一种基于能力的模糊Web服务聚类及预检索算法

赵文栋 张 进 彭来献 田 畅
(解放军理工大学通信工程学院 南京 210007)

摘 要 实现对Web服务的自动聚类,是提高Web服务发现速度的有效方式之一。针对常用聚类算法在实现服务聚类时需要获取网内所有服务或通过服务训练集来发掘领域内服务特征,不适用于动态服务环境的问题,提出了服务能力的概念,并给出了服务能力描述及计算的方法。借助本体技术,提出了一种基于服务能力的聚类算法。无需先验知识或服务间相似度的比较,该算法可将服务能力及功能相似的服务聚类在一起。在此基础之上,提出了一种服务预检索算法。理论分析及仿真结果表明,聚类算法可有效地反映领域内服务基于功能的聚类特征,预检索算法可有效地滤除无关服务,提高服务检索效率。

关键词 Web服务,本体,聚类,检索,能力
中图法分类号 TP393 **文献标识码** A

Ability-based Fuzzy Web Service Clustering and Searching Algorithm

ZHAO Wen-dong ZHANG Jin PENG Lai-xian TIAN Chang
(Institute of Communication Engineering, PLA University of Science and Technology, Nanjing 210007, China)

Abstract With the rapid growth of Web services and the need of quickly finding the right services, automatically clustering Web services become exceedingly important and challenging. The main drawback of current clustering algorithms is needed to get total services or a service training set to mine the service attribute. This does not suit for the dynamic distributed environment. In order to address this problem, this paper first raised the concept of service ability and gave a method to compute the service ability. By means of ontology, this paper presented a novel service clustering algorithm based on service ability. Without priori knowledge or similarity computation between any two services, this clustering algorithm can cluster the services with similar function and service ability. On the basis of this algorithm, a service searching algorithm was proposed. Experimental and theoretical results show that the clustering algorithm can reflect the services' function clustering feature effectively and searching algorithm can filter a lot of unrelated service.

Keywords Web service, Ontology, Cluster, Search, Ability

1 引言

随着Web服务的普及、服务数量的急剧增加,如何在海量服务群中发现满足需要的服务成为影响Web服务进一步发展的瓶颈。

Web服务的检索技术主要分为基于关键词的服务查找技术与基于语义的服务查找技术两类。基于语义的服务查找技术利用本体技术,增强了对Web服务的功能、行为的语义描述,通过逻辑推理、相似度计算等方法来实现Web服务的检索,解决了基于关键字的服务匹配算法准确性不高的问题。

虽然基于语义的服务描述及检索技术提高了服务检索的准确性,但是匹配算法普遍存在设计复杂、计算量大的问题。服务器如果每收到一个服务请求,都要与本地存贮的所有服务完成相似度计算,则会引入巨大的计算量,特别是在Web服务数量剧增的情况下,采用逐个比对的方式会使服务器的检索效率明显降低。

为了提高服务的检索效率,通常将服务聚类[1-10]作为其它复杂服务匹配算法的预处理算法。通过挖掘各服务描述间的关联关系,将相似的服务聚类在一起,从而减少服务的比对范围,以达到减少匹配计算量、提高匹配速度的目的。如文献[2]提出了一种基于特征服务的聚类算法,该算法首先根据服务的内容,将网络中的服务划分成不同的服务簇,并保证簇内服务功能基本相似,其相似度的判定是通过计算本体中各概念的距离来实现的,并以特征服务的方式表示每个簇。服务器收到某个服务发布时,首先计算此服务与各特征服务的相似度,并依据计算结果将服务归入不同的簇,但文献中并没有给出特征服务的产生办法。从作者给出的处理过程可以推断出,服务的聚类过程是属于集中式的,即事先必须知道网内各簇的划分情况。文献[3,4]提出了一种基于神经网络的服务聚类方法。该方法首先对服务注册中心的服务进行语义分析,并获取服务的语义特征向量,然后通过神经网络生成一组标识服务——元服务。当用户查找服务时,首先计算服务请

到稿日期:2012-07-25 返修日期:2012-12-22 本文受江苏省自然科学基金(BK2010103)资助。
赵文栋(1972—),男,博士生,副教授,主要研究方向为计算机网络,E-mail:nj_mouse@163.com。

求与各元服务的余弦相似度，然后按相似度进行排序，并返回给用户前 N 个服务。文献[5]在设计服务聚类算法时，不但考虑了服务的功能，而且考虑了各服务描述词汇的权重；首先采用 PLSA(基于概率的语义分析)或 LDA(隐含狄利克雷分配)的方法，对大量的语义描述文件进行分析，以获取每个词汇的权重；然后生成服务转换矩阵，服务仓库中的服务根据服务转换矩阵可生成不同的服务描述向量，当收到用户的服务请求时，同样根据服务转换矩阵生成请求描述向量；再计算各向量间的多维角，并以此来对服务进行聚类。文献[6]同样采用向量的方法实现对服务的描述，然后采用经典的 K-means 算法对服务进行聚类。文献[7]采用 Minkowski 空间进行服务聚类。文献[8,9]针对一种服务间相似度的估算方法不能很好地反映服务间相似程度的问题，提出了两种基于多种相似度评估标准的服务聚类算法，即首先根据多种相似度标准计算出两服务间的相似度，然后计算加权值，最后通过计算此加权值在输入\输出空间中的分布情况来实现聚类。文献[10]则针对 WSDL 不支持语义描述的问题，对 UDDI 场景首先提取服务描述中的语义并生成服务描述向量，然后通过模式匹配算法实现服务的聚类。

上述聚类算法存在的主要问题是：

1)主要应用于逻辑集中式的服务检索环境。如 UDDI，对于分布式的服务检索环境则不适用。

2)针对基于特征服务聚类的算法，在生成特征服务时，需要获取全网的服务描述文件或者需要大量的训练服务。这适用于服务种类相对固定、变化不大的环境，而对于某些专用服务领域，如战场环境下的军事通信网络或分布式服务管理系统，则不适用。

3)当服务簇生成后，服务的种类相对固定，如果发生变化，则需要重新计算并划分簇。

4)在服务检索过程中，服务器收到用户的服务请求后，首先需要确定请求与哪一类服务比较相似，其计算过程同样属于全局计算过程。如果服务的种类比较多，则其相似度判断过程同样会引入较大的计算量。

鉴于以上存在的问题，本文借助本体技术，提出了一种基于服务能力的服务聚类及预检索算法，该算法通过简单的计算可滤除大量无关的服务，其优点是：

1)各服务在聚合时无需相互比较，即可保证服务能力相近的服务以很大的概率聚合在一起。

2)可支持服务种类的动态变化。

3)服务器在根据用户请求定位服务簇时，无需预先计算与每个服务簇的相似度。

本文第 2,3 节给出了基于本体的服务功能描述及相关定义；第 4,5 节给出服务聚类及预检索算法及性能仿真分析；最后是对全文的总结。

2 服务功能本体及相关定义

从 Web 服务的匹配过程来看，服务的匹配首先是功能的匹配。即查找的服务是否在功能上能满足用户的服务请求。为了简化问题，本文只考虑服务的功能性描述，现构造某领域服务功能本体，如图 1 所示。

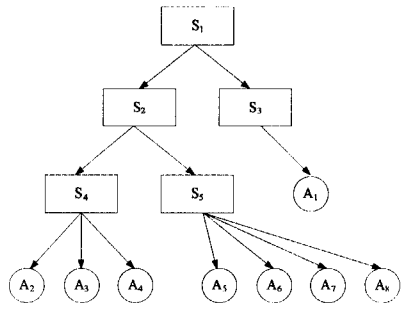

图 1 服务功能本体示意图

图 1 中“○”表示该领域的原子服务功能描述，“□”表示复合服务功能描述，“↓”表示包含关系。在图 1 中以下关系成立：

$$S_2=S_4\cup S_5=A_1\cup A_2\cup\cdots\cup A_8$$

在构建功能本体时，需保证在语义层面上各原子服务功能间是相互独立的。

在给出领域功能本体描述后，此领域所需的所有原子服务功能可以用全集 U 表示，$U=\{A_1,A_2,A_3,\cdots,A_n\}$，其中 A_i 表示一个原子服务功能。本领域所能提供的服务，依据功能本体可表示为各原子服务功能的组合。如，某节点发布服务 $S=S_3+S_4$，则依据图 1，此服务可表示为 $S=A_1+A_2+A_3+A_4$。每个服务功能均采用服务向量进行描述。同时，为了消除服务描述中的二意性，本文假设各原子服务都是可偏序排序的。如果服务 S_1 与服务 S_2 包含相同的原子服务，则其最终会生成相同的服务表示向量。

如图 1 所示，假设服务器 A 发布的服务描述为 $S_1=\{s_3,s_4\}$，服务器 B 发布的服务为 $S_2=\{s_4,A_1\}$，则服务 S_1，S_2 的向量表示为：

$$S_1=[1,1,1,1,0,0,0,0]$$

$$S_2=[1,1,1,1,0,0,0,0]$$

同理，对于任意服务请求的功能需求描述表示为 $Q=\{s_i\mid\forall s_i\in U\}$，其中 s_i 表示服务请求者所请求的服务项。即如果某用户请求服务 $Q=S_5$，则其请求的服务功能可表示为 $Q=A_5+A_6+A_7+A_8$，其服务功能向量表示为：

$$Q=[0,0,0,0,1,1,1,1]$$

为了便于描述，现给出相关定义如下。

定义 1(服务向量) 设领域功能本体共包含 n 个原子服务功能，则其描述服务的向量描述的长度为 n，如果此服务包含某原子服务功能，则对应位置为“1”。

定义 2(服务能力) 服务提供者所能提供的服务中所包含的原子服务功能的数目。它表示每个服务提供者的固有服务提供能力，并用$|S_i|$表示。如果某服务 S_i 包含 n 个原子服务，则$|S_i|=n$。

同时，定义运算如下：

1)$S_1>S_2$。对于任意的服务 S_1 及 S_2，$S_1>S_2$ 表示 S_1 的固有服务能力大于 S_2 的固有服务能力，即 S_1 包含的原子服务个数大于 S_2 包含的原子服务个数。

2)$S_1\cup S_2$。表示服务 S_1 与 S_2 所包含原子服务的并集。

3)$S_1\cap S_2$。表示服务 S_1 与 S_2 所包含原子服务的交集。

4) S_1-S_2。表示属于 S_1 但不属于 S_2 的原子服务集合。

定义 3(服务匹配) 对于任意服务请求 Q,如果式(1)成立,则表示服务 S 与请求 Q 匹配成功。

$$\exists S \wedge \forall s_i \in Q \Rightarrow s_i \in S \tag{1}$$

例如:$Q=[1,0,0,1,0,0,0]$, $S=[1,1,0,1,0,1,0]$,则 Q 与 S 匹配成功。

定义 4(服务功能相似度) 服务 S_1 与 S_2 的相似度定义为:

$$L=\frac{S_1 \cap S_2}{S_1 \cup S_2} \tag{2}$$

为便于描述,简记为 L^{S_1,S_2}。

3 服务能力标识计算定义及特点分析

对于任意给定的服务 S_i,其服务能力标识,即服务能力的数值化表示采用式(3)计算:

$$S_i^A=\sum_{i=1}^{n} v_i \cdot \lg p_i \tag{3}$$

式中,在累加时 n 表示本领域所有原子服务的总数。v_i 表示服务 S_i 的表示向量的第 i 位的取值。p_i 表示所选定的素数序列的第 i 个素数。素数序列的长度与本领域服务总数相同,同样取 n,且此序列是按升序排序的。关于素数序列的选取办法,将在下文介绍。

根据定义的“>”运算规则,素数序列的选取应满足以下原则:

设选取的素数序列为 P,则式(4)必须成立:

$$\begin{cases} \forall p_i, p_j, p_k \in P \Rightarrow p_i \cdot p_j > p_k \\ \lg(\mathrm{Max}(P))-\lg(\mathrm{Min}(P)) \ll \lg(\mathrm{Min}(P)) \end{cases} \tag{4}$$

例如,在素数 507383 与 640307 之间共有 10000 个素数,且最小的两个素数 507383 与 507401 的乘积大于 640307,且 $\lg(640307)-\lg(507383)\approx 0.1 \ll \lg(507383)\approx 5.7053$。如果指定领域的原子服务功能数目小于 10000 个,则可选用此序列。

根据式(3)、式(4),有以下定理成立:

定理 1 如果服务 S_i 的服务能力大于服务 S_j 的服务能力,则以下结论将以很大概率成立:

$$\forall S_i, S_j, S_i > S_j \Leftrightarrow S_i^A > S_j^A \tag{5}$$

证明:根据定义 1,如果服务 S_i 的服务能力大于服务 S_j 的服务能力,则有 $|S_i|>|S_j|$,依据式(3)及限制条件(4),可推出,在服务向量各项计算累计误差范围内,式(5)成立。其累积误差可通过素数序列的选取得以控制。

定理 2 如果服务请求 Q 与服务 S 匹配,则以下结论成立:

$$\begin{cases} \forall Q, S, Q \subseteq S \\ \Rightarrow S^A - Q^A \geqslant n \cdot \mathrm{Min}(\lg p_i) \end{cases}$$

式中,n 表示 S 与 Q 的原子服务数量差值。

证明:如果服务请求 Q 与服务 S 匹配,则依据定义 2 有,Q 请求的任一原子服务,S 均可满足,且在非精确匹配的情况下,至少存在一个属于服务 S 且不属于服务请求 Q 的原子服务。依据计算式(3)、式(4)并综合考虑精确匹配的情况,上式得证。

推论 1 如果服务 S_i 与服务请求 Q_i 能够精确匹配,则以下条件成立:

$$\begin{cases} \forall S_i, Q_j, S_i = Q_j \\ \Rightarrow S_i^A = Q_j^A \end{cases}$$

定理 3 对于任意给定的服务 S_i, S_j, S_k,如果 S_i, S_j, S_k 的服务能力相同,且 $L^{S_i,S_j} > L^{S_i,S_k}$,则以下结论将以很大概率成立:

$$\begin{cases} \forall S_i, S_j, S_k, |S_i|=|S_j|=|S_k| \wedge L^{S_i,S_j} > L^{S_i,S_k} \\ \Leftrightarrow |S_i^A - S_j^A| < |S_i^A - S_k^A| \end{cases}$$

证明:对于任意给定的服务 S_i, S_j, S_k,如果 3 者服务能力相同且 $L^{S_i,S_j} > L^{S_i,S_k}$,则依据定义 3,可推出 $|S_i \cap S_j| > |S_i \cap S_k|$。由素数序列选取条件 5 及式(3),在素数计算累计误差范围内,有 $|S_i^A - S_j^A| < |S_i^A - S_k^A|$。

同样,如果某领域内各服务具有良好的聚类特性,则同一簇内各服务的功能描述向量具有类似的原子功能分布特征,而不同簇之间的服务描述向量的原子功能分布特征差别较大。如果 $|S_i^A - S_j^A| < |S_i^A - S_k^A|$,则根据式(3)可得出,在素数累计误差范围内,$S_i$ 与 S_j 具有相似功能分布特性的可能性要大于 S_i 与 S_k 具有相似功能分布的可能性。即可得出以很大的概率 $L^{S_i,S_j} > L^{S_i,S_k}$。

4 基于能力服务聚类算法

4.1 算法描述

基于服务能力的服务聚类算法的伪代码描述如图 2 所示。

```
Input: Serivce S_i;
       rank para α;
Output: Clustering Service Array
Begin
1. for any Service S_i compute S_i^A
2. Insert S_i into Service Array sorting by S_i^AL
3. Cluster the Service
   if((S_i^A - S_{i-1}^A) < (S_{i+1}^A - S_i^A)) and ((S_i^A - S_{i-1}^A) < α)
   then
     put Service S_i into Cluster that S_{i-1} belongs to
   if ((S_i^A - S_{i-1}^A) > (S_{i+1}^A - S_i^A)) and ((S_{i+1}^A - S_i^A) < α)
   then
     put Service S_i into Cluster that S_{i+1} belongs to
   if ((S_i^A - S_{i-1}^A) > α) and ((S_{i+1}^A - S_i^A) < α)
   then
     create a new cluster and put S_i in
End
```

图 2 基于能力的服务聚类算法

本算法的主要思想是利用服务能力标识完成网络中各服务基于能力的聚类。服务器收到一个服务描述文件时,首先依据功能本体获取此服务的表示向量,并依据式(3)计算其服务能力标识,再按数值大小将服务插入队列,完成排序后,将此标识值与队列中相邻两服务能力标识进行比较,如果与相邻的两标识间的差值均大于算法设计阈值 α,则新建一个簇,否则,将之加入到差值较小的服务所属簇中。依据本文第 3 节的分析,以下结论成立:

1)在服务队列中,各服务按服务能力的升序排序且服务能力相同的服务聚合在一起。

2)在服务能力相同的服务簇中,服务能力相似的服务以很大的概率聚合在一起。

下面以仿真的方式验证本服务聚类算法的准确性及由于素数累计误差给服务聚合带来的影响。

4.2 算法性能仿真

4.2.1 服务能力与服务能力标识变化一致性仿真

本节仿真的目的主要是验证基于各服务的服务能力排序

与基于服务能力标识排序的一致性。采用的仿真工具是matLab,仿真参数如表1所列。

表1 仿真参数

最大原子服务数	仿真次数	最小素数	最大素数
1000	5000	44123	55009

各服务描述通过matLab随机产生,每个服务中所包含原子服务的数目及项目符合正态分布。

图3为排序后各服务的服务能力统计示意图,图4为在相同的排序结果情况下各服务的服务值统计示意图。由图3及图4可以看出,二者的变化趋势完全相同。经统计发现,在仿真的结果中,未出现 $S_1>S_2$ 但 $S_1^{AL}<S_2^{AL}$ 的情况。

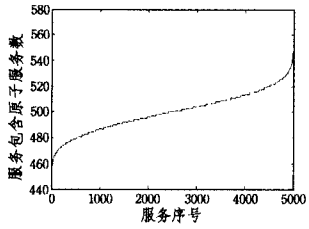

图3 服务包含原子服务能力排序统计示意图

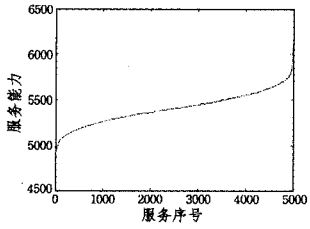

图4 服务能力标识排序统计示意图

仿真结果表明,根据具体的服务领域,在选取素数序列适当的情况下,各服务的服务能力的排序与基于各服务能力标识的排序一致,可依据服务能力标识判断服务能力的大小。

4.2.2 服务聚类仿真测试

本节仿真的目的是测试当服务的服务能力相同时,服务功能相似的服务的聚合情况。首先,仿真的是在所发布的服务聚合特性较理想情况下的服务聚合情况。仿真参数如表2所列。

表2 服务聚合仿真参数

最大原子服务功能数	每个服务包含原子服务数	仿真次数	服务聚类预设值
3030	40	7529	50

仿真时,其服务功能向量长度为3030,将服务向量随机分为50个功能簇,相邻簇之间功能元素最多交叉20个,每个簇包含的原子服务功能数为50~70个,以每个簇为单位,随机生成100~200个服务描述向量,每个向量包含40项原子服务功能。在仿真的过程中,共生成7529项服务,按服务能力标识聚合后,其统计结果如图5所示,图6是图5前400项服务的聚合情况的放大图。

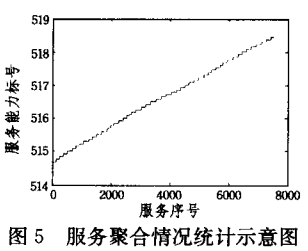

图5 服务聚合情况统计示意图

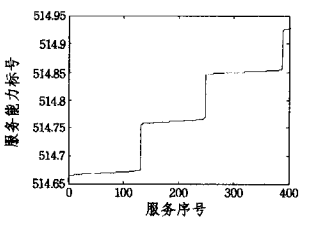

图6 服务聚合部分情况统计示意图

由图5及图6可以看出,服务原子功能落在相同簇内的服务能力标识值非常接近,不同簇间的服务能力标识具有较明显的差别。由此可以看出,采用本文给出的服务聚类算法在服务聚合特性较理想的情况下,能很好地反映出所描述的服务聚合特性。

根据式(4)素数序列的选取条件,本文中采取的素数序列是单调递增的,在计算服务能力标识值时,能出现式(6)描述的情况:

$$\begin{cases} \lg p_1<\lg p_2<\lg p_3<\lg p_4 \\ \lg p_1+\lg p_4\approx\lg p_2+\lg p_3 \end{cases} \tag{6}$$

这将会影响到本文服务聚类算法的性能,我们对此情况做了仿真。其它仿真参数不变,任意选取了8个连续的簇,并生成了4类服务,其中第一类服务来自簇1及簇8,第二类服务来自簇2及簇7,以此类推。服务聚合情况如图7所示。

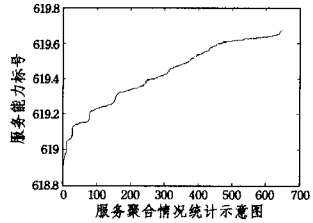

图7 服务聚合情况统计示意图

由图7可以看出,由于计算过程的误差,在这种情况下,服务聚合的效果并不好,通过对排序结果进一步分析发现,虽然聚合在一起的服务功能比较相似,但是并没有较好地生成4个服务簇。这将对进一步的服务过滤带来困难。

在实际应用中,如果出现这种情况,则表明大量领域相关的原子服务功能在编号时并没有编在一起,从而加大了计算过程中的累计误差,这种情况可以通过调整本体树中原子服务功能的排列顺序,得以较好的解决。

5 基于能力的服务检索算法

5.1 算法描述

基于能力的服务检索算法的伪代码描述如图8所示。

Input: Service request Q_i^A;
the service number N that user want to pre-select
Output: Top N services that most like the request
Begin
1. Set set I = ∅
2. search the queue
3. if $Q_i^A = S_j^A$
put the Service into set I
end
Count = 1
4. while selected service number < N
$S_{min}^A = Q_i^A + \sum_{k=LabMin-\alpha-Count}^{LabMin-\alpha} \log P_k$
$S_{max}^A = Q_i^A + \sum_{k=LabMax+\alpha}^{LabMax+\alpha+Count} \log P_k$
scan the service in $[S_{min}^A, S_{max}^A]$
if $Q \subseteq S$
put the service in set I.
end
Count ++
end
End

图8 基于能力的服务查找算法

本算法的选择标准是,首先选择与用户服务请求功能精

确匹配的服务，然后选择可满足用户服务请求且其服务能力与请求最接近的服务。

服务器收到服务请求后，首先依据定义 3 计算 Q_i^A，然后查找服务能力队列中是否有与 Q_i^A 精确匹配的服务，如果存在，则将该服务放入集合 I。同时考虑到，本服务虽然在功能上与请求可精确匹配，但是其输入、输出参数不一定能满足要求，因此依据用户输入的初选服务数量作进一步的选择。

在做进一步选择时，首先计算服务能力选择范围。根据定义 3 服务能力的计算方法及假设条件，在计算能力的查找范围时，采用如下计算方法：

$$S_{\min}^A = Q_i^A + \sum_{k=\mathrm{LabMin}-\alpha-i}^{\mathrm{LabMin}-\alpha} \log P_k$$
$$S_{\max}^A = Q_i^A + \sum_{k=\mathrm{LabMax}+\alpha}^{\mathrm{LabMax}+\alpha+i} \log P_k \quad (7)$$

式中，LabMin 表示服务请求 Q 中最左边“1”对应向量位的编号，LabMax 表示服务请求 Q 中最右边“1”对应向量位的编号。如：$Q=[0,0,0,0,1,0,1,0,1,0,0,1,0,0,0,0]$，则 LabMin=5，LabMax=12。

α 表示服务的扩展范围，可由用户根据实际情况给出。

判断服务 S 是否可以满足服务请求 Q，可采用如下计算方法：

if $(Q\&S)\otimes Q=[0]\Rightarrow Q\subseteq S$

式中，“&”表示服务请求向量 Q 与服务向量 S 的按位与运算，“⊕”表示向量间的异或运算，“[0]”表示全“0”向量。

依据定义 3 及表达式(3)、(4)可知，当算法运行结束后，其预选集合中的 N 个服务是与用户请求最相似的 N 个服务。

5.2 算法性能仿真

仿真参数如表 3 所列。

表 3 仿真参数

服务向量长度	用户请求数	服务数	服务簇数
302	50	863	5

实验中，服务向量的长度为 302，所有的服务向量及服务请求向量均由 MatLab 随机产生，各服务及请求均有良好的聚合特性。为了便于验证算法的正确性，在仿真时，为每个服务请求创建了 10 个匹配服务，匹配条件满足定义 3。预选集合的大小为 5。

图 9 是各服务请求在检索服务的过程中的查找次数统计示意图。由统计结果可以看出，每个服务请求的查找次数在 5～103 范围之内，远远小于服务数目 863。这表明，在服务的预检索过程中，大量的无关服务被滤除。同时，经统计验证，所选择的服务均符合定义 3 的要求。

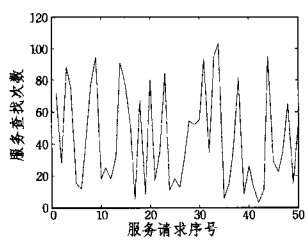

图 9 查找次数统计示意图

结束语 随着 Web 服务数量的急剧增加和应用的日益普及，如何快速、准确和高效地发现满足用户需求的服务已经成为制约 Web 服务发展的瓶颈之一。通过服务聚类的方式，减少在服务检索过程中的搜索空间，提高服务的检索效率是目前常用的一种行之有效的方法。然而，目前的算法主要是针对集中式的服务检索架构设计的，且在服务聚类的过程中，往往需要获取全局的服务描述，或通过大量的服务训练序列来预先确定服务的种类。这类算法不适用于分布式计算环境或服务种类动态变化的应用场景。针对这一问题，本文首先提出了服务能力的概念，并给出了服务能力数据表示的计算方法，并在此基础之上，借助本体技术，给出了基于服务能力的服务类算法，该算法无需计算服务间的相似度，即可实现服务能力及功能相似的服务聚类。并在此基础之上给出了服务检索的预处理算法，采用此算法，可有效地滤除大量无关服务，缩小服务的检索空间，提高服务的匹配效率。下一步的研究工作将考察本算法在分布式网络服务查找环境中的应用。

参 考 文 献

[1] Gao Hao, Yan Jun, Mu Yi. Web Service Selection based on Similarity Evaluation[C]// IEEE International Conference on Services Computing. 2011:322-329

[2] 张莹，黄厚宽，杨冬，等. 基于 Chord 的带有 QoS 的语义 Web 服务发现方法研究[J]. 电子与信息学报，2009，31(9)：711-714

[3] 陈蕾，杨庚，张迎周，等. 基于核 Batch SOM 聚类优化的语义 Web 服务发现机制研究[J]. 电子与信息学报，2011，33(6)：1307-1312

[4] Chen Lei, Yang Geng, Zhang Ying-zhou, et al. Web Services Clustering Using SOM based on Kernel Cosine Similarity Measure[C]//2010 2nd International Conference on Information Sicence and Engineering. 2010:846-850

[5] Cassar G, Barnaghi P, Moessner K. A Probabilistic Latent Factor Approach to Service Ranking[C]// 2011 IEEE International Conference on Intelligent Computer Communication and Processing. 2011:103-109

[6] Xie Ling-li, Chen Fu-zan, Kou Ji-song. Ontology-Based Semantic Web Services Clustering[C]// 2011 IEEE International Conference on Industrial Engineering and Engineering Management. 2011:2075-2079

[7] Gholamzadeh N, Taghiyareh F. Ontology-based Fuzzy Web Services Clustering[C]// International Symposium on Telecommunications. 2011:721-725

[8] Skoutas D, Sacharidis D, Simitsis A, et al. Ranking and Clustering Web Services Using Multicriteria Dominance Relationships [J]. IEEE Transactions on Services Computing, 2010, 3(3):163-176

[9] Skoutas D, Sacharidis D, Simitsis A, et al. Top-k Dominant Web Services Under Multi-Criteria Matching[C]// Proceedings of the 12th International Conference on Extending Technology: Advances in Database Technology. 2009:898-909

[10] Paliwal A V, Shafiq B, Vaidya J, et al. Semantics-Based Automated Service Discovery[J]. IEEE Transactions on Services Computing, 2012, 5(2):260-274