

# 全局视角下的 Web 服务系统模型及推荐策略

刘 轶<sup>1</sup>, 范玉顺<sup>1+</sup>, 黄科满<sup>2</sup>

(1. 清华大学 自动化系, 北京 100084; 2. 天津大学 计算机科学与技术学院, 天津 300072)

**摘要:**考虑到当前的服务推荐策略大多缺乏对系统整体发展需求的关注, 结合服务新增、失效等动态要素构建了服务生态系统仿真模型; 从全局发展的视角出发, 提出了服务系统综合发展指标体系; 结合服务系统模型和综合发展指标体系, 设计了能够保障系统中服务多样性、减小服务失效率的推荐策略, 该策略将服务分为新服务、保护服务和普通服务并分别设计推荐机制; 在基于实际数据构建的仿真模型上进行测试, 验证了推荐策略的有效性。

**关键词:**Web 服务生态系统; 服务失效; 系统发展指标; 服务推荐

中图分类号: TP311

文献标识码: A

## Web service ecosystem model and service recommendation strategy from overall perspective

LIU Yi<sup>1</sup>, FAN Yushun<sup>1+</sup>, HUANG Keman<sup>2</sup>

(1. Department of Automation, Tsinghua University, Beijing 100084, China;

2. Department of Computer Science and Engineering, Tianjin University, Tianjin 300072, China)

**Abstract:** Owing to the most of present recommendation methods were focused on users' perspective only, a service recommendation strategy from the overall perspective was proposed based on a Web service ecosystem model with dynamic factors to consider the development and stability of the whole system. In the proposed strategy, the services were divided into the new service group, the protected service group and the normal group, and the corresponding recommendation mechanisms were design. Based on the real data, the simulation model was constructed to verify the effectiveness and efficiency of the new strategy.

**Keywords:** Web service ecosystem; service deprecation; system developing indicators; service recommendation

## 1 问题的提出

面向服务的体系结构 (Service Oriented Architecture, SOA)<sup>[1]</sup> 的提出革新了软件系统的构建模式, 而 Web 服务则是互联网环境下 SOA 的典型实现, 用户通过标准化定义的接口调用 Web 服务, 只需要对 Web 服务进行组合就可以便捷地构造出功能完善的应用软件系统<sup>[2]</sup>。随着 Web 服务数量的逐渐增长, 以 ProgrammableWeb 为代表的服务平台逐渐取代传统的统一描述、发现和集成协议 (U-

niversal Description, Discovery and Integration, UDDI) 服务注册中心, 成为 Web 服务发布、查找和使用的中介。截止到 2015 年 1 月 30 日, ProgrammableWeb 平台上发布的 Web 服务应用程序接口 (Application Programming Interface, API) 数量已经达到 12 790 个, 涵盖地图、社交、搜索等数百个功能领域, 而平台上基于这些原子服务构建的服务组合 (mashup) 数量达到了 6 164 之多, 经过近 10 年的发展, ProgrammableWeb 已经成为互联网上最大的 Web 服务公共平台之一。

收稿日期: 2015-03-02; 修订日期: 2015-06-01。Received 02 Mar. 2015; accepted 01 June 2015.

基金项目: 国家自然科学基金资助项目 (61174169); 国家 863 计划资助项目 (2012AA02A613); 高等学校博士学科点专项科研基金资助项目 (20120002110034)。**Foundation items:** Project supported by the National Natural Science Foundation, China (No. 61174169), the National High-Tech. R&D Program, China (No. 2012AA02A613), and the Specialized Research Fund for the Doctoral Program of Higher Education, China (No. 20120002110034).

随着平台上服务数量的增加,帮助用户找到合适的服务已成为服务系统运营中最重要的问题<sup>[3]</sup>,而服务推荐也成为服务科学领域的研究热点。服务推荐的核心任务是分别识别用户需求和\*\*服务特征,随后进行用户需求和\*\*服务特征的匹配,而服务特征又可以分为服务功能和服务质量(Quality of Service, QoS)<sup>[4]</sup>两部分。在用户需求识别方面,当前的算法不仅考虑了用户对于服务最基本的功能和质量需求<sup>[5]</sup>,还构建了服务的信誉评价体系,并能够根据用户的地理位置<sup>[6-7]</sup>、上下文使用环境<sup>[8-9]</sup>、服务选择偏好<sup>[10]</sup>等特征进行服务推荐。在服务特征的辨识方面,最初的服务功能识别基于 Web 服务描述语言<sup>[11]</sup>(Web Service Descriptive Language, WSDL)进行,而随着基于 RESTful<sup>[12]</sup>的服务架构的兴起,第三方不再能获取丰富详细的 WSDL 文件,基于服务描述文本识别和服务关联网络<sup>[13,14]</sup>的方式成为当前研究的主流。

在服务平台上,平台运营者、服务提供者、用户、服务和\*\*服务组合等角色相互作用,形成了交互频繁、不断演化的复杂系统,该系统被称为服务生态系统<sup>[15]</sup>。根据生物学中的研究,生态系统的多样性<sup>[16]</sup>极大地影响着系统抵抗冲击的能力,也意味着系统能够更加稳定持续地提供功能。然而通过对 Web 服务系统进行分析后发现,服务平台上的资源分布出现了较严重的不均衡性:即少部分的服务占用了绝大多数的用户资源,大部分的服务则很少得到关注和调用——在 ProgrammableWeb 平台上,只有 22.2%的服务曾经被该平台上发布的\*\*服务组合调用(如图 1)<sup>[17]</sup>。

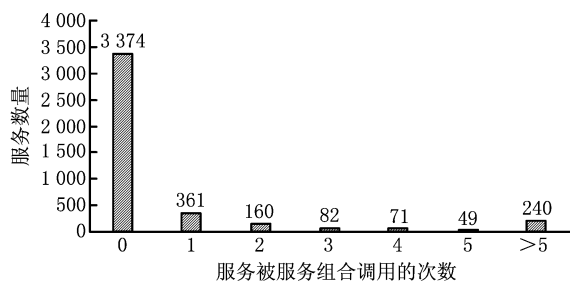


图1 服务被服务组合调用次数的分布<sup>[17]</sup>

当前主要的服务推荐算法从用户需求和服务特征的匹配出发,旨在仅推荐匹配度最高的服务给用户,然而在考虑信誉等累计型要素的情况下,被推荐并使用的服务能够迅速积累口碑而成长,服务之间用户资源的差距将会进一步拉大。从全局的视角来看,这种推荐模式存在以下问题:

(1)新加入系统的 Web 服务由于缺少足够的信誉记录而无法得到推荐机会、实现信誉的积累,新的服务开发者在当前机制下难以激发持续创新的热情;

(2)平台上服务关注度的差距持续增大,长期被推荐系统忽略的边缘化服务由于没有得到预期的收益而失去维护的动力甚至最终退出系统,进而导致服务失效率升高,服务系统的多样性逐渐降低,平台功能的稳定性降低。

针对目前服务推荐算法中的问题,本文提出了一种综合考量系统发展与\*\*服务匹配度的 Web 服务推荐策略。为了设计推荐策略并研究其对系统发展的影响,首先从服务平台的基本结构入手,构建了包含服务失效、信誉变化等动态要素的服务系统仿真模型,随后提出了服务平台发展的多样性和公平性指标,以其为优化目标,设计了保障服务平台稳定发展的服务推荐策略,最后基于实际服务平台数据集,在模型上进行了服务推荐策略的效果评估。

## 2 服务系统基本结构

### 2.1 服务系统结构

服务具有功能特征和质量特征,其中功能特征是能够代表服务本身所提供功能的特征(例如谷歌地图的功能是帮助调用者将地图嵌入 Web 应用中,同时也允许开发者利用 JavaScript 脚本进行应用开发拓展),而服务的质量特征则是对服务的 QoS 指标<sup>[18]</sup>和信誉指标<sup>[19]</sup>的概括,综合表征服务的非功能属性,包括服务的可用性、可靠性、费用和信誉值等。在服务系统中,功能特征相似的服务可以被划分为同一个服务种群,一个服务种群中的服务属于同一个服务功能域,在功能层面彼此具有替代性,但是在服务质量上可能不同。同一个种群中的服务针对需求相似的用户群体,因此相互之间具有一定的竞争关系;而不同种群中的服务可能被服务组合、一同调用,相互协调完成更加复杂的流程,具有合作关系。

一个典型的 Web 服务系统由服务平台运营者、服务提供者、服务组合开发者和\*\*服务(组合)用户等四种角色组成(如图 2)。其中:服务平台运营者是整个 Web 服务平台的运营者,负责管理平台上的服务及\*\*服务组合,需要设计服务推荐系统,针对用户的需求进行服务或服务组合的推荐;服务提供者是 Web 服务的创建者,也是服务本身的管理者和维护者,定义了服务的功能、接口,并决定了服务的质量,同时,服务提供者也可能会根据服务被使用的情况

进行服务的更新或撤销服务;服务组合开发者调用服务、构成服务组合,以实现更加复杂的功能,服务组合也可以被发布到平台上供用户使用;服务用户和服务组合用户则根据自身需求从服务平台上选择服务或服务组合进行调用,在调用完成后,可能会根据试用的情况进行评价。

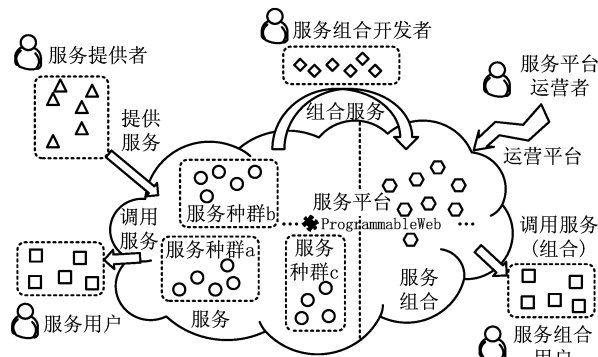


图2 服务生态系统结构

在服务生态系统中,服务的推荐过程是各角色发生交互的核心环节,平台上服务推荐的发生过程如图3所示,服务提供者首先结合自身的业务及技术背景提供服务,并将服务信息发布到服务平台上,服务平台运营者根据服务提供者提供的 Web 服务信息进行 Web 服务的组织,并维护服务质量信息;当服务用户产生调用需求时,服务平台运营者利用用户需求识别系统,根据服务数据和用户需求进行服务推荐,用户从推荐服务列表中选择服务并调用,在使用完成后,用户对服务进行评价。服务组合开发者可以视为服务的用户,因为他们也需要在平台上选取服务来构造新的服务组合并发布到服务平台上。

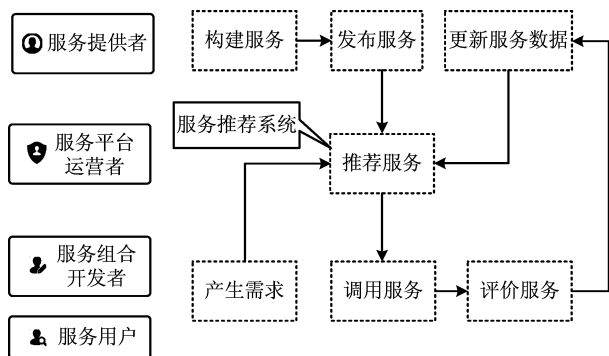


图3 服务生态系统推荐流程

不同的角色在服务生态系统中发挥不同的作用,对于生态系统运行过程的期望也不同。服务提供者和组合开发者都希望有更多用户能够使用

自己的服务,服务用户则仅希望能够发现契合自己需求同时服务质量最高的服务,而对于服务平台运营者来说,则希望在兼顾服务提供者、服务组合开发者和终端用户基本需求的前提下,拥有更多种类、更多数量的服务,保证服务生态系统的持续繁荣发展,相对于当前从用户视角出发的服务推荐模式而言,服务平台运营者视角下的服务推荐策略应该更加注重全局最优,力求综合考虑服务群体和用户群体的整体发展,这也正是本文的出发点所在。

## 2.2 服务生态系统中的动态要素

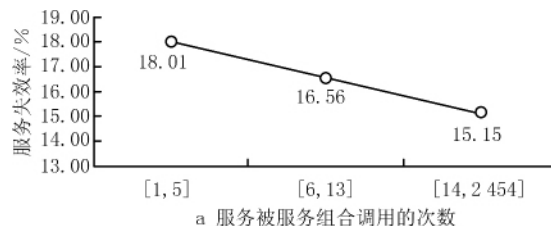
服务生态系统是一个复杂的系统,除了图3中所展示的推荐流程之外,诸多动态要素也会作用于服务系统,对系统的状态产生影响。

### (1) 服务新增

新的服务不断加入系统,这些服务可能属于现有的功能域,也可能是当前系统中所不具有的服务类型,这类创新型的服务会成为系统中新的服务种群。由于新加入系统的服务通常没有信誉记录,需要系统给出对应的初始信誉值,另一方面,由于新服务的累积评价较少,如果不在推荐过程中给予特别的关注,则这些新的服务难以获得足够的关注而成长。

### (2) 服务失效

与生物系统中个体的死亡类似,Web 服务也可能退出系统,这种现象被称为服务失效<sup>[20]</sup>。根据统计,截止到2013年7月,在ProgrammableWeb平台上发布的9 646个服务中,已经有11.2%的服务被标注了“失效”标签,这些失效服务不仅影响了直接调用或通过服务组合使用它们的用户,而且会导致服务平台上的服务种群多样性受到影响。通过分析 ProgrammableWeb 上的数据,可以发现服务的失效与服务本身在系统中得到重视的情况有比较明显的相关关系(如图4):被服务组合调用次数越多的服务以及在服务协作网络中处于更加核心地位的服务,其失效率会相对较低。虽然服务退出系统的原因可能多种多样,但是在平台中持续处于边缘化的服务将会具有更高的失效几率。



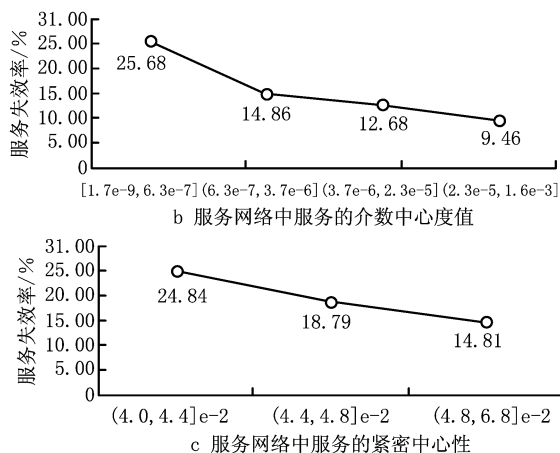


图4 服务失效与服务网络地位

当前的服务推荐模式仅仅从服务本身的功能和质量出发进行推荐,没有从全局发展的角度综合考虑,为其他服务提供一定的展示机会,从而进一步导致服务的失效率提升。针对可能失效的服务,服务平台运营者一方面可以通过收集服务数据训练分类器进行预判,对可能调用的用户予以提示;另一方面可以从积极的角度入手,通过采用新的推荐方式改变当前服务系统中极不均匀的推荐模式,给予边缘化的服务一定的展示机会,从而获得用户,避免服务因长期在系统中的边缘化而失效。

### (3) 服务创新

作为一种特殊形态的软件,Web 服务中也存在相应的迭代更新。这里的服务创新是指 Web 服务平台中,现有的 Web 服务通过功能层面的重新定义和编写而对原有服务功能进行改变的现象。这些经过了创新的服务可能会更加符合用户的需求,或能够开辟新的服务功能领域。服务持续创新是服务具有高度活力的表现,有利于增强系统的多样性。服务平台的运营者可以根据当前系统中已有的服务类型发掘潜在的功能需求,对服务提供者的创新过程提出建议。

### (4) 服务质量变化

服务质量是对服务非功能特性的描述,不论是服务的 QoS 特征还是服务信誉特征,都可能不断发生变化。服务的 QoS 值除了服务提供者自身行为对服务的改变之外,还会受到网络服务等环境因素的影响,表现出较强的波动性。服务 QoS 值的变化将会影响服务推荐时的优先级,从而影响用户对于服务的选用。

服务的信誉特征是对服务可信度的描述,它一

方面来自于服务刚进入系统所设定的初始值,另一方面则依赖于使用过程中用户对于服务的评价,如果能持续收到较高的评价,则服务的信誉值可以不断提升,如果服务一直得不到使用,则其信誉值会随着时间不断衰退。

## 3 服务生态系统建模及仿真

通过构建服务生态系统的模型,可以对系统的运行和演化过程进行仿真,继而考察服务系统在不同推荐模式作用下的发展方向。下面结合服务生态系统的基本结构构建结构模型,同时对服务生态系统中的主要动态要素和运行机制进行建模。服务生态系统是一个由多种类型的个体与诸多环境要素相互作用而形成的复杂系统,多 Agent 仿真模型<sup>[21]</sup>对于这类系统的建模和仿真具有先天的优势,因此下文基于多 Agent 建模的基本框架构造服务生态系统模型。

### 3.1 结构模型

**定义 1 服务个体。**服务个体  $S$  代表服务生态系统中的服务,其属性用多元组表示,  $S = \langle Id, Name, Domain, LiveTime, RpuValue, UseHistory, RcmmdHistory \rangle$ 。其中:  $Id$  和  $Name$  分别表示服务的编号和名称,  $Domain$  表示服务所属的功能域,  $LiveTime$  为服务的存活时间,  $RpuValue$  记录服务的信誉值,  $UseHistory$  为服务个体对于自身调用情况的记录,包含调用时间以及相应的调用者编号,  $RcmmdHistory$  记录服务个体被推荐的情况,包括推荐时间、推荐排名和推荐队列等。

**定义 2 服务组合。**服务组合  $C$  表示服务生态系统中的服务组合,其属性用多元组表示,  $C = \langle Id, Name, LiveTime, RpuValue, UseHistory, SerList \rangle$ 。其中:前五项的定义与服务个体的定义类似,  $SerList$  为对服务组合所调用的服务列表的记录。

**定义 3 服务用户。**服务用户  $U$  表示服务生态系统中的用户,使用多元组表示为  $U = \langle Id, CreTime, UseHistory \rangle$ 。其中:  $CreTime$  表示用户进入系统的时间,  $UseHistory$  为用户的使用记录,记录了当前用户调用的服务  $Id$  和对应的调用时间。

在服务生态系统中,由于服务提供者、服务组合开发者的行为可以分别体现在服务和组合的变化上,本文不进行单独建模。

### 3.2 动态要素建模

在服务生态系统模型中,动态要素推动系统不

断发展演化,以下选取服务新增、服务组合新增、服务用户新增、服务信誉值变化和服务失效作为典型动态要素进行介绍。

#### (1) 服务新增

服务新增代表有新的服务不断加入系统中,新的服务可能属于现有功能域,也可能是新的功能域。在多 Agent 仿真模型中,设定服务生成器(service factory agent)来完成服务新增的工作,根据一定的规律,服务生成器在每个时间步生成若干新的服务,并将服务添加到系统中。

同时,新服务在加入系统后,还由服务平台需要对新的服务进行信誉值初始化设定(boot strap-ping)。在进行初始值设定时,可以采用两种不同的方式:

1) 直接设定默认初始值(DI)方式,将新服务的初始值设定为某个介于最大信誉值和最小信誉值之间的值  $R_{ini}$ ,服务平台运营者可以根据本身的运营策略选择不同的值,如果选择的  $R_{ini}$  高于当前平均值,则可以视为鼓励型的初始值设定方式,反之,如果选择的  $R_{ini}$  低于当前的平均值,则可认为是保守型的初始值设定方式。

2) 适应性设定方式(AI),所有的新加入系统的服务初始信誉值都根据当前系统中所有服务信誉值的平均值进行设定,默认所有的新加入系统中的服务至少具有当前信誉值的平均水平。

#### (2) 服务组合新增

服务组合新增代表新的服务组合加入了系统。每一个新增的服务组合都需要选择用于组合的服务,在仿真模型中,由服务组合生成器(composition factory agent)生成新的服务组合,并随机设定新的服务组合所需要选取的服务类型;随后,服务组合根据服务类型从对应的功能域中选取具体的服务,并将服务的相关信息添加到服务列表  $SerList$  中。

同样地,对于服务组合信誉值也需要进行初始化,初始化的方式与服务初始化的方式类似,既可以采用默认值,也可以按照当前系统中的服务组合信誉平均值进行设定。

#### (3) 服务(组合)用户新增

新的用户不断从服务系统中选取服务或服务组合,每一个新的用户以概率  $P_{ser}$  ( $0 \leq P_{ser} \leq 1$ ) 选择服务,以  $1 - P_{ser}$  选择服务组合,新增的用户由用户生成器(user factory agent)生成。当然,结合实际的模型需要,可以将  $P_{ser}$  设置为 1 或 0(代表用户仅选

择服务或服务组合)。

#### (4) 服务信誉值变化

在每一个时间步内,服务可能被新的服务组合调用,也可能直接被用户调用,每当发生一次调用,服务根据用户调用对于服务的评分或利用服务组合的信誉值来更新当前的信誉值。

若时间步  $t$  内服务  $s_i$  被新的服务组合  $C_j$  调用,对应的服务组合信誉值为  $CT_{t,j}$ ,则可以直接采用加权平均的方式在原有的服务信誉值  $T_{t,j-1}(s_i)$  上进行加权,得到更新后的信誉值  $T_{t,j}(s_i)$ ,表达式如下:  

$$T_{t,j}(s_i) = (1 - w)T_{t,j-1}(s_i) + w \times CT_{t,j}(s_i) \quad (1)$$
 式中:  $w$  ( $0 \leq w \leq 1$ ) 为加权参数,  $w$  值越大,新的服务组合信誉值在权重更新的过程中也更加受到重视。

类似地,若在时间步  $t$  内服务  $s_i$  被新的用户  $U_j$  调用,对应的用户反馈值为  $UF_{t,j}$ ,则可以直接采用加权平均的方式在原有的服务信誉值  $T_{t,j-1}(s_i)$  上进行加权,得到更新后的信誉值  $T_{t,j}(s_i)$ ,表达式如下:

$$T_{t,j}(s_i) = (1 - v)T_{t,j-1}(s_i) + v \times UF_{t,j}(s_i) \quad (2)$$
 式中  $v$  ( $0 \leq v \leq 1$ ) 为加权参数。

另外,由于服务本身的 QoS 值等特征随时间可能会不断发生变化,对服务信誉做出的评价随着时间的推移也应该逐渐失去效力,模型中还对服务的信誉值设置了时间衰减参数  $\lambda$ ,在每一个时间步  $t$  结束时,原有的服务信誉值  $T'_{t,j}(s_i)$  将会被更新为  $T_{t,j}(s_i)$ 。

$$T_{t,j}(s_i) = T'_{t,j}(s_i) \times e^{-\lambda} \quad (3)$$

#### (5) 服务失效

根据 2.2 节中对于服务失效行为的讨论可以发现,在服务系统中处于边缘化地位的服务具有较高的失效概率,随着服务被边缘化的时间越长,服务失效的概率也越大。在传统的电子元器件可靠性研究中,威布尔函数被用来模拟元器件正常工作的时间,在服务系统中,服务未被使用的时间与服务的失效概率呈正相关,因此,使用威布尔函数对服务失效概率与服务未被使用的时间关联进行建模。设当前时刻距离服务上次被调用的时间长度为  $t_f$ ,则对应的失效概率为  $P_f$ ,

$$P_f = 1 - \exp\left(-\left(\frac{t_f}{\theta}\right)^\beta\right) \quad (4)$$

式中  $\theta$  和  $\beta$  是失效参数,  $\theta > 0, \beta > 0$ 。

### 3.3 服务生态系统多 Agent 仿真框架

根据服务生态系统中的基本结构和动态要素,可以构建生态系统的多 Agent 仿真框架,该框架包含服务 Agent(service agent)、服务生成器(service factory agent)、服务组合 Agent(composition agent)、服务组合生成器(composition factory agent)、服务用户 Agent(user agent)和用户生成器(user agent)等 Agent。3.1 节中介绍的结构模型代表了服务 Agent、服务组合 Agent 和服务用户 Agent 的基本属性,而在多 Agent 仿真中,随着仿真时钟的不断推进,系统中的 Agent 定期执行 Step 操作,由不同的 Step 操作反映系统中的动态变化。下面介绍不同 Agent 的属性和主要行为。

(1)服务生成器 服务生成器是服务生态系统模型中执行服务新增操作的 Agent,所包含的属性是服务生成列表(service generate list),其中记录了服务生成器所创建的服务 Agent 以及对应的创建时间;一个服务生态系统模型中仅需要一个服务生成器。在每次按照时间步刷新时,服务生成器的 Step 操作包括以下内容:

1)根据预设规则生成服务 Agent,对服务的 ID、名称、功能域进行初始化;

2)依据 3.2 节中的介绍,使用默认初始值或适应性方式,对新的服务 Agent 信誉初始值进行设定。

(2)服务 Agent 服务 Agent 代表系统中发布的服务,服务的属性如定义 1 的介绍,包括服务 ID、服务名称、服务功能域、服务存活时间、服务信誉值、使用记录和推荐记录等。在每次按照时间步刷新时,服务 Agent 的 Step 操作包括以下内容:

1)根据被推荐情况,更新服务推荐记录(rcmmd history);

2)根据被使用情况,参照式(1)~式(3)更新服务使用记录(use history)和服务信誉值(rpu value);

3)更新服务存活时间(live time)。

(3)服务组合生成器 服务组合生成器是服务生态系统模型中执行服务组合新增操作的 Agent,所包含的属性是服务组合生成列表(composition generate list),其中记录了服务组合生成器所创建的服务组合 Agent 以及对应的创建时间;一个服务生态系统模型中仅需要一个服务组合生成器。在每次按照时间步刷新时,服务生成器的 Step 操作包括

以下内容:

1)根据预设规则生成服务组合 Agent,对服务组合的 ID、名称、调用服务功能域类型、信誉值进行初始化;

2)根据服务功能域类型获取推荐服务列表,从列表选取服务,更新调用服务列表(ser list)。

(4)服务组合 Agent 服务组合 Agent 代表服务组合开发者在系统中发布的服务组合,服务组合的属性如定义 2 的介绍,包括服务组合 ID、服务组合名称、服务组合存活时间、服务组合信誉值、使用记录以及服务组合调用的服务列表等。在每次进行刷新时,服务组合 Agent 的 Step 操作包括以下内容:

1)根据用户的使用情况,更新服务组合使用记录(use history)和服务组合信誉值(rpu value);

2)更新服务组合存活时间(live time)。

(5)用户生成器 用户生成器用于在每个时间步内添加新的用户,并对用户本身的基本属性进行初始化,一个服务生态系统中仅需要包含一个用户生成器。用户生成器的 Step 操作包括以下内容:根据预设规则生成用户 Agent,对用户的 ID、创建时间进行初始化设定。

(6)用户 Agent 用户 Agent 代表系统中的服务用户,具体属性如定义 3 的介绍,包括用户 ID、创建时间、使用记录等。在每次按照时间步刷新时,用户 Agent 的 Step 操作包括以下内容:

1)根据预设规则的生成调用需求,以概率  $P_{ser}$  ( $0 \leq P_{ser} \leq 1$ ) 选择服务,以概率  $1 - P_{ser}$  选择服务组合,从服务(或服务组合)推荐列表中选择服务进行调用;

2)对调用的服务(服务组合)进行评价,评价价值  $UF$  用于更新服务(服务组合)信誉值;

3)更新用户的调用记录(use history)。

在实际仿真时,可以将用户 Agent 的反复调用操作简化成为不同 Agent 的单次操作,即每个用户 Agent 只在加入系统后发生一次调用行为,通过用户生成器反复创建新的用户 Agent,实现多次调用。

## 4 基于生态系统综合发展的服务推荐

### 4.1 服务生态系统综合指标

平台运营者是服务系统整体的管理者,他们关注服务平台整体发展的繁荣度,而繁荣度可以通过系统中的服务数量和服务种类来衡量,具有数量更

多、种类更丰富、不同种类中分布更加均匀的服务系统将更加繁荣,这也是平台运营者所追求的目标。

为了提升(或保障)当前系统中的服务种类,一方面,运营者可以通过服务关联推荐的方式,挖掘潜在的服务组合模式,为调用某些类型服务的顾客提供可选的合作服务,扩展服务被调用的渠道,避免某些服务种群由于受到关注较少而最终从系统中消失;另一方面,通过分析当前系统中的服务种群类型,可以提出服务创新的方向,作为建议提供给服务提供者,帮助他们更好地针对用户需求设计新的服务,促进系统中的服务种类提升。

在保证服务数量方面,平台运营者可以从两个角度进行调控:①创造公平的服务推荐环境,鼓励新服务的加入,避免新服务由于缺乏信誉积累而在推荐中受到歧视;②关注系统中由于长期不被用户调用而可能导致失效的服务,从全局出发进行推荐,降低服务的失效率,减小同一服务功能域中的服务推荐及用户资源分配的不均衡性。

针对服务关联推荐<sup>[22]</sup>和服务功能创新<sup>[23]</sup>,已经有许多学者开展了相关研究。本文将重点介绍从服务平台运营者的视角出发、关注公平性和全局性的服务推荐算法,而不对保障服务种类的推荐算法进行具体分析。

针对服务推荐过程中的公平性和全局性,本文提出以下关键指标:

#### (1) 基尼系数<sup>[24]</sup>

用户分配基尼系数(user gini coefficient)是国际上用来综合考察居民内部收入分配差异状况的一个重要分析指标,基尼系数越大,意味着分配的不均匀性越强。在服务系统中,用户资源是服务提供者最关注的资源,因此将用户调用次数类比为传统基尼系数计算中的居民收入,可以得到用户分配基尼系数 UGI 的计算公式如下:

$$UGI = 1 - \sum_{i=1}^x \left( \frac{|S_{u=i}|}{|S|} \right)^2. \quad (5)$$

按照被用户调用的次数,服务集合  $S$  被分为  $x$  个子集,其中子集  $S_{u=i}$  表示用户调用次数为  $i$  的服务所组成的集合,  $|X|$  表示集合  $X$  所对应的元素数量。

#### (2) 用户分配信息熵

用户分配信息熵(User Entropy, UE)的意义与基尼系数类似,通过计算所有被使用过的服务中用户分布的均匀性来描述整个系统的公平性特征。

其计算公式如下:

$$UE = - \sum_{i=1}^x \frac{|S_{u=i}|}{|S|} \log \left( \frac{|S_{u=i}|}{|S|} \right). \quad (6)$$

对于用户分配信息熵来说,越均匀的分配模式将会带来更高的用户分配信息熵取值。

#### (3) 服务推荐机会指数

服务推荐机会(Recommended Proportion, RP)指数用来考量系统中得到推荐的服务比例,如果系统中更多的服务得到展示机会,则说明系统的推荐机制对于服务来说具有更好的公平性。其计算公式如下:

$$RP = \frac{|RS|}{|S|}. \quad (7)$$

式中:  $RS$  为曾经获得推荐机会的服务所组成的集合,  $RP$  为曾经被推荐的服务占总服务的比例,  $RP$  值越高则具有更好的公平性。

#### (4) 服务使用机会指数

服务使用机会(Used Proportion, UP)指数用来描述系统中被使用的服务比例,更多的服务被使用,说明系统中的资源照顾到了更多的服务,也说明推荐方法在提升系统多样性发展方面具有较好的效果。其计算公式如下:

$$UP = \frac{|US|}{|S|}. \quad (8)$$

式中  $US$  为曾经获得推荐机会的服务所组成的集合。

#### (5) 服务数量及服务失效率

从全局出发的服务推荐方法力求保障现有服务的存活率,因此存活服务数量  $|AS|$  和服务失效率(Deprecated Porportion, DP)都是考量算法效果的关键指标,系统中存活的服务的集合为  $AS$ ,该集合的规模对应当前的存活服务数量  $|AS|$ ;服务失效率  $DP$  是系统中失效服务占服务总数的比例,通过失效服务数量  $|DS|$  和服务总数  $|S|$  进行计算,

$$DP = \frac{|DS|}{|S|}. \quad (9)$$

式中  $DS$  是已经失效的服务集合。

### 4.2 基于生态系统综合发展的服务推荐策略

在服务生态系统中,为了保证系统的繁荣度,需要关注新加入系统的服务,避免新加入的服务由于信誉记录和信誉值初始化的问题而无法得到关注。另一方面,为了减少系统中的不稳定因素、减少服务的失效率,需要关注系统中可能面临失效的服务。为此,下面结合 4.1 节中的服务生态系统发展综合

指标,提出基于生态系统综合发展的服务推荐策略。

由于当前大部分算法仅考虑了服务的功能、质量属性与用户需求的最大匹配,导致推荐资源向少部分服务过度倾斜的现象,如果希望能够在用户需求和服务匹配的基础上关注新加入系统的服务以及系统中面临失效的边缘化服务,则需要分别对普通服务、新服务和边缘化服务进行关注,分别使用不同的推荐队列生成规则,保证新服务和边缘化服务得到一定的展示机会。因此,本算法的核心思路是改变推荐过程中采用统一规则的方式,将服务划分为新服务集合、保护服务集合和普通服务集合,在推荐过程中,结合用户对服务的功能和质量需求,分别从这三个集合中筛选候选服务,然后将这些候选服务整合成为最终的推荐队列,由用户从推荐队列中选取服务。

下面详细介绍对本算法运行过程中涉及到的信誉值初始化、服务分类、服务队列生成和服务选择的问题。

#### (1) 服务信誉值初始化

对于新加入系统中的服务,首先需要进行服务初始化,初始化的方法包括按照默认初始值设定和按照适应性初始值设定两种。设置方式如 3.2 节所述。

#### (2) 服务分类

服务分类是本方法中的关键步骤。传统的服务推荐算法仅按照服务信誉值(或服务质量)进行排序和推荐,没有考虑新服务在初始进入系统时存在的信誉值偏低的问题,以及由于长时间未得到推荐服务而退出系统所带来的稳定性问题。下面分别定义新服务、保护服务和普通服务,以便对不同的服务采取不同的推荐策略。

**定义 4** 新服务集合。新服务集合用来组织新发布的服务。对于服务  $s$  来说,如果  $s$  加入系统的时间不大于时间阈值  $T'_N$ ,并且  $s$  的信誉值不大于信誉阈值  $R'_N$ ,则  $s$  被归入新服务集合  $S_N$ ,即  $S_N = \{s | s.LiveTime \leq T'_N \& s.RpuValue \leq R'_N\}$ 。

**定义 5** 保护服务集合。保护服务集合用来组织那些在当前系统中被边缘化、需要采取保护措施以保证留存率的服务,一般来说这些服务的信誉值相对较低,同时也不在新服务的集合范围内。对于服务  $s$  来说,如果它不在新服务集合中,同时当前时刻距离上一次调用的时间  $s.t_f$  大于时间阈值  $T_p$ ,则  $s$  被列入保护服务集合  $S_P$ ,即  $S_P = \{s | s.t_f \geq T_p \& s \notin S_N\}$ 。

在实际环境中,运营者也可以为保护服务集合

设定信誉值的下限,避免信誉值过低的服务进入推荐列表,影响用户对平台推荐算法的评价。

**定义 6** 普通服务集合。在服务集合  $S$  中,不属于新服务集合和保护服务集合的服务都被归为普通服务  $S_O$ ,即  $S_O = \{s | s \in S \& s \notin S_N \& s \notin S_P\}$ 。

需要注意的是,  $S_N \cup S_P \cup S_O = S$  且  $S_N \cup S_P \cup S_O = \emptyset$ 。根据不同的集合,将分别进行推荐服务的筛选及推荐队列的生成,随着服务信誉值的不断变化,服务所属的服务集合也随之进行更新。

#### (3) 推荐服务队列生成

在对服务集合进行分类后,分别从不同类别的集合中选择候选服务,假设一共需要生成  $k$  个候选服务,则分别从  $S_O, S_N$  和  $S_P$  中选取  $l, m$  和  $n$  个服务,其中  $l+m+n=k$  ( $l, m, n, k \in \mathbb{Z}^+$ ),具体的取值可以根据服务平台运营者当前的战略要求决定。在从  $S_O$  和  $S_N$  中选择服务时,分别选择信誉值最高的  $l$  和  $m$  个服务进行推荐,从  $S_P$  选取  $t_f$  值最高的  $n$  个服务放入推荐列表,设不同的服务集合  $S_O, S_N$  和  $S_P$  中生成的推荐服务列表为  $rS_O, rS_N$  和  $rS_P$ 。

#### (4) 推荐服务展示和选择

在生成三类服务的推荐列表后,服务平台的运营者需要将推荐服务展现给用户,在进行展示时,既可以将三类推荐列表混合成为总的推荐列表呈现给用户,也可以将保护服务集合和新服务集合作为特殊服务列表展现给用户,还可以将三类列表分别展示给用户。

在多列表的情况下,用户可以分别从不同的队列中查看推荐服务并选择。在模型仿真中,用户的选择过程可以用概率进行描述,假定用户按照不同概率首先选择服务列表,然后从服务推荐列表中选择候选服务使用。

(1) 单列表展现:将列表  $rS_O, rS_N$  和  $rS_P$  汇总成为一个大的服务推荐列表  $rS$ ,列表中按照服务的信誉值进行排序,用户从列表中进行选择。

(2) 双列表展现:将列表  $rS_N$  和  $rS_P$  汇总成为一个特殊服务列表  $rS'$ ,将特殊服务列表  $rS'$  与普通服务推荐列表  $rS_O$  分别提供给用户,各列表内部按照服务的信誉值进行排序,用户从列表中进行选择。

(3) 三列表展现:将列表  $rS_O, rS_N$  和  $rS_P$  分别提供给用户,各列表中按照服务的信誉值进行排序,用户按照不同的概率从各列表中选择候选服务,从三个列表中选择服务的概率分别为  $P_O, P_N$  和  $P_P$ ,且  $P_O + P_N + P_P = 1$ 。



## 5 实验设计及结果分析

### 5.1 实验数据集及实验设计

为了对推荐模式的效果进行检验并对推荐模式进行分析,本文基于 ProgrammableWeb 平台数据集设计了实验。数据集截取了 ProgrammableWeb 平台从 2005 年 12 月到 2012 年 8 月之间的服务及服务组合数据,其中共包含 7 076 个服务和 6 627 个服务组合。每一个服务都包含对应的名称、服务功能域、发布时间等数据,而服务组合则包括了名称、创建时间、所需要的服务功能域列表、信誉值等数据。

实验中重点考察新的推荐方法作用下系统综合指标的变化情况,因此,首先根据数据集中的服务和组合数据建立基本模型,然后将新的推荐指标施加于服务系统中,观察对应的指标并进行分析。根据系统中服务和组合发布的时间,整个系统被分为 86 个时间区间,每一个时间区间内都有新的服务和组合加入系统,新的服务会使用选定的方法进行信誉值的初始化,而新的服务组合将会根据所需要的服务功能域列表从制定功能域中获取推荐列表,并根据被推荐服务信誉值的大小随机选择服务。在完成服务调用后,根据式(1)和式(3)更新服务信誉值;随后,根据式(4)计算每个服务的失效概率,并计算服务是否失效,随后开始新一个时间步的仿真。在整个时间区间内仿真完成后,根据式(5)~式(8)计算服务系统发展的综合指标值。需要注意的是,因为本数据集中不涉及普通用户的基本数据,所以在实验中直接将服务组合视作调用服务的用户,忽略与终端用户相关的模型要素。

实验中对整个服务生态系统运行的流程进行建模,由于涉及到的变量较多,为了更加简洁地研究新的服务推荐策略给系统带来的影响,根据实际情况对主要变量做如下设定:

(1)服务信誉值更新公式(式(1))中的服务更新权重参数  $w=0.2$ ,意味着服务被新的服务组合调用后,新的信誉值对原有信誉值的加权强度为 0.2。

(2)服务信誉值衰减公式(式(3))中的衰减参数  $\lambda=0.005$ 。

(3)根据实际的服务失效曲线进行拟合,设定服务失效率公式(式(4))中的参数为  $\theta=4\ 500$ ,  $\beta=1.2$ 。

(4)服务信誉初始值设定采用两种模式,第一种为默认初始值模式,采用初始值  $R_{mi}=0.3$  和  $R_{mi}=0.7$  两种取值,分别代表保守型设定和鼓励型设定;第二种为适应性设定,根据当前系统中全局的信誉

平均值设定服务的初始信誉值。

(5)服务分类中的队列阈值参数设置。新服务的集合时间阈值  $T_N=15$ ,信誉值阈值根据初始信誉值设定的模式进行设置,  $R_N^t$  分别选取 0.5、0.9 和 0.7 以对应保守型设定、鼓励型设定和适应性模式。保护服务集合根据服务上一次被调用的时间进行设定,选取  $T_P=20$  作为保护服务移动的时间阈值。

(6)服务队列生成中,分别从新服务集合、保护服务集合和普通服务集合中选取对应的候选服务,候选服务数量  $l=m=n=10$ ,最终总的推荐服务数量  $k=30$ 。

(7)服务选择过程可以采用不同的方式进行,设定三队列模式下用户从普通服务集合中选取服务的概率为 0.6,从新服务集合和保护服务集合中选取服务的概率分别为 0.2;双队列模式下用户从普通服务集合中选取服务的概率为 0.6,从其他服务集合中选取服务的概率为 0.4。

为了进行推荐效果的对比,还需要进行不使用新的推荐模式下的对照组实验,对照组实验中总的推荐服务数量  $k=30$ ,其他设定与对应的实验组相同。实验仿真采用 Repast Simphony<sup>[25]</sup> 平台进行,每次仿真结果均为随机运行 50 次的基础上计算得到的平均值。

### 5.2 实验结果及分析

#### (1)推荐算法效果分析

在保守型设定、鼓励型设定和适应型设定情况下,分析新的推荐模式对于系统发展综合指标带来的影响。

表 1 是三种初始化模式下实验组与对照组的服务生态系统综合发展指标对比,通过观察表 1 的数据可以发现:不论哪种服务信誉值初始化模式,在使用新的推荐模式后,系统的基尼系数都减少了 8.0%~18.5%,而系统的用户使用信息熵则增加了 11.4%~23.2%,说明所有的被调用服务中,调用次数的分布变得更加均匀,系统内部的服务公平性更好;而就被推荐的服务数量和被使用的服务数量占比来看,服务推荐和使用比例的增长分别达到 73.5%~110.4% 和 76.5%~94.3%,说明不论是被推荐还是被使用,都有更多的服务被纳入考虑的范畴,系统更多地照顾到了整个服务群体的发展。AS 和 DP 两个指标表征了仿真结束时存活的服务数量和服务失效率,可以发现三种初始值情况下,服务失效率都在原来的基础上减少了 30%~40%,说明新的服务推荐算法明显改善了服务失效的情况,使得更多的服务

可以在系统中稳定存活。而对比三种不同的初始值设定的情况可以发现,在适应性模式设定初值

的情形下,当前的服务推荐机制效果不如采用默认值进行初始化的效果。

表 1 推荐算法效果分析

初始化模式	组别	UGI	UE	RP	UP	AS	DP/%
$R_{mi}=0.3$	对照组	0.812 8	0.701 9	0.225 6	0.131 7	634 5	10.33
	实验组	0.662 8	0.864 7	0.474 7	0.255 9	661 5	6.52
$R_{mi}=0.7$	对照组	0.817 1	0.700 1	0.225 8	0.130 7	628 9	11.12
	实验组	0.667	0.863 6	0.470 7	0.251 6	660 2	6.70
适应性设定	对照组	0.817 3	0.698 8	0.225 7	0.132 7	632 6	10.6
	实验组	0.752 3	0.778 7	0.391 7	0.234 2	654 3	7.53

## (2) 服务推荐队列分析

在采用默认信誉值进行服务初始信誉值设定的条件下,选取  $R_{mi}=0.3$  作为服务的初值,研究单队列、双队列、三队列等三种服务推荐队列展现模式下的推荐方法对最终性能的影响,结果如表 2 所示。

表 2 服务推荐队列性能分析

初始化模式	UGI	UE	RP	UP	AS	DP
单队列	0.583 3	0.908 1	0.478 2	0.301 4	6 682	5.57
双队列	0.662 8	0.864 7	0.474 7	0.255 9	6 615	6.52
三队列	0.687 8	0.840 0	0.470 2	0.215 1	6 604	6.67

通过比较各种队列模式可以发现,在各方面表现中,三种队列机制都比较接近,但是单队列模式要好于双队列模式和三队列模式。这是因为在当前的仿真模型中,面对多推荐队列的情形,用户首先根据一定的概率选定服务队列,随后从选定的服务队列中按照信誉值的高低随机选择服务,在多队列机制中所设定的用户对于普通服务队列的偏好值较高,一旦用户初始没有选择新服务队列或保护服务队列,在后面的选择中,新的服务或保护中的服务得到选择的机会就较少;如果采用单队列机制,用户则可能在整个服务列表中进行选取,无疑会增加新服务和保护服务被选择的概率。因此,在实际的服务生态系统运营中,如果运营者在生成推荐列表时不直接告知用户列表中的服务属性,则可以使本推荐策略取得更加显著的效果。

## 6 结束语

随着服务计算的发展,越来越多的服务生态系统开始在多种多样的服务平台上运转,服务提供者、服务、服务组合开发者、服务组合、服务平台运营者以及用户在平台上频繁交互,构成了一个复杂的生态系统。另一方面,由于多种动态要素不断作用于

服务生态系统,使得系统不断发生演进和变化,对于服务平台运营者来说,如何诱导服务生态系统的良性发展成为非常关键的问题。

本文针对当前服务推荐模式基本上仅考虑用户与服务需求的匹配度而忽略系统发展整体需求的现状,提出了平台运营者视角下的 Web 服务推荐策略。首先,结合服务生态系统中的基本结构和服务新增、服务失效、服务信誉更新等动态要素构建了服务生态系统模型,该模型不仅可以用于服务生态系统的建模,还可以用于服务系统仿真,研究系统在不同要素下的作用和发展;其次,结合平台运营者对于系统发展的保障服务多样性、减小服务失效率的需求,提出了服务生态系统发展的综合指标;再次,基于综合指标提出了相应的推荐策略,该策略在当前算法考虑服务功能匹配和服务质量匹配的基础上,将服务划分为新服务、保护服务和普通服务,从三类不同的服务集合中分别选取服务生成推荐列表。在基于 ProgrammableWeb 数据集构建的服务生态系统模型上进行仿真后发现,本文所提出的推荐方法对于提升服务系统的多样性、减少服务失效率具有显著效果。

由于服务生态系统中存在诸多动态要素,除了本文中所考虑的服务新增、服务组合新增和服务失效外,服务创新和服务质量变化也对系统的发展具有十分重要的影响,如何将这些因素与当前的模型进行有效结合,并提出全面考虑动态要素对系统作用下系统综合发展的服务推荐策略,是未来的研究中需要关注的问题。

## 参考文献:

- [1] PERREY R, LYCETT M. Service-oriented architecture[C]// Proceedings of 2003 IEEE Symposium on Applications and the Internet Workshop. Washington, D. C., USA: IEEE, 2003: 116-119.
- [2] HUANG Shuangxi, FAN Yushun, ZHAO Dazhe, et al. Web

- service based enterprise application integration[J]. Computer Integrated Manufacturing Systems, 2003, 9(10): 864-867 (in Chinese). [黄双喜, 范玉顺, 赵大哲, 等. 基于 Web 服务的企业应用集成[J]. 计算机集成制造系统, 2003, 9(10): 864-867.]
- [3] ZHANG Xiuwei, HE Keqing, WANG Jian, et al. A survey of personalized web service recommendation[J]. Computer Engineering & Science, 2013, 39(9): 132-140 (in Chinese). [张秀伟, 何克清, 王 健, 等. Web 服务个性化推荐研究综述[J]. 计算机工程与科学, 2013, 39(9): 132-140.]
- [4] SHAO Lingshuang, ZHOU Li, ZHAO Junfeng, et al. Web service QoS prediction approach[J]. Journal of Software, 2009, 20(8): 2062-2073 (in Chinese). [邵凌霄, 周 立, 赵俊峰, 等. 一种 Web Service 的服务质量预测方法[J]. 软件学报, 2009, 20(8): 2062-2073.]
- [5] MOU Yujie, CAO Jian, ZHANG Shensheng, et al. QoS oriented interactive Web service selection[J]. Computer Integrated Manufacturing Systems, 2006, 12(5): 788-794 (in Chinese). [牟玉洁, 曹 健, 张申生, 等. 面向服务质量的交互式 Web 服务选择[J]. 计算机集成制造系统, 2006, 12(5): 788-794.]
- [6] MUSAEV A, WANG D, CHIEN A C, et al. Landslide Detection Service Based on Composition of Physical and Social Information Services[C]//Proceedings of the 2014 IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2014.
- [7] VON DER WETH C, HEGDE V, HAUSWIRTH M. Virtual location-based services: merging the physical and virtual world [C]//Proceedings of the 2014 IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2014.
- [8] ZHANG M, LIU C, YU J, et al. A correlation context-aware approach for composite service selection[J]. Concurrency and Computation: Practice and Experience, 2013, 25(13): 1909-1927.
- [9] KUMARA B T G S, INCHEON P, OHASHI H, et al. Context-aware filtering and visualization of Web service clusters [C]//Proceedings of the 2014 IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2014.
- [10] ZHANG J, TAN W, ALEXANDER J, et al. Recommend-as-you-go: a novel approach supporting services-oriented scientific workflow reuse[C]//Proceedings of the 2011 IEEE International Conference on Services Computing. Washington, D. C., USA; IEEE, 2011.
- [11] CHRISTENSEN E, CURBERA F, MEREDITH G, et al. Web services description language(WSDL)1.1[Z]. 2001.
- [12] RICHARDSON L, RUBY S. RESTful Web services[M]. Sebastopol, USA; O'Reilly Media, Inc., 2008.
- [13] LIU L, LECUE F, MEHANDJIEV N, et al. Using context similarity for service recommendation[C]//Proceedings of the 2010 4th IEEE International Conference on Semantic Computing. Washington, D. C., USA; IEEE, 2010.
- [14] CHURCH J, MOTRO A. Efficient service substitutions with behavior-based similarity metrics [C]//Proceedings of the 2013 20th IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2013.
- [15] BARROS A P, DUMAS M. The rise of web service ecosystems[J]. IT Professional, 2006, 8(5): 31-37.
- [16] MA Keping, CHEN Lingzhi, YANG Xiaojie. Ecosystem diversity: concept, research contents and progress [C]//Proceedings of the 1st National Biodiversity Conservation and Sustainable Use Conference. Beijing: The Committee on Biodiversity of the Chinese Academy of Science, 1994 (in Chinese). [马克平, 陈灵芝, 杨晓杰. 生态系统多样性: 概念、研究内容与进展[C]//首届全国生物多样性保护与持续利用研讨会. 北京: 中国科学院生物多样性委员会, 1994.]
- [17] HUANG K, FAN Y, TAN W. An empirical study of programmable Web: a network analysis on a service-mashup system[C]//Proceedings of the 2012 19th IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2012.
- [18] LIU Y, NGU A H, ZENG L Z. QoS computation and policing in dynamic web service selection[C]//Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters. New York, N. Y., USA; ACM, 2004.
- [19] WANG Y, VASSILEVA J. A review on trust and reputation for web service selection[C]//Proceedings of the 27th International Conference on Distributed Computing Systems Workshops. Washington, D. C., USA; IEEE, 2007.
- [20] XIA Bofei, FAN Yushun, HUANG Keman. Prediction method of perishing services in Web service ecosystem[J]. Computer Integrated Manufacturing Systems, 2014, 20(8): 2060-2070 (in Chinese). [夏博飞, 范玉顺, 黄科满. Web 服务生态系统中消亡服务的预测方法[J]. 计算机集成制造系统, 2014, 20(8): 2060-2070.]
- [21] FERBER J. Multi-agent systems: an introduction to distributed artificial intelligence[M]. Boston, Mass., USA; Addison-Wesley Reading, 1999.
- [22] HUANG K, FAN Y, TAN W. Recommendation in an evolving service ecosystem based on network prediction[J]. IEEE Transaction on Automation Science and Engineering, 2014, 11(3): 906-920.
- [23] MOKARIZADEH S, KUNGAS P, MATSKIN M. Utilizing Web services networks for Web service innovation[C]//Proceedings of the 2014 IEEE International Conference on Web Services. Washington, D. C., USA; IEEE, 2014.
- [24] HUANG K, LIU Y, NEPAL S, et al. A novel equitable trustworthy mechanism for service recommendation in the evolving service ecosystem[C]//Proceedings of the Service-Oriented Computing. Berlin, Germany; Springer-Verlag, 2014.
- [25] NORTH M J, HOWE T R, COLLIER N T, et al. The repast symphony runtime system[C]//Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms. Argonne, Ill., USA; Argonne National Laboratory, 2005.

## 作者简介:

- 刘 轶(1988—),男,湖北洪湖人,博士研究生,研究方向:面向服务计算、服务生态系统的动态要素、服务失效、服务推荐, E-mail: yi-liu10@mails.tsinghua.edu.cn;
- 范玉顺(1962—),男,江苏扬州人,教授,博士,博士生导师,研究方向:工作流理论与技术、面向服务计算、企业建模与企业集成、通信作者, E-mail: fanyus@tsinghua.edu.cn;
- 黄科满(1987—),男,广东揭阳人,讲师,博士,研究方向:服务计算、Web 服务组合、服务推荐、社交网络分析、数据挖掘, E-mail: keman.huang@tju.edu.com.