面向全局社交服务网的Web服务聚类方法

陆佳炜 马 俊 张元鸣 肖 刚
(浙江工业大学计算机科学与技术学院 杭州 310023)

摘 要 现有的服务聚类方法主要关注服务功能属性或QoS属性,而没有考虑服务在网络中的社交属性,随着服务数量的急速增长,其面临着服务发现效率低等问题。为此,提出一种面向全局社交服务网(GSSN)的Web服务聚类方法。该方法将孤立的服务联结为一种全局社交服务网络,以挖掘服务间的社交相似度。首先,综合REST与SOAP服务,从服务描述信息、领域标签、QoS信息等层面对服务进行相似度计算。其次,结合服务在网络中的社交属性,利用GSSN算法对相似度计算结果进行聚类处理,以提高服务的发现效率。最后,对全局社交服务网进行可视化实现,以展现各服务在全局环境下的服务社交关系,并设计实验用于对提出的方法进行验证。

关键词 服务聚类,全局社交服务网,服务发现,服务可视化
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.03.032

Service Clustering Approach for Global Social Service Network

LU Jia-wei MA Jun ZHANG Yuan-ming XIAO Gang
(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract The existing service clustering approaches mainly focus on functionality or QoS attribute, and they are lack of considering the social attribute in services. The growing number of Web services brings about a series problems of reducing efficiency of service discovery. Thus, this paper proposed a new service clustering approach for global social service network which can connect the isolated service into a social network. First, the similarity of services is calculated according to descriptive information, tag of domain area and QoS attribute in REST and SOAP service. Second, similarity calculations are clustered by combining with social attribute to enhance the services' sociability on a global scale. At last, service visualization of global social service network is given to show the social relationships among realted services. The experimental result shows the effectiveness of the proposed method.

Keywords Service clustering, Global social service network, Service discovery, Service visualization

1 引言

Web服务作为一种潜在的分布式服务架构解决方案,在互联网上具有重大影响。随着云计算的兴起,各类Web服务层出不穷,极大地促进了服务计算领域的发展。然而,目前Web服务并没有发挥其应有的价值,截至2016年11月30日,在Web服务编程网站Programmable Web(PWeb)[1)]上发布的Web服务已经超过16000个,但是发布在服务组合系统中的Web服务不超过4000个[1]。许多已经发布的Web服务的使用效率低下,未能被用户更好地发现、组合及调用,这也为软件开发者有效发现和重用服务资源带来了极大的挑战。

造成以上现象的原因如下:

1)现有的服务描述语言,如WSDL,Web APIs,Ontology Web Language for Service(OWL-S),只将服务作为一个单独的服务孤岛来进行研究,并没有考虑服务与服务之间的社交关系,这导致服务的发现和组合变得十分困难。UDDI提供了一些服务分类系统,但这些分类标准并不统一且较为简单,无法保证所采用的分类方法能够正确反映服务的功能。对于发布在PWeb中的服务,服务消费者只能看到与该服务相关的文本描述信息,无法直接调用该服务,也未能了解其关联服务的组合情况。

2)大部分研究只考虑了服务的功能属性或非功能性属性,并没有考虑服务在网络中的社交属性。Web服务与其他相关服务进行组合才能发挥更强的作用,其功能属性与非功能属性应相互依存,以增强服务的整体表现。同时,引入服务的社交属性也能提高服务消费者的使用满意度,例如:通过学习该服务的历史社交关系以进一步改进服务发现的质量。

1) http://www.programmableweb.com

到稿日期:2016-12-02 返修日期:2017-04-16 本文受浙江省科技厅公益性技术应用研究项目(2014C33071,2014C31078),浙江省重大科技专项(2014C01048)资助。

陆佳炜(1981—),男,硕士,讲师,主要研究方向为云计算、软件复用,E-mail:viivan@zjut.edu.cn;**马 俊**(1991—),男,硕士生,主要研究方向为云计算;**张元鸣**(1977—),男,博士,副教授,主要研究方向为软件体系结构;**肖 刚**(1965—),男,博士,教授,主要研究方向为云制造等,E-mail:xg@zjut.edu.cn(通信作者)。

全局社交服务网[2]反映了服务之间的依赖关系，可用来支持服务发现、服务推荐。然而在全局社交服务网中，隐藏在服务之间的聚类信息还没有被充分挖掘，即若服务之间的依赖关系相似，即共同依赖着大部分相同的其他服务，则这些服务同属于一类的可能性较大。

将全局社交服务网与服务聚类技术相结合，预先对服务进行聚类，可缩小服务检索的空间和范围，提高服务的搜索能力，进一步挖掘出各个服务之间的关联关系。并且，将相似的服务聚类到一起，可缩小服务之间的对比范围。

为了有效地解决上述服务中存在的问题，本文提出了一种面向全局社交服务网的 Web 服务聚类方法，该方法的主要思想在于：

1）现有服务聚类方法大都针对 WSDL 文档或 OWL-S 文档等单一类型的服务描述文档，并且这些服务大都遵循 SOAP 协议，对通过自然语言文本描述的 REST 服务的关注相对较少[3]。综合 REST 服务与 SOAP 服务，从服务描述信息、领域标签、QoS 信息等层面对服务进行相似度计算，以提高服务的查找效率。

2）在 Internet 环境下，网络化软件以协作的方式组合而成，因此单个 Web 服务不是孤立存在的，而是作为服务群体的一部分。社交属性有利于消除领域障碍，即不同领域的 Web 服务也可能存在组合关系，为服务推荐奠定了基础。将全局社交服务网与聚类进行结合，利用服务社交属性可进一步提高聚类效果，同时能够支持更好的服务社交活动，为服务发现以及推荐提供依据；另外，基于该方法设计软件，可实现软件的可视化实现。

2　相关工作

目前，学术界已有许多对聚类和服务社交关系方面的研究，主要体现在以下几方面。

1）在服务聚类研究方面，文献[3]提出了一种面向主题的领域服务聚类方法，该方法在对服务进行领域分类的基础上，结合概率、融合领域特性的领域服务聚类模型 DSCM；同时，基于该模型提出了一种面向主题的聚类方法。文献[4]提出了面向领域标签辅助的服务聚类方法，该方法在建立 DT-WSC 服务聚类模型的基础上提高了聚类效果。Liu 和 Wong[5]从 WSDL 文档中提取了内容、上下文、主机名和服务名称 4 个特征，以便使用树遍历算法对 Web 服务进行聚类，通过归一化 Google 距离（NGD）来测量内容和上下文之间的相似性。

2）在服务社交关系研究方面，文献[2]提出通过构建全局社交服务网来实现更高 QoS 的服务发现，根据所提出的已连接的特定服务原则来构建全局社交服务网。文献[6]结合复杂网络来分析服务依赖网络的拓扑性质，如小世界、无标度以及社区结构等特性。文献[7]提出了一种面向服务 Petri 网模型及其结构化语义操作，针对服务的各种组合方式，根据所提出的组合算子来构建面向服务 Petri 网模型——扩展开放网。

但是在上述文献中，对聚类的研究只停留在服务的功能属性、QoS 属性或者领域标签属性上，并没有考虑服务的社交属性，而服务社交关系的研究多侧重于基于图论的理论研究。为此，本文提出了一种面向全局社交服务网的 Web 服务聚类方法，将全局社交服务网和聚类技术进行有效结合，利用社交属性来提高聚类的精度，并通过可视化技术予以实现。

3　方法框架

为方便讨论和理解，给出聚类所涉及的几个基础性概念。

定义 1（原子服务）　原子服务（Atomic Service，AS）指可被独立调用且功能不可再分的 Web 服务，可以用四元组 $AS=\{AS_{name},AS_{des},AS_{in},AS_{out}\}$ 来进行描述。其中，AS_{name} 表示 Service Name，描述 Web 服务的名称；AS_{des} 表示 Service Description，描述 Web 服务的文本信息描述，详细说明了 Web 服务的功能；AS_{in} 表示 Service Input，描述 Web 服务的输入信息；AS_{out} 表示 Service Output，描述 Web 服务的输出信息。

定义 2（服务描述模型）　服务描述模型（Service Describe Model，SDM）是对 AS 的定义和表达，包括功能属性（Function Attribute，FA）和非功能性属性（Quality of Service，QoS）。即 SDM＝{FA，QoS}，FA 和 QoS 所包含的详细属性分别如表 1、表 2 所列。

表 1　功能属性表

Table 1　Description of functional properties

功能参数	参数描述
服务名称	服务的名称，发布时指定
服务描述	服务的文字性功能描述
服务输入	对服务输入参数有关的信息进行描述，包括参数名称、参数类型、关联的本体信息等
服务输出	对服务输出参数有关的信息进行描述，包括参数名称、参数类型、关联的本体信息等
服务领域	服务所属的领域标签信息
服务来源	服务的发布机构、行业领域等信息

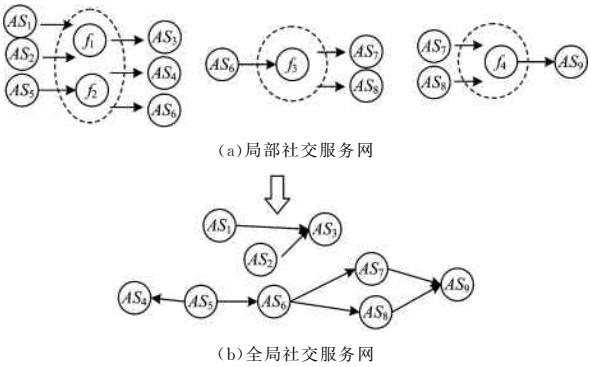
表 2　非功能属性表

Table 2　Description of non-functional properties

QoS 参数	参数描述
响应时间	服务从开始执行到调用结束所花费的时间
可用性	服务被成功调用的概率
吞吐量	单位时间内，服务能够处理的最大服务请求数量
可靠性	服务成功执行，返回正确结果的概率
价格	服务的价格
信誉度	用户对服务的评价

定义 3（全局社交服务网）　全局社交服务网（Global Social Service Network，GSSN）是一个开放的有向图 $GSSN=\{V,E\}$，由节点和有向边组成，用于描述服务的社交情况。其中每个节点代表一个 AS，每一条边代表 AS 之间的输入、输出参数的依赖关系，即前一个 AS 的输出参数中至少存在一个参数是后一个 AS 的输入参数的依赖。GSSN 表明了服务的社交状态并能为服务社交活动的推测、规划、协作提供依据。

图 1（a）给出了局部社交服务网，不同的服务通过映射关系 $f_i(i=1,2,\cdots)$ 相互组合，f_i 代表了不同服务之间的输入输出接口的依赖关系。如服务 AS_1 和 AS_2 通过映射关系 f_1 与服务 AS_3 组合，AS_1 的输出参数 a、AS_2 的输出参数 b 分别与 AS_3 的输入参数 c 和 d 相互依赖，即 $f_1=\{\langle AS_1.a,AS_3.c\rangle,\langle AS_2.b,AS_3.d\rangle\}$。通过关联各个局部的社交服务网，从而便组成了全局社交服务网，如图 1（b）所示。



(a)局部社交服务网

(b)全局社交服务网

图 1 全局社交服务网

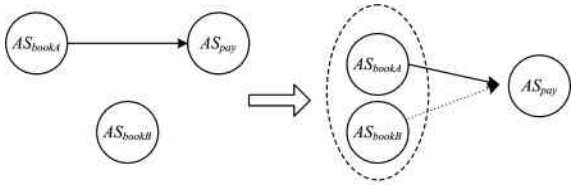
Fig. 1 Global social service network

万维网随着 Web 站点的发布和消亡能够不断地自我演化,并通过超链接关系来实现站点与站点间的互相发现。GSSN(如万维网)一般具有相似的演化特性,如随着新服务组合关系的产生,不断有新的 AS 及有向边更新到 GSSN 中。随着失效 AS 的消亡,其对应的节点及边关系也将从 GSSN 中移除。因此,将 GSSN 融入到服务研究工作中,有利于促进服务发现和服务组合。

定义 4(社交属性) 社交属性(Social Attribute,SA)指该服务与其他服务进行组合的能力及趋势,使用二元组 $SA=\{HSA,FSA\}$来描述。其中,HSA 和 FSA 分别代表历史社交域和未来社交域,HSA 指目前该服务所具备的服务组合能力,FSA 指未来该服务与其他服务进行组合的趋势。HSA 和 FSA 的具体定义如下。

定义 5(历史社交域(History Social Area,HSA)) 在 GSSN 中,从服务节点 AS_i 到服务节点 AS_j 的有向边记为 $\langle AS_i,AS_j\rangle$,其中,$AS_i,AS_j\in V$;HSA 定义为从 AS_i 出发且路径长度为 n 的所能到达的服务节点集合,记为 $HSA(AS_i)^n$,其中 n 为正整数,代表所经过路径的长度。如图 1(b)所示,服务节点 AS_6 的历史社交域可推导为:$HSA(AS_6)^n=HSA(AS_6)^1\cap HSA(AS_6)^2=\{AS_7,AS_8\}\cap\{AS_9\}$。

定义 6(未来社交域(Future Social Area,FSA)) FSA 是指在 GSSN 中,目前没有与 AS_i 建立社交关系,但通过聚类后可能与 AS_i 建立社交关系的服务节点集合,记为 $FSA(AS_i)$。如图 2(a)所示,有分别来自服务提供者 A 和 B 的预订服务 AS_{bookA},AS_{bookB},其中 AS_{bookA} 与支付服务 AS_{pay} 存在社交关系,而 AS_{bookB} 目前不存在类似关系。若 AS_{bookB} 与 AS_{bookA} 聚类后同属于一个服务簇,则表明未来 AS_{bookB} 存在与 AS_{pay} 发生社交关系的可能性,如图 2(b)所示,即 $AS_{pay}\in FSA(AS_{bookB})$。



(a)聚类前 (b)聚类后

图 2 未来社交域

Fig. 2 Future social area

研究服务的社交属性有助于在 GSSN 中更好地实现服务发现及推荐,了解该服务曾经关联的服务并进一步推导出将来与之关联的服务,获取该服务在 GSSN 中的社交地位等信息,如连接多个服务簇的枢纽节点具有跨簇传播功能,表示其具有较高的社会地位。

为了实现面向全局社交服务网的服务聚类,综合 REST 和 SOAP 服务信息进行服务注册,建立服务组合日志和服务运行 QoS 信息库,计算服务相似度并基于 GSSN 进行聚类;为了更好地展现聚类效果,最后对结果进行可视化分析。面向全局社交服务网的服务聚类方法框架如图 3 所示,其主要包括服务注册、服务运行信息采集、服务聚类、服务可视化 4 个模块。

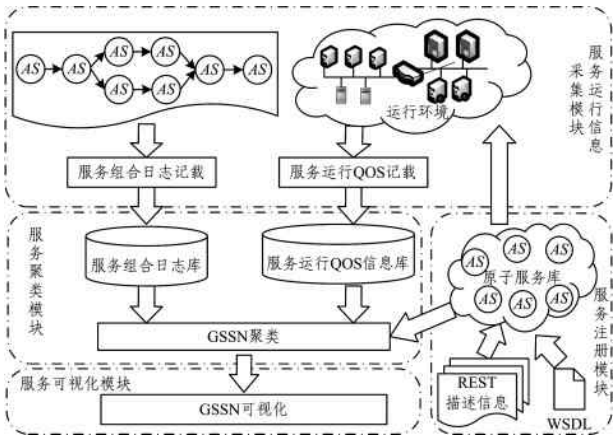


图 3 面向全局社交服务网的服务聚类方法框架

Fig. 3 Framework of service clustering method oriented to global social service network

1)服务注册模块:该模块实现 REST 服务及 SOAP 服务信息的注册,将原子服务注册到服务库中,为服务运行 QoS 采集、服务组合、服务聚类提供相关的服务资源。

2)服务运行信息采集模块:该模块主要采集原子服务的组合日志以及原子服务的运行 QoS 信息,并分别将其记录到服务组合日志库、服务运行 QoS 信息库中。

3)服务聚类模块:核心模块,主要结合原子服务库中的 AS、服务组合日志库、服务运行 QoS 信息库,采用 GSSN 聚类算法进行聚类,为服务可视化做准备。

4)服务可视化模块:该模块通过可视化操作界面,基于 GSSN,提供结合聚类的可视分析功能,辅助服务消费者更为直观地挖掘服务背后的隐藏信息。

4 聚类流程

4.1 流程解析

由于 Web 服务是由不同的组织机构开发的,带有很强的随意性,因此服务资源描述呈现多样化特征。如 PWeb 上的服务描述语言有 WSDL,OWL-S,WADL 等,而占服务总数最多的 REST 服务则采用自然语言来描述。服务规模的剧增和服务描述的多样化为用户准确、高效地发现服务资源增加了难度。针对上述问题,选取 REST 服务及 SOAP 服务作为研

究对象，图 4 展示了整个 GSSN 服务聚类的处理流程：首先将爬取的 REST 服务和 SOAP 服务注册到原子服务库中，提取出相关描述信息特征；其次，分别对服务进行功能相似度、Tag 相似度、QoS 相似度计算，其中 QoS 相似度来源于服务 QoS 信息库采集的信息；然后，生成综合相似度和相似矩阵；最后，进一步结合服务组合日志库采集到的服务组合信息生成 GSSN，利用 GSSN 聚类算法优化 GSSN 并实现聚类，为用户提供可视分析。

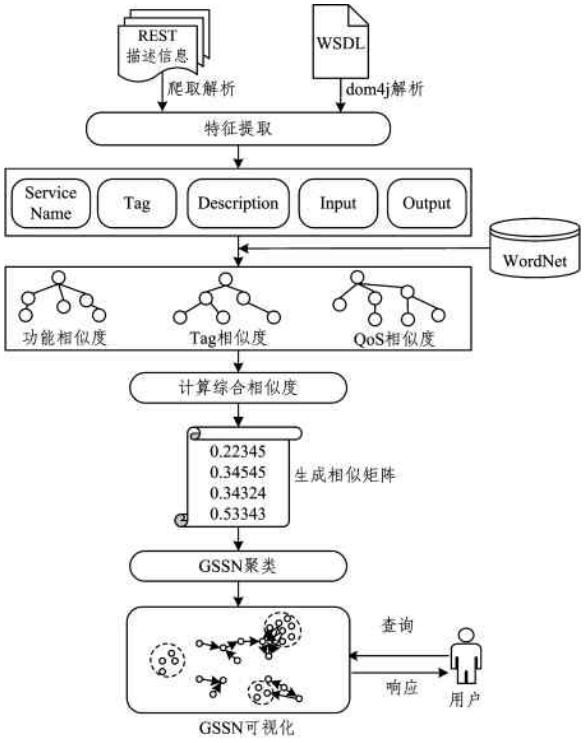


图 4　GSSN 聚类流程图

Fig. 4　Procedures of GSSN clustering

针对 REST 服务，PWeb 在开发者注册服务时允许开发者对所注册的服务添加标签，即添加与该服务所属领域、功能相关的标签，添加标签后的服务如图 5 所示。图 5 中的服务是一个天气查询服务，其中包含与该服务相关的描述信息及领域标签等信息。

图 5　天气服务相关描述信息

Fig. 5　Description of Weather Source API

PWeb 上的服务详情页满足特定的编码规则，如服务名称一般使用“h1”“header”等语义化 HTML5 标签表示，tag 标签使用 CSS 样式类“tags”指明。利用这些规则，建立爬取规则库，爬取 REST 服务的相关描述信息并进行特征提取，其中，特征提取算法如算法 1 所示。首先，遍历 REST 服务相关网页，将句子划分为词语，移除无意义的词语，如“a”“the”等。其次，建立表示输入、输出、标签、服务描述等规则的数据词典。进一步，在遍历文本过程中，若遇到如“output”“input”等数据词典中表征输入输出信息的词语，则对其后面的句子进行输入输出特征提取；若遇到如“provide”“allow”等数据词典中表征服务功能描述信息的词语，则对其后面的句子进行描述信息提取；若遇到如“category”“tags”等数据词典中表征标签信息的词语，则对其后面的句子进行 Tag 特征提取。最后返回 REST 服务特征信息。

算法 1　REST 服务特征提取算法

Input：REST Web Page

Output：REST Service Array

1. AS＝null，REST Service Array＝new Array；
2. create Service Feature Dictionary D；
3. for each word in REST Web Page
4.　delete meanless word such as “a”“the”；
5.　switch(word)
6.　　case(match D_{name}) extract AS_{name}；
7.　　case(match D_{des}) extract AS_{des}；
8.　　case(match D_{in}) extract AS_{in}；
9.　　case match (D_{out}) extract AS_{out}；
10.　　case(match D_{tags}) extract tags；
11.　end
11. insert AS into REST Service Array；
12. return REST Service Array.

针对 SOAP 服务，爬取 WSDL 描述信息后，基于 dom4j 技术对 XML 文档进行解析。由于一个 WSDL 中可能包含多个 *AS* 信息，因此提取出所有 *AS* 的服务名称、Tag 标记信息、服务的描述文本、输入输出参数。

SOAP 服务的 WSDL 解析算法如算法 2 所示。

算法 2　WSDL 解析算法

Input：WSDL

Output：SOAP Service Array

1. AS＝null，SOAP Service Array＝new Array；
2. for each Service AS in WSDL
3.　AS＝Service Name N_{AS}；
4.　calculate Tag via tf-idf；
5.　for each port p in AS
6.　　AS＝AS ∪ Service Location LAS；
7.　　get PortName，PortType；
8.　　for each Operation op in PortType
9.　　AS＝ AS ∪ OperationName OAS；
10.　　insert AS into SOAP Service Array；
11.　end
12.　end
13. end
14. return SOAP Service Array.

首先，获得 WSDL 的根节点，解析出所有的 Service 节点，获得每个 AS 的服务名称，根据 WSDL 中的服务描述信息提取词语，移除停用词，通过计算词频得到 Tag 标记信息，然后解析出每个 AS 的方法名及其对应的输入、输出参数。

其次，解析 REST 服务和 SOAP 服务得到的服务信息，结合服务运行时日志中的 QoS 信息分别进行 Web 服务相似性计算。相似性计算主要包括以下 4 个过程：

1)功能相似度计算；

2)领域标签相似度计算；

3)QoS 相似度计算；

4)综合前三步得到的结果，计算生成综合相似度。

在服务综合相似度集成后构造服务相似度矩阵，然后利用 GSSN 聚类算法对其进行聚类并可视化，为服务发现及推荐提供全局可视分析。

4.2 功能相似度度量

在服务的聚类过程中首先是功能聚类，即功能相同的服务被聚为一簇，而功能大多采用自然语言描述。语义 Web 服务是 Web 服务的扩展，能够更加准确地表达 Web 服务的功能含义，增强人与机器、机器与机器之间的交互性。通过语义描述，Web 服务成为了机器可读、可理解、可操作的实体，而 WordNet 是比较详尽的词语语义知识词典，用于度量不同词汇之间的语义相似度。当两个词汇的距离越大，其相似度越低；反之，若两个词汇的距离越小，则其相似程度越大。

如图 6 所示，概念 O_1 与 O_2 分别位于本体概念树的不同层次中，两者的语义相似度可用式(1)来度量。$Dis(O_1,O_2)$ 代表两个概念之间的距离，指 O_1 与 O_2 之间的最短路径长度，在图 6 中，$Dis(O_1,O_2)=3$。

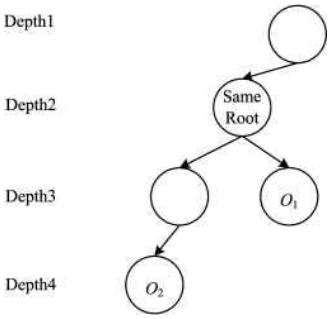


图 6 本体概念层次结构片段

Fig. 6 Section of ontology conceptual structure

$$sim_{onto}logy(O_1,O_2)=1-\frac{Dis(O_1,O_2)}{2(Depth-1)} \tag{1}$$

针对服务功能聚类问题，对特征提取后的服务名称、服务描述信息、服务的输入输出参数进行概念相似度计算，采用 WordNet 语义词典构建领域本体层次结构。Web 服务功能的相似度计算方法如下：

$$sim_{func}(As_i,As_j)=W_N\times sim_{name}(As_i,As_j)+W_D\times sim_{des}(As_i,As_j)+W_I\times sim_{in}(As_i,As_j)+W_O\times sim_{out}(As_i,As_j) \tag{2}$$

其中，Sim_{name} 表示服务名称相似性，Sim_{des} 表示服务功能信息描述相似性，Sim_{in} 表示服务输入匹配度，Sim_{out} 表示服务输出匹配度。W_N，W_D，W_I，W_O 分别表示为对应的权重，取值范围为 0～1。

4.3 领域标签相似度度量

领域标签信息也属于 Web 服务的功能性描述，如服务所属的领域、服务的来源等，这些标签信息能够有效提高服务聚类的精度及查询效率。

给定 Web 服务 AS_i 以及其对应的标签集合 T_i，根据 Jaccard 系数，计算出两个 Web 服务 AS_i 和 AS_j 之间的标签相似度：

$$sim_{tag}(AS_i,AS_j)=\frac{N(T_i\cap T_j)}{N(T_i)+N(T_j)-N(T_i\cap T_j)} \tag{3}$$

其中，$N(T_i\cap T_j)$表示同时拥有的标签数目。

4.4 QoS 相似度度量

现有的语义 Web 服务聚类方法主要从服务的功能属性出发，缺乏对 QoS 的考虑。随着服务数量的快速增长，以及由于服务质量的参差不齐，QoS 成为了用户在使用 Web 服务时考虑的重要指标。如何快速地从海量的服务中找到既能满足用户需求又具有最优 QoS 的服务是服务发现的研究重点。

根据 W3C[1] 于 2003 年给出的 13 个 Web 服务的 QoS 属性，选取其中便于度量的属性对 Web 服务的 QoS 进行度量，建立的 QoS 向量如下：

$$V_{QoS}=\{a_1,a_2,\cdots,a_n\} \tag{4}$$

其中，a_n 代表 QoS 属性，可分为连续型和离散型；连续型包括响应时间、服务价格等，离散型包括吞吐量、可用性、可靠性、信誉度等；n 代表可度量属性的个数。

考虑到不同的 a_n 取值范围有着很大的差别，如价格为 100 元，响应时间为 0.01s，需要对各个离性值进行标准化计算，将其都转化为[0,1]之间的数。

对于离散型，利用如下公式进行归一化计算。

$$a_n'=1-\frac{index}{num-1} \tag{5}$$

其中，num 代表 a_n 属性取值的个数，$index$ 代表 a_n 属性取值在所有离散取值范围中的索引。如服务的信誉度 QoS 属性，它由用户评价，评价等级的取值范围是{5,4,3,2,1}，当 a_n 为 3 时，其对应的索引位置为 2，则标准化计算后的值为 1−2/4=0.5。

对于连续型，利用最小-最大规范法进行归一化计算：

$$a_n'=\frac{a_n-min_{a_n}}{max_{a_n}-min_{a_n}} \tag{6}$$

其中，max_{a_n} 代表功能相同的服务簇中 a_n 属性的最大值，min_{a_n} 代表功能相同的服务簇中 a_n 属性的最小值。

接着，针对归一化计算后的 QoS 向量，计算两个 Web 服务的 QoS 相似度：

$$Sim_{QoS}(AS_i,AS_j)=\frac{V_i\cdot V_j}{|\vec{V_i}||\vec{V_j}|}=\frac{\sum_{k=1}^{n}a_{ki}a_{kj}}{\sqrt{\sum_{k=1}^{n}a_{ki}^2}\cdot\sqrt{\sum_{k=1}^{n}a_{kj}^2}} \tag{7}$$

4.5 综合相似度集成

综合原子服务的功能相似度、领域标签相似度、QoS 相似度，获得两个原子服务的综合相似度，其计算方法如下：

[1] http://www.w3c.or.kr/kr-office/TR/2003/ws-qos

$$sim(AS_i,AS_j)=\alpha * sim_{func}(AS_i,AS_j)+\beta * sim_{tag}(AS_i,AS_j)+\lambda * sim_{QoS}(AS_i,AS_j) \quad (8)$$

其中，α,β,λ 为权重，取值在 0 到 1 之间，根据综合相似度即可得到服务相似矩阵，为 GSSN 聚类做好准备。

4.6 GSSN 聚类算法

服务组合日志库中的信息代表着多个不同的局部社交服务网，记录着每个局部社交服务网中服务节点之间的连接关系。算法先通过将各个不同的局部社交服务网关联到一起以生成一个初始 GSSN，之后再结合聚类算法对 GSSN 进行进一步优化。

下面给出 GSSN 聚类所用到的基础性概念：

定义 7（初始 GSSN（Primitive Global Social Service Network，PGSSN））　PGSSN 由多个局部社交服务网关联推导形成，是 GSSN 的初始集合。

定义 8（强关系历史社交域（Strong Relation History Social Area，SRHSA））　对于 $HSA(AS_i)^n$，其社交关系的稳定性与所经过的路径长度成正比。n 越小，AS_i 与 $HSA(AS_i)^n$ 之间的社交关系越稳定；n 越大，所经过的服务节点数越多，当某一服务节点失效时，信息无法到达更远的服务节点，因此其社交关系越不稳定。当 n 为 1 时，定义 AS_i 的强关系历史社交域为 $SRHSA(AS_i)$，$SRHSA(AS_i)=HSA(AS_i)^1\subseteq HSA(AS_i)^n$。

定义 9（社交相似度（Social Similarity，SS））　两个服务节点 AS_i 和 AS_j 在自己所在 SRHSA 中所能达到的服务集重合度越高，表明这两个服务的社交相似度越大，属于同一服务簇的可能性也越大，记为 $SS(AS_i,AS_j)$。

$$SS(AS_i,AS_j)=\frac{|SRHSA(AS_i)\cap SRHSA(AS_j)|}{\sqrt{|SRHSA(AS_i)|\cdot|SRHSA(AS_j)|}} \quad (9)$$

定义 10（同簇服务（Same Cluster Service，SCS））　在 GSSN 中，如果服务 AS_i 与服务 AS_j 的社交相似度大于或等于社交相似度相似度阈值 ε，则定义服务 AS_i 与服务 AS_j 互为同簇服务。

$$SCS_\varepsilon(AS_i)=\{AS_j\in SRHSA(AS_i)\,|\,SS(AS_i,AS_j)\geqslant\varepsilon,\ \varepsilon>0\} \quad (10)$$

其中，ε 是用于划分同簇与非同簇的相似度阈值。当一个服务拥有较多的同簇服务时，认为其足够活跃，并将其定义为簇心服务，用于扩大服务簇。

定义 11（簇心服务（Cluster Center Service，CCS））　若服务 AS_i 的 SCS 个数超过某一临界值，则服务 AS_i 为簇心服务，定义为：

$$CCS_{\varepsilon,\mu}(AS_i)\Leftrightarrow|SCS_\varepsilon(AS_i)|\geqslant\mu \quad (11)$$

其中，$\mu(\mu>0)$ 用于判定簇心服务的阈值。

在 GSSN 中，由服务主动发起的关联服务与该服务自身的功能属性相关，而被动发起的关联服务则表明了其他服务对该服务的社交兴趣。物以类聚，人以群分，历史社交域相似的服务往往聚为一类，同属于一个服务簇，即若两个服务所能达到的服务集重合度越高，则两个服务属于同一簇的可能性越大。

GSSN 聚类算法在 K-means 聚类算法的基础上，结合服务在 PGSSN 中的社交属性，利用服务的社交相似度来进一步提高服务聚类的精度，同时利用 FSA 为服务推荐奠定基础。

GSSN 聚类算法分为 3 个阶段：

第 1 阶段　根据服务组合日志建立服务间的局部社交服务网，通过局部社交服务网的互相关联推导出 PGSSN，初始 PGSSN 建立的依据为历史数据，仅能展现已有的服务关系。

第 2 阶段　对于 PGSSN 中的 AS，统计 AS 的 SRHSA，计算社交相似度，根据社交相似度进行聚类以不断扩大服务簇。

第 3 阶段　对于原子服务库中的 AS，采用 K-means 算法，并基于综合相似度进行聚类后，根据服务簇间的相似度阈值将其划分至 PGSSN 中相似的服务簇中，融合后的新服务簇即为优化后的 GSSN。

GSSN 聚类算法的伪代码如算法 3 所示，具体步骤如下：

步骤 1　将现有的服务关系存储至服务组合日志库 L 中，$L=\{N,R\}$，其中 N 为各局部社交服务网节点的集合，R 为各局部社交服务网社交关系的集合，即对于任一局部社交服务网，其节点集合 $N_k=\{AS_1,AS_2,\cdots,AS_i\}$，社交关系集合 $R_k=\{AS_m,AS_n\in N_k\,|\,\langle AS_1,AS_2\rangle,\cdots,\langle AS_i,AS_j\rangle\}$，有 $N=\{N_1\cap N_2\cap\cdots\cap N_k\}$，$R=\{R_1\cap R_2\cap\cdots\cap R_k\}$。其中，$k$ 为正整数，代表局部社交服务网编号。由于现有的服务社交关系都记录在 L 中，因此读取 L 可推出各个局部社交服务网，并以此为基础生成 PGSSN。

步骤 2　遍历 PGSSN 中的 AS，统计每个 AS 的强关系历史社交域并计算社交相似度，得到同簇服务。如 $SRHSA(AS_1)=\{AS_2,AS_3,AS_4,AS_5\}$，$SRHSA(AS_6)=\{AS_2,AS_3,AS_4,AS_7\}$，则 $SS(AS_1,AS_6)=3/\sqrt{4*4}=0.75$。假设社交相似度阈值 ε 为 0.5，由于 0.75 大于 ε，因此 AS_1 与 AS_6 互为 SCS。

步骤 3　根据同簇服务，得到簇心服务，遍历所有簇心服务，将每个簇心服务的所有同簇服务聚为一个簇，根据簇中的簇心节点重复步骤 3，再次扩展服务簇，直到没有新的服务加入服务簇。

步骤 4　对于原子服务库中的 AS，根据综合相似度，采用 K-means 算法进行聚类，并将其划分至 PGSSN 中对应的服务簇中，把 PGSSN 优化为 GSSN。

算法 3　GSSN 聚类算法

```
Input：服务组合日志库 L，原子服务库 ASD，阈值 s
Output：聚类后的 GSSN 及各个 AS 的 FSA
1. L ⇒ PGSSN；//根据 L 生成 PGSSN
2. for each AS in PGSSN
3.     calculate SRHSA(AS)
4.     if(CCS_{ε,μ}(AS))
5.         tag AS as C_i//标记 AS 的簇编号为 C_i
6.         add all SCS_ε(AS) to queue Q
7.         while(Q.size! =0){
```

```
8.        Node m=Q.peek();//将 m 的同簇服务 n 加入到同一服
          务簇中
9.        for each Node n in SCSε(m)
10.         if (n 未聚类 ‖ !NCN)
11.           将 n 分为 Ci;
12.       Q.remove(m);
13.     }
14.   else AS 为无簇节点 NCN;
15. end for
16. for each AS in ASD
17.   使用 k-means 聚类 Cj;
18. If dist (Ci,Cj)>s
19.   合并 Ci,Cj 为一个新的簇;
20. end for
21. 输出聚类后的 GSSN 及各个 AS 的 FSA.
```

GSSN 聚类将在 PGSSN 中的服务划分为若干个子服务簇，使得具有相似社交相似度的服务归于同一簇，将在原子服务库中经 K-means 聚类后的服务划分至 PGSSN 中的相似服务簇。优化后的 GSSN 结合服务社交属性进行聚类，有助于提高聚类的准确度，并为服务组合和服务推荐提供依据。

5 实验验证

5.1 实验准备

为了验证所提出的面向全局社交服务网的 Web 服务聚类方法的有效性，基于课题组研发的“云通”服务平台来进行测试。该平台通过 JAVA 语言编程实现，可将爬取到的 SOAP 服务与 REST 服务进行注册，注册后能自动生成客户端代码并进行调用。“云通”支持原子服务的组合调用，能记录服务运行日志及相关 QoS 数据。其运行环境为：Inter 酷睿 i5 4200M 处理器，4GB 内存；操作系统为 Windows2008，MyEclipse10，JDK 1.7；数据库为 Mysql5.6+MongoDB3.0.3。

考虑到在实际应用中来自不同领域的服务数据源很难进行大范围组合，为了更好地展示不同服务之间的社交关系及 GSSN 聚类效果，实验采用的数据集取自 PWeb 爬取的服务，香港中文大学 WS-DREAM[1] 提供的服务，以及课题组参与的浙江省重大科技专项（特种设备云设计服务平台）中所实现的关于电梯、扶梯设计计算流程的服务。其中，SOAP 服务大约占 43%，REST 服务占 57%，每个领域的服务数量及其部分核心标签如表 3 所列。

表 3 服务数据

Table 3 Information of services

服务领域	服务数量	核心标签
Traffic	207	railway,bus,bike,tools
Financial	213	finance,trade,tax,payment
Weather	218	weather,rain,sunshine,cloud
Social	227	social,facebook,twitter
Cloudmanu	355	elevator,staircase,cloudmanu

[1] http://wsdream.github.io

[2] http://js.cytoscape.org

5.2 实验结果

通过“云通”服务平台，首先将爬取到的服务注册到平台上，考虑到某些服务的描述信息比较少，会影响服务间的相似度计算，在注册时对这类服务的功能属性进行适当补充，如对部分描述简单的领域标签进行扩充。注册完毕后，平台生成客户端程序并自动测试服务，收集服务的运行日志和 QoS 信息。之后，平台从服务描述信息、领域标签、QoS 信息等层面对服务进行相似度计算，获取综合相似度。最后，基于 GSSN 聚类算法将计算结果进行处理，划分服务簇，以展现各服务在全局环境下的服务社交关系。

实验结果如图 7 所示，可视化模块采用 Cytoscape 技术[2] 来实现，Cytoscape 基于 JavaScript 语言实现，能在网页上呈现动态的网络图。“云通”服务端调用服务后将计算结果以 JSON 的格式注入到前端页面中，并将其作为 Cytoscape 的初始参数，Cytoscape 基于该输入值进行网络图的动态绘制和可视化交互。

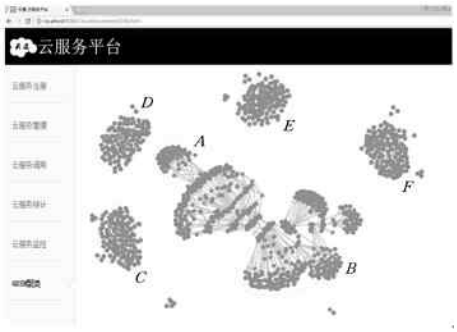


图 7 GSSN 聚类可视化

Fig. 7 Visualization of GSSN clustering results

采用 GSSN 聚类算法后的服务簇分别用 A,B,C,D,E,F 在图 7 中进行了标记。簇 A 和簇 B 分别是扶梯、电梯设计计算过程中某类功能的服务簇，因为这些 AS 有着相似的 SRHSA，所以其社交相似度也接近，被聚为一个服务簇。对于未在 GSSN 中的服务节点，依据综合相似度进行 K-means 聚类后，根据相似度阈值将经过聚类后的服务簇与 GSSN 的服务簇进行比较，最后将相似的服务簇划分至 GSSN 的服务簇中，形成簇 A 和簇 B。

图 8 展现了簇 A 放大后的局部效果图，簇 A 中的服务用于计算扶梯扶手带驱动力。以 handrailForceS0 和 handrailForceS2 为例，采用 K-means 等聚类算法聚类时，离群点的存在会影响聚类精度，而这两个服务具有相似的 SRHSA，社交相似度较高，GSSN 聚类算法可利用社交相似度将离群点重新划分至对应服务簇中，提高聚类精度。没有边与服务 handrailForceS1 相连，表明其位于原子服务库中，尚未被组合，因此根据相似度阈值将其划分至簇 A，当 handrailForceS0 失效时，可推荐 handrailForceS1 来替换，同时，handrailForceS0 现有的 SRHSA 可作为 handrailForceS1 的 FSA，为 handrailForceS1 服务组合进行推荐。因此，将 GSSN 与聚类进行

融合，结合服务的社交属性，利用社交相似度，不仅有利于提高聚类的精度，而且当簇 A 中存在社交关系的节点失效时，有助于从簇 A 中选取目前没有社交关系但功能相似的服务来替换失效的服务，为服务发现和服务推荐提供依据。

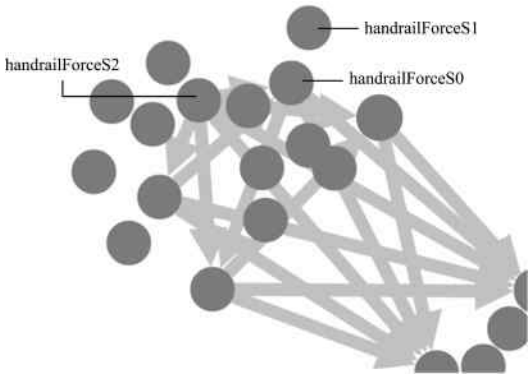


图 8　簇 A 局部放大的效果图

Fig. 8　Partial enlargement of cluster A

簇 C，D，E，F 分别代表 Traffic，Financial，Weather，Social 这 4 个领域。由于这 4 个领域相对独立，只有少部分 AS 在自己领域内有社交关系，这少部分节点因社交相似度而被聚类到一起。再根据综合相似度，将相似的服务节点划分至对应的服务簇，利用社交相似度可以提高聚类的准确性，同时为服务组合和服务推荐提供依据。

5.3　实验评价

实验 1　聚类效果分析

将聚类精度（*Precision*）、召回率（*Recall*）和 F 值（*F-measure*）作为聚类结果评价的标准。聚类精度和召回率广泛用于信息检索领域，衡量检索系统的查准率和查全率。将实验聚类后的簇 C_i 与人工分类的簇 M_j 进行对比，精度越高，表明聚类结果越好。

$$P(C_i,M_j)=\frac{|C_i\cap M_j|}{|C_i|} \tag{12}$$

$$R(C_i,M_j)=\frac{|C_i\cap M_j|}{|M_j|} \tag{13}$$

F-measure 是聚类精度 P 和召回率 R 的调和平均数。

$$F\text{-}measure=\frac{2*P(C_i,M_j)*R(C_i,M_j)}{P(C_i,M_j)+R(C_i,M_j)} \tag{14}$$

K-means 聚类算法是一种应用最广泛的基于划分的聚类算法，文献[8-9]采用该方法对服务进行聚类。Agglomerative 算法是一种自下而上的层次聚类算法，文献[10-11]采用该方法进行聚类分析。使用 K-means 算法、Agglomerative 算法与本文聚类方法进行对比的结果如图 9、图 10 所示。

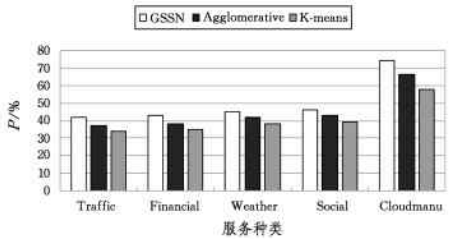


图 9　聚类精度的比较

Fig. 9　Comparison of clustering precision

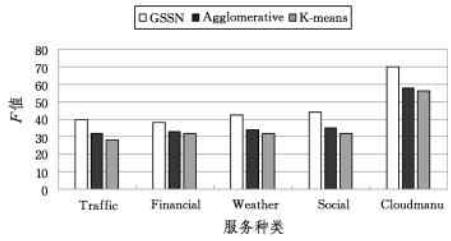


图 10　F 值的比较

Fig. 10　Comparison of *F-measure*

实验结果表明，相比于传统的 K-means 聚类和 Agglomerative层次聚类算法，GSSN 聚类算法可以有效提高聚类精度，尤其当位于全局社交服务网的服务节点数量越多时，每个服务的社交能力越强，聚类精度越高。图 9、图 10 展现了聚类精度和 F 值的实验结果，由于 Traffic，Financial，Weather，Social 这 4 个领域相对独立，只有少部分服务在自己领域内有社交关系，且其召回率较低，使得 F 值也较低；采用 GSSN 聚类算法后，聚类精度只提高了 1 到 3 个百分点。而 Cloudmanu 领域由于有 300 个以上的服务展现在 GSSN 中，并且有大量的协作服务来完成机械零部件产品的分析与设计计算功能，其对应的社交能力也表现得更强，在综合社交相似度计算后，结合 GSSN 聚类算法，最终的聚类精确度提高了 8 个百分点左右。我们相信随着 GSSN 中服务数量的提升和服务社交能力的提高，本方法能更有效地改善聚类效果。

实验 2　聚类时间分析

图 11 给出了 3 种聚类算法针对 5 个领域的聚类时间。

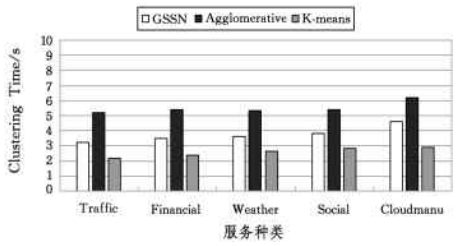


图 11　聚类时间的比较

Fig. 11　Comparison of clustering time

从图 11 中可以看出，K-means 算法消耗的时间最少，GSSN 算法次之，Agglomerative 算法所需时间最长。GSSN 算法比 K-means 算法所耗时间更长是因为 GSSN 算法是在 K-means 算法的基础上结合服务社交属性进行改进，利用社交相似度重新划分服务簇来提高聚类精度。K-means 算法的时间复杂度为 $O(n*k*t)$，其中，n 为服务个数，k 为类别个数，t 为迭代次数。服务簇划分过程需遍历 GSSN 有限个数的节点，遍历过程中得出节点的 SRHSA，从而利用 SS 进行聚类，聚类时间复杂度为 $O(n)$。因此，GSSN 聚类算法的时间复杂度为 $O(n*k*t)+O(n)$，近似于 K-means 算法的时间复杂度。

结束语　Web 服务的大量涌现使得用户面临着服务发现困难的问题，如何快速、准确和高效地发现满足用户需求的 Web 服务是现阶段亟需解决的关键问题之一。基于此，本文提出了一种面向全局社交服务网的 Web 服务聚类方法，对

REST 服务和 SOAP 服务提取相关特征，从服务描述信息、领域标签、QoS 信息等层面对服务进行相似度计算，进一步结合服务的社交属性，采用 GSSN 算法将计算结果进行聚类处理，划分服务簇，以展现各服务在全局环境下的服务社交关系。最后，以 PWeb 等系统上真实的服务集进行实验，实验结果显示该方法能够有效改善 Web 服务聚类效果及提高服务聚类的精度，具有较好的实际应用价值。

在接下来的工作中，我们将在现阶段研究的基础上进一步完善软件平台。在理论上，也还有许多问题有待于进一步解决。例如，在动态环境下，还需研究基于聚类后的全局社交服务网实现最优 QoS 的服务发现。此外，当某个服务无法正常工作时，在 GSSN 环境下，如何实现相同服务簇内最优服务的自动推荐也将是今后的研究重点。

参 考 文 献

[1] AL-MASRI E, MAHMOUD Q H. Investigating Web Services on the World Wide Web[C]// International Conference on World Wide Web(WWW 2008). Beijing, China, 2008: 795-804.

[2] CHEN W, PAIK I, HUNG P C K. Constructing a Global Social Service Network for Better Quality of Web Service Discovery [J]. IEEE Transactions on Services Computing, 2015, 8(2): 284-298.

[3] LI Z, WANG J, ZHANG N, et al. A Topic-Oriented Clustering Approach for Domain Services[J]. Journal of Computer Research and Development, 2014, 51(2): 408-419. (in Chinese)
李征，王健，张能，等. 一种面向主题的领域服务聚类方法[J]. 计算机研究与发展，2014，51(2)：408-419.

[4] TIAN G, HE K Q, WANG J, et al. Domain-Oriented and Tag-Aided Web Service Clustering Method[J]. Acta Electronica Sinica, 2015, 43(7): 1266-1274. (in Chinese)
田刚，何克清，王健，等. 面向领域标签辅助的服务聚类方法[J]. 电子学报，2015，43(7)：1266-1274.

[5] LIU W, WONG W. Web service clustering using text mining techniques[J]. International Journal of Agent-Oriented Software Engineering, 2009, 3(1): 6-26.

[6] CHERIFI C, LABATUT V. Web Services Dependency Networks Analysis[C]// International Conference on New Media and Interactivity. 2010: 115-120.

[7] GUO F, WEI G, DENG M M, et al. Service Oriented Petri Net Model and It's Structural Operational Semantics[J]. Journal of Chinese Computer Systems, 2013, 34(12): 2739-2743. (in Chinese)
郭峰，魏光，邓蒙蒙. 一种面向服务 Petri 网模型及其结构化操作语义[J]. 小型微型计算机系统，2013，34(12)：2739-2743.

[8] WANG X, WANG Z, XU X. Semi-empirical Service Composition: A Clustering Based Approach[C]// IEEE International Conference on Web Services (ICWS 2011). Washington DC, USA, DBLP, 2011: 219-226.

[9] CHEN L, WANG Y, YU Q, et al. WT-LDA: User Tagging Augmented LDA for Web Service Clustering[M]// Service-Oriented Computing. 2013: 162-176.

[10] KUMARA B T, PAIK I, KOSWATTE K R. Ontology learning with complex data type for Web service clustering[C]// IEEE Symposium on Computational Intelligence and Data Mining (CIDM). IEEE, 2014: 129-136.

[11] NAYAK R, LEE B. Web Service Discovery with additional Semantics and Clustering[C]// IEEE/WIC/ACM International Conference on Web Intelligence. IEEE, 2007: 555-558.

[12] WU J, CHEN L, ZHENG Z, et al. Clustering Web services to facilitate service discovery[J]. Knowledge & Information Systems, 2014, 38(1): 207-229.

[13] XIE L L, CHEN F Z, KOU J S. Ontology-based semantic web services clustering[C]// 2011 IEEE 18Th International Conference on Industrial Engineering and Engineering Management (IE&EM). IEEE, 2011: 2075-2079.

[14] KUMARA B T G S, YAGUCHI Y, PAIK I, et al. Clustering and Spherical Visualization of Web Services[C]// IEEE International Conference on Services Computing. IEEE Computer Society, 2013: 89-96.

(上接第 177 页)

[7] PU K Q, ZHU Y. Efficient indexing of heterogeneous data streams with automatic performance configurations[C]// Proceedings of Scientific and Statistical Database Management Conference. 2007: 34-34.

[8] JIN C Q, QIAN W N, ZHOU A Y. Analysis and Management of Streaming Data: A Survey[J]. Journal of Software, 2004, 15(8): 1172-1179. (in Chinese)
金澈清，钱卫宁，周傲英. 流数据分析与管理综述[J]. 软件学报，2004，15(8)：1172-1179.

[9] LU H, NG Y Y, TIAN Z. T-tree or B-tree: Main memory database index structure revisited[C]// Proceedings of Australian Database Conference. 2000: 65-73.

[10] BERCKEN J, SEEGER B. An evaluation of generic bulk loading techniques[C]// Proceedings of VLDB Conference. 2001: 461-470.

[11] CIACCIA P, PATELLA M. Bulk loading the M-tree[C]// Proceedings of Australasian Database Conference. 1998: 15-26.

[12] LO M L, RAVISHANKAR C V. The design and implementation of seeded trees: An efficient method for spatial joins[J]. IEEE TKDE, 1998, 10(1): 136-152.

[13] ARGE L, HINRICHS K H, et al. Efficient bulk operations on dynamic R-trees[J]. Algorithmica, 2002, 33(1): 104-128.

[14] JERMAINE C, DATTA A, OMIECINSKI E. A novel index supporting high volume data warehouse insertion[C]// Proceedings of VLDB Conference. 1999: 235-246.

[15] KHOLGHI M, KEYVANPOUR M R. Comparative evaluation of data stream indexing models[J]. International Journal of Machine Learning and Computing, 2012, 2(3): 257-260.

[16] SHIVAKUMAR N, GARCIA-MOLINA H. Wave-indices: indexing evolving databases[C]// Proceedings of SIGMOD Conference. 1997: 381-392.