基于概念松弛的高效 Web 服务查询方法

欧伟杰[1] 曾 承[1),3)] 项小明[2)] 彭智勇[2)] 李德毅[1),4)]

[1)](武汉大学软件工程国家重点实验室 武汉 430072)

[2)](武汉大学计算机学院 武汉 430072)

[3)](清华大学软件学院 北京 100084)

[4)](中国电子工程系统研究所 北京 100141)

摘 要 随着云计算技术的发展,面向服务的应用在互联网上呈现快速增长趋势,开放平台中基于云服务的组合服务也如雨后春笋般大量涌现,这给用户快速、精确定位所需服务带来了巨大挑战.尽管传统服务查询方法在查全率和查准率方面已取得较大进步,但仍无法适用于动态的互联网环境下大规模服务发现的要求.文章根据概念之间的语义关系,提出了基于概念松弛的相似性服务查询方法,它通过计算无关概念与服务对查询结果的影响,不仅改善了服务查询的效果,而且满足海量服务查询的高效性要求.经实验证明,文中提出的方法不仅在性能上优于传统方法,且满足服务查询的可扩展性.此外,该方法已经应用于上线的按需服务平台中.

关键词 服务计算;服务发现;语义相似性;二分图匹配

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2011.02381

Efficient Web Service Query Approach Based on Concept Relaxation

OU Wei-Jie[1] ZENG Cheng[1),3)] XIANG Xiao-Ming[2)] PENG Zhi-Yong[2)] LI De-Yi[1),4)]

[1)](*State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)

[2)](*School of Computer Science*, *Wuhan University*, *Wuhan* 430072)

[3)](*School of Software*, *Tsinghua University*, *Beijing* 100084)

[4)](*Institute of Electronic System Engineering of China*, *Beijing* 100141)

Abstract Cloud computing and service-oriented applications on the Internet are growing rapidly. On the other hand, open platform speed up the emergence of composited services, which is based on Cloud service. How to discover the desired services for users efficiently has become a significant challenge. Although traditional approaches have made progress in recall rate and precision. But they are not suitable for large-scale service discovery under dynamic environment. In this article, a novel approach for service query based on concept relaxation is proposed, which employs the semantic relation of concepts from hierarchical ontology. The unrelated concepts and services are figured out to improve the efficiency of algorithm. The method has been implemented in a prototype of on-demand service platform. The results of experiments illustrate that the proposed approach not only outperform the traditional ones, but also satisfy the scalability of service discovery.

Keywords service computing; service discovery; semantic similarities; bipartite graph matching

---

收稿日期:2011-06-26;最终修改稿收到日期:2011-11-01.本课题得到国家"九七三"重点基础研究发展规划项目基金(2007CB310806)资助.**欧伟杰**,男,1981 年生,博士研究生,研究方向为服务计算.E-mail: oweijie@gmail.com. **曾 承**,男,1978 年生,博士,副教授,研究领域为服务计算、信息推送、信息检索. **项小明**,男,1987 年生,硕士研究生,研究方向为 Web 服务. **彭智勇**,男,1963 年生,博士,教授,博士生导师,研究领域为 Web 数据管理、复杂数据管理、可信数据管理. **李德毅**,男,1944 年生,博士生导师,中国工程院院士,主要研究领域为复杂网络、数据挖掘、人工智能.

1 引 言

云计算已从概念逐步走向成熟,而 Web 服务作为云计算实现的关键技术受到了越来越多的关注.面向不同领域的具体服务可以满足个人用户的日常需求,具体服务之间的互操作可以为企业建立各种新颖的业务流程,实现按需组合.服务作为一种软件资源为企业和个人用户带来了巨大的便利,但随着互联网上服务数量的快速增长,传统的服务查询方法已不能满足用户的需要.如何实现面向 Web 服务的高效查询已成为一个严峻的挑战,引起不少研究者的关注[1].

传统服务发现无论是 UDDI(Universal Description Discovery and Integration)还是搜索引擎,都是通过关键字查询实现.但调研显示绝大部分服务描述仅包含 30～40 个词语[2],因此返回结果可能出现两种情况:当搜索关键字较为特殊时,由于没有包含关键字的服务而导致查全率低;当关键字为常用词时,返回大量假阳性的结果致使查准率不足.虽然也有部分研究通过信息检索技术(向量空间模型,tf-idf 等)改善这种情况,但对于服务描述这类短文本,实际效果并不理想.于是语义方法被引入服务计算,通过领域本体中的语义信息建立需求与服务之间的桥梁.面向语义服务描述语言(OWL-S,WSMO 以及 SAWSDL 等),出现了基于概念匹配度的服务发现方法.这类方法可以在很大程度上改善服务发现的性能,但计算代价较高,不适合大数量级服务的应用.另外,基于语义的方法限于特定领域之内,且本体构建和基于本体的服务描述扩展均需要手工完成,也是这类方法无法广泛使用的原因.

目前,为了更好地适应 WSDL 服务,摆脱领域束缚,众多基于概念相似性的服务查询方法被提出来[3-10].核心思想是通过相似性衡量服务与需求的匹配程度,其中概念的相似性计算依靠外部信息(如通用层次本体①[11]、搜索引擎[12])实现,这些方法在实验中均得到较高的查准率和查全率.但应用于实际互联网时,仍存在以下不足:

(1)计算概念的语义距离时,并未考虑不同概念之间的语义关系,如 is-a、part of 等;

(2)无法支持大规模服务应用,大部分方法计算单个服务相似性的响应时间在百毫秒级别;

(3)部分方法通过预计算来提高算法的整体效率,但当外部信息频繁变化时,预计算带来巨大的计算开销;

(4)无法适用于组合服务的查询.组合服务根据组合方法的不同有着不同的描述,面向具体服务的查询方法将失效.

为了解决上述问题,本文提出了基于概念松弛的两段式服务查询方法.在概念相似性计算时考虑了泛化和特化两种不同的策略,充分利用了概念之间的语义关系;基于概念映射的相似性查询算法,支持动态环境下的及时更新.我们主要的贡献主要表现在以下 3 点:

(1)提出了基于层次本体的概念松弛算法,该方法不仅根据概念之间语义关系设计了不同的松弛策略,而且能够适应于实时领域切换、领域本体更新等动态环境;

(2)基于概念映射的服务查询方法,较传统方法有更高的计算效率,且能够在无预计算情况下实现实时语义查询.

(3)在原型系统中实现了面向接口的服务相似性查询方式,适应具体服务与组合服务混合的查询;

本文第 2 节介绍服务查询的相关研究工作;第 3 节将分析相似性服务查询存在的问题,并给出基本概念以及定义.两段式服务查询方法将在第 4 节详细说明,包括方法的整体流程、概念松弛以及相似性计算;第 5 节是针对本文提出方法的具体实验和结果分析.

2 相关研究

服务发现方法从最初基于 UDDI 的关键字查询发展到面向语义服务的概念匹配方法,虽然查全率和查准率均有一定提升,但本体的构建、服务描述的语义扩展都需要大量人工参与,无法广泛应用.而基于相似性的服务查询借鉴了语义服务发现中概念匹配的思想,通过概念间的语义距离计算用户需求与服务之间的语义相似度,引起了不少研究者的关注.这方面的工作主要可以分为 3 类:基于实例的查询、二分图匹配以及预计算聚类服务,也有部分方法同时运用以上多种策略.

基于实例的服务查询将用户请求以 WSDL 表示,通过相似性反映服务对需求的满足程度.最具代表性的工作是 Woogle[3] 和 URBE[6].Woogle 中所提出的方法首先聚类服务集合中的概念,根据不同

① Wordnet,http://wordnet.princeton.edu/

语义的概念类分别计算 Web 服务之间的 3 个相似性：输入/输出参数相似性、操作描述的相似性、服务名及描述相似性. 通过实验确定 3 个相似性的权重，最后得到 Web 服务整体的相似性结果. 由 Pierluigi 等提出的 URBE，将相似性计算分为两部分：语义相似性和语法相似性. 其中语义方面与该 Web 服务所要完成的功能有关，表现为整个服务描述、操作以及参数的名称. 语法方面与该服务的输入、输出接口之间的关系以及接口的数据类型相关. 其中值得注意的是，在语义相似性计算时也利用了二分图匹配，作者还讨论了两个接口之间相似性的不对称性，这种不对称来自于两接口的概念数量不一致.

基于二分图匹配的服务查询是通过概念集合来表示服务，首先得到概念之间的语义距离，通过二分图建模用户需求和具体服务，最大权匹配[13]的结果反映了服务与请求的相符程度. 浙江大学吴朝晖等提出的方法[9]关注点在于服务发现，通过服务、操作以及接口参数三个层次讨论需求与服务的关系. 其中接口参数相似性计算中利用了二分图匹配，并加入了输出与输入的依赖关系. 但对于最大权匹配应用于不平衡二分图的情况，仅通过添加虚拟结点处理. 文献[7]中指出 WordNet 不如搜索引擎所支持的概念广泛，而且不能适应新生概念. 因此利用搜索引擎的返回结果作为两个概念之间相似性的度量. 分析了目前二分图算法中每个服务接口的概念都是相对孤立的，没有考虑未匹配节点的语义. 因此作者提出了几种相似性计算公式，将未匹配结点的最大权边加入相似性计算. 总的来说，二分图匹配可以提高相似性计算的准确率但其计算复杂度较高，不适用于大规模服务的应用.

而基于服务聚类的服务查询的主要思想是首先计算任意两个服务之间的相似性，通过聚类得到不同语义的服务类，将查询空间限制于特定类中，提高计算效率. 文献[8]中是利用 Google 距离作为概念相似性的度量，同时考虑了服务的结构信息，综合得到两个服务之间的语义距离. 文献[10]利用本体中概念之间的不同关系来聚类服务集合，不用给出特定的阈值. 但当服务库中服务频繁变化时，以上预计算方法得到的聚类将失效，需要重新计算服务类，维护代价过高.

综上所述，基于相似性的服务查询比关键字查询和语义匹配方法有更好的适应性和性能. 但概念相似性的获取和二分图匹配带来的计算开销严重制约了具体应用中服务的规模，基于聚类的方法虽然可以在一定程度上解决该问题，但不适合动态环境下的服务发现.

3　服务查询的定义

3.1　基本定义

对于服务查询，用户需求与服务如何表征关系到具体方法的设计. 具体服务可以同时考虑服务的结构与语义，因此基于实例的查询方法可以充分利用这部分信息. 但对于组合服务而言，不同组合方法最终生成的结果有较大差异性，大部分组合算法的结果仅为包含特定输入/输出接口的服务流程，流程中任一服务均不能充分代表当前组合服务的实际能力. 为了统一具体服务和组合服务的查询，下面给出服务的形式化定义.

定义 1.　服务. 一个服务 S 可以通过 1 个 4 元组来表示：$S=\{n_s, d_s, I, O\}$，其中：

(1) n_s 是服务名称；

(2) d_s 表示服务的描述信息；

(3) $I=\{i_1, i_2, \cdots, i_n\}$ 是该服务的输入概念集合；$O=\{o_1, o_2, \cdots, o_m\}$ 为该服务的输出概念集合.

但组合服务仅包含服务名和输入/输出概念集合. 本文定义用户的服务请求为如下定义.

定义 2.　服务请求. 一个服务请求 R 可表示为一个 3 元组 $R=\{I, O, \theta\}$，其中：

(1) I 是用户提供的输入概念集合；

(2) O 是期望得到的输出概念集合；

(3) $0<\theta\leq 1$ 是用户设定的相似度阈值，用来表示用户可接受的服务与当前服务请求的最低相似度.

根据上面定义，基于相似性的服务查询即根据用户提出的服务请求 R，通过服务匹配算法在服务库中找到与用户需求的相似度满足阈值 θ 的服务集合. 目前大部分匹配算法均采用二分图表示需求和服务中的概念集合，二分图中边的权值反映了概念之间的语义关系，将需求与服务接口的相似度计算转化为求二分图的最大权匹配. 下图 1(a)就是一个二分图的具体例子，左子图代表服务请求 $Request=\{Author, Book\}$，而右子图则为服务 $Service=\{Publication, Writer, Title\}$，其中图中的边代表了概念之间的相似度. 虽然不同的服务发现方法对于概念间语义关系的表示和获取各有不同，但整体思路都是一致的，概念间语义距离越远，则两个概念相似度越低.

用户请求和服务建立概念间的语义关系后，就

可以计算当前二分图的最大权匹配 M. 图 1(b)中加黑的边表示了例子中二分图的最大权匹配 $M=\{(Author:Writer),(Book:Title)\}$. 下面给出最大权匹配基本公式：

$$max\_value=\max\left\{\sum_{C_i\in R,C_j\in S} w_{i,j}\right\},\text{其中 } 0<w_{i,j}\leq 1 \quad (1)$$

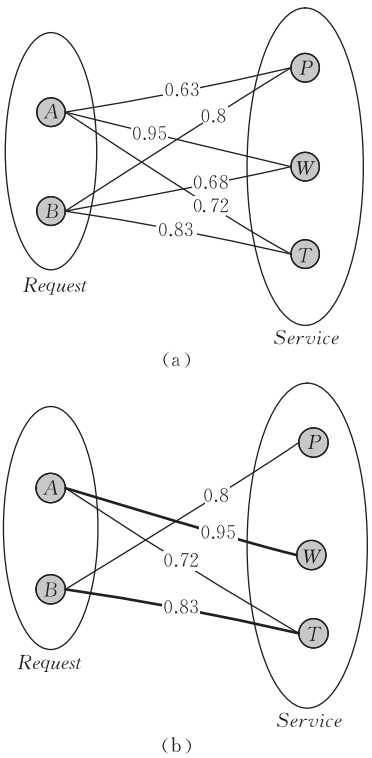


图 1 需求与服务的二分图表示和最优匹配

在得到二分图的最大权匹配 M 后，就可以根据匹配中边的数量 $|M|$ 得到整体的相似度，但在服务发现中服务与需求的相似性和用户请求的概念数 $|R|$ 相关. 上面例子中整体相似度为 $(0.95+0.83)/2=0.89$，形式化公式如下：

$$simRS=\frac{max\_value}{|R|},\text{其中 } 0<simRS\leq 1 \quad (2)$$

3.2 问题描述

根据式(1)和(2)，不难发现多数情况下最大权匹配仅与二分图中权值较大的边相关，这些边连接了服务描述中与需求强相关的概念. 如图 1(b)所示当阈值 $\theta=0.7$ 时，原二分图中两条权值较低的边被裁剪，并不影响最大权匹配以及相似度计算的结果. 这里引入无关概念的形式化定义.

定义 3. 无关概念. 设一个概念 C 与用户的服务请求 R 满足，$\forall C'\in R sim(C,C')<\theta$，则称该概念 C 为 R 的无关概念.

在之前的例子中并不存在无关概念，服务中每个概念都有满足阈值的语义关系. 对于包含无关概念的服务来说，这类概念对服务的整体相似度没有贡献. 当服务中所有概念都属于无关概念时，则该服务必定不满足用户的阈值要求，称其为当前需求的无关服务.

现有服务发现方法并不能预先对无关概念和服务加以区分，因此大部分计算开销用于无关服务的比较，随着服务数量的不断增加这个比例将继续提高. 也有部分研究者考虑通过缓存概念之间的相似度来避免概念相似性的重复计算. 但无论是搜索引擎还是通用本体并非一成不变，最典型的例子就是 yago2 知识库[14]，它是基于 WordNet 和维基百科构建的层次化本体. 它能随维基百科的变化自动添加概念和实体. 因此当外部信息发生变化时，相似度的缓存策略将带来不必要的计算和空间开销.

另一方面，现有方法在计算概念相似度时并未考虑语义的方向性. 如 Vehicle 和 Car 这类的概念在传统语义服务匹配方法[15]中语义关联可表示为：Vehicle 为 Car 的父概念，因此 Vehicle 相对 Car 的匹配度为 Plug-in；而 Car 对于 Vehicle 的匹配度仅为 Subsume，低于前者. 目前概念间的相似度计算不能反应这种差异，因为基于搜索引擎方法[7-8]，通过搜索结果度量语义距离，无法获取概念间的包含关系；而基于通用本体方法[6]也忽视了层次结构所包含的语义信息.

4 基于概念松弛的服务查询

针对现有方法的不足，本文提出了基于概念松弛的服务查询方法. 图 2 为该方法的整体框架，主要包含基于层次本体的概念松弛和服务相似度计算两个部分.

图 2 中层次本体是概念松弛的基础，管理概念之间的层次关系，概念松弛方法利用层次本体中概念之间的语义关系计算概念相似度. 该方法是根据用户的需求动态计算概念相似度，适用于各种层次本体，不仅是 WordNet，GeneOntology 等传统本体，也包括 Yago2 这类可以不断演化的新型知识库. 虽然搜索引擎的结果也可作为概念语义的度量，但返回结果会随搜索算法的调整发生改变，而且基于返回结果的度量无法反映概念间的包含关系. 因此本文并不考虑这种方式.

对应服务库存储了服务的 WSDL 文件，解析其中各部分的信息用于服务的组合与调用. 还管理了

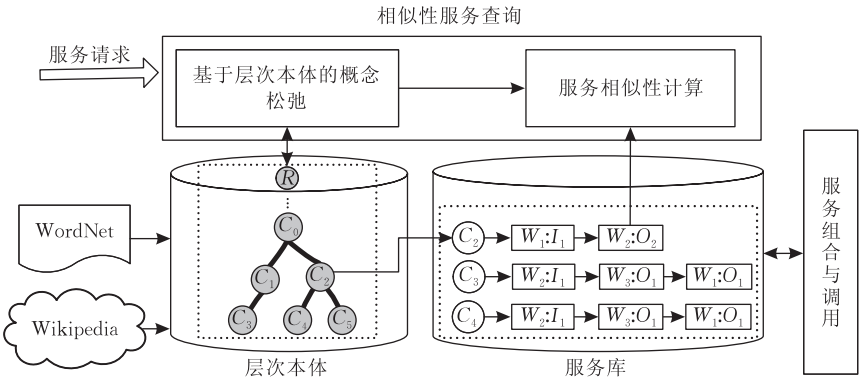


图 2　服务查询整体框架

概念和服务接口的映射关系，这种映射关系是通过对服务接口参数的自动分词以及概念识别得到的. 这里的服务不仅有从互联网中爬取到的具体服务，也包括自动生成的组合服务，根据接口参数统一管理和查询. 这种映射关系可通过 MapReduce 方式支持海量服务的语义管理并及时响应服务库的频繁变化.

服务查询处理模块由两部分构成：概念松弛模块主要针对用户请求所包含的概念进行语义计算，用户请求可以是简单概念集合或指明输入/输出接口所包含的概念. 根据层次本体中概念之间的语义关联，得到近似概念集合，过滤当前服务请求的无关概念. 而服务相似度计算根据第一步得到的相关概念集以及概念相似度，计算相关服务与用户请求的相似度，最终返回相关服务集合，并支持结果的动态扩展.

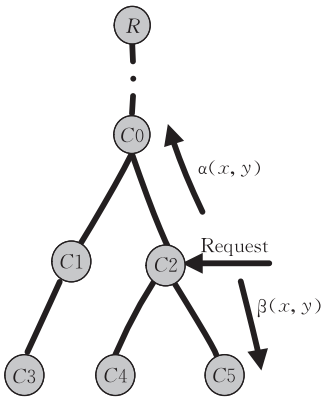
4.1　概念松弛

服务查询时，为了确定任一服务描述与当前服务请求是否相关，首先要计算两子图中概念之间的语义距离. 为了避免大量无关概念的相似度计算，本文提出了概念松弛方法在计算概念相似度时根据层次本体的结构信息，通过遍历算法确定满足阈值的相关概念集合.

设用户的服务请求为 R，针对 R 中的每个概念 C 进行松弛操作. 在介绍概念松弛方法之前需要先解释两个概念：概念的泛化和特化，其定义如下.

定义 4.　概念泛化. 对于当前概念 C，概念泛化是查找层次本体 O 中 C 的直接父概念 $C_A = Ancestor(C,O)$. 这一操作可以看作是在层次本体中的上行遍历. 例如 Vehicle 可以看作是 Car 的泛化. 当父概念不是唯一时，$Ancestor(C,O)$ 的结果是概念集合.

定义 5.　概念特化. 概念特化是遍历 C 的所有直接子概念 $C_D = Descendent(C,O)$. 可表示为层次本体 O 中的下行遍历操作. 而 Vehicle 的特化可能包括 Car、Boat、Rocket 等概念.



(a)

Concepts	n	APS
C_3	0	1/2
C_4	0	1/2
C_5	0	1/2
C_1	1	1/3
C_2	2	1/4
C_0	5	1/7
R	d	$1/(2+d)$

(b)

图 3　本体和对应概念 APS 示例

为了区别概念泛化和特化的不同影响，本文根据文献[16]提出的 APS 模型度量概念在层次本体中遍历时发生的语义变化. 设概念 C 的子孙概念总数为 n，则 C 的 $APS(C)=\frac{1}{2+n}$. 当概念 C 向上遍历得到父概念 B 时，其语义变化为它们 APS 的比值 $\alpha(C,B)=\frac{APS(B)}{APS(C)}$；相应的，概念 C 特化得到子概念 D 时，其语义变化为它们 APS 之差 $\beta(C,D)=APS(D)-APS(C)$. 图 3 是一个本体片段的示例，图 3(b)中给出了本体中各概念的 APS 值. 在图 3(a)本体中，对 C_2 进行概念泛化时，上行遍历到 C_0 的语义变化可以表示为：$\alpha(C_2,C_0)=4/7$；而 C_2 下行遍历到 C_5 时，特化的语义变化为 $\beta(C_2,C_5)=1/4$. 虽然概念之间的语义变化满足传递性，但对于不同的方向语义变化的传递性也是不同的：

$$\alpha(C_5,C_2)\times\alpha(C_2,C_0)=\frac{APS(C_2)}{APS(C_5)}\times\frac{APS(C_0)}{APS(C_2)}=\frac{APS(C_0)}{APS(C_5)}=\alpha(C_5,C_0) \tag{3}$$

$$\begin{aligned}\beta(C_0,C_2)+\beta(C_2,C_5)&=APS(C_2)-APS(C_0)+\\&\quad APS(C_5)-APS(C_2)\\&=APS(C_5)-APS(C_0)\\&=\beta(C_0,C_5)\end{aligned}\tag{4}$$

因此可以通过迭代方式不断扩展当前概念的近邻概念得到所有满足阈值的近似概念集合 $S(C,\theta)$. 值得注意的是,由于语义变化的方向性,因此 $\alpha(C_2,C_0)-\beta(C_0,C_2)\neq 0$.

概念之间的语义距离通过语义变化的规范化得到,这里给出概念泛化和特化的语义距离式(5),其中用于规范化的 $maxD$ 是通过计算任意叶子概念到根的上行距离得到. 而概念特化的语义距离较小,因此下行扩展时,会得到较多的近似概念. 这与语义匹配中匹配度 *plugin* 大于 *subsume* 是一致的. 具体推导过程可参见文献[16],本文由于篇幅原因不详细介绍.

$$D(C,C')=\begin{cases}-\dfrac{\log(\alpha(C',C))}{maxD}, & \text{当 } C' \text{为 } C \text{ 的父概念时}\\[2ex] \dfrac{\log(1+2\beta(C,C'))}{maxD}, & \text{当 } C' \text{为 } C \text{ 的子概念时}\end{cases}\tag{5}$$

下面给出概念松弛方法的伪代码.

算法 1. 基于层次本体的概念松弛算法.

输入:需求概念集合 RC,对应层次本体 O,相似性阈值 θ

输出:相关概念集合 RS

```
1.  RS=∅;//初始化
2.    for each C in RC to do //对于需求中的每个概念
3.  add C to AncQueue and DesQueue;
                  //将需要松弛的概念分别加入两个队列
4.  for each C' in the AncQueue to do//上行遍历
5.  if ((α=D(C,C')<=1-θ)&&(C' not in S(C,θ))
6.  then add (C',α) into S(C,θ);
7.          add Ancestor(C',O) to AncQueue;
                  //将 C'的所有直接父概念加入上行队列
8.  for each C'' in the DesQueue to do//下行遍历
9.  if ((β=D(C,C'')<=1-θ)&&(C'' not in S(C,θ))
10. then add (C'',β) into S(C,θ);
11. add Descendent(C'',O) to DesQueue;
                  //将 C''的所有直接子概念加入下行队列
12. return RS=RS∪S(C,θ);
```

分析可知遍历按方向的不同分为两部分:第4~7行对于上行遍历队列中的概念进行判定,若满足阈值要求则加入近似概念集合,并继续判断其父概念;第8~11行是相应的下行遍历操作. 对于每个概念而言,最坏情况下是对本体中每个概念遍历一次. 设需求 R 中存在 n 个概念,本体 O 中有 m 个概念,则计算复杂度为 $O(n\times m)$,其中 n 和 m 均为常量.

最终得到任意概念 C 的近似概念集合 $S(C,\theta)$, $S(C,\theta)$ 中每一个概念与 C 的语义距离都满足阈值 θ,对应需求 R 中所有概念的近似集合 $RS=\bigcup\limits_{C\in R}S(C,\theta)$. 根据服务库中概念与服务的映射可以得到所有与近似集合 RS 相关的服务,就是用户需求潜在可满足的服务集合. 那么根据前面的定义就可以判定当前需求的无关服务.

定理 1. 不包含相似集合 RS 中任意概念的服务,即为当前用户需求的无关服务.

证明. 根据定义 3 可知,若概念 C 与需求 R 中任一概念相似度满足阈值,则必然属于当前需求的近似概念集合 RS. 若服务 S 中不包含近似集合中的任何概念,即证明服务 S 中概念均为无关概念,因此可判定服务 S 为需求 R 的无关服务.

4.2 相似性计算

概念松弛不仅裁剪了与用户需求无关的服务,而且得到了服务与需求中概念的相似度,这是进行二分图匹配的基础. 但在进行二分图匹配之前,还需要对相关服务集合进行过滤以提高匹配计算的效率.

首先,考虑相关服务中所包含的 3 类不同概念:匹配概念、非匹配概念和无关概念. 无关概念已在第 3 节中定义,匹配概念和非匹配概念定义如下.

定义 6. 匹配概念. 对于服务 S 中的概念 C' 满足 $\exists C\in R, E(C',C)\in M$, M 为二分图的最大权匹配,则称 C' 为匹配概念.

定义 7. 未匹配概念. 对于服务 S 中的概念 C' 满足 $\exists C\in R, D(C',C)\leq 1-\theta$,但 $\forall C\in R, E(C',C)\notin M$,则称 C' 为未匹配概念.

由以上定义可知,对于特定用户需求,服务中的匹配概念和未匹配概念反映了与需求的语义关联,而无关概念是没有意义的,应对服务整体相似性不造成影响. 在传统服务匹配中,笼统地将这三类概念全部与需求进行匹配计算,导致假阳性问题.

在计算相似度之前,借鉴文献[11]的思想估计服务相似度上限过滤不满足阈值的相关服务,上限估计是利用服务中所有相关概念的最大相似度实现,具体公式如下:

$$UBsim=\frac{\sum\limits_{C_i\in S}\max\{w_i\}}{\min\{|R|,|S|\}},\ \text{其中 } \theta\leq w_i\leq 1\tag{6}$$

这里的 w_i 代表了对应服务中概念 C_i 的相似度,由于概念 C_i 可能与需求中多个概念的相似度都满

足阈值，因此取其最大值. 由于不再考虑无关服务的影响，对式(1)和(2)作出调整，即二分图匹配算法中仅考虑满足阈值的概念相似度作为匹配的基础. 根据上限估计和式(1)、(2)可以得到以下定理.

定理 2.　对于上限估计 UBsim 不满足阈值的服务 S，其整体相似性 simRS 也不满足阈值.

证明. 比较式(6)和(1)不难发现，$\sum_{C_i \in S} \max\{w_i\} \geq max\_value$，由于二分图最大权匹配时并不是所有服务中的概念都能取得最大相似度，同时 $\min\{|R|, |S|\} \leq |R|$，因此 $UBsim = \dfrac{\sum_{C_i \in S} \max\{w_i\}}{\min\{|R|, |S|\}} \geq \dfrac{max\_value}{|R|} = simRS$，当 $UBsim$ 小于阈值 θ，$simRS$ 必然无法满足阈值.　　证毕.

分析可知，二分图匹配得到的最大相似度仅考虑了匹配概念的贡献，但未匹配概念的作用被忽略了. 而服务的语义相似性需要通过服务中所有与需求相关概念反映，所以在相似性计算中也要加入未匹配概念的贡献. 设未匹配概念集合为 UM，相似度计算公式如下：

$$simRSP = \frac{2max\_value + \sum_{C_i \in UM} \max\{w_i\}}{|R| + |S|},$$
$$\text{其中 } \theta \leq w_i \leq 1 \qquad (7)$$

该公式根据二分图中每个概念的贡献综合得到需求与服务的相似度，下面给出具体排序算法的伪代码，如算法 2 所示.

算法 2.　基于概念松弛的服务相似性算法

输入：用户需求 R，对应相关服务集合 RS，相似性阈值 θ

输出：排序服务集合 $SL(S, simRS)$

```
1. for each S in RS to do
                    //对于相关服务集合中的任一服务 S
2. SL=∅;UM_max=0;
3. if UBsim(S,R)>=θ   //上限估计过滤
4. then remove Unrelated Concepts in S
5. Max_value=KM(S,R);
                    //利用二分图匹配得到最大相似性式(1)
6. for each Unmatched ConceptC in S to do
                    //处理未匹配概念
7. UM_max=UM_max+max(C);
8. simRS=Sim(Max_value,UM_max);
                    //根据式(7)计算整体相似性
9. Insert (S,simRS) into SL;
10. return SL;
```

已知计算二分图最大权匹配的算法时间复杂度为 $O(V \times E \times n)$，其中 n 为相关服务个数，V 代表了图中相关概念的数目，而 E 为概念之间边的数目. 在实际服务中概念数 V 和概念之间的边数 E 均为常数，因此排序算法的整体复杂度与相关服务数呈线性关系. 另外根据上限估计过滤，相关服务 n、相关概念 V 以及概念之间的边 E 均大大减小.

5　实验与结果分析

5.1　实验环境及数据

本文选择与传统服务查询比较验证方法的性能和可扩展性. 实验数据采用文献[15]中给出的标准测试集 OWL-S TC，其中包含了 1083 个服务，划分为 9 个不同领域. 还在按需服务平台 DigService① 中实现了本文提出的相似性服务查询，针对服务库中包含的 1 万个真实 WSDL 服务进行性能测试. 主要比较对象及数据情况如表 1 所示，其中由于 OWLSMX 仅能处理 OWL-S 服务，因此无法在真实服务集上进行测试. 标准测试集采用查全率 *recall* 和查准率 *precision* 作为服务查询性能的评价标准，还引入了 $F\text{-}measure = \dfrac{2 \times precision \times recall}{precision + recall}$ 作为方法整体性能的度量. 实验在 Intel Core i5 760 2.8 GHz 处理器和 4 GB 内存的主机上运行，操作系统为 Windows 7. 测试代码为 Java 编写，Eclipse3.6 编译通过.

表 1　实验数据集及对比文献

	数据规模	性能指标	对比文献
标准测试集	1083 个服务，分为 9 个领域	查准率 查全率 *F-measure*	URBE[6] M_max[7]
真实服务集	10000 个服务，划分 6 组不同规模的服务集合	响应时间 匹配效率	OWLSMX[15]

5.2　实验结果与分析

和传统方法比较之前，需要确定概念松弛的策略以及相应的用户阈值. 概念松弛策略包括两方面：首先要确定语义变化度量，根据之前的介绍语义变化有 α 方法、β 方法以及混合方法 3 种；另外还要分析概念松弛是否考虑当前概念的兄弟概念，即上行遍历得到的近似概念是否需要遍历其子孙概念，对应两种不同的遍历策略(回溯遍历和不回溯遍历). 而对于阈值的选择将影响返回结果的规模. 因此首先根据标准测试集，给出 3 种不同语义变化度量对应的服务查询结果，如图 4 所示. 由比较结果可以发

① www.mydigservice.com

现,α 方法返回结果虽然查准率较高,但查全率不足;β 方法的情况正好相反.而混合方法是整体上最好的,这是因为上下行不同的语义度量符合层次本体中概念的语义变化.

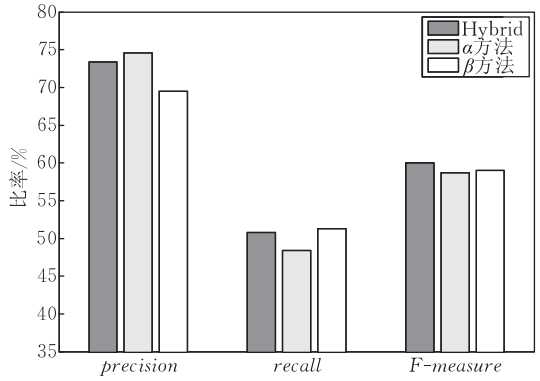


图 4 应用不同语义变化度量的查询效果

图 5 是根据不同的遍历策略得到的查询结果对比.可以发现回溯遍历不论是查全率还是查准率都要优于不回溯遍历,这一点和文献[14]中匹配时考虑本体中的兄弟概念是一致的.而图 6 反映了不同用户阈值 $\theta(0.6\sim0.9)$ 下服务查询的性能比较,可以发现根据测试集提供的基准,随着阈值的上升,查准率提高,但查全率降低.而用户阈值为 0.7 时取得最好的综合性能.

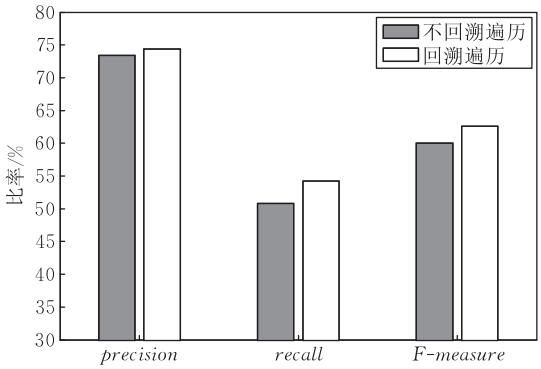


图 5 不同遍历策略下的查询效果

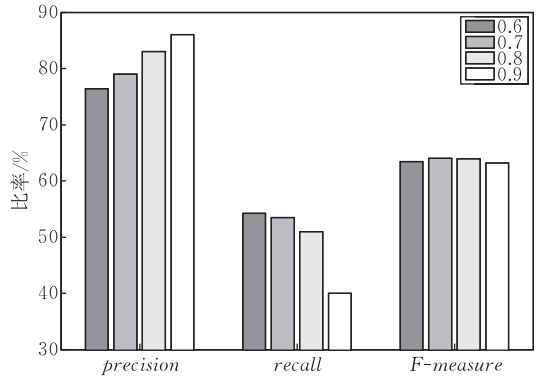


图 6 不同阈值对查询结果的影响对比

图 7 中给出了本文提出方法 simRSP(式 7)、M_max以及 URBE 三种不同服务相似性计算方法得到的查询结果对比.通过结果可见 *simRSP* 方法优于其它方法.证明上限估计可以有效的过滤假阳性结果得到更好的查准率,而未匹配概念能反映服务的整体语义.3 种方法整体性能并不高是因为只考虑了服务描述中输入输出参数所包含的概念相似性,也没有利用特定领域本体中概念的语义相关性.

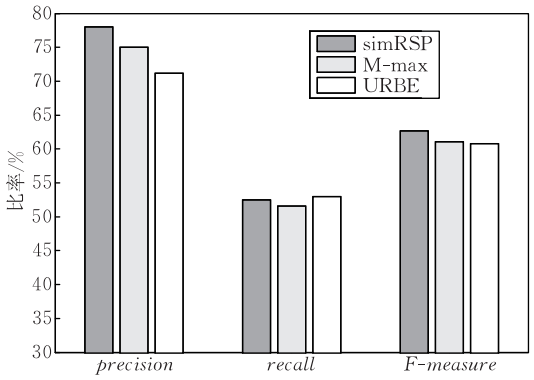


图 7 与现有服务查询方法的对比

为了测试方法的可扩展性,在按需服务平台 DigService 中实现了基于概念松弛的相似性查询.可扩展性是指该方法在不同规模的服务库中的可用性,主要考察服务查询方法的计算时间.服务库中包含了从互联网上获取的 WSDL 服务近 2 万个,用户可以通过高级查询设定相似性阈值,启动相似性服务查询.

为衡量本文方法的可扩展性从服务库中提取了 6 组不同规模(1000~10000)的服务集合进行查询,比较本文方法与传统方法的响应时间.通过图 8 对比可发现,若设查询的最大响应时间为 100 s,那么 URBE 方法仅能支持 1000 个服务以内的查询.而缓存概念相似性后,虽然一定程度上提高了查询的效

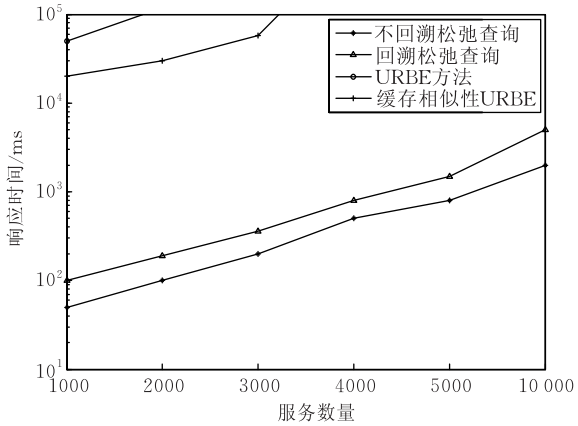


图 8 与传统方法的查询效率对比

率，但也不能适用于 4000 个服务的应用场景. 而概念松弛方法的执行时间基本保持在毫秒级别，即使是回溯遍历的方法，处理 1 万个服务的查询时间也不到 10 s. 在可扩展性方面，本文提出的方法具有明显优势.

最后比较 3 种方法 2 分图匹配的计算效率，如图 9 所示，证明概念松弛和上限过滤有效地缩小了需匹配的服务数量. 本文方法在匹配阶段的计算开销较传统匹配方法节省了 90%.

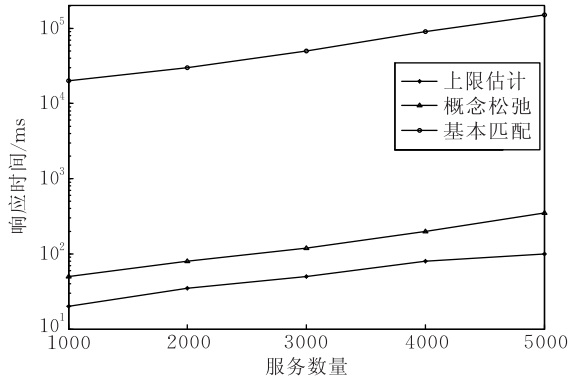


图 9　二分图匹配的计算效率对比

根据测试数据和真实服务上的实验结果可证明，本文所提出的基于概念松弛的服务相似性查询，不仅通过考虑层次本体中概念之间的语义关系，提高了服务查询的查准率和查全率；还能极大提高相似性服务查询的可扩展性，以适应云计算环境下服务数量急剧增加的应用场景. 实时计算得到用户需求的概念相似度和相关服务能满足互联网环境下动态性的要求.

6　结　论

综上所述，Web 服务是云计算从理念到实现的关键支撑技术之一，而如何提高基于相似性的 Web 服务查询的准确性以及效率是本文研究的重点. 本文首次提出面向层次本体的概念松弛方法，不仅充分利用概念之间的语义关系，并能裁剪大量与需求无关的服务. 相似性计算考虑服务中不同概念的相似度贡献得到查询结果. 经实验证明该方法在概念松弛时考虑不同方向的松弛策略是符合实际情况的，且提出的相似性查询方法无论是查全率还是准确率均优于传统方法，算法效率可以支持动态环境下的大规模服务应用. 下一步将考虑通过 MapReduce 方法进一步提高概念松弛的性能，这对云计算中海量基于服务的应用具有现实意义.

参　考　文　献

[1] Zhang L-J, Zhang J, Cai H. Services Computing. New York and Beijing: Springer and Tsinghua University Press, 2007

[2] Zhang Rong, Zettsu K, Kidawara Y, Kiyoki Y. Context-sensitive query expansion over the bipartite graph model for Web service search//Proceedings of the 16th International Conference on Database Systems for Advanced Applications (DASFAA'11). Berlin, German, 2010: 418-433

[3] Dong Xin, Halevy A, Madhavan J, Nemes E, Zhang Jun. Similarity search for Web services//Proceedings of the 13th International Conference on Very Large Data Bases(VLDB'04). 2004: 372-383

[4] Wu J, Wu Z H. Similarity-based Web service matchmaking//Proceedings of the International Conference on Services Computing(SCC2005). Orlando, FL, USA, 2005, 1: 287-294

[5] Chen Lei, Yang Geng, Wang Dong-Rui, Zhang Ying-Zhou. WordNet-powered Web services discovery using kernel-based similarity matching mechanism//Proceedings of the 2010 5th IEEE International Symposium on Service Oriented System Engineering (SOSE'10). 2010: 64-68

[6] Plebani P, Pernici B. URBE: Web service retrieval based on similarity evaluation. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(11): 1629-1642

[7] Liu Fang-Fang, Shi Yu-Liang, Yu Jie, Wang Tian-Hong, Wu Jing-Zhe. Measuring similarity of web services based on WSDL//Proceedings of IEEE 8th International Conference on Web Services (ICWS2010). Washington, DC, USA, 2010: 155-162

[8] Elgazzar K, Hassan A E, Martin Pa. Clustering WSDL documents to bootstrap the discovery of Web services//Proceedings of the 2010 IEEE International Conference on Web Services (ICWS'10). Washington, DC, USA, 2010: 147-154

[9] Deng Shui-Guang, Yin Jian-Wei, Li Ying, Wu Jian, Wu Zhao-Hui. A method of semantic Web service discovery based on bipartite graph matching. Chinese Journal of Computers, 2008, 31(8): 1364-1375(in Chinese)
(邓水光，尹建伟，李莹，吴健，吴朝晖. 基于二分图匹配的语义 Web 服务发现方法. 计算机学报，2008，31(8)：1364-1375)

[10] Dasgupta S, Bhat S, Lee Yugyung. Taxonomic clustering and query matching for efficient service discovery//Proceedings of 2011 IEEE International Conference on Web Services (ICWS2011). Washington, DC, USA, 2011: 363-370

[11] Cilibrasi R L, Vitanyi P M B. The Google similarity distance. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3): 370-383

[12] Schickel-Zuber V, Faltings B. OSS: A semantic similarity function based on hierarchical ontologies//Proceedings of the 20th International Joint Conference on Artificial Intelligence. San Francisco, USA, 2007: 551-556

[13] Lovasz L, Plummer M. Matching Theory. Amsterdam: North-Holland, 1986

[14] Suchanek F M, Kasneci G, Weikum G. Yago: A core of semantic knowledge//Proceedings of the 16th International Conference on World Wide Web (WWW2007). NY, USA, 2007: 697-706

[15] Klusch M, Fries B, Sycara K. OWLS-MX: A hybrid semantic Web service matchmaker for OWL-S services. Web Semantics: Science, Services and Agents on the World Wide Web, 2009, 7(2): 121-133

[16] On B W, Koudas N, Lee D, Drivastava D. Group linkage// Proceedings of IEEE 23rd International Conference on Data Engineering (ICDE'07). Istanbul, Turkey, 2007: 496-505

OU Wei-Jie, born in 1981, Ph. D. candidate. His research interests include service computing and information retrieval.

ZENG Cheng, born in 1978, Ph. D., associate professor. His research interests include service computing, information recommendation, information retrieval.

XIANG Xiao-Ming, born in 1987, M. S. candidate. His research interests include service computing and database.

PENG Zhi-Yong, born in 1963, Ph. D., professor, Ph. D. supervisor. His research interests focus on database management.

LI De-Yi, born in 1944, Ph. D. supervisor, academician of Chinese Academy of Engineering. His main research interests include complex network, data mining, and artificial intelligence.

Background

With the fast development of Cloud computing, the quantity of Web services explodes rapidly while the requirement of consumer increases sharply, which subsequently brings great challenges to the accurate, efficient and automatic retrieval of target services for consumers. Some hot topics in service computing areas include service discovery, service composition, and quality of service. In this paper we focus on problems brought by service discovery.

Service discovery usually need to be translated to the problem of optimal matching for bipartite graph. The solutions of bipartite graph matching in the literature involve similarity that represents the semantic relation between various concepts. But for large-scale service, complete bipartite graph matching is too expensive. Although some concepts and services are unrelated to the requirement, most of traditional approaches couldn't prune any services in advance. Clustering based methods were proposed to improve the efficiency of service discovery, but they are not suitable for dynamic environment.

In order to make up for the shortcoming of the above methods, this paper proposes a new method named simRSP for service query via concept relaxation, which uses hierarchical ontology to figure out similar concepts. Mathematical analysis for concept relaxation is presented in this paper as well. The theoretical and practical pruning effect is presented in this paper. We conduct extensive experiments to evaluate the performance of the method. Experimental results show that simRSP has better performance in precision and recall rate than others and obtains an order of magnitude speed-up comparing to URBE.

This work is supported in part by the National Grand Fundamental Research 973 Program of China. The foundations focus on the research of various areas of service computing. Our group has been working on the research of service computing for many years, and many good papers have been published in worldwide conference and transactions, such as SCC, WWW, ICDE, TKDE et al. This prototype system introduced in this paper has obtained the best demo award in NDBC'2011. This method is a new idea for the service discovery study and has wide range of applications.