

基于聚类的 QoS 语义 Web 服务发现研究

武彩红, 李蜀瑜

(陕西师范大学 计算机科学学院, 陕西 西安 710062)

摘要:如何动态地选择出适合用户需求的 Web 服务正在引起相关研究者的关注。为了提高 Web 服务查找的效率, 提出了一种支持 QoS 的语义 Web 服务发现框架。首先根据 Web 服务本体分别计算服务描述、输入、输出、前提条件和结果这五个层面的语义相似性, 然后利用聚类技术, 将相似度高的服务聚为一类, 过滤掉与服务请求完全不同类别的服务, 形成候选服务集, 最后进行 QoS 比较, 得到一个服务排序, 为请求者选择 QoS 综合值最高的服务。仿真实验验证了该方法的可行性和有效性。

关键词:语义 Web 服务; 聚类; 服务质量; 服务发现

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2011)03-0132-05

Web Service Discovering Approach Supporting QoS Using Clustering Algorithm

WU Cai-hong, LI Shu-yu

(School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)

Abstract: How to dynamically select Web services that can best meet the requirements of consumers is an ongoing research direction in Web service community. A new semantic Web service discovery approach supporting QoS is proposed to improve the efficiency of service discovery. Firstly, the semantic similarity of service description, input, output, precondition and effect is computed based on Web service ontology, and Web services are divided into groups by applying clustering technology. Then a service sequence is presented by calculating the QoS distance, and the highest value is chosen as the indicator of the best service. At last, experiment results have shown the feasibility and effectiveness of the proposed method.

Key words: semantic Web service; clustering; QoS; service discovery

0 引言

随着 Web 服务的大量涌现, 提供类似功能的 Web 服务数目日益增多, 如何快速地发现满足需要的服务已经成为一个重要问题。随着 Web 服务数量的增加, 服务注册信息库也在不断膨胀, 这导致了查找服务费时, 结果不准确等问题。因此很多学者将数据挖掘^[1-4]的方法, 如聚类, 应用于 Web 服务以便提高服务发现效率。

文献[1]采用服务聚类的思想对相似 Web 服务分组, 来改善服务发现, 其中服务相似度的计算是采用关键词在服务描述文档中出现频率的思想进行的, 计算方式较为简单, 相应地准确度不高; 文献[2]采用 q-

grams 方法计算字符串相似性, 基于 WSDL 的服务描述完成对服务的聚类; 文献[3]针对网格服务, 提出了基于语义相似度的服务本体聚类方法, 该方法从服务输入、输出以及功能相似性来进行计算; 文献[4]提出了一个语义 Web 服务发现框架, 从服务的功能相似和过程相似两个层面对服务进行聚类。

总体来说, 现有的利用聚类技术来改善服务发现的这些研究, 大多是根据服务描述、输入输出来计算服务的相似度以及匹配, 主要从语义的角度考虑服务的功能相似, 而考虑服务质量(QoS)的较少。我们在分析了语义网技术和相关聚类服务发现机制的特点后, 提出了一种基于聚类的 QoS 语义 Web 服务发现框架, 该框架的主要思想在于: 从服务描述、输入、输出、前提条件和结果这五个层面上评估服务之间的相似性, 充分考虑其语义信息; 利用聚类技术, 对服务库中海量的服务进行预处理, 将相似度高的服务聚为一类, 形成服务簇; 在服务聚类的基础上, 实现基于 QoS 的 Web 服务匹配。

收稿日期: 2010-06-10; 修回日期: 2010-09-05

基金项目: 国家自然科学基金(60671063); 陕西师范大学校级优秀科技预研项目(200702018)

作者简介: 武彩红(1986-), 女, 山西文水人, 硕士研究生, 研究方向为语义 Web 服务; 李蜀瑜, 副教授, 博士, 研究领域为嵌入式系统、Web 服务。

1 基于语义的服务相似度计算

服务发现是当前网络及语义 Web 中的一个热门的研究课题,随着 Web 服务应用的普及,Web 服务的数量急剧增加,面对数量庞大的服务群,如何从提供相似功能的可用 Web 服务集中,过滤出满足需要的 Web 服务已成为亟待解决的问题。为了加快过滤过程,将服务进行聚类预处理可以缩小可用服务的范围。过滤过程不仅仅基于功能方面,还应基于非功能方面,通过预先对服务进行聚类处理,将具有相似功能和相似接口的服务聚类在一起,在服务发现时,大大的缩小了服务的查找空间,从而提高了匹配效率。

1.1 计算基本描述的相似性

在语义 Web 服务框架中,Web 服务本体的语义结构定义为四元组:服务描述^[5]、公共属性、接口属性、QoS。

定义 1. Web 服务本体: $WS = (D, CP, IOPE, QoS)$, 其中, WS 表示 Web 服务的名称, D 为服务描述, CP 为服务公共属性,如服务 ID、服务提供者 ID、服务提供者名称、服务类型、版本等非功能属性, $IOPE$ 表示接口属性,包括输入、输出、前提条件和结果, $IOPE = Input(WS) \cup Output(WS) \cup Precondition(WS) \cup Effect(WS)$, QoS 为服务质量。

定义 2. 领域本体 $O = (C, A^C, R, A^R, H, X)$, 其中, C 为概念集合, A^C 为概念的属性集合, R 为关系集合, A^R 为关系的属性集合, H 为概念层次集合, X 为公理集合。

目前,关于 Web 服务领域的语义相似性已有不少研究,但暂时还没有一个统一的标准。文中采用如下定义:

定义 3. 在一个本体分类体系中,概念 C_1 和概念 C_2 之间的本体距离 $Dis(C_1, C_2)$ 定义为在本体树中连接它们的最短路径的边数。

定义 4. 在一个本体分类体系中,概念 C_1 和概念 C_2 之间的语义相似性^[6-9] 定义为:

$$Sim(C_1, C_2) = \frac{\alpha * (l_1 + l_2)}{(Dis(C_1, C_2) + \alpha) * \max(|l_1 - l_2|, 1)} \quad (1)$$

其中 l_1, l_2 是概念 C_1, C_2 分别在本体分类中所处的层次,最上层定义为 1, $Dis(C_1, C_2)$ 是它们的本体距离, α 是相似度为 0.5 时 C_1, C_2 之间的距离, α 是一个可调节的参数,一般 $\alpha > 0$ 。

定义 5. Web 服务的服务描述相似性直接采用基于两个概念之间的本体距离计算:

$$Sim_{des}(S_i, S_j) = \frac{\alpha * (l_1 + l_2)}{(Dis(S_i, S_j) + \alpha) * \max(|l_1 - l_2|, 1)} \quad (2)$$

1.2 计算基于接口属性的相似性

Web 服务通常有多个输入参数,假定服务 SP 有 m 个输入参数 $SP_{input} = (IP_{p1}, IP_{p2}, IP_{p3}, \dots, IP_{pm})$, 服务 SQ 有 n 个参数 $SQ_{input} = (IP_{q1}, IP_{q2}, IP_{q3}, \dots, IP_{qn})$ 。关于参数最优配对问题,已有一些成熟的方法和技巧,可以直接利用。参数直接配对情况记为 $IP = \{ \langle IP_{p1}, IP_{q1} \rangle, \langle IP_{p2}, IP_{q2} \rangle, \dots, \langle IP_{pl}, IP_{ql} \rangle \}$, 其中 $IP_{p1} \neq IP_{p2} \neq \dots \neq IP_{pl} \in SP_{input}, IP_{q1} \neq IP_{q2} \neq \dots \neq IP_{ql} \in SQ_{input}, l = \min(m, n)$ 。

文中采用 L. Rips 提出的基于多维属性的语义相似度来定义输入参数的相似度,即:

$$Dis(SP_{input}, SQ_{input}) = \sqrt{\sum_{i=1}^l (IP_{pi} - IP_{qi})^2} \quad (3)$$

结合语义距离以及参数之间的匹配程度服务间输入参数相似度计算如下:

$$Sim_{input}(SP_{input}, SQ_{input}) = \frac{\alpha * (l_1 + l_2)}{(Dis(SP_{input}, SQ_{input}) + \alpha) * \max(|l_1 - l_2|, 1)} * \frac{1}{m+n-1} \quad (4)$$

类似地,输出参数相似度为 $Sim_{output}(SP_{output}, SQ_{output})$ 、前提条件相似度 $Sim_{pre}(SP_{pre}, SQ_{pre})$ 、结果相似度 $Sim_{effect}(SP_{effect}, SQ_{effect})$ 的计算方法与输入参数的相似度计算类似。

1.3 计算 Web 服务的相似性

当采用公式(2)、(4)计算出各项相似度后,计算 Web 服务 S_1, S_2 的综合相似度:

$$SimWS(S_1, S_2) = w1 * Sim_{des}(S_1, S_2) + w2 * Sim_{input}(S_1, S_2) + w3 * Sim_{output}(S_1, S_2) + w4 * Sim_{pre}(S_1, S_2) + w5 * Sim_{effect}(S_1, S_2) \quad (5)$$

其中 $w1, w2, w3, w4, w5$ 分别是服务描述、输入、输出、前提条件、结果的权值, $\sum w_i = 1.0$, 且 $0 \leq w_i \leq 1$ 。可由用户指定,如果用户没有对权值的要求,则可采用默认的平均权值,即这五个方面同等重要。两个服务的综合相似度值越大表明服务的匹配度越高。

2 基于聚类服务发现框架

为了有效地解决服务发现中存在的问题,提出了一个基于聚类的 QoS 语义 Web 服务发现框架(如图 1 所示)。该框架由三部分组成:服务提供者、服务请求者和扩展的 UDDI 注册中心。

该框架根据服务的服务描述和接口属性相似度,借助于聚类技术,对服务库中的 Web 服务进行聚类;在服务发现过程中,实现了基于 QoS 的服务匹配。服务提供者通过 Web Service Provider Interface 发布且更新他们描述的服务,这些描述包括服务的功能属性和服务质量属性。Service Management 用于管理语义和

QoS 信息,并且通过计算服务的语义相似性来实现服务的聚类,形成服务簇。服务请求者可以通过 Web Service Requester Interface 发送一个目标描述信息给服务发现组件,请求消息中同样也包含了服务的功能属性和服务质量属性的描述,在服务发现组件中将服务请求者的请求与注册中心存储的服务簇心进行比较,然后在此基础上进行基于 QoS 的匹配,匹配完成后将返回一组满足要求的候选服务列表,我们使用 Ranking Component 组件对这组候选服务依据 QoS 得分进行降序排序。

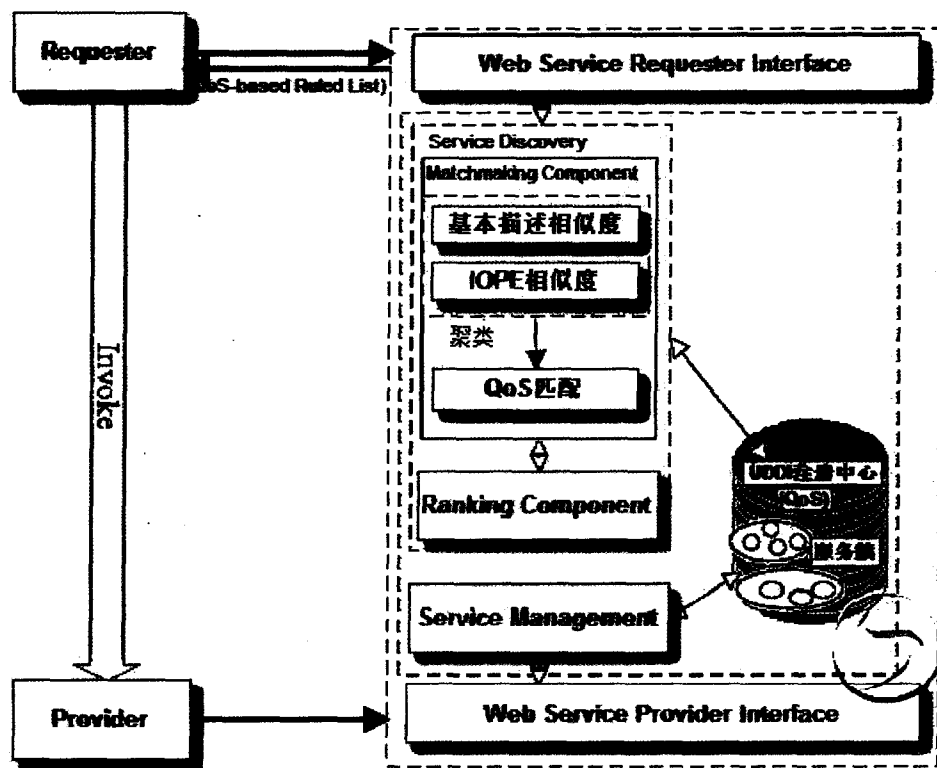


图 1 基于 QoS 的语义 Web 服务发现框架

3 基于服务聚类的 Web 服务发现

Web 服务发现是 Web 服务系统架构中的一个重要部分。Web 服务发现的传统解决方案是 UDDI,即通过对 UDDI 上的服务注册信息进行关键词匹配和简单分类进行服务发现,由于注册库中服务的规模和数量是相当庞大的,服务的查找效率成为 Web 服务发展的一个瓶颈。同时由于缺乏对服务质量 QoS 的描述而难以为用户选择最佳服务。因此,本文在对服务进行聚类预处理之后,过滤掉与服务请求完全不同类别的服务,避免了在相似度较低甚至无相似的匹配计算上浪费时间,提高了匹配效率。通过在候选服务集中进行 QoS 比较,得到一个服务排序,为请求者选择 QoS 综合值最高的服务。基于 QoS 的服务发现是建立在服务聚类结果基础之上的。

3.1 Web 服务 QoS 描述

目前关于 QoS 的描述有很多,如吞吐量、可靠性、可扩展性、并发处理能力、响应时间、服务价格、信誉度、可用性、准确性、安全性等。这些 QoS 属性对于 Web 服务来说同样重要,他们分别从不同的角度反应了 Web 服务性能。本文只考虑三个服务质量属性:服务执行时间、服务执行价格、服务信誉度。

Web 服务 QoS 向量定义为: $QoS(WS) = \langle T, C, R \rangle$

其中 T (Time) 为服务执行时间; C (Cost) 为服务执行价格; R (Reputation) 为服务的信誉度。此 QoS 模型

可以根据用户的需求进行扩展。

3.2 QoS 的相似度计算

假设通过服务聚类后,候选服务集中有 n 个满足请求者功能需求的服务,每个服务有 m 个 QoS 属性,则得到一个 $n * m$ 阶的决策矩阵 Q ,这里选择第 i 个服务的第 j 个 QoS 属性 $Q_{i,j}$ 为评价对象。由于每个 QoS 属性有各自的度量单位,需要先对各个属性进行规范化处理,使其范围映射到 $[0, 1]$ 区间。QoS 属性分为效益型和成

本型两类。效益型属性是指属性值越大越好的属性,成本型属性是指属性值越小越好。本文 QoS 属性的规范化分为:数值型属性规范化和区间型属性规范化。

1) 数值属性的 QoS 规范化。如果属性描述为精确型,用下面的公式进行规范化。

效益型属性:

$$V_{i,j} = \begin{cases} \frac{Q_{i,j} - Q_j^{\min}}{Q_j^{\max} - Q_j^{\min}} & \text{if } (Q_j^{\max} - Q_j^{\min} \neq 0) \\ 1 & \text{if } (Q_j^{\max} - Q_j^{\min} = 0) \end{cases} \quad (6)$$

成本型属性:

$$V_{i,j} = \begin{cases} \frac{Q_j^{\max} - Q_{i,j}}{Q_j^{\max} - Q_j^{\min}} & \text{if } (Q_j^{\max} - Q_j^{\min} \neq 0) \\ 1 & \text{if } (Q_j^{\max} - Q_j^{\min} = 0) \end{cases} \quad (7)$$

其中 $V_{i,j}$ 是第 i 个服务第 j 个属性的规范化后结果, Q_j^{\max} 是矩阵 Q 第 j 列的最大值,即 n 个候选服务第

j 个 QoS 属性的最大值, Q_j^{\min} 是矩阵 Q 第 j 列的最小值, 即 n 个候选服务第 j 个 QoS 属性的最小值。

2) 区间属性的 QoS 规范化。如果使用了区间型描述, $Q_{i,j} = [Q_{i,j}, \overline{Q_{i,j}}]$, 其中 $Q_{i,j}$ 是区间的下界, $\overline{Q_{i,j}}$ 是区间的上界, 则使用下面的公式进行规范化。

效益型属性:

$$V_{i,j} = \begin{cases} \frac{Q_{i,j} - Q_j^{\min}}{Q_j^{\max} - Q_j^{\min}} & \text{if } (Q_j^{\max} - Q_j^{\min} \neq 0) \\ 1 & \text{if } (Q_j^{\max} - Q_j^{\min} = 0) \end{cases} \quad (8)$$

$$\overline{V_{i,j}} = \begin{cases} \frac{\overline{Q_{i,j}} - \overline{Q_j^{\min}}}{\overline{Q_j^{\max}} - \overline{Q_j^{\min}}} & \text{if } (\overline{Q_j^{\max}} - \overline{Q_j^{\min}} \neq 0) \\ 1 & \text{if } (\overline{Q_j^{\max}} - \overline{Q_j^{\min}} = 0) \end{cases} \quad (9)$$

$$V_{i,j} = \frac{V_{i,j} + \overline{V_{i,j}}}{2} \quad (10)$$

成本型属性:

$$V_{i,j} = \begin{cases} \frac{Q_j^{\max} - Q_{i,j}}{Q_j^{\max} - Q_j^{\min}} & \text{if } (Q_j^{\max} - Q_j^{\min} \neq 0) \\ 1 & \text{if } (Q_j^{\max} - Q_j^{\min} = 0) \end{cases} \quad (11)$$

$$\overline{V_{i,j}} = \begin{cases} \frac{\overline{Q_{i,j}} - \overline{Q_j^{\min}}}{\overline{Q_j^{\max}} - \overline{Q_j^{\min}}} & \text{if } (\overline{Q_j^{\max}} - \overline{Q_j^{\min}} \neq 0) \\ 1 & \text{if } (\overline{Q_j^{\max}} - \overline{Q_j^{\min}} = 0) \end{cases} \quad (12)$$

$$V_{i,j} = \frac{V_{i,j} + \overline{V_{i,j}}}{2} \quad (13)$$

其中公式(8)是对 $Q_{i,j}$ 下界的评价公式, $V_{i,j}$ 是评价结果, Q_j^{\max} 是该属性的所有区间下界的最大值, Q_j^{\min} 是该属性的所有区间下界的最小值。公式(9)是对 $Q_{i,j}$ 上界的评价公式, 公式(10)是计算公式(8)、(9)结果的均值。对于成本型属性公式(11-13)各解释一样。

把请求服务经过规范化后的 QoS 各项均与提供服务的规范矩阵进行计算, 可得到综合相似度, 我们使用加权的欧几里得距离来测量两个服务质量之间的相似度, 可以表示为如下:

$$\text{dis}(Q_i, Q_j) = \sqrt{w_1 |Q_{i1} - Q_{j1}|^2 + w_2 |Q_{i2} - Q_{j2}|^2 + \dots + w_p |Q_{ip} - Q_{jp}|^2} \quad (14)$$

其中 $\sum_{i=1}^k w_i = 1$, 表示不同 QoS 属性所占的权重值。函数返回一个 0 到 1 之间的实数, 返回值越接近 0, 两个 QoS 属性就越匹配。

3.3 Web 服务发现算法

基于聚类^[10-12]的 Web 服务发现算法基本思想

是: 首先通过计算服务描述和接口属性的语义相似度, 对 Web 服务进行聚类预处理, 将具有相似服务描述以及接口属性的服务聚合在一起, 形成服务簇; 然后根据服务请求从各个簇中找出与服务请求最相似的簇; 计算服务请求与簇内的每个服务之间的相似度, 相似度越大, 表明越符合请求, 当相似度大于一定阈值 δ 时, 将其留下, 形成一个新的集合; 最后计算服务请求与集合中的每个服务之间的 QoS 距离, 距离越小的, 说明越符合要求, 选取距离最小的服务。

算法: ServiceSelectAlgorithm

输入: 服务集 (WS), 簇的数目 k , 服务请求 (Req), 阈值 δ

输出: 满足服务请求者 QoS 需求的一个最佳服务 BestS
//使用公式(5)计算服务集 WS 中服务的相似度

Assigning k service as the cluster center $C = \{C_1, C_2, \dots, C_k\}$

Foreach service $s_i \in WS$ do / * 服务聚类 */

For each cluster center $C_j \in C$

Calculate the $\text{sim}(s_i, C_j)$

Computer new center

End for

Endfor

//计算服务请求与每个簇心的相似度

For each cluster center $C_i \in C$

do $\text{RQSimC}_i = \text{CalculateSimWS}(\text{Req}, C_i)$

if $\text{RQSimC}_i > \delta$ then do

CS CS. append(RQSimC_i) //将大于阈值的簇心加入候选集 CS 中

Endif

Endfor

//计算服务请求与候选集中每个服务的 QoS 距离

For each $cs_i \in CS$ do

$\text{QoSDistance}_{\text{Req-cs}} = \text{CalculateQoS}(\text{Req}, cs_i)$

Endfor

//找出 QoSDistance 数组中值最小元素的下标, 如果有多个, 则从其中任意挑选 1 个

$\text{BestS} = \text{FindMinDis}(\text{QoSDistance}_{\text{Req-cs}})$;

//找出与服务请求者 QoS 需求距离最短的服务

Return BestS

4 实验结果及分析

鉴于目前没有相关的标准平台和标准测试数据集, 本文以酒店服务预定为对象, 生成服务测试用例, 实验的数据均采用模拟的方式生成, 形成 1000 个语义 Web 服务, 每个语义 Web 服务对应构建一个领域本体, 并设计了常用的 10 个服务请求。

将本文提出的方法同 UDDI 中心提供的关键字服务发现方法和基于 OWL-S/UDDI 的服务发现方法进行了比较, 采用三种不同方法所获得的查全率统计如

图 2 所示,查准率如图 3 所示,其中 $\alpha=0.5, \delta=0.6$,所有的权重值都取平均值。

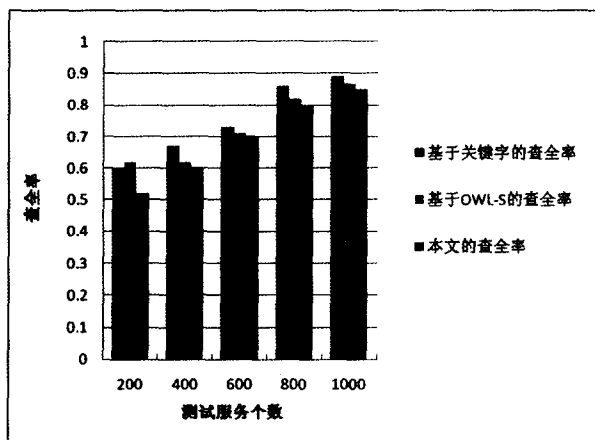


图 2 三种方法的查全率比较

测试结果显示:基于关键字服务发现方法、基于 OWL-S 服务发现方法和本文提出的服务发现方法的平均查全率分别为 0.75、0.73 和 0.69,平均查准率分别为 0.60、0.71 和 0.81。从图 2、图 3 和平均数据看出,本文提出的方法综合考虑了服务的描述和接口属性的语义相似性,以及服务质量匹配的方法,在查准率方面有明显提高,由于采用聚类技术,过滤掉很多不相关的服务,同时由于考虑了服务质量属性,在查全率方面有所降低。

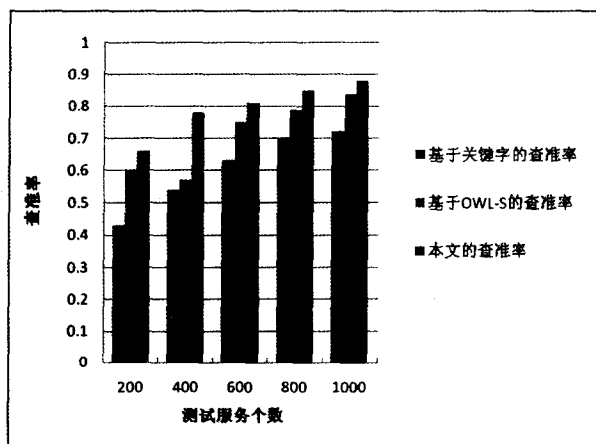


图 3 三种方法的查准率比较

5 结束语

文中针对服务发现提出了基于聚类的 QoS 语义 Web 服务发现方法,首先采用聚类技术,基于服务描述和接口属性两方面将相似服务聚合在一起,以此来提高发现效率。基于此基础上,针对服务请求者对服务

质量的需求,对满足需求的候选服务进一步筛选,从而提高了服务发现的查找效率和查准率。实验结果证明了本文服务发现方法的有效性。

在今后的工作中,将进一步提高基于 QoS 的服务发现算法的效率。此外,实现服务的模糊聚类也是将来的一个研究重点。

参考文献:

- [1] Nayak R, Lee B. Web service discovery with additional semantics and clustering[C]//Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. Washington, DC: [s. n.], 2007: 555-558.
- [2] Ram Sudha, Hwang Yousub, Zhao Huimin. A clustering based approach for facilitating semantic Web service discovery [C]//Proceedings of the 15th Annual Workshop on Information Technologies & Systems. Las Vegas, USA: [s. n.], 2006: 1-6.
- [3] Sudha Rajagopal, Thamarai Selvi S. Semantic grid service discovery approach using clustering of service ontologies//Proceedings of IEEE TENCON 2006. Hong Kong, China: [s. n.], 2006: 1-4.
- [4] 孙萍, 蒋昌俊. 利用服务聚类优化面向过程模型的语义 Web 服务发现[J]. 计算机学报, 2008, 31(8): 1340-1353.
- [5] 胡建强, 邹鹏, 王怀民, 等. Web 服务描述语言 QWSL 和服务匹配模型研究[J]. 计算机学报, 2005, 28(4): 505-513.
- [6] Amos T. Feature of similarity[J]. Psychological Review, 1977, 84(4): 327-352.
- [7] 钟福金. 语义 Web 服务发现及其应用研究[D]. 合肥: 合肥工业大学, 2005.
- [8] Rodriguez A, Egenhofer M. Determining Semantic Similarity Among Entity Classes from Different Ontologies [J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(2): 442-456.
- [9] 吴健, 吴朝晖, 李莹. 基于本体论和词汇语义相似度的 Web 服务发现[J]. 计算机学报, 2005, 28(4): 595-602.
- [10] 刘克非, 王红, 王卫玲. 基于语义相似度的 Web 服务发现研究[J]. 计算机技术与发展, 2007, 17(2): 16-19.
- [11] 梁循. 数据挖掘算法与应用[M]. 北京: 北京大学出版社, 2006.
- [12] Shahpurkar S S, Sundareshan M K. Comparison of self-organizing map with k-means hierarchical clustering for bioinformatics applications[C]//International Joint Conference on Neural Networks. Hungary: IEEE Press, 2004: 1221-1226.

《计算机技术与发展》欢迎投稿, 欢迎订阅!