: Top- $k$  服务组合

输入: 描述 workflow 的串 OL 及  $k$  值

输出: 组合实例  $cs$  及虚拟任务列表 VAL

Initial an object stack OS;

for each object in OL

{

if (object != “”)

OS.push(object);

else

{

$A_1 = OS.pop()$ ;

Pattern = OS.pop();

$A_2 = OS.pop()$ ;

OS.pop();

$VA \leftarrow \text{Combine}(A_1, \text{Pattern}, A_2)$ ;

OS.push(VA);

VAL.add(VA);

}

}

$VA \leftarrow OS.pop()$ ;

from CS(VA), search the cs with best utility;

return cs

## 3 寻找替换服务

基于案例推理(case-based reasoning, CBR)策略, 文献[30]提出了一种自治愈合框架, 其中一个关键问题就是当组合方案中的某个服务运行出现异常时, 如何高效地为其寻找到一个合适的替代服务。尽管算法 1 实际上可以给出  $k$  组组合方案, 但这些组合方案往往都涉及某些服务, 当这些服务中的某个出现异常时, 就不能相互替换了。因此, 本文通过算法 2 为组合方案中涉及的每个服务寻找一个替代服务, 其中  $cs(A)$  表示任务  $A$  的候选服务集,  $UCS(cs)$  表示组

合实例cs涉及的原子服务的集合。该算法基于如下的假设: 对于一个虚拟任务的两个优选候选服务, 如果它们所涉及的原子服务中仅有一对是不相同的, 则这对原子服务是可以相互替换的。例如, 对于工作流( $P \odot A$ ), 若存在优选的组合实例( $P_{11}, A_{11}$ )与( $P_{11}, A_{12}$ ), 则 $A_{11}$ 与 $A_{12}$ 是可替换的。

设一个虚拟任务的优选候选服务个数为 $k$ , 它们可以形成 $k(k-1)/2$ 对服务组合, 故算法2的时间复杂度为 $O(nk^2)$ ,  $n$ 为工作流中涉及的任务个数。该算法并不能总为每个服务寻找到至少一个可替代服务, 但当 $k$ 稍大时, 能够成功找到的概率是很大的, 后面的测试将证明这一点。

算法2: 寻找替换服务

输入: 组合实例cs, 虚拟任务列表VAL

输出: 替换服务列表AS

```

For each  $WS_i$  appeared in cs
{
    Look for  $A_i$  where  $cs(A_i) \ni WS_i$ ;
    From VAL, search the first  $VA_j$  which
    combined by  $A_i$  in turn;
    For each pair of  $cs_1$  and  $cs_2$  in  $CS(VA_j)$ 
        If  $((UCS(cs_1) - UCS(cs_2)) \ni \{WS_i\})$  then
            add  $UCS(cs_2) - UCS(cs_1)$  to  $AS_i$ .
}
Return AS

```

### 4 实验分析

为了验证该方法的有效性, 本文分别从 $k$ 值对效用值的影响、算法的求解质量、时间开销以及可替代性能4个方面进行分析。

#### 4.1 实验环境

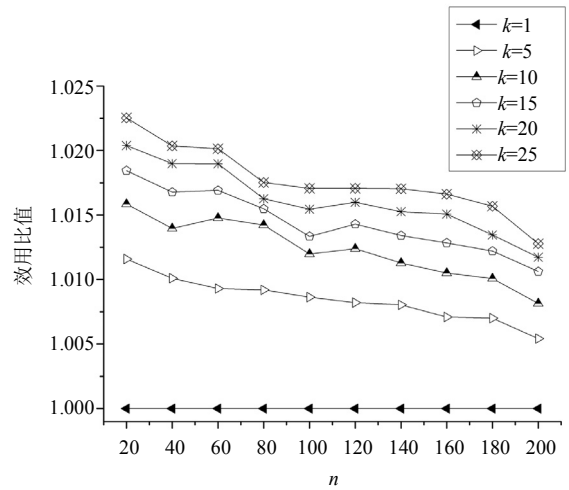
该实验是在一台拥有Intel(R) Core (TM) i5-2 430 M、2.4 GHz Intel Xeon双核处理器、4 GB RAM、Win7操作系统的Lenovo PC机上实现的, 实验代码用Java 7编写。测试时, 包含顺序、选择和并发模式的复杂工作流是随机产生的, 其中并发、选择模式的数量各占1/4。分配给每个候选Web服务的QoS值来源于QWS数据集<sup>[31-32]</sup>, 它包含了2 508个真实Web服务, 每个服务拥有10个质量属性, 测试时选用了具有代表性的响应时间、吞吐量、可靠性等3个属性。每种规模测试20次后取平均值。

#### 4.2 实验结果

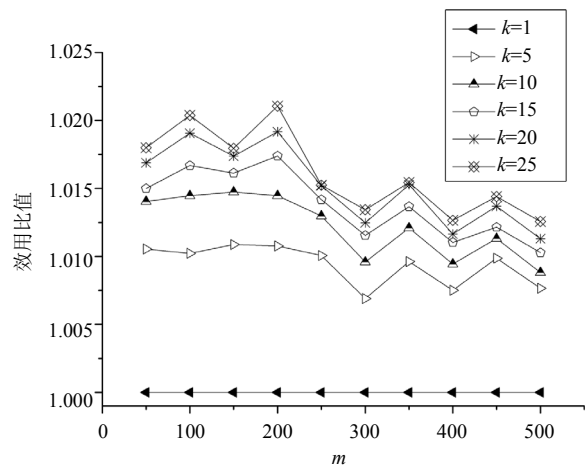
##### 4.2.1 $k$ 值对效用的影响

图3为不同规模、不同 $k$ 值时的效用, 为了使不

同情况下效用值的对比效果更明显, 突出 $k$ 值对效用的影响, 这里以 $k=1$ 时的效用值为参考标准(即设 $k=1$ 时的效用比值为1), 其它 $k$ 值时的效用比值为其实际效用值与 $k=1$ 时的实际效用值之比。从图中可以看出: 1) 相同问题规模下, 效用比值随 $k$ 值的增大而缓慢增大; 2) 当 $k=10$ 时, 已经可以得到较为理想的优化效果; 3) 同样的 $k$ 值时, 效用随问题规模的增大而略有降低。



a.  $m=100$ ,  $n$ 从20递增到200



b.  $n=50$ ,  $m$ 从50递增到500

图3 不同 $k$ 值下的效用比值

##### 4.2.2 算法的求解质量

图4对不同问题规模下Top- $k$ 优选( $k=10$ )、局部选择、遗传算法(genetic algorithm, GA)等3种方法的效用值进行了比较。该图的纵坐标是以Top-10方法得到的效用值为参考标准的, 即设Top- $k$ 方法的效用比值为1, 而其它情况为其效用值与Top-10方法效用值之比。从图中可以看出, Top- $k$ 优选的效用值明显高于局部选择算法和GA算法。

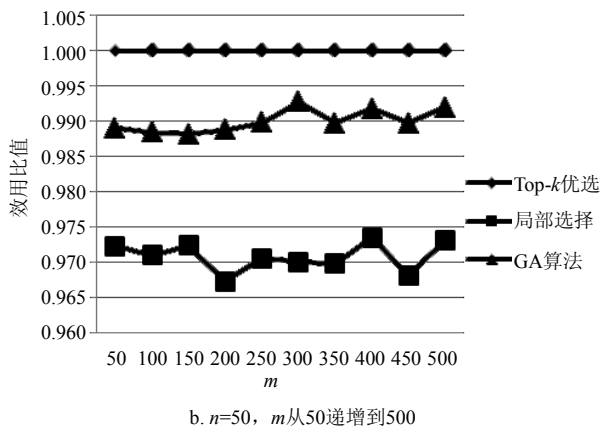
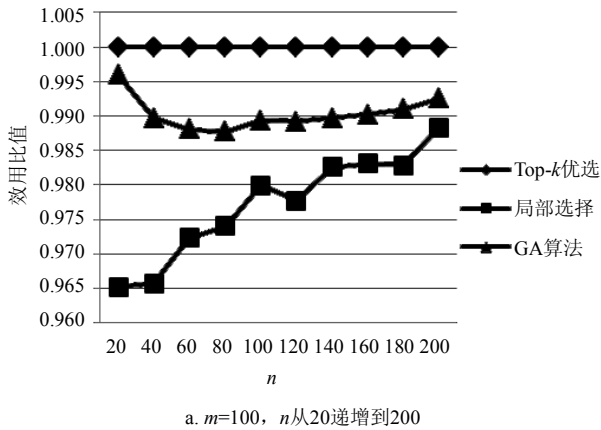


图4 算法求解质量的比较

### 4.2.3 时间开销分析

图5给出了不同 $k$ 值、不同规模下, Top- $k$ 方法的时间开销情况。从中可以发现, 该方法的时间开销大致与 $n^2$ 、 $m^2$ 成正比, 与 $k$ 值关联不大。测试代码仍有一定优化空间, 整体时间开销较为理想。

### 4.2.4 可替代性能分析

本文从能够寻找到替代方案的概率 $r/t$ ( $r$ 表示方案中所提到的Web服务个数,  $t$ 表示至少寻找到一个可替代服务的Web服务个数)及替代方案的质量两个方面评价算法的可替代性能。

从图6可以看出, 可替代概率随 $k$ 值的增大而增大; 当 $k=12$ 时, 该概率可达80%。

在替代方案的质量测试中, 本文将 $k=10$ 、 $k=20$ 时的替代效果与 $M_2$ (局部最优)、 $M_3$ (QoS效用差距最小<sup>[30]</sup>)、 $M_4$ (随机选择)3种情况进行了比较(见图7, 以 $M_2$ 的效用值作为参考)。  $M_2$ 是选择候选服务中综合效用最高的(局部最优),  $M_3$ 是选择候选服务中与需要替换的服务的QoS差距最小的,  $M_4$ 是从候选服务中随机选择。该算法的效果略优于其他算法。

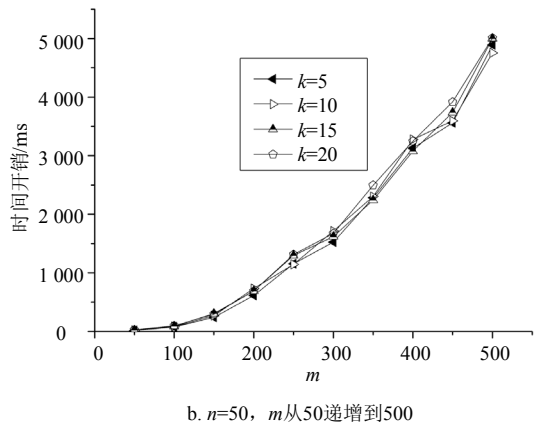
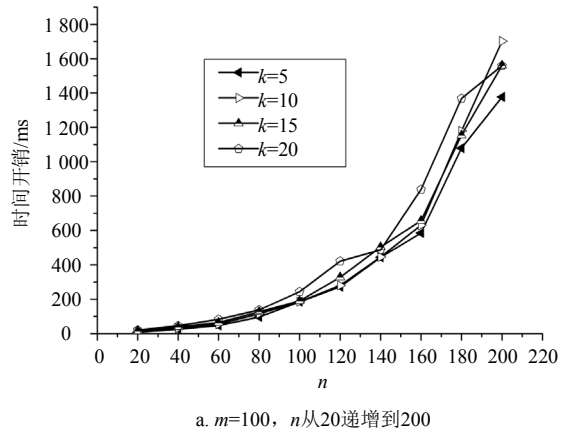


图5 不同 $k$ 值下的时间开销

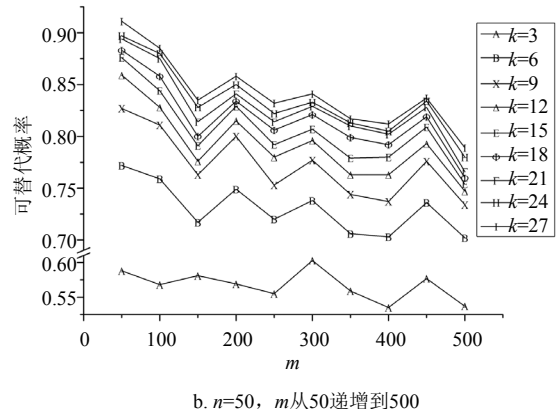
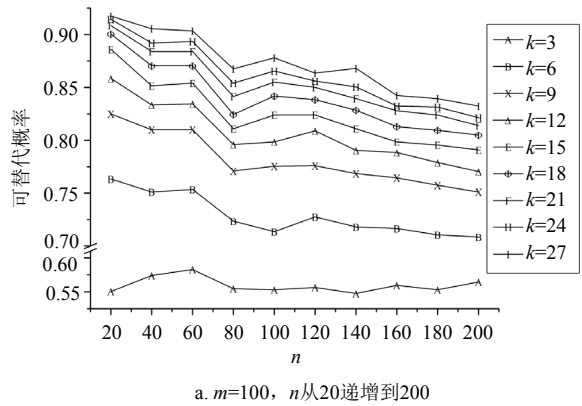
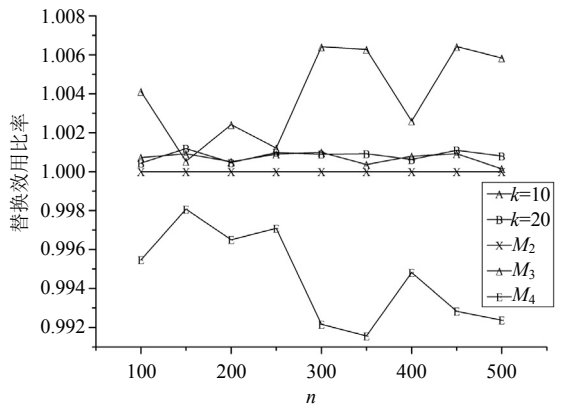
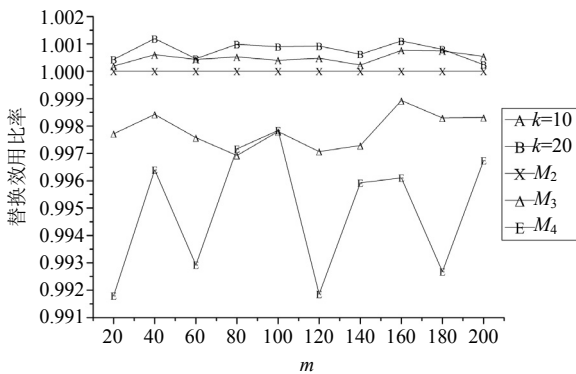


图6 不同 $k$ 值下的可替代概率



a. m=100, n从20递增到200



b. n=50, m从50递增到500

图7 替代效果的比较

### 5 结束语

通过将 workflow 描述为一个符号串, 本文提出了一种由里及外逐步合并虚拟任务的 Web 服务组方法; 为了降低时空复杂度, 在每次合并虚拟任务时采用了 Top-k 优选策略。这种方法兼顾了局部优选与全局优选的特点, 可以以较小的时间复杂度获得较优的组合方案。同时, 本文给出了一种新的寻找可替换服务的算法, 支持一定的自愈能力。

### 参 考 文 献

[1] LIU Xuan-zhe, HUANG Gang, HONG Mei. Discovering homogeneous web service community in the user-centric web environment[J]. IEEE Transaction on Services Computing, 2009, 2(2): 167-181.  
 [2] 张佩云, 黄波, 孙亚民. 面向服务组合的服务语义匹配机制[J]. 电子科技大学学报, 2008, 37(6): 917-921.  
 ZHANG Pei-yun, HUANG Bo, SUN Ya-min. Service semantic matching mechanism for web services composition[J]. Journal of University of Electronic Science and Technology of China, 2008, 37(6): 917-921.  
 [3] ZOU Guo-bing, LU Qiang, CHEN Yi-xin, et al. QoS-Aware dynamic composition of web services using numerical temporal planning[J]. IEEE Transactions on Services Computing, 2014, 7(1): 18-31.

[4] WANG Peng-wei, DING Zhi-jun, JIANG Chang-jun, et al. Constraint-aware approach to web service composition[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2014, 44(6): 770-784.  
 [5] 陈文宇, 张忠全, 向涛, 等. 基于相似度的语义 Web 服务发现技术研究[J]. 电子科技大学学报, 2010, 39(6): 896-899.  
 CHEN Wen-yu, ZHANG Zhong-quan, XIANG Tao, et al. Research on similarity-based semantic web services discovery[J]. Journal of University of Electronic Science and Technology of China, 2010, 39(6): 896-899.  
 [6] YU Tao, ZHANG Yue, LIN Kwei-jay. Efficient algorithms for web services selection with end-to-end QoS constraints[J]. ACM Transactions on the Web Service, 2007, 1(1): 1-26.  
 [7] ARDAGNA D, PERNICI B. Adaptive service composition in flexible processes[J]. IEEE Transactions on Software Engineering, 2007, 33(6): 369-384.  
 [8] ZENG Liang-zhao, BENATALLAH B, DUMAS M, et al. QoS-aware middleware for web services composition[J]. IEEE Transactions Software Engineering, 2004, 30(5): 311-327.  
 [9] ALRIFAI M, RISSE T. Combining global optimization with local selection for efficient QoS-aware service composition[C]//Proceedings of the 18th International Conference on World Wide Web. New York: ACM, 2009: 881-890.  
 [10] LUO Yuan-sheng, QI Yong, HOU Di, et al. A novel heuristic algorithm for QoS-aware end-to-end service composition[J]. Computer Communications, 2011, 34(9): 1137-1144.  
 [11] JAEGER M C, MUHL G, GOLZE S. QoS-aware composition of web services: an evaluation of selection algorithms[C]//OTM Confederated International Conference. Berlin Heidelberg: Springer, 2005: 646-661.  
 [12] 冯建周, 孔立富. 基于 QoS 的 Web 服务组合中多路径全局优化方法的研究[J]. 小型微型计算机系统, 2013, 34(7): 1493-1497.  
 FENG Jian-zhou, KONG Li-fu. Research on the method of multi-path global optimization in QoS-based Web service composition[J]. Journal of Chinese Computer Systems, 2013, 34(7): 1493-1497.  
 [13] GABREL V, MANOUVRIER M, MURAT C. Web services composition: Complexity and models[J]. Discrete Applied Mathematics, 2014, 196(2): 67-82.  
 [14] KLEIN A, ISHIKAWA F, HONIDEN S. Efficient heuristic approach with improved time complexity for QoS-aware service composition[C]//IEEE International Conference on Web Services. Washington, USA: IEEE Computer Society, 2011: 436-443.  
 [15] DING Zhi-jun, LIU Jun-jun, SUN You-qing, et al. A transaction and QoS-aware service selection approach based on genetic algorithm[J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2015, 45(7): 1035-1046.  
 [16] YU Yang, MA Hui, ZHANG Meng-jie. An adaptive genetic programming approach to QoS-aware web services

- composition[C]//IEEE Congress on Evolutionary Computation. Cancun: IEEE Computer Society, 2013: 1740-1747.
- [17] 温涛, 盛国军, 郭权, 等. 基于改进粒子群算法的Web服务组合[J]. 计算机学报, 2013, 36(5): 1031-1046.  
WEN Tao, SHENG Guo-jun, GUO Quan, et al. Web service composition based on modified particle swarm optimization[J]. Chinese Journal of Computers, 2013, 36(5): 1031-1046.
- [18] YU Qing, CHEN Ling, LI Bin. Ant colony optimization applied to web service compositions in cloud computing[J]. Computer and Electrical Engineering, 2015, 41(1): 18-27.
- [19] YILMAZ A E, KARAGOZ P. Improved genetic algorithm based approach for QoS aware web service composition[C]//IEEE International Conference on Web Services. Anchorage: IEEE Computer Society, 2014: 463-470.
- [20] 张燕平, 荆紫慧, 张以文, 等. 基于离散粒子群算法的动态Web服务组合[J]. 计算机科学, 2015, 42(6): 71-75.  
ZHANG Yan-ping, JING Zi-hui, ZHANG Yi-wen, et al. Dynamic web service composition based on discrete particle swarm optimization[J]. Computer Science, 2015, 42(6): 71-75.
- [21] YIN Hao, ZHANG Chang-sheng, GUO Ying, et al. An improved evolutionary multiobjective service composition algorithm[C]//The International Symposium on Computational Intelligence and Design. Hangzhou, China: IEEE, 2013, 1: 269-272.
- [22] 苏凯, 马良荔, 郭晓明, 等. 一种QoS感知的服务全局优化选择算法[J]. 华中科技大学学报(自然科学版), 2014, 42(4): 72-76.  
SU Kai, MA Liang-li, GUO Xiao-ming, et al. QoS-aware service selection global optimization algorithm[J]. Journal of Huazhong University of Science and Technology. (Natural Science Edition), 2014, 42(4): 72-76.
- [23] ZHANG Wei, CHANG C K, FENG Tai-ming, et al. QoS-based dynamic web service composition with ant colony optimization[C]//37th Annual IEEE International Computer Software and Applications Conference. Seoul: IEEE Computer Society, 2010: 493-502.
- [24] 夏亚梅, 程渤, 陈俊亮, 等. 基于改进蚁群算法的服务组合优化[J]. 计算机学报, 2012, 35(2): 270-281.  
XIA Ya-mei, CHENG Bo, CHEN Jun-liang, et al. Optimizing services composition based on improved ant colony algorithm[J]. Chinese Journal of Computers, 2012, 35(2): 270-281.
- [25] 王尚广, 孙其博, 杨放春. 基于全局QoS约束分解的Web服务动态选择[J]. 软件学报, 2011, 22(7): 1426-1439.  
WANG Shang-guang, SUN Qi-bo, YANG Fang-chun. Web service dynamic selection by the decomposition of global QoS constraints[J]. Journal of Software, 2011, 22(7): 1426-1439.
- [26] MARDUKHI F, NEMATBAKHS N, ZAMANIFAR K, et al. QoS decomposition for service composition using genetic algorithm[J]. Applied Soft Computing, 2013, 6(5): 3409-3421.
- [27] SUN S X, ZHAO Jing. A decomposition-based approach for service composition with global QoS guarantees[J]. Information Sciences, 2012, 199(15): 138-153.
- [28] LIU Zhi-zhong, XUE Xiao, SHEN Ji-quan, et al. Web service dynamic composition based on decomposition of global QoS constraints[J]. Int J Adv Manuf Technol, 2013, 69(9): 2247-2260.
- [29] TAN Wei, ZHOU Meng-chu. Business and scientific workflow: a web service-oriented approach[M]. IEEE Press Series on Systems Science and Engineering. Wiley: IEEE, 2013.
- [30] LI Guo-qiang, LIAO Le-jian, SONG Dan-dan, et al. A fault-tolerant framework for QoS-aware web service composition via case-based reasoning[J]. International Journal of Web and Grid Services, 2014, 10(1): 88-99.
- [31] AL-MASRI E, MAHMOUD Q H. Discovering the best web service[C]//Proceedings of the 16th International Conference on World Wide Web. New York: ACM, 2007: 1257-1258.
- [32] AL-MASRI E, MAHMOUD Q H. QoS-based discovery and ranking of web services[C]//Proceedings of the 16th International Conference on Computer Communications and Networks. Honolulu: IEEE, 2007: 529-534.

编辑 蒋晓