

基于模糊匹配的语义网格服务发现

刘 立, 陈志刚, 杨 博, 胡春华

(中南大学信息科学与工程学院, 长沙 410083)

摘 要: 匹配是网格服务动态发现和集成的关键问题。针对“网格服务发现”对服务底层数据抽象和分类不足的问题, 该文给出了一种基于语义的模糊匹配网格服务发现框架算法。仿真实验结果表明, 该算法提高了服务发现的查全率和查准率。

关键词: 网格服务; 匹配相似度; 查全率

Semantic Grid Services Discovery Based on Fuzzy Matching

LIU Li, CHEN Zhi-gang, YANG Bo, HU Chun-hua

(College of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】 Matching is considered as one of the crucial factors to ensure dynamic discovery and composition of grid services. Current matching methods are inadequate given their inability to abstract and classify the underlying data of grid services. This paper proposes a semantic grid services discovery arithmetic which exploits FMD in order to abstract and classify the underlying data of grid services. Experimental results show FMD has more precision ratio and recall ratio.

【Key words】 grid services; degree of matching; precision ratio

1 概述

网格服务是一种特殊的Web服务, 是网格计算和Web服务相结合的计算环境^[1]。网格服务以Web服务为基础, 服务发现是“Web/网格服务”首要解决的问题。

UDDI是“Web服务发现”的标准。在发布Web服务时, 虽然UDDI允许发布者提供服务的类别信息, 通过唯一的标识符UUID来发现和定位服务, 但是UDDI存在2个限制^[2]:

(1)使用NAISC, UNSPSC等的分类机制, 分类描述的是整体服务而不包括子服务, 服务底层的数据抽象和分类能力不足, 甚至有些内容没有定义。服务消费者不能通过UDDI来搜索到满足请求的服务细节, 而且搜索结果也是粗糙的。

(2)UDDI使用XML来描述数据模式, 因为缺少对服务内容的语义描述, 所以这种数据不能被机器所理解, 难以将子服务区分开来^[3], 因此, 使它成为UDDI发现机制的一个障碍。

此外, UDDI需要服务消费者在搜索的服务中, 手动地选择服务, 但这很难做到服务选择和服务合成的自动化, 而网格服务发现的任务就是提高效率 and 自动化程度^[3]。

网格服务发现的关键是匹配。目前网格服务发现主要是基于关键字来匹配的, 这会导致服务查询的低效率和低精确度, 原因在于没有考虑2个关键字之间的语义关系。现有的UDDI机制缺乏对网格服务能力的精确描述^[2], 因此, 实现现有UDDI发现机制实现对语义描述的支持是当前网格服务发现的研究方向。

OWL-S是DAML-S的新版本, 是一种基于语义的服务, 把一个网格服务看作由2个相互区别的部分组成:(1)服务实体;(2)子服务集。这是目前定义语义网格/Web服务的标准。它提供一套核心的构造来描述Web服务的能力和属性, 这种描述是明确的, 是计算机能够理解的形式, 网格服务的OWL-S表示能够使用网格服务, 包括自动发现、执行、合成和互操作。OWL-S包含3个组件:(1)ServiceProfile表示Web

服务能力, 用Web服务的input, output, precondition, effects属性(即IOPE指标)来表示;(2)ServiceModel表示Web服务的实现;(3)ServiceGrounding提供WSDL~OWL-S的一致性映射。

在ServiceProfile, ServiceModel, ServiceGrounding, Services 4个部分中, ServiceProfile用来描述该服务内容, 主要由IOPE指标来表示该服务的能力, ServiceModel描述该服务的工作方式, 并使用服务的高层信息来描述服务, 而用低层的子类、处理来定义服务之间的交互细节; ServiceGrounding描述服务的实现机制以及OWL-S到WSDL之间的映射。

虽然利用OWL-S基于服务能力的搜索机制能够克服UDDI的限制, 并产生较好的结果集, 但是OWL-S不能直接用于查询服务的注册和发现, 还是要借助UDDI来实现, 因此, 要实现语义服务必须将OWL-S表示的服务映射到UDDI上, 利用OASIS组织给出的WSDL到UDDI的映射机制, 可以实现OWL-S表示的服务能力到UDDI的映射。利用语义的分类机制和基于能力的搜索可以得到更接近用户请求的结果。

2 相关工作

Kuo-Ming Chao^[4]利用OWL-S对服务的数据进行分类和抽象, 将服务抽象层的语义信息嵌入到中UDDI中, 并通过UDDI的机制发布到注册服务器, 服务的剩余部分使服务请求者能够在抽象层使用模糊条件来查询, 匹配思想分为2个步骤:

(1)将顶层表示服务能力的内容嵌入到UDDI, 作为发布信息, 用于服务发现, 剩下的底层部分存储在OWLJESSKB

基金项目: 国家自然科学基金资助项目(60573127)

作者简介: 刘立(1970-), 男, 博士研究生、讲师, 主研方向: 网格计算中间件; 陈志刚, 教授、博士生导师; 杨博、胡春华, 博士研究生

收稿日期: 2006-11-07 **E-mail:** llyan_liu@163.com

文件用于匹配；

(2)当服务消费者查询服务时候，首先是能力匹配，如果匹配，模糊推理将模糊请求映射到适当的服务上，如果不匹配，就引入 OWLJESSKB 文件中的支持信息，运用近似推理来判断模糊条件之间的关系。

文献[4]存在的主要问题是：没有说明服务是如何分类和抽象的，现有服务和语义服务之间映射的机制不够明确，如何对现有的服务进行语义转换，转化是否需要对现有服务结构进行改造等；通过模糊匹配发现适当服务之后，怎么实现检索结果的自动选取，是服务发现之后和服务调用之前一个重要的步骤，文献[4]没有解决自动选择的问题，缺少查询效率的分析。

文献[5]对现有的服务描述机制进行语义处理，说明了如何将语义表示的服务映射到 UDDI 上的过程。在匹配方面，文献[6]给出了请求输出 OutR 和注册服务公告 OutA 之间的 4 种匹配度：Exact,fail,plug in,subsum。

3 网络服务的模糊发现

针对关键字匹配的搜索效率低下的缺点，笔者采用一种基于语义的服务模糊匹配发现机制，这种机制通过对语义表示的服务能力进行模糊聚类，减少了搜索空间、提高了发现效率，而且支持服务的动态和弹性发现，能使误报率(false positives)和错报率(false negatives)尽可能小，由于采用语义匹配，因此请求者和提供者的匹配能够更加接近，同时还可以对匹配结果集自动选择最合适的服务，减少了手动选择的麻烦。

3.1 模糊集理论

定义 1 定义一个模糊集 $\tilde{A}$ ，对任何 x ，其隶属度函数 $\mu_{\tilde{A}}(x) \in [0,1]$ 是 x 到成员空间的映射， X 是 x 表示的对象集合，可以用式(1)来表示模糊集 $\tilde{A}$ 。

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\} \quad (1)$$

定义 2 1 个模糊集函数的补集 $\mu_{\mu_{\tilde{A}}}$ ，定义为

$$\mu_{\mu_{\tilde{A}}} = 1 - \mu_{\tilde{A}}(x), x \in X \quad (2)$$

定义 3 2 个模糊集的交集表示它们之间的公共元素，设交集的隶属度函数为 $\mu_{\tilde{C}}(x)$ ，即 $\tilde{C} = \tilde{A} \cap \tilde{B}$ ，可以表示为

$$\mu_{\tilde{C}}(x) = \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in X \quad (3)$$

定义 4 2 个模糊集的并集为两个模糊集的和，即

$$\tilde{C} = \tilde{A} \cup \tilde{B}$$

设隶属度为 $\mu_{\tilde{C}}(x)$ ，定义为

$$\mu_{\tilde{C}}(x) = \max\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in X \quad (4)$$

定义 5 2 个服务之间的整体相似度用式(5)来表示，即

$$Sim(\tilde{A}, \tilde{B}) = \frac{\sum_{i=1}^n (\mu_{\tilde{A}}(A_i) \wedge \mu_{\tilde{B}}(B_i))}{\sum_{i=1}^n (\mu_{\tilde{A}}(A_i) \vee \mu_{\tilde{B}}(B_i))} \quad (5)$$

定义 6 (最大隶属度原则 1) 设 $S_1, S_2, \dots, S_n \in F(S)$ 为 n 个服务， $S_0 \in U$ 是待识别的服务，如果 S_i 满足条件

$$\mu_{\tilde{S}}(S_0) = \max\{\mu_{\tilde{S}}(S_0), \mu_{\tilde{S}}(S_0), \dots, \mu_{\tilde{S}}(S_0)\} \quad (6)$$

则认为 S_0 相对隶属于 S_i 。

定义 7 (最大隶属度原则 2) 设 $A \in F(U)$ ， $\mu_1, \mu_2, \dots, \mu_n \in U$ 是 n 个满足条件的服务的隶属度，如果 μ_i 满足条件

$$\mu_{\tilde{A}}(\mu_i) = \max\{\mu_{\tilde{A}}(\mu_1), \mu_{\tilde{A}}(\mu_2), \dots, \mu_{\tilde{A}}(\mu_n)\} \quad (7)$$

则应该优先选用 S_i 。

3.2 模糊发现框架

服务模糊发现框架(见图 1)由 OWL-S 编辑器、模糊分类器、模糊匹配引擎、OWL-S 语义识别数据库、OWL-S2UDDI 转换器、OWL-S/UDDI 注册服务器等组成。

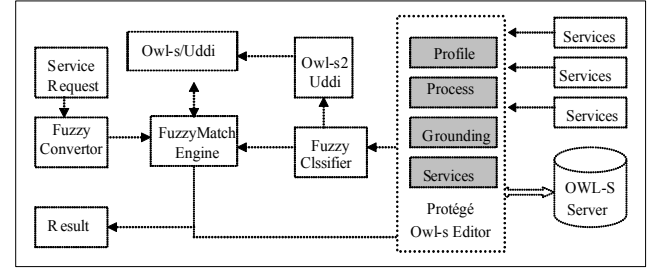


图 1 模糊发现框架

OWL-S editor 将服务提供者要发布的网格/Web 服务 WSDL 文件转换为 OWL 文件，通过 fuzzy classifier，将服务按照服务能力进行分类，形成一种服务分类层次结构，高层用 Profile 部分描述服务实体的整体属性，低层通过 process 部分的 IOPE 指标来描述的子服务；通过 OWL-S/UDDI，将包含语义信息映射到 UDDI 上；FuzzyMatch Engine 是针对服务请求的条件、注册服务器中注册的服务高层信息、低层的子服务进行模糊匹配信息的；通过 Fuzzy Convertor，将服务请求的查询条件转化为支持模糊匹配的条件；通过 OWL-S Server，将以语义表示的服务低层信息保存在基于语义数据库中，供语义匹配使用。

3.3 模糊发现过程

支持语义的模糊服务发现机制的工作为：服务到语义服务的转化，按照服务能力对服务进行抽象聚类，前端查询条件的处理，查询条件和服务的匹配等，具体步骤如下：

(1)将要注册服务转化为能表示服务能力的 OWL-S 语义形式文件。可以采用 OWL-S Editor，将网格服务的 WSDL 文件转化成 OWL 文件。

(2)依据模糊聚类方法，按照服务能力对服务进行抽象聚类，形成语义服务的层次结构。基于服务能力和服务聚类方法步骤如下：

1)建立模糊相似矩阵 R 。设 $S = \{S_1, S_2, \dots, S_n\}$ 是 n 个待分类的服务对象，各自的服务能力有 m 个特征属性，用 $R = (r_{ij})_{n \times n}$ 来表示模糊相似矩阵。模糊相似矩阵是一个对角线为 1 的模糊矩阵， r_{ij} 表示服务 S_i 和 S_j 按照 m 个服务能力属性特征相似的程度，对每一个服务来说，可以用一组数据 $x_{i1}, x_{i2}, \dots, x_{im}$ 来表示它的 m 个特征。采用贴近距离来求得服务间的相似系数 r_{ij} ，见式(8)。

$$r_{ij} = d(\tilde{S}_i, \tilde{S}_j) \quad (8)$$

模糊矩阵表示为

$$R = \begin{bmatrix} 1 & r_{12} & \dots & r_{1j} & \dots & r_{1n} \\ r_{21} & 1 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \dots & 1 & \vdots & \vdots & \vdots \\ r_{i1} & \dots & \dots & r_{ij} & \vdots & r_{in} \\ \vdots & \dots & \dots & \dots & 1 & \vdots \\ r_{n1} & r_{n2} & \dots & \dots & \dots & 1 \end{bmatrix} \quad (9)$$

2)服务聚类。由 R 的等价矩阵在给定 $\lambda \in [0,1]$ 的强截矩阵 R_λ ，可得到服务的分类。假如 $X = \{x_1, x_2, x_3, x_4, x_5\}$ 在给定模糊相似矩阵 R 下，其强截矩阵 R_λ 和对应的聚类分别为

$$R = \begin{bmatrix} 1 & 0.8 & 0.6 & 0.1 & 0.2 \\ 0.8 & 1 & 0.8 & 0.2 & 0.85 \\ 0.6 & 0.8 & 1 & 0 & 0.9 \\ 0.1 & 0.2 & 0 & 1 & 0.1 \\ 0.2 & 0.85 & 0.8 & 0.1 & 1 \end{bmatrix}$$

当 $\lambda \in (0.8, 0.85)$ 时， X 可以分为 3 类，即

$$\{x_1\}, \{x_2, x_3, x_5\}, \{x_4\}$$

3) 对于一个新加入的服务来说, 如需确认服务类型, 则要遵守定义 6。

(3) 服务能力注册。因为现有语义不支持 UDDI 的注册, 通过 UDDI 将服务注册到注册服务器上, 所以, 需要实现 OWL-S~UDDI 的映射, 通过使用 OWL-S 来扩展 UDDI 以及扩展 UDDI API, 就可以将语义表示服务高层的 OWL-S Profile 信息嵌入到 UDDI 的 Advertisement 中, 实现基于语义的服务注册。文献[4,5]实现了这种映射机制。将低层的服务保存在支持语义的语义数据库中, 见图 1。

(4) 模糊查找。要支持对服务查询条件的模糊匹配, 需要对用户输入的查询条件进行模糊(更窄或者宽)匹配处理, 可以借助前面介绍的模糊理论进行处理。

在查询时, 给定一个查询条件以及要达到的隶属度, 设为 $\mu_A(x)$, 更窄的模糊的查询为 $(\mu_A(x))^2$, 更宽的条件为 $(\mu_A(x))^{1/2}$ 。在查询结果集过多的情况下, 查询采用更窄条件进行修改。相反则采用更宽的条件来查询。

(5) 模糊匹配。模糊匹配分为 2 个层次: 1) 高层的服务能力匹配; 2) 低层的子服务匹配。在匹配的过程中, 首先匹配高层的服务能力, 当注册服务器中没有服务能力匹配时, 就查询语义数据库, 进行低层的子服务匹配。

(6) 服务的自动选择。由于满足查询条件的结果可能是一个结果集, 因此“选择最合适的结果”是实现程序自动化首要解决的问题。根据模糊集理论对查询的结果进行自动选择, 并遵守定义 7, 就可以在搜索结果中选择最合适的服务。可以给定一个阈值来确定查找的服务是否满足要求, 从而自动挑选最合适的服务, 提供给服务请求者。

3.4 匹配算法的描述

语义模糊匹配包括 2 级匹配, 首先是高层的服务能力匹配, 若匹配则完成, 当服务能力匹配失败时, 转向匹配低层的子服务。在算法中, req 是服务请求, adv 是注册表示能力的服务, Capmatch 表示满足最低隶属度 μ 匹配的服务, subsmatch 用于子服务匹配。

算法具体描述如下:

```

Sim= $\mu$ ;
match(req){
  Capmatch=empty list;//服务能力匹配
  For all adv in advertisements do{
    If match(req, adv,  $\mu$ ) then
      Capmatch.add(adv,  $\mu$ );
      Sort(Capmatch,  $\mu$ );
    }
  else{//子服务匹配
    For all subservices in owl base do{
      if match(req, subs,  $\mu$ ) then
        subsmatch.add(subs,  $\mu$ );
        Sort(subsmatch,  $\mu$ );
      }
    }
  }
}

```

由算法可知, 在给定的隶属度下, 服务匹配首先匹配已经注册的高层服务能力, 当高层服务不满足条件时, 就匹配低层的子服务, 这种算法降低了服务发现的漏报率和误报率。

4 仿真实验

查全率和查准率是度量服务发现的 2 个主要性能指标: 查全率指查询返回符合查询条件的服务数量与查询返回服务总数量之间的比率; 查准率指查询返回符合查询条件的服务与测试样本集中符合查询条件的服务的比率。

笔者设计了几种服务, 包括酒店预订(5)、房间价格(14)、周围风景(12)、房间设备(21)、服务质量(18)。选取的服务发现方法为: 基于关键字匹配, 2 种基于语义匹配^[7,8], 基于语义模糊匹配的网格服务发现方法。这些方法的平均查全率分别为 23%, 77%, 81%, 92%; 平均查准率分别为 19%, 73%, 78%, 86%。具体测试结果见图 2、图 3。

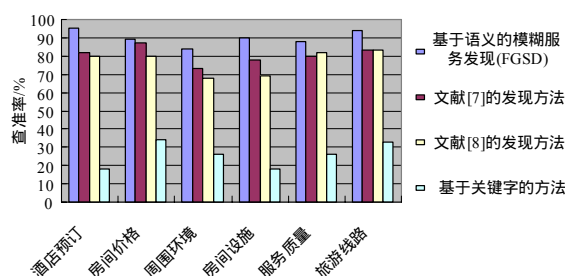


图 2 查准率比较

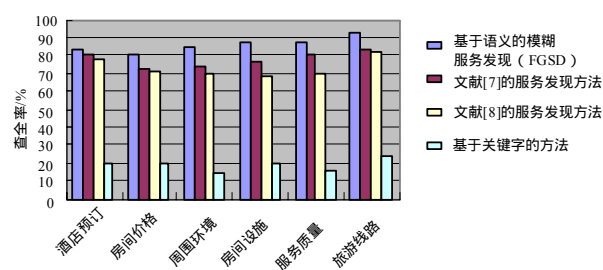


图 3 查全率比较

基于语义的发现机制的查全率和查准率显著优于基于关键字的发现方法, 而基于模糊的语义发现方法又要优于其他的语义发现方法, 而且查全率和查准率一般都在 80% 以上。

5 结束语

对现有的网格服务进行抽象, 形成层次结构。将表示网格服务能力的高层信息注册到服务器上, 将低层服务数据保存在语义数据库中, 在进行“服务发现”时, 进行能力匹配, 由于减少了搜索空间, 因此可以提高服务匹配的 efficiency。当服务消费者在注册服务器中找不到注册的服务时, 通过保存低层服务的语义数据库来发现更小粒度的服务(子服务), 特别是在服务粒度的划分缺少统一标准的情况下, 这种模糊匹配方法具有实际意义。

参考文献

- Ludwig S A. Review of Service Discovery Systems[R]. Uxbridge Middlesex: Department of Electrical and Computer Engineering, Brunel University, Technical Report: TR-DGRG695, 2002.
- Srinivasan N, Paolucci M, Sycara K. Adding OWL-S to UDDI Implementation and Throughput[C]//Proc. of the 1st International Workshop on Semantic Web Services and Web Process Composition, San Diego, California, USA. 2004.
- Majithia S, Walker D W, Gray A. Automated Composition of Semantic Grid Services[C]//Proceedings of the UK E-science All Hands Meeting. 2004.
- Chao K M, Younas M, Lo C C, et al. Fuzzy Matchmaking for Web Services[C]//Proceedings of the 19th International Conference on Advanced Information Networking and Applications. 2005.
- Paolucci M, Kawamura T, Payne T, et al. Semantic Matching of Web Services Capabilities[C]//Proceedings of the 1st International Semantic Web Conference. 2002.

(下转第 183 页)