一种基于点击率索引的语义Web服务发现方法

张彦彦 熊海灵 朱明强

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 为了解决现有语义Web服务发现方法查询效率不高的问题,提出了一种基于点击率索引的Web服务发现方法。通过为统一描述、发现和集成注册库中点击率较高的Web服务建立索引,使得可以在查询过程中首先匹配访问率较高的服务,实现用户所需服务的快速发现。实验结果表明,该方法既可保证服务发现的准确率,又可明显提高服务发现的效率。

关键词 语义Web服务,索引,服务发现,点击率

中图法分类号 TP393.09 **文献标识码** A

Semantic Web Service Discovery Method Based on Click Rate Index

ZHANG Yan-yan XIONG Hai-ling ZHU Ming-qiang

(School of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract A Web service discovery method based on the click rate index was presented to deal with lower efficiency of query in current semantic Web service discovery methods. By indexing the Web services with higher click-through rate in the universal description discovery and integration repository, we could firstly match the services with a higher access rate in the query process. Experimental results show that the method can guarantee the accuracy and improve the service discovery efficiency obviously.

Keywords Semantic Web service, Index, Service discovery, Click rate

随着Internet技术的快速发展与Web服务的日益增多,如何在众多的Web服务中定位到满意的服务是实现服务共享、复用的前提。传统的通过对UDDI(Universal Description Discovery and Integration)上的注册信息进行关键词查询来发现服务的服务匹配方法,在查准率和查全率方面都无法达到令人满意的效果[1]。语义Web服务是在Web服务中加入语义信息,通过服务语义的匹配来发现服务。为了实现基于语义的服务查询,目前多采用W3C组织提出的语义Web服务描述语言OWL-S与UDDI结合的Web服务发现框架[2],该框架很好地实现了基于语义的服务查询,可以提高Web服务的查全率和查准率。

目前,关于语义Web服务发现的研究主要集中在提高查准率和查全率上[2-5]。这些方法在服务发现的准确率方面有明显提高,但却没有考虑到服务查找匹配的效率。这样,在文本Web服务数量增多的情况下,如何快速地从众多服务中找到目标服务就成为服务发现的难点。针对这个问题,徐德智等提出了带Cache的语义Web服务发现方法,该方法在一定程度上提高了服务发现的效率,但是命中率不是很高[6]。这主要是因为在查询过程中,往往需要匹配UDDI库中的大多数服务方能找到符合要求的服务,这无疑大大降低了查询的效率。而带缓存的语义Web服务发现方法,由于缓存中保存的只是近期被访问的服务而不能保证缓存的命中率。

针对上述问题,我们通过对服务查询特点的分析,提出了一种基于点击率索引的Web服务发现方法,即通过在传统Web服务发现框架中增加索引管理模块,为UDDI注册库中点击率较高的Web服务建立索引,使得可在查询过程中首先匹配点击率较高的服务,从而提高Web服务查询的效率,而且索引中只是保存现有服务的ID,占用的内存空间较少,不需要为维护服务的一致性花费时间。

1 基于索引的语义Web服务发现框架

1.1 语义Web服务描述

OWL-S是Web服务的本体语言,它包括3个组件:Service Profile描述服务的功能;Service Model描述服务的具体实现细节;Service Grounding描述如何访问服务。结合Web服务的本体描述,本文依据服务的点击率搜索服务的算法特点,及现有的Web服务描述模型[1],为其增加服务点击率的描述。我们可以利用定义1来确定本文的Web服务描述模型。

定义1 一个Web服务可以描述为$WS=\langle D,F,C\rangle$

(1)D为服务的基本描述,包括服务分类、服务ID、服务名称、服务文本描述、服务提供者、版本等。

本文受国家自然科学基金(40740420660),西南大学博士基金(SWUB2008073),中央高校基本科研业务费一般项目(XDJK2010C032)资助。

张彦彦(1987—),女,硕士生,主要研究方向为数据库与信息系统,E-mail:yy67787711@126.com;**熊海灵**(1971—),男,博士,副教授,硕士生导师,主要研究方向为形式语言与自动机、数据库与信息系统,E-mail:xionghl@swu.edu.cn(通信作者);**朱明强**(1985—),男,硕士生,主要研究方向为数据库与信息系统。

(2)F 为服务功能描述，包括输入参数、输出参数、前提和结果。

(3)C 为一个 Web 服务的点击率，该值反映了一个 Web 服务的重要性。

1.2 带索引的语义 Web 服务发现框架

根据点击率是一个服务重要性的最好预测[7]和程序局部性原理，服务请求者对于某些 Web 服务的使用次数要大于其他服务。这说明服务请求者对服务的访问需求具有很大的时间和空间局限性，点击率较高的服务可能会被很多用户经常访问，访问过的服务可能马上会被再次访问。基于以上思想，可以为 UDDI 注册库中点击率较高的 Web 服务建立索引，使得可以优先在索引中匹配服务请求者所需的服务，从而快速找到一些使用率比较高的服务，比如一些经常被使用的网站，这样就保证了服务访问的空间局限性。为了实现基于点击率的索引，在以往的 OWL-S/UDDI 框架[5]的基础上，提出了具有点击率索引机制的 Web 服务发现框架。该框架的结构如图 1 所示。

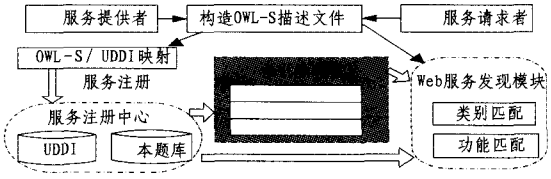

图 1 带索引的 Web 服务发现框架

该 Web 服务发现框架主要包括以下几个模块：

(1)OWL-S 服务文件构造：该模块实现将用户与服务提供者提交的服务信息转换成语义 Web 服务的描述文件 OWL-S。

(2)OWL-S/UDDI 映射：该模块应用 OWS-S 与 UDDI 一一对应的机制将服务描述文件中的 Service Profile 内容置入 UDDI。

(3)UDDI 和本题库：该模块用于存储注册的 Web 服务。本体库是指语言学本体库。

(4)索引管理：该模块用于保存点击率较高的 Web 服务的 ID，并采取一定的更新机制对索引内容进行更新。由于索引文件中保存的是服务的 ID，因此不必为同步索引与 UDDI 库中的服务花费时间和资源。

(5)Web 服务发现：该模块是 Web 服务发现框架的核心，它将 Web 服务的发现分为两步，首先取索引文件中指向的服务与服务请求者所需的服务进行服务匹配，服务匹配过程分为两步：1)服务的类别匹配，2)服务的功能匹配。如果索引文件中不存在满足要求的服务，再从 UDDI 中选择服务进行匹配。

1.3 索引更新机制

若索引文件中不存在满足条件的服务，Web 服务发现模块将从 UDDI 库中匹配到符合服务请求者需求的服务 S，将该服务返回给用户后，需要对索引文件进行更新。更新的过程中，不管当前服务点击率的大小，都将索引文件中点击率最低的服务替换为当前服务，这就在一定程度上保证了服务访问的时间局限性。同时，为了避免一些不再被使用的服务长期占用索引空间，在将新的服务放入索引文件时，将索引文件中原有服务的点击率减 1。具体更新流程如图 2 所示。

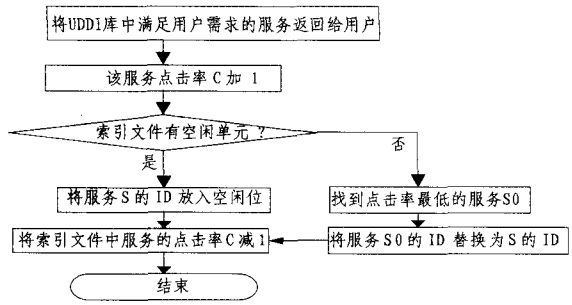

图 2 索引更新流程图

我们可以看出，索引文件的建立是在用户提交查询之前完成的，同时索引文件的更新是在返回满足用户需求的服务后进行的，所以索引文件的建立与更新与用户所需服务的检索过程是相对独立的，对索引文件的维护不会影响用户的查询效率，这就为查询效率的提高提供了保证。同时由于我们的索引文件是根据点击率的高低建立的，因此服务查找的命中率就在一定程度上得到了保证。

2 语义 Web 服务匹配

Web 服务发现是建立在服务匹配的基础上的，通过将用户请求服务的信息与服务注册信息进行匹配，发现满足用户需求的服务。

本文借鉴文献[2]的关键字相似度值确定方法，当提供服务的一个输入参数 P_i 和请求服务的一个输入参数 R_i 在 UDDI 中同一个本体中定义时，有以下 4 种可能出现的匹配等级。

(1)精确匹配(exact)：Exact(R_i,P_i)=$\{R_i$ equivalent of $P_i\}$

(2)包含匹配(subsume)：Subsume(R_i,P_i)=$\{R_i$ subclass of $P_i\}$

(3)插入匹配(plug in)：Plug in(R_i,P_i)=$\{P_i$ subclass of $R_i\}$

(4)匹配失败(Fail)：Fail(R_i,P_i)=$\{P_i$ Disjoint Class $R_i\}$

由于采用匹配等级不能给出两个服务间的精确匹配度，因此利用了表示本体的分类树中两个不同节点之间的距离来衡量节点之间的相似度。对于分类树中任意两个节点 p、r，如果 p 是 r 的父节点或祖先节点，则称 p 到 r 有路径。这样我们可以将两个节点之间的距离 $distance(p,r)$ 定义如下：

(1)如果 p 与 r 为树中相同节点，则 $distance(p,r)=0$。

(2)如果从节点 p 没有路径到达 r，且从节点 r 也没有路径到达 p，则 $distance(p,r)=\infty$。

(3)如果从节点 p 到达节点 r 有路径，则 $distance(p,r)$ 为从 p 到达 r 的路径的长度。

(4)如果从 r 到达 p 有路径，则 $distance(p,r)$ 为从 r 到达 p 的路径长度的负数。

令节点 p 表示概念 n_p，节点 r 表示概念 n_r，则可以定义以下公式来计算树中两个节点所表示的概念的精确匹配度：

$$sim(n_p,n_r)=\begin{cases}1, & distance(p,r)=0\\ \dfrac{1}{distance(p,r)+1}, & distance(p,r)>0\\ \dfrac{1}{2}+\dfrac{1}{|distance(p,r)|+1}, & distance(p,r)<0\\ 0, & distance(p,r)=\infty\end{cases}\quad(1)$$

2.1 类别匹配

语义 Web 服务的类别是用 OWL-S 中的 Service Category 进行描述的，在服务匹配的过程中，首先利用上面的匹配度计算方法来计算两个服务类别匹配度，当提供服务与请求服务的类别相似度小于一个最小相似度阈值 S_{min}（S_{min} 可根据用户对服务查准率的要求来确定，准确率要求越高，S_{min} 取值越大）时，则不对该服务进行功能匹配。这样可以大大提高服务查找的效率。

2.2 功能匹配

OWL-S Profile 通过 Input、Output、Precondition、Effect 4 个属性来描述服务的功能。这样服务的功能相似度计算公式可以定义如下：

$$sim_{fun}(p,q)=w_1 sim_{in}(p_{in},q_{in})+w_2 sim_{out}(p_{out},q_{out})+w_3 sim_{pre}(p_{pre},q_{pre})+w_4 sim_e(p_e,q_e) \quad (2)$$

式中，$sim_{in}(p_{in},q_{in})$、$sim_{out}(p_{out},q_{out})$、$sim_{pre}(p_{pre},q_{pre})$、$sim_e(p_e,q_e)$ 分别表示服务功能输入参数、输出参数、前提和结果的相似度，w_i 表示功能属性的权重，且 $0\leqslant W_i\leqslant 1$，$w_1+w_2+w_3+w_4=1$，可由用户决定，如果用户没有对权值的要求，则可采用平均权值，即这 4 个方面是同等重要的。

3 基于索引的服务发现算法

在图 1 所示的服务发现框架的基础上，利用我们提出的服务发现方法，可以按图 3 所示的算法流程来完成一次服务搜索。

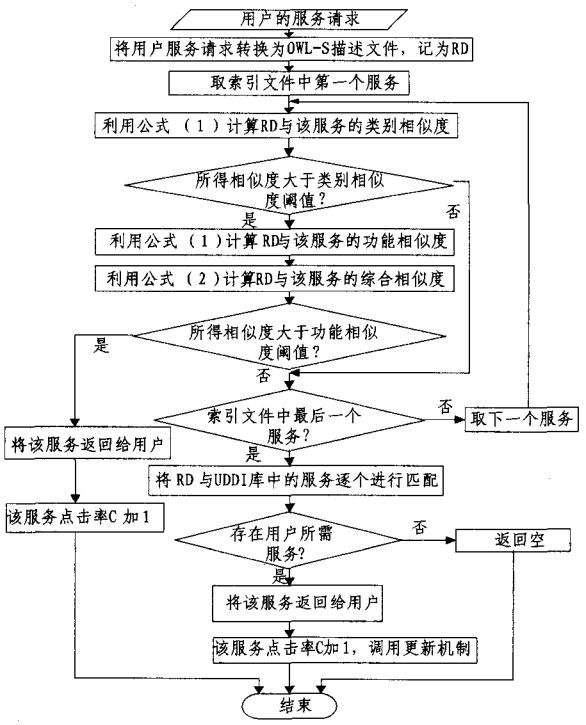

图 3 服务发现流程图

4 性能分析

为了验证基于点击率索引的 Web 服务发现算法（click-method）的有效性，将其与带 cache 的 Web 服务发现方法（cache-method）以及传统的 Web 服务发现方法（traditional-method）进行了比较。所有实验的硬件环境都相同：CPU 为 Pentium 3.00GHz；内存为 2G，操作系统为 Windows XP Professional。实验中使用 Protégé 工具建立本体，包括类、子类和层次关系等信息，存放在 oracle 数据库中。在服务注册库中存放用 OWL-S 描述的服务，用前面定义的 Web 服务描述模型，对服务的功能特性和基本信息进行描述。每个服务的输入输出参数个数为 2～6 个，并随机设定各个服务的点击率。取点击率位于前 20%的若干个服务将其 ID 存放于索引文件中。实验分别取 100、300、500、700、900、1100 个服务构成 UDDI 注册库，来比较几种发现方法的效率。图 4 是随着服务个数的增加几种服务发现方法平均查找时间的比较结果，每次试验分别执行 50 次。

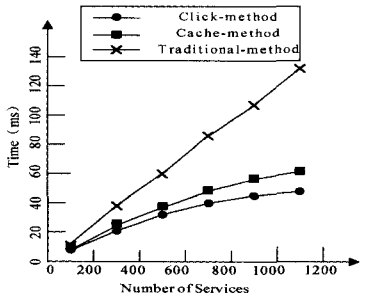

图 4 不同服务发现算法的查找时间对比图

从图 4 可以看出，采用索引策略后，Web 服务发现的效率明显高于传统 Web 服务发现方法。并且随着访问的 Web 服务个数的增加，这种优势更明显。同时，与带 cache 的 Web 服务发现方法相比，该方法的效率也有所提高。这是因为基于点击率的索引使得在服务查找过程中，用户可以很快找到那些使用率较高的服务，而这些服务占用户所需服务的很大一部分。

结束语 为了提高语义 Web 服务发现的效率，本文在对语义 Web 服务特点进行分析的基础上，提出了一种基于点击率索引提高 Web 服务发现效率的方法。该方法在传统 Web 服务架构中引入索引机制，为 UDDI 注册库中点击率较高的 Web 服务建立索引。最后对该方法的 Web 服务发现过程进行了仿真，实验结果验证了本文方法能有效提高 Web 服务发现的效率。未来的工作主要在于怎样在提高查找效率的同时，考虑用户对服务质量的要求，设计更精确的服务匹配规则，进一步提高服务发现的效率和质量。

参 考 文 献

[1] 李春梅，蒋运承. 具有 QoS 约束的语义 Web 服务发现的研究[J]. 计算机科学，2007，34(6)：116-121

[2] 张慧明，唐慧佳. 语义 Web 服务匹配策略研究[J]. 计算机应用，2010，30(4)：1083-1085

[3] Paolucci M，Kawamura T，Sycara K，et al. Semantic Matching of Web Services Capabilities[J]. Lecture Notes in Computer Science，2002，2342：333-347

[4] 吴健，吴朝晖，李莹，等. 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

[5] 杨永齐，符云清，余伟. 基于多阶段匹配的语义 Web 服务发现框架[J]. 计算机科学，2010，37(9)：164-167

[6] 徐德智，陈稀伟，陈建二. 带 Cache 的语义 Web 服务发现研究[J]. 计算机科学，2010，37(8)：129-132

（下转第 317 页）

的查询 $q_2=\{(ProductId,\{C_1,C_2\}),(StoreId,\{T_1,T_2,T_3\})\}$。我们有 $q_1 \bar{R} q_2$。

由于实物化视图的目的是保存数据供查询访问，因此视图的表示与查询是一致的。我们常将查询和视图混用。

系统所收集的查询包括查询的集合 Q 和每个查询 q 的发生频率 $f(q)$。

完成了以上的基础定义，现在正式给出基于查询频率更新的算法。

在给定存储空间 Memory 的限制下，基于查询频率更新问题就是选择那些查询频率更高的查询，用以更新物化视图，来最大限度地提高总体查询的效率。

为了合理利用空间，我们采用贪心算法来求解。在系统所记录的查询 Q 中选出将被实物化的视图，选择标准是单位空间的频率，即 $f(q)/|q|$。其中 $f(q)$ 是已知的，而 $|q|=|D_q|\times\prod_1^m|q.A_i|/\prod_1^m|q.D_i|$，其中 m 为数据集的维数，$|q.D_i|$ 表示级别 D_i 中不同元素的个数，$|q.A_i|$ 表示 A_i 的基数。

算法描述(F-US)：

Step1　输入查询集合 Q；

Step2　判断内存空间是否大于 0，否则退出循环；

Step3　选择最大的 $f(q)/|q|$，将 q 里的数据集合赋给 V 集合；

Step4　如果内存空间 $M\geqslant|V|$，将 q 从 Q 集合中去除，否则 $M=0$。

返回 Step2。

该算法的时间复杂度为 $O(_n\log 2n)$，通过附加的存储器，可以大大提高查询效率，如对 f(q)/|q|建立索引等[4]。

2.2　算法分析的结果评测

为了进一步证明该算法的优点，我们进行了一系列的实验，性能改善达到了预期的效果。实验中，为了更好地与该领域中已有的研究成果进行比较，我们采用 BPUS 算法[5]作为视图选择准则 M，并采用相似的实验方法，即对于多维数据，我们也设计 4 个维 D_1，D_2，D_3 和 D_4，它们分别有 4，3，4，3 个级别，各个级别的成员个数如表 1 所列。

表 1　维层次和维成员个数

维层次	维数			
	D_1	D_2	D_3	D_4
3	5		5	
2	125	5	25	5
1	250	125	125	50
0	500	250	250	250

多维数据的基视图 DB 共有 500k 行，并且假设多维数据的分布是均匀的，估算出数据集尺寸为 15×106 行，存储空间取为数据集尺寸的 20%。我们设计了 90%访问多维数据格中 10%的结点。对于任意被访问的级别，在访问它的所有查询中，有 80%的查询访问其中 20%的成员。实验中共使用了 10000 个查询。这些查询被分为 10 组，其中每一组按所述比例在结点和维成员间重新随机分布。然后取这些查询的平均代价作为查询的代价。

我们比较了在不同的存储空间内对实物化视图查询的代价。存储空间取数据集总尺寸的 0%～20%。比较的结果如图 2 所示。

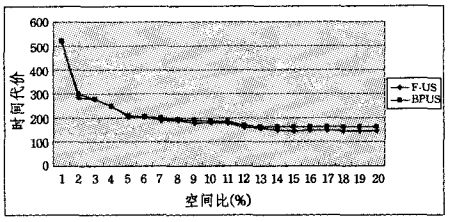

图 2　F-US 与 BPUS 算法代价比较

从图 2 中看出，随着查询数据占存储空间的增加，两种算法的代价都在下降，但是基于查询频率的算法总体优于 BPUS。基于查询频率的物化视图更新算法能够使系统获得更精确的实物化视图，节省了用户的时间，较好地满足了 OLAP 系统的需求。

结束语　通过以上分析论述，可以总结 F-US 算法的几个优点：

(1)根据用户对系统的查询，使得物化视图能得到最大效率的使用。通过最大频率的查询，可以解决用户查询均匀分布的不合理假设。

(2)可以准确地计算并保存表连接或聚集等耗时较多的操作的结果，这样，在执行查询时，就可以避免进行这些耗时的操作，从而快速地得到查询结果。

(3)目前的数据仓库容量已经增长到兆兆字节，精确的实物化视图意味着终端用户现在需要读取的行数很少，所以基于查询频率的实物化视图的更新算法，可以明显地缩短查询响应时间。

本文的研究实验基于特定的项目所涉及的数据，因此难免存在一定的局限性，对于算法的推广应用还有待进一步的研究，希望专家和同行提出宝贵的意见。

参 考 文 献

[1] Inmon W H. Building the Data Warehouse[M]. 王志海，译. 北京：机械工业出版社，2007：20-21

[2] 王丽珍，等. 数据仓库与数据挖掘原理及应用[M]. 北京：科学出版社，2005：109-110

[3] 林宇，等. 数据仓库原理与实践[M]. 北京：人民邮电出版社，2003：204-206

[4] 谭红星，周龙骧. 多维数据实视图的动态选择[J]. 软件学报，2002，13(6)：1090-1096

[5] Harinarayan V，Rajaraman A，Ullman J D. Implementing Data Cubes Efficiently[C]// Jagadish，SIGMOD' 96，Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. Montreal：ACM Press，1996：205-216

(上接第 311 页)

[7] Fox S，Karnawat K，Mydland M，et al. Evaluating implicit measures to improve Web search[J]. ACM Transactions on Information Systems，2005，23(2)：147-168

[8] Srinivasan N，Paolucci M，Sycara K. An efficient algorithm for OWL-S based semantic search in UDDI[J]. lecture Notes in ComputerScience，2005，3387：96-100

[9] Mokhtar S B，Preuveneers D，Georgantas N，et al. EASY：Efficient SemAntic Service Discovery in Pervasive Computing Environments with QoS and Context Support[J]. Journal of Systems and Software，2007，81(5)：785-808

[10] 岳昆，王晓玲，周傲英. Web 服务核心支撑技术研究综述[J]. 软件学报，2004，15(3)：428-442