ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.8, August 2007, pp.1922−1932
DOI: 10.1360/jos181922
© 2007 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种双层P2P结构的语义服务发现模型*

刘志忠[1+], 王怀民[1,2], 周 斌[1]

[1](国防科学技术大学 计算机学院 网络与信息安全研究所,湖南 长沙 410073)

[2](国防科学技术大学 计算机学院 并行与分布处理国家重点实验室,湖南 长沙 410073)

A Two Layered P2P Model for Semantic Service Discovery

LIU Zhi-Zhong[1+], WANG Huai-Min[1,2], ZHOU Bin[1]

[1](Institute of Network and Information Security, School of Computer, National University of Defense Technology, Changsha 410073, China)

[2](National Laboratory for Parallel and Distributed Processing, School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Fax: +86-731-4512504, E-mail: liuzane@gmail.com, http://www.nudt.edu.cn

Liu ZZ, Wang HM, Zhou B. A two layered P2P model for semantic service discovery. *Journal of Software*, 2007,18(8):1922−1932. http://www.jos.org.cn/1000-9825/18/1922.htm

Abstract: In open Internet environment, it is inevitable that multiple ontologies coexist. Centralized service discovery mechanism becomes the bottleneck of SOC (service oriented computing), which results in poor scalability of system. Aiming at solving these problems, a two layered P2P based model for semantic service discovery is proposed in this paper. The model is based on ontology community and integrates iVCE (Internet-based virtual computing environment) core concepts into a P2P model. Based on this model, a service discovery algorithm composed of two stages and three steps is proposed. It matches services across communities as well as within community. Within a community, algorithm locates registers holding service information with a high probability of satisfying a request firstly. Then it captures semantic matching between service advertisements and service requests by logical reasoning. Service discovery across communities occurs according to some policies. The model is suitable for opening environment with coexistent multiple ontologies. Experimental results show that given an appropriate setting, the model can make a tradeoff between recall and responding time. In addition, the model will release the mean load of registers efficiently while holding recall.

Key words: P2P; semantic service discovery; ontology community

摘 要: 在开放的 Internet 环境下,多本体共存是一种必然.同时,集中式的语义服务发现机制是整个面向服务计算的瓶颈,导致系统的可扩展性差.为了支持多本体共存并提高系统的可扩展性,提出一种双层 P2P 语义服务发现模型.该模型以本体社区为核心,将 iVCE(Internet-based virtual computing environment)的核心概念集成到 P2P 模型中.基于该模型,提出一种分两个阶段、3 个步骤的服务发现算法.除了在本体社区内以外,算法还可以实现跨社区的服

* Supported by the National Basic Research Program of China under Grant No.2005CB321800 (国家重点基础研究发展计划(973)); the National Natural Science Foundation of China under Grant Nos.90412011, 60625203 (国家自然科学基金)

Received 2007-03-01; Accepted 2007-04-26

© 中国科学院软件研究所 http://www.jos.org.cn

务发现.在本体社区内,算法首先根据语义相似性选定相应的注册服务器,然后再利用逻辑推理来实施精确的服务匹配.而跨社区的服务发现则按照一定的策略实施.该模型适用于多本体共存的开放环境.实验结果表明,通过合理的参数配置,模型能在查全率和服务发现响应时间之间加以折衷,并取得较好的结果;另外,模型能够在不降低服务查全率的情况下降低注册服务节点的平均负载.

关键词: P2P;语义服务发现;本体社区

中图法分类号: TP393 **文献标识码:** A

作为计算机技术与通信技术相融合的产物,互联网正由一般意义下的计算机通信平台,逐步演变成为广泛存在的虚拟计算环境.网络资源的聚合与协同是虚拟计算环境的核心.我们所研究的虚拟计算环境iVCE (Internet-based virtual computing environment)就这些问题的解决途径进行了有效的探索[1].iVCE通过自主元素、虚拟共同体、虚拟执行体等核心概念和相关机制为基于Internet的资源聚合与协同提供框架支持.但是,iVCE中的自主元素主要强调的是资源的自主性,而对于资源的发布与发现等相关技术没有进行深入讨论.面向服务计算(service oriented computing,简称SOC)以资源服务化为基础,通过服务的发布、发现、调用等相关机制,为开放环境下的资源聚合提供强有力的支持.因此,将面向服务计算和iVCE相结合可以方便资源的发布、发现,从而更好地实现资源的按需聚合.

但是,目前广泛采用的基于关键字的服务匹配机制面临服务匹配精度低以及服务发现、调用、组合自动化程度低等问题.就这些问题,研究人员在Web服务的基础上结合语义Web技术提出了基于语义的服务发现.语义服务发现在服务描述中引入语义信息,并在服务匹配过程中利用这些语义信息来提高服务匹配的准确性[2−4].因此,本体是语义服务发现的基础.但是,iVCE最终将面向开放的网络环境,每个虚拟共同体的发布都可能拥有自己的语义模型(即本体).也就是说,iVCE中的语义服务发现模型必须能够支持多个本体的共存以及跨本体的语义服务发现.

另一方面,随着应用规模的不断扩大,传统SOA中的集中式注册中心(如UDDI)最终将成为系统的瓶颈.为此,UDDI3.0 采用了逻辑集中物理分布的模式,允许多个注册中心共存,但是中心的数据需要保持完全一致.这种一致性维护比较复杂.因此,有人提出了物理和逻辑都分布的模式.该模式将注册中心分为几个独立的中心,各个中心独立地维护自身数据[5−8].在这种情况下,各注册中心数据的划分和管理是一个关键.本体作为语义服务发现的基础,也可以是数据划分的一个参照标准.一般情况下,为了方便本体的开发以及维护,一个完整的本体往往会包含几个逻辑上相对独立的本体,这几个本体组成一个逻辑上一致的统一本体[9].结合模块化本体的思想,将注册中心的数据按照模块化本体进行划分,从而实现一个物理和逻辑上都完全分布的服务发现模型,将有助于提高系统的可扩展性、降低注册中心的负载.

本文结合 iVCE 的核心概念和机理,提出了一种以本体社区为基础的双层 P2P 语义服务发现模型.首先,在社区内,不同的注册中心形成一层 P2P 结构;然后,社区间再形成一个 P2P 结构.本体社区以本体为基础,因此,该模型可以支持多本体的共存,并基于社区之间的联系实现跨本体的服务发现.社区内的注册服务器根据模块化本体的思想,由多个注册服务器形成一个逻辑上相对分布、物理上完全分布的注册中心.该模型不仅具有传统分布式服务发现模型的特点,并且通过引入本体社区层的 P2P 结构,可以消除服务发现语义层次上的异构.同时,按照本体模块对数据实施划分,还将保证注册中心之间的关联简单,而内部的服务关联紧密.据此,iVCE 中的服务发现过程分为社区内的服务发现和跨社区的服务发现两个阶段.社区内的服务发现也分为两步:首先通过基于概念实施简单的语义相似性匹配,从而确定所有可能含有满足要求的服务描述的注册服务器;在定位相应的服务器后,服务器再通过逻辑推理匹配所有可能的服务描述,并将结果返回.实验表明,通过提高社区间的请求转发概率可以提高服务发现的查全率,但这一点是以增加服务发现的平均响应时间为代价的;另外,在单个本体社区内,通过设置合适的简单语义相似度的阈值,可以在不降低服务查全率的情况下降低注册服务器的平均负载.

本文第 1 节从 P2P 环境下的面向服务计算角度出发,介绍 iVCE 的核心概念与机理.第 2 节在本体社区的基

© 中国科学院软件研究所 http://www.jos.org.cn

础上,详细描述双层 P2P 的网络拓扑结构.在第 2 节的基础上,第 3 节详细给出服务发现算法中的注册中心选择和服务转发策略.第 4 节对相关工作进行总结与比较.最后总结全文,并对未来的研究工作进行展望.

1 iVCE 与面向服务体系结构

1.1 虚拟计算环境iVCE

为了实现互联网上资源的按需聚合和自主协同,iVCE主要在 3 个层次上进行研究[1].(1) 构建资源的主体化模型.iVCE通过自主元素对互联网上的资源进行抽象和封装,将被动的、静态的网络资源转变为主动的、动态的自主元素(autonomic element,简称AE),从而有效地屏蔽和适应资源的多样性特征.(2) 构建利益共同体,支持按需聚合.iVCE中通过虚拟共同体这一概念来实现.所谓虚拟共同体(virtual commonwealth,简称VC)是指一组具有相同兴趣并遵从共同原则的自主元素的集合.为了支持任务的完成,虚拟共同体需要提供一定的资源元信息管理设施以及相应的资源发布与发现机制.本文的主要关注点就在于如何构建这样的基础设施.(3) 构建分布执行机制,支持资源间的自主协同.在iVCE运行态,需要根据任务选择和请求自主元素的服务,实现自主元素的动态绑定.iVCE中通过虚拟执行体(virtual executor,简称VE)的概念来完成这一过程.所谓虚拟执行体是指协同承担同一任务的相关自主元素,为完成该任务而形成的状态空间的总和.自主元素、虚拟共同体以及虚拟执行体分别在不同的层次上为资源聚合提供支持.这些概念为实现iVCE提供了必要的支持,但是为了更好地实现资源聚合,需要更高层次的资源封装与协同机制.

1.2 iVCE与面向服务体系结构

iVCE 将资源封装为自主元素,从而把被动资源转变成为主动资源.但是,这个封装主要关注的是资源的内部行为与决策过程,而没有关注资源的外部呈现.也就是说,自主元素并未涉及自身如何被其他自主元素发现和使用.为了方便资源的发现与使用,自主元素还需要对外提供一种与系统无关的资源呈现模式.在此,我们借助服务的概念对自主元素实施服务化封装.服务化封装基于 XML 按照一定的规范(WSDL,OWL-S 等)将资源的功能与相关属性对外呈现,以方便其他自主元素或用户访问.图 1 所示为自主元素的服务化封装示意图.服务关注的是自主元素对外所提供的功能,而不再关注自主元素的内部实现与决策策略.从不同的角度出发,自主元素都可以封装为不同的服务.比如在流媒体下载的应用中,一个自主元素既提供资源下载,也从其他资源元素下载资源.在服务封装时,我们可以根据需要将两者封装在不同的服务内,也可以将其封装成为一个服务.

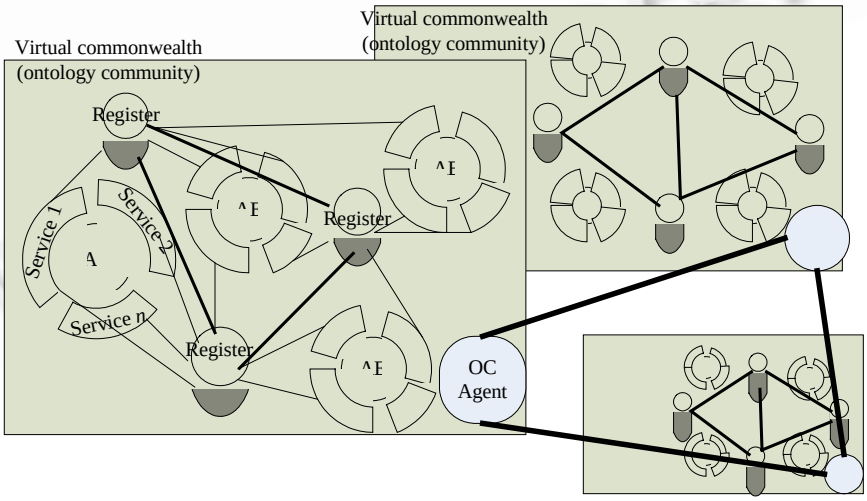

Fig.1 iVCE and SOA

图 1 iVCE 与面向服务计算体系结构

© 中国科学院软件研究所 http://www.jos.org.cn

iVCE以虚拟共同体为基础,并假定同一共同体内部不存在语义异构.在此,我们采用本体作为虚拟共同体的共享语义模型,并将遵循相同本体的自主元素构成的虚拟共同体作为一个本体社区(ontology community,简称OC),简称社区[10].为了实现跨社区的服务请求匹配,每个社区都由一个社区代理(OC Agent)来代表,整个社区都通过该代理和其他社区进行交互.但是,由于iVCE面向开放的Internet环境,多本体共存是一种必然.这样,多个社区的共存会带来新的语义异构问题.为了解决这个层次上的语义异构问题,需要使用语义研究领域的一些相关知识.在此,我们通过建立本体之间的映射关系来解决本体社区之间的语义异构.图 1 中本体社区代理之间的连线表示不同本体之间的映射关系,同时也是本体社区之间的联系.另外,由于不同的本体之间可能会存在重叠,因此,同一自主元素可能会同时隶属于多个不同的社区,如图 1 所示.

发现并绑定相应的服务是服务发现机制的核心任务.目前,主流的服务注册与发现均采用物理或(和)逻辑集中的模式,如 UDDI,ebXML.这些模式在可扩展性、容错性等方面存在明显的不足,加上基于语义的服务匹配更为复杂,所以,在基于语义的服务发现中,集中式服务注册与发现更可能成为整个系统的瓶颈.为了缓解这个瓶颈,在一个社区内,我们允许多个注册中心的存在,如图 1 所示.这些注册中心不仅在物理上分布,也在逻辑上分布.根据模块化本体的思想,各个注册中心仅仅负责与某些概念相关的服务的注册.注册中心之间的联系是一般的非结构 P2P 网络中的随机连接.

2 双层网络拓扑

上一节已经就 iVCE 与 Web 服务的结合进行了相关的讨论.本节在详细介绍本文所提出的用于语义服务发现的双层 P2P 网络拓扑结构之前,先简要介绍本体社区的概念.

本体社区以本体为基础,负责管理基于此本体表示的所有服务[10].社区内的服务遵循统一的本体模型,因此,服务之间不存在异构.一般情况下,一个完整的本体都是由多个较小的本体组成.因此,本体社区之间可以共享某些小本体,并且这些本体将成为联系这些社区的纽带.下一节中介绍的社区间P2P结构将利用这些关系来建立本体社区之间的联系.

2.1 社区代理及社区间的P2P结构

在开放环境下,多个本体共存是一种必然,因此,多个本体社区的共存也是一种必然.这些社区彼此相互独立,但又存在一定的联系.为了支持用户实现跨社区的服务发现与匹配,需要根据本体之间的联系建立社区之间的关系.这些联系在服务发现以前可以由人工给定、也可以借助相应的本体映射算法建立,并且可以在服务发现过程中不断进行修正.但是,该结构是相对稳定的.

每个本体社区由一个本体社区代理来代表.本体社区代理既是社区内注册中心与其他社区交互的中介,也是其他社区访问该社区的入口,其功能包括:(1) 将本社区内的请求按照一定的策略转发到相关的其他本体社区以实现跨社区的语义服务匹配;(2) 接收其他社区的服务发现请求并将请求转发到本本体社区以实施语义服务匹配.本体社区代理的结构框图如图 2 所示.这两方面的功能都需要本体转换的支持.本体转换将利用不同本体社区之间的本体映射关系,将服务请求在不同的本体之间实施转换.

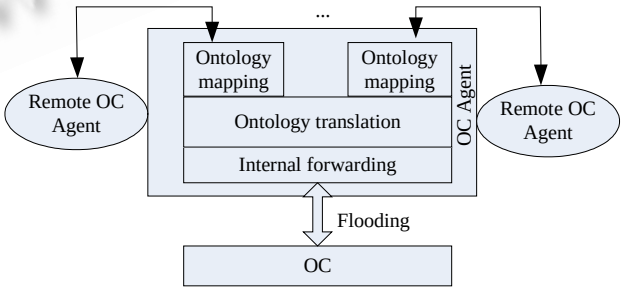

Fig.2 Structure of OC Agent

图 2 本体社区代理结构

© 中国科学院软件研究所 http://www.jos.org.cn

由于本体社区是通过社区所遵循的本体来刻画的,因此,每个Peer也就可以通过这组内部一致的本体来刻画,如$OC_i=\{O_{i1},O_{i2},\ldots,O_{im}\}$.这$m$个本体构成一个逻辑本体$O_i$,如果$OC_i\cap OC_j\neq\varnothing$,则这两个本体社区之间存在逻辑连接,该连接关系可以用本体映射来表示,即$Mapping:O_i\rightarrow O_j$.该本体包含若干概念,本体社区内的注册服务器将根据这些概念之间的关系来对本体进行模块化划分.

2.2 社区内的P2P结构

本体社区的引入可以很好地解决服务发现中的语义异构以及跨本体的服务发现问题.但是,在单个本体社区内,物理或逻辑集中的注册中心将成为系统的瓶颈.为了缓解物理集中的注册中心的瓶颈问题,同时回避逻辑集中式注册中心的复制协议和一致性维护等问题,我们在本体社区内使用物理和逻辑上都分布的 P2P 结构来组织社区内的多个注册中心,每个注册中心都只负责某一部分服务的注册.在服务发布与发现时,算法首先需要根据相应的原则选定合适的注册服务器,这个问题将在下一节给出.在此,我们先讨论注册中心的划分以及表示等问题.

我们知道,本体是由一组概念以及概念之间的关系组成的.模块化本体的思想将本体根据概念之间关系耦合的紧密性分为若干个模块,并基于依赖图建立模块之间的依赖关系.我们以模块化本体为基础,针对本体中的每一个模块设置一个对应的注册中心.该注册中心将负责与该本体模块紧密相关的服务的注册与发现,如图 3 所示(每个黑点代表一个本体概念,每个圆框为一个本体模块,针对每个模块都有一个相应的注册服务器).每个注册服务器由m个特征概念来表征,即注册服务器$R_i=\{C_{i1},C_{i2},\ldots,C_{im}\}$.在服务注册与发现时,请求也通过模板的输入/输出概念来刻画,即请求$Q=\{q_1,q_2,\ldots,q_n\}=I_Q\cup O_Q$.服务发布与发现时的注册中心的选择将依据这两个概念之间的语义相似性来确定.本体社区内的语义相似性计算和服务发现算法将在后续章节中详细给出.

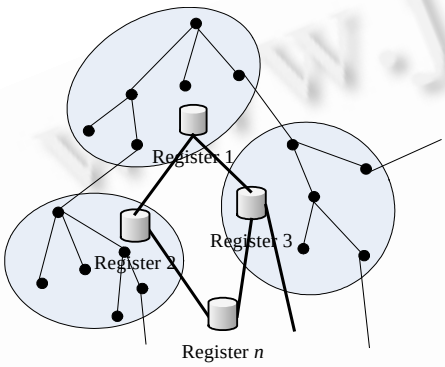

Fig.3 P2P network of register within OC

图 3 社区内注册服务器 P2P 网络

结合两个层次上的 P2P 网络结构图,可以得到如图 4 所示的结构图(灰色小圆表示服务注册中心,黑色大圆表示本体社区代理).在本体社区内,多个注册中心以 P2P 结构组织在一起,为整个社区提供完整的注册与发现服务.同时,社区通过社区代理组织成一个较高层的 P2P 网络以支持跨本体社区的服务匹配与发现.

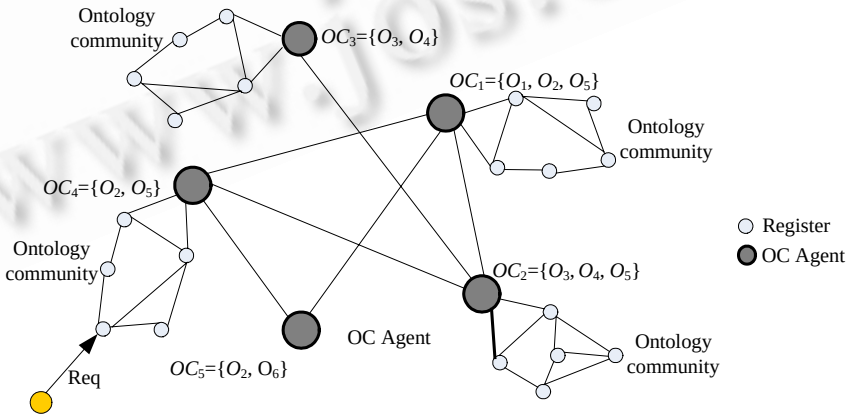

Fig.4 Two layered P2P network

图 4 双层 P2P 网络

© 中国科学院软件研究所 http://www.jos.org.cn

3 基于双层拓扑的服务发现算法

双层 P2P 模型下的服务发现算法包括两个阶段:社区内的服务发现和跨本体社区的服务发现.社区内的服务匹配首先通过语义相似性计算来定位相应的注册中心,然后再利用复杂的逻辑推理来实现精确的语义匹配.我们在文献[10]中就本体社区内基于逻辑推理的服务匹配与社区间的请求转换进行了深入研究,本文主要讨论服务发现中基于语义相似性的注册中心选择以及服务请求转发策略两个问题.

3.1 相关定义

语义相似性(semantic similarity). 给定两组概念$C_1=\{c_{11},c_{12},\ldots,c_{1n}\}$和$C_2=\{c_{21},c_{22},\ldots,c_{2m}\}$,两组概念之间的语义相似性$SemSim(C_1,C_2)$定义如下[11]:

$$SemSim(C_1,C_2)=\mathrm{Max}\left(\sum_{i=1}^{n}\sum_{j=1}^{m}Sim(c_{1i},c_{2j})\right)/\mathrm{Min}(n,m),$$

其中,$Sim(c_{1i},c_{2j})$表示本体内两个概念类之间的语义相似性,包括语法层次上的相似性Sim_T、内部结构相似性Sim_{IS}、外部结构相似Sim_{ES}以及外延相似性Sim_{EX}[12].$Sim(c_{1i},c_{2j})$按照加权求和计算($\omega_1+\omega_2+\omega_3+\omega_4=1$)得到,即

$$Sim(c_{1i},c_{2j})=\omega_1 Sim_T(c_{1i},c_{2j})+\omega_2 Sim_{IS}(c_{1i},c_{2j})+\omega_3 Sim_{ES}(c_{1i},c_{2j})+\omega_4 Sim_{EX}(c_{1i},c_{2j}),$$

其中,Sim_T为基于最大相同子串得到的相似性;Sim_{IS}通过比较概念的属性来计算概念相似性;Sim_{ES}通过比较概念的父概念和子概念来计算概念之间的语义相似性;Sim_{EX}主要比较概念的实例之间的相似性,从而得到概念之间的相似性.在我们的研究中,概念不存在实例,故设定$\omega_4=0,\omega_1=\omega_2=\omega_3=1/3$.

3.2 服务发现算法

社区内注册中心的选择采用 P2P 中经典的泛洪策略来实施服务搜索.在此,我们采用泛洪策略主要出于两个方面的考虑.一方面,由于本体自身的目的就是为了实现领域的共享,为领域提供一个统一的领域模型,本体所包含的本体模块一般都比较少并且它们之间关系简单,这样,社区内的注册服务器之间的 P2P 网络也比较简单.因此,经典泛洪算法所带来的网络开销在社区内的注册服务器之间的 P2P 网络中并不显得突出.另一方面,在本体社区内,本体模块之间的关联在一定程度上已经表示了不同注册中心之间的关联性.因此,在这个意义上说,通过经典泛洪算法来定位相应的注册中心实际上是一种非常有效的方法.

节点通过计算注册中心表征概念和服务请求所使用概念之间的语义相似性来确定注册节点是否存在可能需要的服务.为了方便介绍,先给出如下记号:

- *ST*:服务请求;
- *SearchRegister*(*sq*,*ST*,*rp*,*TTL*,*Flag*):服务请求者向注册服务器 *sq* 发送查询服务 *ST* 的请求,*rp* 为已经请求的节点列表,*TTL* 为消息的存活周期,并且,目前为止找到匹配服务的标志为 *Flag*(找到为 True,没找到为 False);
- [*SearchRegister*(*sq*,*ST*,{Reg_1,Reg_2,…,Reg_m},*TTL*−1),True]*g*:按照泛洪策略转发请求,当没有邻居节点时转发给社区代理.
- *SemanticServiceMatching*(*ST*):在服务器上通过逻辑推理匹配请求 *ST*.

节点按算法 1 处理请求.针对服务中心R_i,如果$Semsim(R_i,C_{ST})>\xi$(ξ为设定的阈值,C_{ST}为模板*ST*中的概念集),则认为该服务器满足条件,在将消息进行转发处理后,该服务器将执行*SemanticServiceMatching*(*ST*);否则,该服务器对请求进行一定的处理后将实施[*SearchRegister*(*sq*,*ST*,{Reg_1,Reg_2,…,Reg_m},*TTL*−1),True]*g*.当消息跳数大于*TTL*或节点无其他邻居节点转发时,该消息将转发给本体社区代理.代理将根据策略判断是否转发到其他社区以实施跨社区的服务发现.在此算法中,参数ξ的设置与注册服务节点的平均负载和服务的查全率密切相关.它们之间的关系将在下一节通过实验给出.

© 中国科学院软件研究所 http://www.jos.org.cn

算法 1. *SearchRegister*(*sq*,*ST*,{Reg_1,Reg_2,…,Reg_m},*TTL*,*Flag*). //在本体社区内寻找合适的注册服务器来实施基于逻辑的语义服务匹配

1. *TTL*=*TTL*−1
2. if *sq* in {Reg_1,Reg_2,…,Reg_m} //已经收到过消息
3. if *TTL*=0 then //至今为止未找到合适的注册服务器,且消息不能继续传播
4. [*SearchRegister*(*OCAgent*,*ST*,{Reg_1,Reg_2,…,Reg_m},*TTL*,*flag*)]$_g$ //消息转发给社区代理
5. else
6. *Sim*=*SemSim*(*sq*,*ST*) //注册服务器和服务请求之间实施简单的相似性匹配
7. If *Sim*>ξ then //该注册中心满足要求
8. [*SearchRegister*(*sq*′,*ST*,{Reg_1,Reg_2,…,Reg_m,*sq*},*TTL*−1,True)]$_g$
9. Else
10. [*SearchRegister*(*sq*′,*ST*,{Reg_1,Reg_2,…,Reg_m,*sq*},*TTL*−1,False)]$_g$

当在社区内实施注册服务器选择时,注册服务器将那些已经达到最大存活时间的请求或没有邻居可转发的请求转发给社区代理.由于社区内的服务请求采用泛洪策略,所有同一个用户请求在社区内可能会存在多个副本.本体社区代理将根据自身所收到的没有找到匹配注册中心的服务请求数和自身的转发策略来决定是否对服务请求进行转发.一般情况下,主要策略包括:

- 所有请求都转发;
- 本体社区代理收到了未找到合适注册服务器的请求时实施转发;
- 本体社区代理收到超过 *n* 个未找到合适注册服务器的请求时实施转发;
- 本体社区代理收到超过 *p*%未找到合适注册服务器的请求时实施转发.

通过调节最后一种策略的 *p* 值可以达到其他几种策略的效果,因此,我们采用最后一种策略.服务发现的平均查全率和平均响应时间都与 *p* 密切相关.通过实验可以设定最佳的 *p* 值.与社区内注册服务器间的请求转发不同,在该层 P2P 网络上的消息转发只进行一步转发,即本体代理只接收其他社区代理的请求,并不将其转发,转发请求仅发给自身的所有邻居.代理除了要根据收到的内部转发请求将服务请求转发到其他社区外,还需要将其他社区的服务请求在本社区内按泛洪策略转发.根据使用的策略,本体社区代理对内部请求的转发算法如算法 2 所示.

算法 2. *ForwordReq*(*ST*, *p*). //本体社区代理根据策略将本社区的请求转发到其他社区

1. Foreach *SearchRegister* do //计算未找到匹配注册服务器的请求的比率
2. If *SearchRegister*.*flag*=false then
3. *NotFound*++
   End if
4. *Total*++
   End for
5. If *NotFound*/*Total*>*p* then //根据策略实施转发
6. Foreach *neighbourofOC* do
7. *ST*′=*Transform*(*ST*,*OC*,*neighbourofOC*)
8. [*SearchRegister*(*sq*,*ST*′,∅,*TTL*,False)]$_g$
   End for
   End if

3.3 实验分析

从上面的分析可知,ξ与 *p* 的取值对整个算法的性能影响较大.通过对ξ的控制,可以控制各注册节点的平均负载和服务请求的平均查全率.*p* 值的设定将影响服务请求的平均响应时间和查全率.为了确定这两个值对性

© 中国科学院软件研究所 http://www.jos.org.cn

能的影响,我们分别做了两组实验.实验使用 OWL-S/UDDI Matchmaker 作为服务匹配引擎.

实验的样本采用 OWL-TC2.1 中的本体和服务样本.第 1 组实验使用了 Economy,Education,Food 以及 Medical 这 4 个领域的 120~300 个服务以及 20 个服务请求.整个虚拟计算环境包括 4 个本体社区.各个本体社区内根据本体的模块化方法,分别包含 7,9,8,8 个注册服务器.所有服务和请求都均匀地分布在各个虚拟共同体内.第 2 组实验仅以 Economy 领域的服务和请求作为研究对象.所有实验的结果均为所有请求的统计结果.

第 1 组实验主要考虑不同规模下,参数 p 对系统的平均响应时间和系统查全率的影响.实验结果如图 5 所示.从图中可知:

- 服务的数量 n 对 p 与查全率、响应时间的关系影响不大,也就是说,应用的规模对服务的影响不大.
- 在所有的实验中,系统的查全率都将随着 p 的增加而降低,最终将达到只考虑一个社区的情况.随着 p 的减小,服务的查全率将最终趋向于把所有服务放在一个社区时所能到达的查全率.
- 在响应时间上,由于 P2P 结构把所有的服务分布在多个注册服务器上,从而使得服务发现的响应时间随服务的规模的变化缓慢,但是有增大的趋势.整体上说,服务发现的响应时间随着 p 的增大而减小,并趋近于限定在一个社区内的服务发现响应时间;随着 p 的减小,服务发现的响应时间将增大,并最终趋近于所有的请求都实施跨社区的服务匹配所需要的时间,大约为最小值的 2 倍.当 p=0.6 左右时,模型在服务的查全率和平均响应时间之间有一个较好的折衷(*Recall*=0.8,响应时间只有最大时的一半左右).

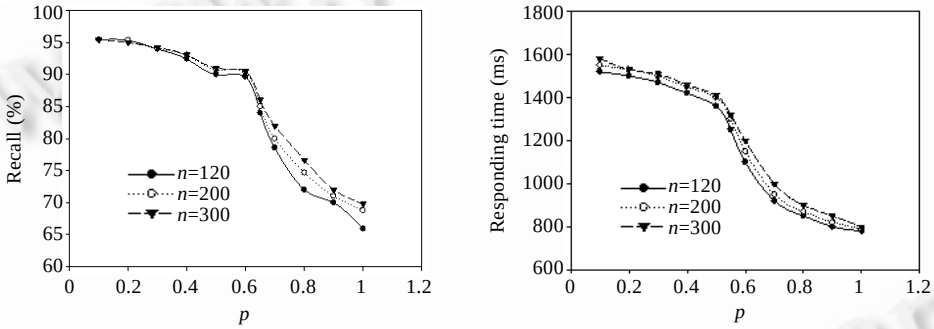

Fig.5 Relation between p and recall and responding time

图 5 p 与查全率和响应时间之间的关系

第 2 组实验主要考虑简单语义相似性的阈值ξ的设置与社区内服务的查全率和各服务注册节点的负载之间的关系.服务注册节点的负载包括两部分:简单语义相似性计算和基于逻辑推理的服务匹配,其中,逻辑推理部分是节点的主要负载.因此,在评价过程中我们主要考虑这一部分负载.为了简单起见,我们通过节点所处理的逻辑推理的次数来评价节点的负载.节点的查全率也是指本体社区内的查全率,即不考虑跨本体领域的服务发现.实验结果如图 6 所示,从图中可以看出:

- 服务的查全率将随着ξ值的增大而减小,但是当ξ较小时,这种变化并不明显.只有当ξ超过某个阈值(ξ>0.4)时,服务的查全率才开始明显下降.这主要是由于如果ξ设置较低,则很多注册服务器都将成为合适的注册服务器,因此,基本上所有发布的服务都要与请求执行基于逻辑推理的语义相匹配,从而达到较高的查全率.查全率随ξ值的变化与服务的数量关系不大.
- 另一方面,随着ξ的增大,注册服务节点的平均负载将显著降低.因为随着ξ的增大,只有较少的符合程度比较高的注册服务器才响应服务请求,执行基于逻辑推理的语义服务匹配.因此,服务器的平均负载下降.当设定ξ=0.4 时,服务仍然可以保持较高的查全率(0.8),而注册服务器的平均负载只有最大时的 10/20=50%左右).

© 中国科学院软件研究所 http://www.jos.org.cn

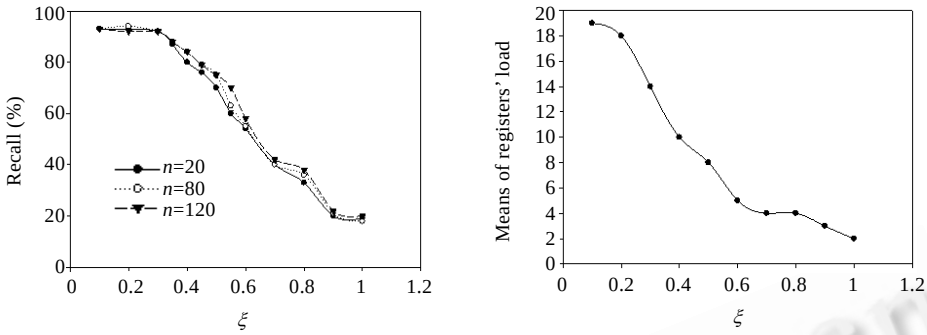

Fig.6 Relations between ξ and recall and mean load of registers

图 6 ξ与查全率、注册节点平均负载的关系

4 相关工作

在所有分布式语义服务发现机制的研究中,Meteor-S项目的MWSDI中所使用的方法与我们所提出的方法最为相似.MWSDI使用扩展注册本体(XTRO)来保存各个本体注册中心的元信息,通过将本题注册中心和本体定义中的领域相关联来实现分布式的服务发现.针对某一个领域,MWSDI中存在多个注册中心,从而形成注册中心联邦地来实现服务的发现[8].在WSMDI中,所谓的领域是相对狭隘的一个概念,实际上它还限定在同一的本体模型之内,只是将本体模型中的 1 个或几个概念作为一个领域.因此,它所研究的语义服务发现还是限定在单个同一的语义模型下,因此不支持多本体环境下的语义发现.MWSDI中的领域实际上是针对我们所提出的模型中的一个本体社区内的Peer,并且在该Peer环境下,通过多个注册中心组成联邦来实现语义的服务发现.

法国的Tarek ESSAFI等人在DEAL项目中提出了一种基于P2P的语义服务发现方法[6].在DEAL中,服务发现也分为两个阶段:确定相应的服务器和匹配相应的服务.他们将具有相似服务描述的服务放在相同的服务器上.在服务发现时,对服务描述基于本体实施编码,基于此编码来确定服务信息所在的服务器.该方法存在两方面的不足:首先,通过编码来确定服务所在的服务器的条件过于苛刻,这将降低服务的查全率;其次,该方法局限于只有 1 个本体存在的情况,不能适用于多本体共存的开放环境.

瑞典Linköping大学的Daniel Elenius等人基于JXTA和OWL-S提出了一种在P2P网络中基于本体的服务发现方法Oden[13].在该方法中,服务查找分为两个阶段:首先是查找满足某种类型条件的Peer,然后在这些节点上基于OWL对服务进行推理匹配.在这一点上,该方法与我们在本体社区内的语义服务发现过程很类似.但是,我们并不是针对每个概念设立相应的Peer,而是基于模块化本体的思想,针对一组密切相关的关键字建立Peer.这样可以减小Peer发现过程中的网络带宽需求,使得本体规模较大时系统仍然能够正常地工作.

为了有效地实施服务发现,优化网格信息服务的搜索性能,北京航空航天大学在CROWN系统中提出了S-Club机制[14].S-Club机制的基本思想是,在网格信息服务(GIS)网络上构建一个层叠网络,该层叠网络将提供相同类型服务的GIS组成一个Club.当用户需要查找某个服务时,请求首先在此层叠网上传播.S-Club通过将相同类型的服务进行聚类,能够有效地提高服务发现的响应时间并减小网络流量.但是,由于没有相应的语义模型,在规范服务类型时可能存在语义歧异,即具有相同服务类型的服务并不能提供完全一致的服务,而提供相同服务的提供者并不一定就拥有相同的服务类型.另外,动态的构建和维护S-Club也需要一些额外的开销.

另外,国防科学技术大学的胡建强博士提出了一种双层网络发现模型[15],上层P2P结构采用结构化的Chord网络,下层则采用非结构化的P2P网络.该模型没有应用语义模型作为基础,因此,采用了较一般的Chord模型作为骨干网络.在我们的研究中,我们认识到了本体之间的联系,并基于这些联系建立起相对固定的主干网络拓扑,这样有利于提高服务发现的响应时间.郭得科[16]等人提出了一种基于Kautz图的分布式服务发现模型.该模型以Kautz图为基础,从而实现分布式服务注册.

© 中国科学院软件研究所 http://www.jos.org.cn

5 结 论

面向开放的 Internet 环境,iVCE 通过虚拟共同体来实现有效的资源聚合.结合 Internet 环境下多本体共存的现实,我们结合虚拟共同体和本体理论提出了本体社区的概念.另外,传统的集中式服务发现机制缺乏必要的可扩展机制,随着系统规则的增大,将成为系统的瓶颈.我们结合 P2P 结构的优点,针对服务发现,建立本体社区层和社区内双层 P2P 结构上的语义服务发现模型.该模型具有良好的可扩展性,服务发现的响应时间随着服务规模的增大而增长缓慢.该模型通过本体社区以及本体社区之间的映射消除服务发现中的语义异构,从而实现跨本体的服务发现.因此,该模型能够很好地适应开放 Internet 环境下多本体共同的现实.另外,通过实验可知,针对不同的要求可以通过调节相应的参数来控制社区代理的转发策略,以在服务查全率和服务发现响应时间之间得到折衷.

今后,我们将重点研究服务的动态聚类、整个系统的容错性以及模块化本体理论等一系列问题.

References:

[1] Lu XC, Wang HM, Wang J. Internet based virtual computing environment (iVCE): Concepts and architecture. Science in China (Series E), 2006,36(10):1081−1099 (in Chinese with English abstract).

[2] Paolucci M, Kawamura T, Payne TR, Sycara K. Importing the semantic Web in UDDI. In: Bussler C, Hull R, McIlraith S, Orlowska ME, Pernici B, Yang J, eds. Proc. of Web Services, E-Business and Semantic Web Workshop, CAiSE 2002. Toronto: Springer-Verlag, 2002. 225−236.

[3] Paolucci M, Kawamura T, Payne TR, Sycara K. Semantic matching of Web services capabilities. In: Goos G, Hartmanis J, van Leeuwen J, eds. Proc. of the Int'l Semantic Web Conf. (ISWC). LNCS 2342, Sardinia: Springer-Verlag, 2002. 333−347.

[4] Burstein M, Bussler C, Zaremba M, Finin T, Huhns M, Paolucci M, Sheth A, Williams S. A semantic Web services architecture. IEEE Internet Computing, 2005,9(5):52−61.

[5] Banaei-Kashani F, Chen CC, Shahabi C. WSPDS: Web services peer-to-peer discovery service. In: Arabnia HR, Chatterjee S, eds. Proc. of the Int'l Symp. on Web Services and Applications. Las Vegas: CSREA Press, 2004. 733−743.

[6] Essafi T, Dorta N, Seret D. A scalable peer-to-peer approach to service discovery using ontology. In: Proc. of the 9th World Multiconference on Systemics, Cybernetics and Informatics. Orlando, 2005. http://www.math-info.univ-paris5.fr/~seret/artgd.pdf

[7] Schmidt C, Parashar M. A peer-to-peer approach to Web service discovery. World Wide Web, 2004,7(2):211−229.

[8] Verma K, Sivashanmugam K, Sheth A, Patil A, Oundhakar S, Miller J. METEOR-S WSDI: A scalable p2p infrastructure of registries for semantic publication and discovery of Web services. Journal of Information Technology and Management, 2005,6(1): 17−39.

[9] Stuckenschmidt H, Klein M. Modularization of ontologies—Wonderweb: Ontology infrastructure for the semantic Web. 2003. http://wonderweb.semanticweb.org/ deliverables/documents/D21.pdf

[10] Liu ZZ, Wang HM, Zhou B. A scalable mechanism of semantic service discovery in multi-ontology environment. In: Cerin C, Li KC, eds. Proc. of the GPC 2007. LNCS 4459, Berlin, Herdenberg: Springer-Verlag, 2007. 136−146.

[11] Wu J, Wu ZH, Li Y, Deng SG. Web service discovery based on ontology and similarity of word. Chinese Journal of Computers, 2005,28(4):595−602 (in Chinese with English abstract).

[12] Euzenat J, Bach TL, Barrasa J, Bouquet P, de Bo J, Dieng R, Ehrig M, Hauswirth M, Jarrar M, Lara R, Maynard D, Napoli A, Stamou G, Stuckenschmidt H, Shvaiko P, Tessaris S, Acker SV, Zaihrayeu I. D2.2.3: State of the art on ontology alignment. 2004. http://www.starlab.vub.ac.be/publications/kweb-223.pdf

[13] Elenius D, Ingmarsson M. Ontology-Based service discovery in P2P network. In: Proc. of Workshop on Peer-to-Peer Knowledge Management. Boston, 2004. http://citeseer.ist.psu.edu/711664.html

[14] Hu CM, Zhu YM, Huai JP, Liu YH, Ni LM. Efficient information service management using service club in CROWN grid. In: Proc. of the 2005 IEEE Int'l Conf. on Service Computing (SCC 2005). Washington: IEEE Computer Society, 2005. 5−12. http://portal.acm.org/citation.cfm?id=1097875.1098401

[15] Hu JQ. Research on some key technologies of Web service discovery [Ph.D. Thesis]. Changsha: National University of Defense Technology, 2005 (in Chinese with English abstract).

© 中国科学院软件研究所 http://www.jos.org.cn

[16] Guo DK, Ren Y, Chen HH, Luo XS. A QoS-guranteed and distributed model for Web service discovery. Journal of Software, 2006, 17(11):2324−2334 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/2324.htm

附中文参考文献:

[1] 卢锡城,王怀民,王戟.虚拟计算环境 iVCE:概念与体系结构.中国科学(E 辑),2006,36(10):1081−1099.

[11] 吴健,吴朝晖,李莹,邓水光.基于本体论和词汇语义相似度的 Web 服务发现.计算机学报,2005,28(4):595−602.

[15] 胡建强.服务发现若干关键技术研究[博士学位论文].长沙:国防科学技术大学,2005.

[16] 郭得科,任彦,陈洪辉,罗雪山.一种 QoS 有保障的 Web 服务分布式发现模型.软件学报,2006,17(11):2324−2334. http://www.jos.org.cn/1000-9825/17/2324.htm

刘志忠(1980－),男,江西吉安人,博士生,主要研究领域为面向服务计算.

周斌(1971－),男,博士,副研究员,主要研究领域为面向服务计算,软件工程.

王怀民(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布式计算,Agent 技术.

✦✦✦