

# 多 Chord 环境下基于规则的服务推荐

崔 涛, 张抗抗, 杨 峰

(山东财政学院计算机信息工程学院, 济南 250014)

**摘 要:** 传统的服务发布和搜索难以支持语义表示和服务推理。针对该问题, 提出一种基于概念划分和规则推理的服务推荐方法, 通过服务分类本体的构建、Chord 网络与服务分类本体的动态映射以及本体规则推理, 动态生成服务推荐关系, 实现服务的语义扩展查询以及对潜在需求业务的服务推荐。实验结果证明, 该方法具有较高的查全率和查询效率。

**关键词:** Chord 网络; 服务推荐; 服务本体; ROAD 算法; 服务匹配

## Rule-based Service Recommendation in Multi-Chord Environment

CUI Tao, ZHANG Kang-kang, YANG Feng

(School of Computer & Information Engineering, Shandong University of Finance, Jinan 250014, China)

**【Abstract】** Aiming at the problem that traditional service can not support semantic representation and service reasoning during publishing and searching, a service recommendation based on classification of concept is proposed, which includes classification based on concept of service, a dynamic mapping between Chord network and service classification ontology and ontology rule-based reasoning, so dynamic service relationships for recommending are built which implement semantic query expansion and recommendation of potential business services. Experimental results show that the method has high recall and query efficiency.

**【Key words】** Chord network; service recommendation; service ontology; ROAD algorithm; service matching

DOI: 10.3969/j.issn.1000-3428.2011.19.009

### 1 概述

目前, 如何让用户方便准确地选到想要的服务仍然是一个棘手的问题。传统集中式的服务注册和更新使服务器开销很大, 易出现单点故障, 搜索到的服务也无法保证即时有效, 同时缺乏必要的语义支持, 查询只能依靠关键词匹配, 造成查全率偏低, 也很难再挖掘服务间的关系, 生成相关或预测服务。因此, 目前大多数研究都从服务网络构建和语义描述角度出发解决集中式服务管理所带来的服务发现问题。

文献[1]提出了一种基于服务关系的服务生成方法, 从 3 个视点出发对服务关系进行分析, 挖掘服务的动态语义, 生成相关服务, 但无法有效支持具有业务逻辑关系的服务。也有研究通过服务社区划分或计算历史案例相似度的方法来提供服务资源。本文提出了多 Chord 环境下一种基于规则的服务推荐方法: 先使用 Chord 对服务进行组织, 以确保在线服务是有效服务。同时, 服务本体与多 Chord 网络建立起动态映射关系, 为服务推荐提供语义转发支持。通过建立相关服务领域的本体规则, 对查询消息进行推理, 导出多个推荐服务实例。推荐服务可以体现本体规则中的相关关系和业务逻辑, 经匹配筛选后以分类的方式向用户推送。

### 2 服务的 P2P 网络

本文针对 P2P 网络实现了一个服务规则推荐系统 (Service Rules Recommendation System, SRRS), 它由服务平台和服务节点中间件构成。使用该系统对整个网络进行管理和推理查询。

#### 2.1 SRRS 系统框架

系统的整体框架如图 1 所示。其中, 服务平台上保存着服务领域本体, 它负责表述领域中所有服务概念的属性和关

系。服务网络管理层根据领域本体情况对多 Chord 网络进行相应的更新操作。在进行服务推荐查询时, 用户可能提供单一或不完整的服务条件信息, 服务推理层通过调用规则推理引擎, 使用不同的条件信息来匹配本体规则库中的相关规则, 生成推理服务信息。再把服务信息加入到查询消息中再重发给请求节点, 进行查询工作。

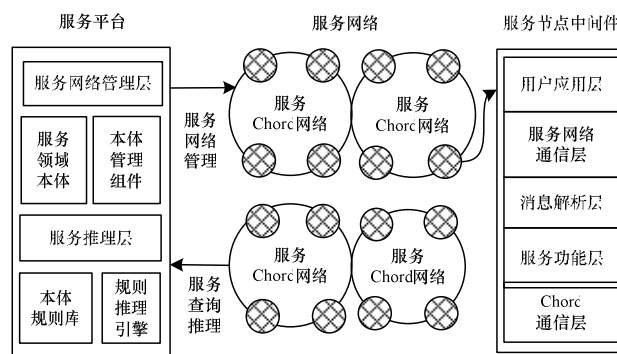


图 1 SRRS 系统框架

服务节点中间件负责服务节点、服务平台和用户三者之间的信息交互, 被安装在服务节点上。Chord 通信层使用了经典的 Chord 算法进行实现。服务功能层负责对服务信息进

**基金项目:** 国家自然科学基金资助项目(60603070); 山东社会科学规划基金资助项目(09DJGZ18); 山东财政学院博士科研启动基金资助项目

**作者简介:** 崔 涛(1986—), 男, 硕士研究生, 主研方向: 分布式系统, 软件工程; 张抗抗、杨 峰, 副教授、博士

**收稿日期:** 2011-02-28 **E-mail:** cruelsor@yahoo.com.cn

行解析并提供服务选取功能。消息解析层对接收到的消息进行解析, 根据消息内容对服务功能层进行调用。服务网络通信层负责与其他节点中间件和服务平台进行通信, 可以根据 Chord 提供的 Finger 表信息来实现查询组播算法。用户应用层提供了用户配置和控制接口, 实现了可视化界面。

## 2.2 服务分类本体的构建

服务分类本体是服务领域本体中的主干部分, 它对所涉及的领域服务进行分类。图 2 就是一个本体片段, 本体树的上下位关系表示了服务概念的父子关系, 底层的概念可以称为服务原子概念。每个服务概念都有通过关系相联系的同义概念和概念特征属性, 它们都是对这个服务概念的一种扩展描述。

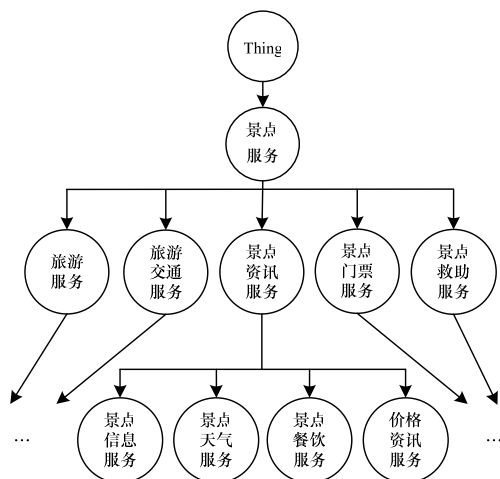


图2 服务分类本体片段

用户在节点中间件中根据提供服务的功能性领域为每个 Web 服务选择服务概念后, 通过向服务平台请求合适的入口信息, 加入到对应的 Chord 网络中。例如, 一个提供景观介绍信息的服务属于景点信息服务概念, 加入网络时要加入到代表景点信息服务的 Chord 网络中, 但是如果这个网络不存在, 它会先查询父概念映射的 Chord 网络是否存在, 如不存在, 自己创建。

## 2.3 服务网络的构建

服务网络中的 Chord 会与服务分类本体中的领域本体建立一一映射关系<sup>[2]</sup>, 如图 3 所示。

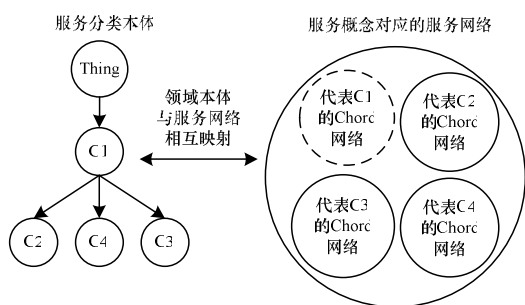


图3 由服务本体映射的 Chord 网络

一个 Chord 中的服务节点都是属于同一服务概念的, 但它们具体提供的功能和接口信息可能不同。每个节点都是平等的, 任意节点可以发起查询和跨网络转发消息, 这也可以保证每个节点加入、退出的自由和负载均衡, 使整个系统更加健壮。把一类服务归入一个 Chord 网络, 可剔除大量的无关服务, 在服务查询时匹配选取效率大大提高。如果服务规模较小, 可以把几个规模较小的服务概念网络合并建立成一

个 Chord 网络, 用它们的父概念进行映射, 系统性能不会受到大的影响。知名节点选取算法如下:

### 算法1 selectFamouseNode(Msg)

**输入** 概念、IP 及端口信息 Msg{SC; IP; Port}

**输出** 加入消息 NodeMsg

(1) IF Msg.SC in List{SC} //List{SC} 是否存在服务概念

{C=List{SC}[i];} //匹配成功

(2) ELSE

{If(IsExistExtendConcept(List{SC}, Msg.SC)) //对概念的扩展词

//匹配

C=List{SC}[i];

(3) IF(IsExistChordInfo(C)) //判断是否存在代表 C 的 Chord 信息 {NodeMsg.Order="join";} //加入操作

(4) ELSE

{NodeMsg.Order="create"

//建立操作

InsertFamouseNodetable(Msg,C);} //保存知名节点

(5) Return NodeMsg;

算法 1 把节点的服务概念映射为相应的 Chord 网络代表的服务概念, 节点映射到服务分类本体上的概念可能是 Chord 网络代表的服务概念或只是它的子类。这个粒度是由平台根据实际环境进行设置的, 这样既可以控制网络的规模又可以提高匹配效率。

## 2.4 服务网络的动态调整

服务本体概念在开放式的互联网环境下不是一成不变的, 要根据实际情况对服务本体进行调整。调整可以分为 4 种操作: 加入, 删除, 分裂和合并。加入并不需要对 P2P 网络进行特意调整, 删除也只需移除对应服务的 Chord 网络和删除服务概念。但发生一个本体概念分裂成多个新本体概念或多个本体概念合并成一个新本体概念时, 就需要对网络中的服务重新组织, 以保证服务节点和 Chord 网络都能及时正确地与服务分类本体相映射。服务本体分裂和合并后进行的拓扑调整算法如下:

### 算法2 拓扑调整算法

**输入** 被调整的本体  $L1=\{A1,A2,\dots,A_n\}$ ; 调整后的本体  $L2=\{B1,B2,\dots,B_n\}$

(1) 分裂操作

$C=L1.next()$ ; //获得被分裂的本体

$Msg[]=\text{chooseConcept}(C)$ ; //向代表 C 的 Chord 发送重选服务概念消息

While(Msg.length) //对返回消息, 进行重新选取知名节点

{SelectFamouseNode(Msg[i]);}

RemoveChord(C); //旧 Chord 移除

(2) 合并操作

$C1=\text{getMaxNumber}(L1)$ ; //获取最大节点数目的服务概念

$L1.remove(C1)$ ; //移除 C1, 保留 C1 的 Chord

ModifyFamouseNodetable(C1,B1); //更新 C1 的知名节点信息

NodeMsg=FamouseNodetable(C1).Msg; //生成节点加入信息

While(L1.hasNext()) //组播发送

{

broadcastChord(L1.next(),NodeMsg);}

## 3 基于规则推理的服务推荐

本文使用 Jena 引擎和本体推理规则库来生成服务关联关系, 根据关系查询出相关服务推荐给用户。推理规则的建立需要领域专家根据实际情况进行编制, 具有很大的灵活性, 同时减少了无用信息的干扰。

### 3.1 服务关系

在某一领域或业务流程中,各种服务往往具有一定的逻辑关系,同样从用户角度出发,他们前后搜索的服务之间也可能是具有内在联系的。通过服务间关系可以方便地把各类相关的服务推荐给用户,尤其是当用户对相关领域或业务流程不熟悉时,可以快捷地得到想要的后继或相关服务。本文定义了3种基本关系类型:

(1)相关关系(related):表达了服务之间具有相似性或者可替代性,在功能上可能都满足用户的需求。

(2)推荐关系(recommend):表达了服务之间具有一定的使用关联性,使用其中的一个服务可能会引起对另一种服务的使用。例如,景点信息服务可以推荐出景点天气服务。

(3)流向关系(goto):表达了服务之间在业务流程中具有次序性和约束性。例如,用户要网上预订酒店时,只有使用酒店会员注册服务后,才有权利调用酒店预订服务,所以,在这个流程中2种服务具有明显的流向和先后约束关系。

### 3.2 基于规则的服务关系推理

在服务领域本体建立好后,单一的服务概念无法细致地表达出服务间的差别。本文通过服务本体实例来解决这个问题,利用对象关系(servicetype)对服务实例的特征进行描述,如功能类型、服务属性或参数概念,这些特征类在领域本体中都是作为 servicetype 对象关系的值域进行约束的。服务的实例所具有的特征数目不尽相同,每个实例都应该能准确地代表一种服务概念下的一类服务。本文根据这些典型的服务实例编写规则,而其中的特征值成为推理规则的约束条件。

在服务平台启动时,把规则加载到 Jena 的推理机中,当查询消息到来时,服务推理层先取得要查询的服务概念和条件特征,然后生成相应的3类基本关系的查询语句,交给 Jena 引擎进行查询,Jena 引擎会利用加载了的规则进行推理筛选,得出具有相关、推荐或流向关系的服务概念和实例,最后发送给对应概念的 Chord 网络进行匹配选取。推理过程如图4所示。

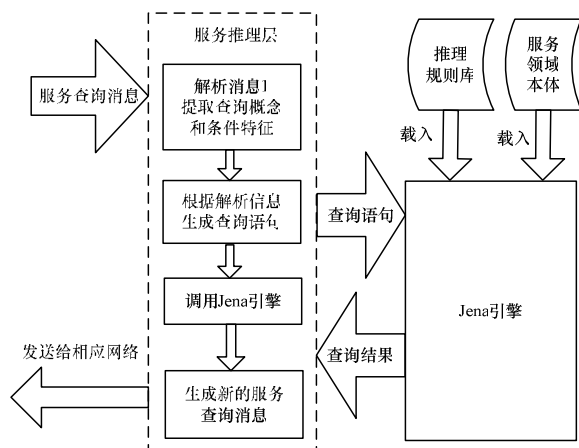


图4 服务推理过程

本文使用 SWRL(Semantic Web Rule Language)编写本体规则。SWRL 是本体语言的扩展公理集,可以直接使用本体模型中的类、属性、实例和关系等元素,从而完美实现对本体的规则推理。规则中左半部分是规则的前件,由一系列的原子约束进行合取,每条约束是本对对本体元素的关系描述或定义。规则的右半部分是推理结果。约束可以有多种形式,如:  $C(?x)$  表示  $x$  是  $C$  代表的本体类;  $P(?x,?y)$  表示  $x$  和  $y$  之间具有  $P$  这种本体关系属性;  $type(?x,C)$  表示  $x$  是本体类  $C$  的

实例。

本文把推理规则分为2类:

(1)直接关系规则:规则中直接定义出2类服务的关系。

例如:

$servicetype(?x,?n) \wedge servicetype(?z,?m) \rightarrow related(?x,?z)$

(2)间接关系规则,通过推理为2个没有直接关系的服务建立关系。例如:

$servicetype(?x,?n) \wedge related(?x,?z) \wedge goto(?p,?x) \rightarrow goto(?p,?z)$

通过这2类基本推理规则,可以定义不同类型的服务关系。只要制定出优秀的规则库,就可以动态地把服务关系展现给用户,满足用户的潜在需求。这里的推理是概念层次上的推理,从一个查询概念服务可以迅速扩展到其他概念服务,有效地覆盖业务流程或用户行为涉及的各个方面,既降低了进行推理时的复杂性,避免了涉及过多的服务细节推理时可能导致的服务平台计算压力过大,又可以全面覆盖可能涉及的服务领域,保证了系统的稳定性和可用性。

## 4 服务的选取

在 Chord 中使用 ROAD 算法<sup>[3]</sup>来实现消息的高效率组播。同时对节点上的服务接口进行描述,根据服务匹配度决定是否选取。

### 4.1 消息组播和服务匹配

ROAD 的转发策略如图5所示,其中,虚线为第1次转发;实线为第2次转发。 $N1$  作为 Chord 中的首个节点,把消息转发给 Finger 表中存储的节点,即节点  $N2$ 、 $N4$ 、 $N7$ ,并把相邻的下一节点作为上一节点的再次转发域。例如  $N4$  收到消息后会根据 Finger 表信息选出在  $N4 \sim N7$  区间内的节点作为再次转发域进行转发。以此类推,最终完成整个环上的消息组播。

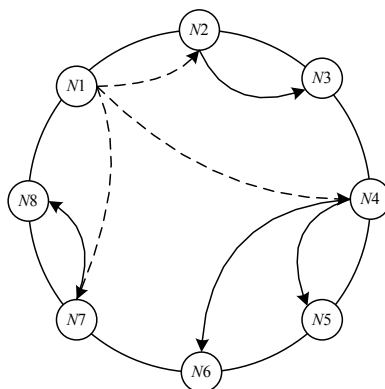


图5 ROAD 算法示意图

本文借鉴文献[4-5]提出的模型和方法对服务进行语义描述和匹配。把 Web 服务抽象成条件( $P$ )、效果( $E$ )、输入参数( $I$ )、输出参数( $O$ )和服务质量参数( $Q$ )来表示。服务匹配相似度主要通过计算本体概念间的上下位关系获得。其计算公式为

$Sim(c1,c2) = \frac{\alpha}{\alpha + k}$ , 其中,  $c1$  为目标服务概念;  $c2$  为匹配服务概念;  $\alpha$  是一个调节参数;  $k$  是一个整数,  $k$  的取值取决于2个概念间的关系。最终根据各分量计算出来的相似度在乘上设置的权重,相加得到服务匹配度。再把匹配度和用户查询时所设置的阈值  $p$  进行比较,超过阈值的可以作为选取节点返回给请求节点。

### 4.2 实验分析

实验使用 Java 编程构建不同服务规模下的 P2P 环境,对

(下转第37页)