

一种基于推荐网络和蚁群算法的服务发现方法

谢晓芹¹⁾ 宋超臣^{1),2)} 张志强¹⁾

¹⁾(哈尔滨工程大学计算机学院 哈尔滨 150001)

²⁾(黑龙江省电子信息产品监督检验院 哈尔滨 150090)

摘 要 服务分散存储在互联网上,随着互联网上 Web 服务数量的迅速增长,自动、准确、快速地搜索 Web 服务已经成为面向服务计算中的难点和关键问题,文中针对在开放、动态环境下现有的服务发现研究中存在的搜索效率不高、负载不均衡和语义欠缺等问题,提出了一种基于推荐网络和蚁群算法的服务发现方法.首先,该方法构建一个自组织服务推荐网络模型,并给出了相关策略,从而适合大规模开放、动态的网络环境,为服务发现提供了搜索空间和基础.其次,在自组织服务推荐网络模型的基础上,给出了一种基于蚁群算法的服务发现方法,该方法使用推荐有效地提高了服务发现的成功率和查全率,引入蚁群算法的思想有效地解决了服务发现中的网络负载均衡问题.最后,实验结果证明了该方法的正确性和可行性.

关键词 服务发现;推荐网络;蚁群算法;负载均衡

中图法分类号 TP312 **DOI 号**: 10.3724/SP.J.1016.2010.02093

A Service Discovery Method Based on Referral Network and Ant-Colony Algorithm

XIE Xiao-Qin¹⁾ SONG Chao-Chen^{1),2)} ZHANG Zhi-Qiang¹⁾

¹⁾(College of Computing Science and Technology, Harbin Engineering University, Harbin 150001)

²⁾(Heilongjiang Province Electronic & Information Products Supervision Inspection Institute, Harbin 150090)

Abstract Services are archived dispersively on Internet. With the rapid increasing number of Web services, how to discovery the desired Web service automatically and accurately has been a critical hot issue in service-oriented computing. Aiming at such problems as the low search efficiency, load imbalance and lacking semantic understanding existing in current service discovery researches, this paper proposes a new service discovery method based on referral network and ant-colony algorithm. Firstly, a self-organizing service referral network model-SSRNM is constructed. SSRNM defines related policies and provides the search space for service discovery, thus it can be fit for the dynamic and open environment. Secondly, based on SSRNM, an ant-colony algorithm-based service discovery method ABSDA is presented. This method makes use of recommendation to improve the success and recall rate of service discovery, and adopts the idea of ant colony algorithm to solve the network load balancing problem effectively. Finally, the experimental results prove that this method is correct and feasible.

Keywords service discovery; referral network; ant-colony algorithm; load balance

收稿日期:2010-06-08;最终修改稿收到日期:2010-08-31. 本课题得到国家自然科学基金(60803037,60803036)、国家“八六三”高技术研究发展计划项目基金(2009AA01Z143)、哈尔滨市科技创新人才研究专项基金(RC2010QN010024)和中央高校基本科研业务费专项资金(HEUCFZ1010,HEUCF100602)资助. 谢晓芹,女,1973年生,博士,副教授,主要研究方向为服务计算、网络模式下知识处理、社会网络和信息检索等. E-mail: xiexiaoqin@tsinghua.org.cn. 宋超臣,男,1979年生,硕士研究生,主要研究方向为服务计算. 张志强,男,1973年生,博士,教授,主要研究领域为数据库、信息检索等.

1 引 言

面向服务的计算是一种新的计算模型, Web 服务及其组装也成为数据和信息集成的有效机制. 但是随着 Internet 上 Web 服务数量和种类的快速增长, 自动、准确地搜索 Web 服务已经成为面向服务计算中的难点和关键问题.

由于互联网是一个开放、动态的环境, 这给 Web 服务发现技术带来了新的难题. 首先, 因为服务提供者、服务使用者以及服务本身的行为和数量都是无法预知的, 他们可以随时加入或退出系统. 这就要求服务发现技术能够适应网络的动态拓扑结构. 而推荐网络可以很好地适应动态开放的环境^[1]. 其次, 集中式服务发现技术无法满足互联网服务搜索的需要. 采用 UDDI 等集中式 Web 服务搜索技术虽然具有实现简单、易于操作等优点, 但随着网络的发展, 集中式服务搜索技术的性能瓶颈、单点失败等缺点逐渐显现出来. 尤其是在 Web 服务语义描述迅速发展的今天, 集中式服务搜索技术已经不能满足语义匹配的需求. P2P 能高效地实现服务的集成及资源的自治, 可以构建一种基于智能体的 P2P 环境下的服务推荐网络^[1], 采用分布式服务发现技术来解决集中式服务发现技术的缺点. 但目前研究中服务发现和搜索技术还存在以下几方面问题:

(1) 搜索效率方面. 集中式服务发现具有性能瓶颈, 不支持语义查询. 与集中式服务发现不同, P2P 环境下服务随机分布在各个节点上, 服务定位尤其是稀疏服务资源的定位非常困难. 搜索过程往往需要遍历较多的节点才能获得较高的查全率, 导致网络承受很大的带宽压力. 为了保证比较高的查全率, 无结构 P2P 网络通常采用泛洪算法, 这容易造成网络拥塞, 而采用随机游走算法查全率又得不到保证. 有结构 P2P 网络虽然可以准确地实现资源定位, 但其在节点加入或离开时需要大量的修复操作, 不适合动态、开放的环境.

(2) 负载均衡方面. 由于基于 P2P 网络的服务发现具有一定的盲目性, 造成一些节点遭到频繁访问而消耗较多的主机资源和带宽资源, 而另一些节点却长期处于空闲状态. 如何调动网络中更多的节点来均摊负载是目前的一个难点问题.

(3) 语义查询方面. 由于进行语义匹配时, 对资源消耗极大, 因此集中式服务发现技术已经不能满足要求. 目前, 一些基于 P2P 网络的服务搜索技术

仍然使用关键字或描述文件等描述服务, 不支持语义查询请求. 而支持语义查询的 P2P 网络, 在大范围进行语义匹配的情况下, 会造成主机资源极大的浪费. 因此, 如何在 P2P 网络中缩小语义匹配范围是执行语义匹配问题的关键.

本文就主要针对以上 3 方面问题, 研究了一种基于推荐网络和蚁群算法的服务发现方法. 本研究中的服务是指能完成某种功能、可重用的存储在互联网上的各种软件资源, 例如可以是软件构件、Web 服务等实体. 本文通过将服务映射为相应的服务节点, 在此基础上构建了自组织服务推荐网络, 通过推荐解决了现有研究中服务发现具有盲目性的问题; 还基于服务推荐网络提出了搜索方法 ABSDA, 利用蚁群算法实现了网络的负载均衡. 本文提出的服务发现方法和模型能够用于互联网中各种软件资源的共享和查找, 可以应用到相关搜索引擎中, 为实现互联网下服务发现和组装提供技术保证.

本文第 2 节给出自组织服务推荐网络模型的描述和构建方法; 第 3 节给出基于推荐网络和蚁群思想的服务发现方法; 第 4 节给出实验数据用以验证本方法的可行性; 第 5 节总结本文并指出下一步的工作.

2 自组织服务推荐网络模型

大量的服务资源分散存储在互联网上, 实现组装的前提是要实现对这些服务资源准确、自动的发现. 但目前的服务发现方法存在效率不高、负载不均衡等问题, 缺乏对这些服务资源有效的管理方法. 借鉴推荐思想, 本研究建立了一个自组织服务推荐网络 (Self-organization Service Referral Network Model, SSRNM), 为后续的服务发现提供了搜索空间, 也提供了一种互联网下服务资源的共享管理模型, 并解决了存在的上述问题. 自组织服务推荐网络模型是一种无结构 P2P 网络, 具有较强的动态性且不需要路径恢复的算法, 不存在单点失败、性能瓶颈等问题. 本节将给出模型的描述和构建方法.

2.1 模型描述与相关概念

定义 1. 自组织服务推荐网络模型可表示为 $SSRNM = (Ns, Rs, Policies)$. 其中, $Ns = \{node\}$ 是服务节点 $node$ 的集合; $Rs = \{SimilarityRelation, NeighborRelation\}$ 是服务节点之间关系的集合, 包括相似关系和相邻关系. $Policies$ 表示模型定义的所有策略的集合, 包括服务标记策略、服务集划分策

略、邻居选择策略和邻居列表更新策略。具体策略描述将在 2.2 节给出。

定义 2. 服务节点是一个 7 元组：

$$Node = (type, SSynsetIDs, IP, Agent, NeighborList, ServiceSet, SKeywords),$$

其中 $type \in \{SP, SC\}$ ，用来描述服务节点类型， SP 表示服务提供者类型， SC 表示服务消费者类型。 $SSynsetIDs$ 是服务的语义标识符集合。 IP 表示服务所在主机的 IP 地址。 $SKeywords$ 是描述服务的一组关键字。 $NeighborList$ 表示邻居列表。 $Agent$ 是运行在节点上的智能体，它负责发布 $SSynsetID$ 并响应服务查询请求，给出匹配信息或推荐。 $ServiceSet$ 表示服务集。

定义 3. 邻居。假设当服务搜索成功时的搜索路径 $Path = \langle sn_m, sn_{m+1}, \dots, sn_n \rangle$ ，称 sn_i 和 sn_{i+1} ($m \leq i < n$) 之间具有相邻关系， sn_{i+1} 是 sn_i 的邻居。形式化表示为

$$NeighborRelation = \{ \langle sn_i, sn_{i+1} \rangle \mid sn_i \in Path \cap sn_{i+1} \in Path \cap m \leq i \leq n-1 \}.$$

定义 4. 邻居列表 ($NeighborList$)。当前服务节点的所有邻居构成的集合称为邻居列表。每个节点维护一个邻居列表。

定义 5. 服务集 ($ServiceSet$)。设 sn 为一个服务节点，它的服务集是指与该节点具有相同 $SSynsetID$ 值的所有服务节点的集合，可表示为

$$ServiceSet = \{ node_k \mid (node_k.SSynsetIDs \cap sn.SSynsetIDs \neq null) \cap 1 \leq k \leq n \},$$

其中 n 表示 SSRNM 中的服务节点个数。

定义 6. 相似关系。设 sn_i, sn_j 为两个服务节点，如果 sn_i, sn_j 属于同一个服务集，或者一个节点在另一个节点的服务集中，则称 sn_i 和 sn_j 是相似的。形式化表示为

$$SimilarityRelation = \{ \langle sn_i, sn_j \rangle \mid$$

$$(sn_i \in sn_j.ServiceSet \cup sn_j \in sn_i.ServiceSet \cup (sn_i \in sn_k.ServiceSet \cap sn_j \in sn_k.ServiceSet)) \cap 1 \leq i \leq n \cap 1 \leq j \leq n \cap 1 \leq k \leq n \cap i \neq j \neq k \},$$

其中 n 表示 SSRNM 中的节点个数。

服务节点的 7 个元素之间相互约束，具有以下依赖关系：

(1) 标记依赖。设 sn 为一个服务节点，通过服务标记策略， $sn.SKeywords$ 被映射为 $sn.SSynsetIDs$ ，表示为

$$sap: sn.SKeywords \rightarrow sn.SSynsetIDs.$$

(2) 类型依赖。设 sn 为一个服务节点， $type$ 的

不同取值决定了 $SSynsetID$ 和 $ServiceSet$ 的取值，具体约束条件为

if $sn.type = SC$ then

$sn.SSynsetIDs$ 和 $sn.ServiceSet$ 为空；

if $sn.type = SP$ then

$sn.SSynsetIDs$ 和 $sn.ServiceSet$ 不为空；

定义 7. 服务信息度 (SIG) 是指一个节点包含某一服务的可能程度。设服务节点 sn_i 包含了服务 s_j ，则 sn_i 节点的服务信息度表示为 SIG_j^i 。一个节点包含几个服务，则可以有几个服务信息度。

在本研究中，服务信息度是一个变量，类似于蚁群算法中的信息素的概念。在进行网络初始化时，为其指定初始值。随着搜索次数的增多其值会不断变化。

定义 8. 服务信息向量 (SIV)。设服务节点 sn 包含 n 个服务 s_1, s_2, \dots, s_n ，则 sn 节点的服务信息向量 $SIV = (SIG_1, SIG_2, \dots, SIG_n)$ 。

定义 9. 社交度 (SD)。设服务节点 sn 包含 n 个服务 s_1, s_2, \dots, s_n ，则 sn 所包含的所有 SIG 之和称为该节点的社交度。表示为

$$SD(sn) = \sum_{i=1}^n SIG_i \quad (1)$$

每个节点利用定义的策略提供或消费服务，整个过程是自组织的。服务消费者节点只能根据本地邻居列表的服务信息度给出推荐，而服务提供者节点既可以响应服务请求也可以给出到其它节点的推荐。

2.2 模型策略

SSRNM 模型中定义的相关策略用于模型的构建和管理，下面阐述各个策略。

策略 1. 服务标记策略。

服务关键字是用来描述服务的一组关键字。服务标记策略用于将描述服务的关键字映射为语义标识符 $SSynsetID$ ，实现服务的语义标记，从而消除服务关键字的歧义性。在 WordNet 中最主要的关系是“is-a”关系，每个同义词集共享同样的属性定义，并使用 $synsetID$ 来唯一标记。本研究通过“is-a”关系或同义词集将服务关键字映射为 $SSynsetID$ 。本策略包含两个规则，设 sn 是服务节点， key_1 和 key_2 是两个关键字， $key_1 \in sn.SKeywords$ ， $key_2 \in sn.SKeywords$ 。

规则 1. 若 key_1 是 key_2 的下位概念，且 key_1 和 key_2 在 WordNet 中对应的标号分别为 $synsetID_1$ 和 $synsetID_2$ ，则用 $synsetID_1$ 来标记服务，即

$$sn.SSynsetIDs = sn.SSynsetIDs \cup synsetID_1.$$

规则 2. 若 key_1 和 key_2 属于同一个同义词集, 对应的同义词集标号为 $synsetID$, 则使用 $synsetID$ 来标记该服务, 即

$$sn.SSynsetIDs = sn.SSynsetIDs \cup synsetID.$$

使用服务标记策略不仅可以有效地消除关键字的歧义性, 而且还有助于建立服务集.

策略 2. 服务集划分策略.

服务集划分策略用于将服务提供者节点加入到相应的服务集中, 有利于提高搜索过程中的查全率. 在 SSRNM 的构建过程中, 本研究使用 WordNet 将关键字映射为 $SSynsetID$, 并对映射后的服务节点进行分类, 将具有相同 $SSynsetID$ 的服务节点组成服务集. 在 SSRNM 中的每个服务节点维持一个服务集, 这样当一个服务节点被搜索到时, 可以对该服务节点的服务集内的所有其它服务节点采用泛洪算法进行二次语义查询, 从而能进一步满足搜索算法的查全率要求. 而且, 每个服务集不需要超级节点进行代理, 不存在单点故障.

服务集划分策略表示为: 设 sn_1, sn_2 为两个服务节点, 如果满足下列条件: $sn_1.SSynsetIDs \cap sn_2.SSynsetIDs \neq \text{null}$, 则执行下列操作:

$$(1) sn_1.ServiceSet = sn_1.ServiceSet \cup sn_2.$$

$$(2) sn_2.ServiceSet = sn_2.ServiceSet \cup sn_1.$$

策略 3. 邻居选择策略.

节点使用邻居选择策略决定服务发现过程中下一跳的地址. 邻居选择策略是自组织服务推荐网络的关键策略, 用于减少服务发现过程中的盲目性, 提高搜索成功率. 设 sn 为服务节点, 邻居选择策略表示为

(1) 查看 sn 的邻居列表中, 是否存在拥有被查询服务信息的节点. 如果存在, 选择服务信息度最大且未被查询过的邻居节点作为查询节点.

(2) 如果不存在拥有被查询服务信息的节点, 则选择社交度最大的节点作为查询节点.

SSRNM 中的每个节点维持一个邻居列表, 并使用 **SIV** 描述一个给定的邻居. 使用该列表, 一个节点应用邻居选择策略决定向哪个节点发布或搜索服务.

策略 4. 邻居列表更新策略.

邻居列表更新策略主要用于邻居的增加与删除, 实现网络的动态管理. 典型的 P2P 网络搜索算法是, P2P 节点向其它节点广播查询请求, 其它节点继续向前广播请求^[2]. 与其不同, 在推荐网络中, 节点接收到查询请求后, 并不向前广播请求而

是向发出请求的节点返回一个推荐, 发出查询请求的节点根据推荐调整搜索策略继续搜索. 在本文中, SSRNM 模型中的每个节点维护一个邻居列表用以保存其它节点的服务信息度, 并通过邻居列表寻找满足需求的服务提供者. 节点之间通过邻居列表相互通信. 邻居列表的大小是固定的, 节点通过对邻居列表的调整实现网络拓扑的改变. 当邻居数量达到一定阈值时, 通过剔除一些使用率低的邻居节点来实现邻居列表的更新.

邻居列表通过搜索过程或服务发布过程得到更新, 不直接依靠其它节点. 邻居列表更新过程操作简单, 不需要定义路径策略. 采用以下 2 种方法对邻居列表进行更新:

(1) 当一个节点 sn_1 收到另一个节点 sn_2 发来的查询请求或查询服务信息时, 如果发布请求的节点不在 sn_1 的邻居列表当中, 即满足下列条件时:

$$sn_2 \notin sn_1.NeighborList,$$

这个节点 sn_2 将会被加入到 sn_1 的邻居列表中.

(2) 当一个节点 sn_1 收到一个推荐 sn_2 时, 如果被推荐的节点 sn_2 不在 sn_1 的邻居列表当中, 即满足下列条件时:

$$sn_2 \notin sn_1.NeighborList,$$

那么这个推荐 sn_2 将会被加入到 sn_1 的邻居列表中.

使用邻居列表更新策略可以实现网络的自组织, 而且节点之间的交互是异步的和自治的, 一个被请求的节点可以提供服务或者给出推荐以响应另一个节点的请求.

2.3 SSRNM 构建过程

SSRNM 的构建过程和服务搜索过程是自组织的. SSRNM 的构建过程实质也是一个初始化过程, 主要完成服务发布和服务分类 2 部分功能. 在初始化过程中, 每个节点被随机指定若干个邻居. SSRNM 初始化过程^[3]如下:

1. 参数初始化. n 是网络节点数, γ 是第一个服务节点发布服务信息时, 包含该服务信息的节点的覆盖率.

2. 随机选择一个服务提供者节点, 该节点使用服务标记策略将关键字映射为同义词集标号 ($synsetID$), 发布数量 (具有某一服务 $synsetID$ 信息的节点的数量) 为 0.

3. 如果接收到推荐, 更新服务提供者节点的邻居列表, 检查发布数量是否达到 $n\gamma$ (γ 为调节参数). 如果已达到, 向服务集内所有节点发送服务集信息, 转到步 6. 如果未达到, 使用邻居选择策略选择一个邻居, 并将服务的 $synsetID$ 发布到该邻居节点上, 发布数量加 1. 在使用邻居选择策略选择邻居节点时, 如果邻居节点在本节点的服务集中, 那么该邻居节点将被忽略.

4. 当邻居节点接收到服务信息,如果服务提供者节点不在该邻居节点的邻居列表中,服务提供者节点将会被追加到该邻居节点的邻居列表中.邻居节点更新邻居列表中服务提供者节点的信息度.检查邻居节点的 *synsetID* 与服务提供者节点的 *synsetID* 是否相同.如果相同,返回服务相同标志;否则,邻居节点返回一个推荐.

5. 服务提供者节点检查返回结果,如果是推荐转到步 2,如果是服务相同标志,则这两个节点应被划分到同一服务集中.服务提供者节点将相应的邻居节点加入到服务提供者节点的服务集中,并将服务集信息发送到邻居节点中.该邻居节点变成新的服务提供者节点,继续发布服务的 *synsetID*,转到步 2.

6. 检查是否还存在未发布服务 *synsetID* 的节点.如果有,从中选取一个服务节点发布服务 *synsetID*,转到步 1.如果没有,说明所有节点都完成了发布,初始化过程结束.

3 基于 SSRNM 的服务发现方法

3.1 问题描述

SSRNM 给出了服务搜索空间,因此服务发现问题可以表示为:给定 SSRNM,假设服务查询请求是一个或多个关键字 *keywords*,要在 SSRNM 中找到相应的服务节点 *Node*.

3.2 算法基本思想

为了阐述方便,我们将基于 SSRNM 的服务发现方法分为两部分进行论述.首先先阐述基于推荐的服务发现基本算法(Basic Service Discovery Algirhtm, BSDA),在此基础上,将推荐和蚁群算法相结合,提出基于蚁群的服务发现算法(Ant-Algorithm-Based Service Discovery Algorithm, ABSDA).

服务发现基本算法 BSDA 的基本思想是:首先,查询请求中的关键字 *keywords* 通过语义匹配被映射为一个或多个语义标注 *SSynsetID*.然后,使用邻居选择策略选择接受节点.若接受节点满足查询请求,则查询结束.否则,接受节点返回若干推荐.搜索节点将这些推荐更新到邻居列表后,继续查找.如果所有邻居都已被搜索,则查询失败.

然而这个基本方法虽然能有效地提高服务搜索的效率,但在使用邻居选择策略进行邻居选择时,即使选择的邻居负荷过重或者网络拥塞,仍然会向该邻居节点发出查询请求.针对这一现象,本研究引入蚁群算法以避免网络拥塞,从而提出基于 SSRNM 的服务发现的增强方法:ABSDA. ABSDA 算法优点在于将推荐网络和蚁群算法进行了有效地结合,

并且蚂蚁只根据目标节点列表进行寻路,操作简单且每只蚂蚁行为是自组织的.

ABSDA 方法利用蚁群算法的思想,将整个服务发现的过程类比为蚂蚁觅食的过程,并将服务提供者节点发布的服务信息度存放到搜索路径上的各节点中,相当于是蚁群算法中的“信息素”,根据蚂蚁觅食行为的特性,通过蚂蚁释放信息素的正反馈机制来指导搜索前进的方向,从而大大提高随机游走算法的查询效率.传统的蚁群算法在查询过程中,蚂蚁只根据本地信息素来决定下一跳节点,类似于马尔可夫过程.而本研究中蚂蚁在查询过程中,不仅根据本地服务信息度还考虑到蚂蚁本身携带的信息来决定下一跳节点.这样,实现了蚁群算法和推荐网络的有效结合,进一步提高了查找效率.蚂蚁在搜索到目标节点后,生成一只回溯蚂蚁沿原路径返回并更新路径上各节点的服务信息度,在返回到源节点后便死亡.

本研究借鉴了文献[4]的信息素更新的思想,对于某一查找成功的服务,使用以下公式更新该服务在查询路径上各节点的服务信息度:

$$SIG^k(n) = SIG^k(n-1)(1-\tau) + \tau \quad (2)$$

$$SIG^k(n) = SIG^k(n)(1-\rho) \quad (3)$$

式中: τ 为常量且 $0 < \tau < 1$, ρ 是信息度挥发因子且 $\rho < 1$, $SIG^k(0)$ 为常量, k 表示查找成功路径上的节点, n 表示第 n 只回溯蚂蚁返回该节点.

文献[4]中,信息素初始值设置为 1,且当有命中消息包从某个邻居返回时,节点对所有邻居的信息素进行更新.而在本文中,只对该邻居相应的服务信息度进行更新,且服务信息度初始值设置为 0.2.

本研究中 $0 < SIG^k(0) < 1$. 假设 $0 < SIG^k(n-1) < 1$, 由式(2)得

$$\begin{aligned} SIG^k(n) &= SIG^k(n-1)(1-\tau) + \tau \\ &= SIG^k(n-1) + \tau[1 - SIG^k(n-1)]. \end{aligned}$$

因为 $0 < \tau < 1$ 且 $0 < SIG^k(n-1) < 1$, 所以 $1 > SIG^k(n) > SIG^k(n-1)$. 可见式(2)可以有效地避免服务信息度的无限累加.通过服务信息度的挥发,可以使邻居列表中一些不常使用的邻居的服务信息度逐渐减少.当邻居的服务信息度低于某一阈值时,节点便将该邻居从邻居列表中删除.

在邻居列表中,邻居的服务信息度采取类似蚁群算法中信息素的更新方法.服务信息度不可能无限大,因此当服务信息度增大到一定程度后便接近于某一定值.根据式(2)本研究中服务信息度最大值无限接近于 1,并且服务信息度每隔一定时间段根

据式(3)进行挥发。

在传统的蚁群算法中,蚂蚁根据当前节点的信息素决定下一跳的地址,整个搜索过程可以被视为马尔可夫过程.本研究为了进一步提高搜索效率,将推荐引入到蚁群算法中.在服务发现的过程中,蚂蚁保存有一个类似于节点的邻居列表的目标节点列表.当蚂蚁到达一个节点后,节点根据邻居选择策略给出一些推荐,蚂蚁将这些推荐更新到蚂蚁的目标节点列表中.蚂蚁查看目标节点列表中是否存在拥有目标服务的节点信息,如果不存在,选择社交度最大的节点作为下一跳地址;如果存在,使用式(4)计算 P_i ,以概率 P_i 选择下一跳的地址:

$$P_i = \frac{SIG^i}{\sum_{k=1}^N SIG^k} \quad (4)$$

式中: N 表示蚂蚁的目标列表中包含目标服务的节点的个数。

这样即使蚂蚁不知道目标节点的信息,也可以通过社交度消除搜索的盲目性,进一步提高查找效率.在知道目标节点的情况下,可以通过式(4)实现网络流量的调节,充分利用网络资源。

3.3 ABSDA 算法

基本 BSDA 服务发现算法的形式化描述如下。

算法 1. 基本的 BSDA 算法。

输入:自组织服务推荐网络 SSRNM,查询请求 *keywords*

输出:服务提供者节点 *Node*

```
{
  将查询请求 keywords 映射为 synsetIDs;
  选择一个服务消费者节点 consumer, 使得
    synsetIDs ∈ consumer.SSynsetIDs;
  int searchCount = 0;
  while(consumer.neighborlist != null &&
    searchCount < n *  $\gamma$ ) {
    receiveNode = consumer.selectNode();
    //使用邻居选择策略选择接受节点
    consumer.sendQuery(receiveNode);
    //向目标节点发出查询请求
    if(receiveNode.satisfy()) {
      //如果接受节点满足请求的服务
      return (receiveNode); //查询成功
    } else {
      referral = receiveNode.selectNode();
      //接受节点使用邻居选择策略返回推荐
      consumer.updateNeighborList(referral);
      //消费节点更新邻居列表
    }
  }
  searchCount ++;
}
```

而增强的 ABSDA 算法过程为:首先,请求被映射为一个或多个语义标识符.生成查询蚂蚁,节点使用邻居选择策略选取服务信息度较大的几个节点(或社交度较大的几个节点)更新到查询蚂蚁的目标节点列表中.查询蚂蚁根据式(4)决定下一跳的地址.接受节点如果满足查询请求,查询成功.否则,查询蚂蚁将访问的节点从目标节点列表中去除.访问节点根据邻居选择策略给出推荐,蚂蚁将这些推荐更新到目标节点列表中,继续查找.如果蚂蚁的目标节点列表都已被搜索,则查询失败.蚂蚁在运动过程中,保存经过的路径.当蚂蚁到达目标节点后,由回溯蚂蚁沿着原路返回并更新路径上的服务信息度.返回源节点后,蚂蚁死亡.具体算法如下。

算法 2. 增强的 ABSDA 算法。

输入:自组织服务推荐网络 SSRNM,查询请求 *keywords*

输出:节点 *Node*

```
{
  将 keywords 映射为 synsetIDs;
  选择一个服务消费者节点 consumer, 使得
    synsetIDs ∈ consumer.SSynsetIDs;
  创建查询蚂蚁 searchAnt;
  selectedNodes = consumer.selectMaxNodes();
  searchAnt.updateNodelist(selectedNodes);
  //查询蚂蚁根据节点的邻居列表更新蚂蚁的目标
  //节点列表
  int searchCount = 0;
  while(searchAnt.nodelist != null &&
    searchCount < n *  $\gamma$ )
  {
    nextHop = searchAnt.selectNextHop();
    //查询蚂蚁选择下一跳地址
    searchAnt.move(nextHop);
    //查询蚂蚁移动到下一跳
    if (nextHop.satisfy()) {
      //满足查询请求置查询成功标志
      searchAnt.updatePath();
      //查询蚂蚁更新路径上的服务信息度
      searchAnt 死亡;
      return(nextHop);
    } else {
      searchAnt.updateNodelist(nextHop.referrals());
      //更新蚂蚁目标节点列表
      searchAnt.removeNode(nextHop);
      //从目标节点列表移除当前节点
      searchAnt.reservePath(nextHop); //保存路径
    }
    searchCount ++;
  }
  searchAnt 死亡;
}
```

上述服务发现算法模拟蚂蚁寻路行为,具有良好的动态性.在搜索开始时,蚂蚁根据服务节点发布的服务信息度,查找目标节点.查找成功后又沿原路径返回并更新服务信息度,使以后的蚂蚁能够沿着该路径前进.与传统的蚁群算法不同,本研究在蚂蚁寻路的过程中,不仅根据本地的服务信息度决定下一跳的地址,还记录了其它经过的节点上的服务信息度.这样路径上的节点给出推荐,蚂蚁将这些信息素更新到目标节点列表中以便更好地进行查找.

3.4 算法特点

ABSDA 搜索方法在均衡网络负载、网络动态性及搜索效率等方面都得到了改进,总的来说具有以下几方面特点:

(1) 搜索过程朝着更加有利的方向行进.主要表现在:首先,使用邻居选择策略减少了搜索过程的盲目性.一个节点接收到搜索请求后,在邻居节点列表中进行搜索,如果存在请求的服务的 *synsetID* 信息,则返回一个到该邻居的推荐.否则,返回邻居列表中社交度最大的节点作为推荐.这样无论在邻居列表中是否存在请求的服务的 *synsetID* 信息,都会返回一个推荐,进行有针对性的搜索.其次,是使用推荐网络能充分利用“小世界”原理,通过推荐寻找短链.再次,蚂蚁在寻路时,只有当查询成功时才更新服务信息度,这样就更具有针对性.因为蚂蚁在查询过程中,只保留路径信息,而不对路径上的节点进行服务信息度更新.当蚂蚁查询成功时,沿原路径返回并对路径上节点的邻居列表的服务信息度进行更新.这样失败的查询蚂蚁不会影响其它蚂蚁寻路,而成功的查询蚂蚁却可以引导其它蚂蚁找到正确的服务.

(2) 提高了查找效率.主要表现在:首先,算法是完全分布的,不存在单点故障.其次,在服务发现过程中,蚂蚁可以充分利用访问过的节点的服务信息,而不是仅局限于某一节点,这样有利于查找效率的提高.因为蚂蚁本身携带一定的目标节点的信息,当蚂蚁到达某个节点时,该节点根据本地邻居列表的服务信息度使用邻居选择策略给出若干推荐,蚂蚁将这些推荐更新到目标节点列表中.

(3) 更加适合动态、开放的环境,且有利于 SSRNM 网络进化为稳定的拓扑结构.主要表现在:首先,算法是自组织的,不需要定义路径恢复等算法,从而能适应各种网络拓扑的改变.其次,服务信息度的挥发是周期进行的,避免某一路径上长期没

有蚂蚁经过时服务信息度得不到更新的情况,且避免出现无限增长的情况.再次,蚂蚁之间的交互是异步的和自治的,蚂蚁之间不是直接交流,而是通过服务信息度的作用来完成搜索过程.另外,服务发现过程中蚂蚁的行为不需要全局控制,蚂蚁对路径上节点邻居列表的服务信息度进行更新时,只考虑加强因素的影响.

(4) 搜索过程能实现负载均衡.算法是流量自适应的,蚂蚁可以根据网络的拥塞程度来决定下一跳的地址.如果某条路径发生拥塞,那么这条路径相应的信息素就会减少以达到流量调节的目的.

4 实验结果及分析

本研究设计了一个仿真实验来模拟整个服务发现过程,以验证本研究提出的服务发现方法的可行性和有效性.

4.1 实验设计

在本研究中,理论上可以使用任何关键字对服务进行描述.为了便于实验,本研究中的服务标记策略以 WordNet 为基础,从 WordNet 中截取了 10 个数据片段,构成关键字数据集.在进行服务标记时,随机选择一对关键字作为服务的关键字;而在进行服务发现时,随机选取一对关键字进行查询.

本实验中,首先初始化相关参数,然后构建 SSRNM 网络. SSRNM 构建完成后,执行查询请求,并记录每个请求的跳数.在实验中,参数设置如下:节点数目 n 为 10000,节点的初始邻居数为 4,每个搜索请求最大查询次数为 10000γ , γ 为调节参数,表示第一个服务节点发布服务信息时,包含该服务信息的节点的覆盖率,实验中设置为 5%.

实验过程如下:

(1) 无结构 P2P 网络初始化参数如下:生成 10000 个节点,其中 2.5% 指定为服务提供者节点,其余的是服务消费者节点.为每个节点随机指定 4 个邻居,服务提供者节点随机分布在 P2P 网络上.服务提供者节点随机选择一对关键字作为服务的标记;

(2) 执行 SSRNM 构建算法;

(3) 根据查询请求随机选择 SSRNM 网络上的服务消费者节点,并记录每个查询请求执行的跳数.生成 1000 次查询请求.

为了使实验结果更加客观,每种实验执行 5 次,取 5000 次查询结果的均值进行分析.

4.2 评价指标

本研究采用以下两个指标对 BSDA 和 ABSDA 方法进行评价:

(1) 成功率

搜索成功率是指节点搜索成功的次数与搜索请求的次数之比. 成功率是查准率的一种变形. 搜索成功率越高, 说明搜索算法越有效. 搜索成功率的高低取决于查询请求在网络中的跳数. 为了防止查询请求在网络上无限转发, 通常采用限制最大跳数或生存时间的办法来防止网络资源的浪费. 在本研究实验中, 最大查询跳数限制为 $n\gamma=500$. 其中, n 是网络节点数, γ 为调节参数. 成功率的计算公式为

$$\text{成功率} = \frac{\text{搜索成功的服务查询请求数}}{\text{发出的服务查询请求总数}} \quad (5)$$

由于本研究中的方法使用了推荐, 当查询跳数较大时说明正在搜索的服务的数量稀少或该种类服务的服务信息度分布不均匀, 因此较大的跳数有利于服务信息度的均匀分布.

(2) 查全率

本研究借鉴了文献[5]提出的 P2P 系统中查全率的计算方法, 定义查全率计算公式为

$$\text{查全率} = \frac{\text{搜索出符合搜索条件的服务数}}{\text{符合搜索条件的服务总数}} \quad (6)$$

4.3 实验结果分析

实验 1. 发现算法的有效性.

本实验是将本文方法与随机游走算法进行对比, 目的是为了验证本研究提出的服务发现算法的可行性并检验算法的效率. 对本研究提出的算法 1 和算法 2 这两个服务发现算法分别进行了 5 次实验, 并计算出平均查询准确率. 20.2% 是推荐网络构建完成后, 服务提供者节点及包含服务信息的服务消费者节点占 SSRNM 网络所有节点的比例.

图 1 给出了实验结果. 其中, X 轴是搜索的跳数, Y 轴是查找成功率, 每个点代表当前查找成功的准确率. 第 1 跳查找成功的概率是 20.2%, 这表明服务提供者节点数和包含服务信息的服务消费者节点数占 SSRNM 网络总节点数的 20.2%, 随机游走算法的第 1 跳搜索成功率就是来自这个数据结果. 从第 2 跳开始由于推荐的作用, BSDA 方法查找成功率高于随机游走算法 2.74 个百分点, 而 ABSDA 方法则要高出 7.74 个百分点. 此后虽然 3 种方法搜索成功率上升的都较快, 但 ABSDA 与 BSDA 方法始终都高于随机游走算法. 另外, 对于 BSDA 方法, 图 1 表现出搜索成功率的增幅具有一定的波动性,

这是由于 BSDA 方法不具有服务信息度挥发的策略, 网络动态性较差, 但搜索成功率也好于随机游走算法. 从第 7 跳开始, 搜索成功率趋于稳定. 最终, ABSDA 方法的搜索成功率稳定在 97.1%; BSDA 方法稳定在 95.2%; 而随机游走算法只有 89.53%. 由于 ABSDA 是对随机游走算法的改进, 因此 ABSDA 在查询过程中产生的信息量与随机游走算法相当, 但却可以有效地提高查询效率. 实验结果表明: 本研究提出的服务发现方法可以有效地提高搜索效率.

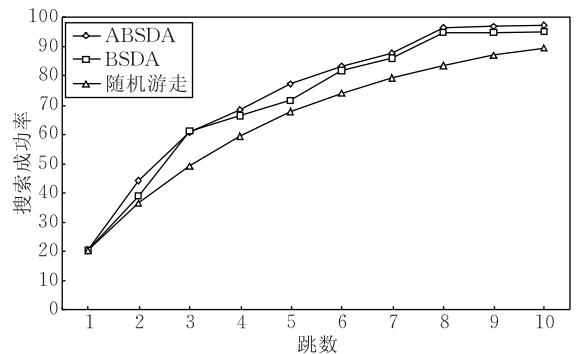


图 1 发现算法的有效性

实验 2. 搜索次数分析.

本实验是验证推荐网络的使用的确降低了搜索次数. 本实验中, 最大搜索次数是 500. 以表 1 中的第 5 次实验数据为例, ABSDA 与 BSDA 方法由于推荐的作用, 最大搜索次数为 37 和 35, 与随机游走算法的 48 相比分别减少了 22.9% 和 27.08%. 而 ABSDA 方法与 BSDA 方法由于都采用了推荐策略, 因此最大搜索次数相差不大.

表 1 最大搜索次数

	最大搜索次数		
	ABSDA	BSDA	随机游走
第 1 次实验	31	29	41
第 2 次实验	19	15	38
第 3 次实验	13	27	35
第 4 次实验	24	22	46
第 5 次实验	37	35	48

实验 3. 查找过程的稳定性.

本实验对初始化后的 SSRNM 使用 ABSDA 算法和随机游走算法进行搜索, 观察每次搜索的跳数随搜索次数变化而变化的关系. 为了使实验客观, 消除噪声数据, 本研究分别对两种算法进行 5 次实验, 并取其均值.

图 2 给出了 ABSDA 方法的搜索跳数与搜索次数的对应关系. 从图中可以看出, 在自组织服务推荐

网络构建完成初期,由于服务节点的服务信息分布不均匀,搜索成功的跳数不但较大而且波动幅度也很大.随着查询次数的不断增多,服务信息逐渐聚集到社交度较高的节点,因此查询跳数不断下降.网络稳定后,查询跳数基本稳定在 1~5 之间.图 3 给出了基于随机游走算法的搜索跳数与搜索次数的对应关系.从图中可以看出,随机游走算法的搜索效率并不会随着搜索次数的增加而提高,而是持续的在 1~15 跳的范围内波动.由此可以得出,ABSDA 算法可以适应动态变化的环境.当网络拓扑发生变化时,自组织服务推荐网络会通过搜索服务来适应拓扑变化,使变化后的网络趋于稳定以提高查找效率.该实验反映出自组织服务推荐网络的两个优点:(1) SSRNM 是自组织的.(2) SSRNM 能够适应动态、开放的环境.

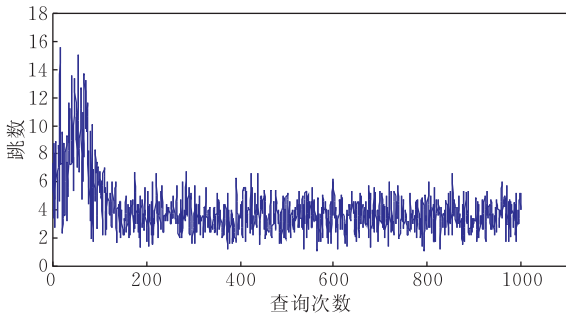


图 2 ABSDA 的查询次数与跳数对应关系

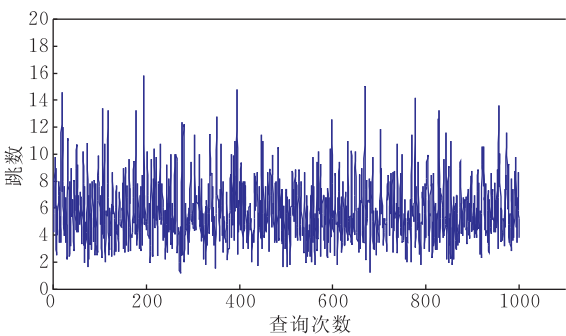


图 3 随机游走算法的查询次数与跳数对应关系

5 相关工作比较

目前有不少研究将推荐网络应用到服务的发现方面,但大部分基于信任网络的分布式协同推荐机制都忽略了服务所存储的网络的开放性和用户的多样性等问题^[6]. Pinar Yolum 等^[1]研究了能实现可信的服务选择的自组织推荐网络的拓扑结构,给出了推荐网络的设计准则.文献[2]和文献[7]分别给出了基于推荐的网络体系结构和匹配系统,但网络

的节点类型是单一的,未能体现服务发现所需要的节点类型多样化和语义化的特征. Yu Bin 等^[8]使用推荐搜索社会网络,并构建了原型系统协助用户搜索领域专家.潘静等^[6]提出一种基于声誉的服务推荐者发现方法.他们的研究重点在于对领域专家或服务推荐者等单个角色的搜索,没有涉及如何发现服务本身,而本研究是在服务的推荐网络建模时就综合考虑了服务提供商、服务开发者和服务本身的多角色问题,最终的目标是发现服务.另外,本研究也使用了这种推荐思想来进行服务的发现和搜索,而且还使用了语义信息对服务进行标记.本研究还使用了服务分类方法,以提高服务发现的查全率.

由于传统的 SOA 体系结构不能满足基于语义的服务发现的要求,许多研究者对其进行语义扩展. Paolucci 等^[9]对 UDDI 进行语义扩展,将基于 DAML-S 描述的 Web 服务的语义描述信息发布并存储到数据库中,通过 DAML-S/UDDI 转换器建立 DAML-S 服务和 UDDI 中存储服务的关系映射. Juan 等^[10]开发了一种 BeMatch 平台,基于行为匹配来对服务进行排序,评价服务和用户请求之间的语义距离.而本研究是通过 SSRNM 模型中的标记策略实现服务的语义标注,使用邻居选择策略和服务信息度更新等方法进行服务的发现.

有不少研究将蚁群算法应用于一些特殊网络的最优路径搜索上.例如, Laura Rosati 等^[4]提出了一种基于蚁群的启发、分布式的包路由算法 DAR 来实现在 Ad Hoc 网络中的数据图路由.崔艳荣等^[11]提出了一种基于蚁群优化的传感器网络中能量均衡数据查询协议 EBDQ.余建平等^[12]针对传感器网络能量资源极为有限的问题提出了带正反馈特征的分布式多蚁群实时查询处理算法 ARTQP.这些研究都是利用了蚁群算法的自组织和正反馈等特征,这点和本文的研究类似.但文献[4]没有考虑网络节点本身具有的语义特征;文献[11]侧重于考虑传感器节点的能量和通信链路的能量消耗方面,文献[12]重点在于提高查询处理算法的节能性、实时性及查询请求接受率,他们都没有考虑在服务发现过程中网络节点类型的多样性、节点可能动态退出或加入以及网络节点特有的语义属性等特点.本文研究主要是针对在动态开放的服务计算环境中如何更好地发现各类服务的问题,并考虑了服务本身的语义特征和服务存储环境的开放动态属性.

Gkantsidis Christos 等^[13]量化分析了在构建无结构 P2P 网络时随机游走算法的有效性,但并没有

讨论如何改进相关的搜索算法. Huang Ye 等^[14]使用基于蚁群的移动 Agent 给出了一种网格社区中的资源发现解决方案. 但他们的研究重点在于体系结构的建立上, 并没有给出具体的搜索算法. 王勇等^[15]针对成员服务提供的服务质量可能发生动态变化的情况和成员服务存在提供服务的上限阈值的情况, 采用蚁群系统设计了两种动态选择方法. 他们的研究在解决负载均衡问题和提高服务质量方面和本文工作类似, 但他们没有给出服务所存储的真实网络空间的描述, 也没有考虑服务本身具有的语义属性, 实验规模也偏小, 而本研究则给出了 SSRNM 模型, 对服务的存储空间、不同类型节点和服务的语义等信息都进行了描述. 李绍滋等^[16]提出了一个非结构化 P2P 共享原型系统, 该系统利用改进的蚁群算法进行路由搜索, 使信息搜索总是倾向于有利的方向, 并通过推荐减少盲目性. 这种推荐和搜索相结合的思想和本研究类似, 但他们的研究没有涉及如何在真实的网络环境中部署和如何均衡负载, 且每个网络节点的资源主要是文件, 也没有涉及文件资源的语义信息. 本研究针对的是互联网上的服务发现, 提出的方法综合了推荐网络和蚁群算法的优势, 给出了服务的搜索空间定义, 综合考虑了互联网环境下服务发现过程中的负载均衡和语义处理等问题.

6 小 结

针对现有服务发现和搜索方法研究中存在的效率不高、负载不均衡和语义欠缺等问题, 本文提出了一种基于推荐网络和蚁群算法的服务发现方法. 该方法给出了一种服务资源的共享组织模型, 并改进了现有服务发现方法的效率, 尤其是更加适合大规模动态开放的互联网环境, 能够被应用到相关搜索引擎中, 为实现自动、智能、高效的服务发现和组装提供保证.

首先, 建立了自组织服务推荐网络模型 SSRNM. 该模型使用邻居选择策略选择节点进行服务发布, 并使用服务标记策略对服务节点进行分类, 为服务发现提供了搜索空间.

其次, 将推荐和蚁群算法相结合, 提出了 ABSDA 方法. 该方法具有流量自适应、正反馈的特点, 可以有效地均衡网络负载, 并具有良好的查询效率, 适用于大规模开放、动态的环境, 能够有效地支持大规模分布式服务发现和搜索.

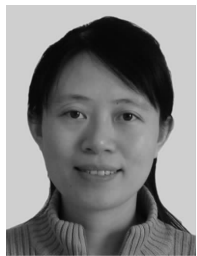
最后, 通过实验验证了自组织服务推荐网络 SSRNM 及基于该网络的服务发现方法的可行性和正确性. 实验结果表明, 本研究提出的服务发现方法可以有效地提高查询效率.

在未来的工作中, 尚需要在以下几方面进行更深入的研究: 采用更多领域和更大规模的服务来验证所提出的模型和方法. 另外, 要对服务进行可信评价, 为每个服务节点添加可信度, 实现可信的服务选择.

参 考 文 献

- [1] Pinar Y, Singh M P. Engineering self-organizing referral networks for trustworthy service selection. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 2005, 35(3): 396-407
- [2] Leonard N, Foner Y. Yenta: A multi-agent, referral-based matchmaking system//*Proceedings of the 1st International Conference on Autonomous Agents*. Marina del Rey, California, United States, 1997: 301-307
- [3] Xie Xiaoqin, Song Chaochen, Zhang Zhiqiang, Long Yu. A service search method based on referral network//*Proceedings of the International Conference on Internet Computing for Science and Engineering 2009*. Harbin, China, 2009
- [4] Laura R, Matteo B, Gianluca R. On ant routing algorithms in Ad Hoc networks with critical connectivity. *Ad Hoc Networks*, 2008, (6): 827-859
- [5] Ling Bo. Information retrieval based on peer-to-peer computing [Ph.D. dissertation]. Fudan University, Shanghai, 2004(in Chinese)
(凌波. 基于对等计算的信息检索技术[博士学位论文]. 复旦大学, 上海, 2004)
- [6] Pan Jing, Xu Feng, Lv Jian. Reputation-based recommender discovery approach for service selection. *Journal of Software*, 2010, 21(2): 388-400(in Chinese)
(潘静, 徐锋, 吕建. 面向可信服务选取的基于声誉的推荐者发现方法. *软件学报*, 2010, 21(2): 388-400)
- [7] Yathiraj B U, Singh M P. Information sharing among autonomous agents in referral networks//*Proceedings of the 6th International Workshop on Agents and Peer-to-Peer Computing (AP2PC)*. Honolulu, Hawaii, USA, 2007: 13-26
- [8] Yu Bin, Singh M P. Searching social networks//*Proceedings of the 2nd International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. Melbourne, Australia, 2003: 65-72
- [9] Paolucci M, Kawamura T, Payne T R. Importing the semantic Web in UDDI//*Proceedings of the Web Services, E-Business and Semantic Web Workshop*. Toronto, Canada. LNCS 2512. Springer-Verlag Berlin Heidelberg, 2002: 225-236

- [10] Juan C, Daniela G, Mokrane B. BeMatch: A platform for matchmaking service behavior models//Proceedings of the EDBT'08. Nantes, France, 2008: 695-699
- [11] Cui Yan-Rong, Li Ke-Qing. Data query protocol based on ant colony optimization for wireless sensor networks. *Journal of Software*, 2010, 21(4): 793-801(in Chinese)
(崔艳荣, 李克清. 传感器网络中基于蚁群优化的数据查询协议. *软件学报*, 2010, 21(4): 793-801)
- [12] Yu Jian-Ping, Lin Ya-Ping. Real-time query processing for sensor networks based on ant algorithm. *Journal of Software*, 2010, 21(3): 473-489(in Chinese)
(余建平, 林亚平. 传感器网络中基于蚁群算法的实时查询处理. *软件学报*, 2010, 21(3): 473-489)
- [13] Gkantsidis Christos, Mihail Milena, Saberi Amin. Random walks in Peer-to-Peer networks//Proceedings of the INFOCOM. Hong Kong, China, 2004: 7-11
- [14] Huang Ye, Bessis Nik, Kuonen Pierre et al. Using metadata snapshots for extending ant-based resource discovery service in Inter-cooperative grid communities//Proceedings of the 2009 1st International Conference on Evolving Internet. Cannes, French Riviera, France, 2009: 89-94
- [15] Wang Yong, Dai Gui-Ping, Hou Ya-Rong. Dynamic methods of trust-aware composite service selection. *Chinese Journal of Computers*, 2009, 32(8): 1668-1675(in Chinese)
(王勇, 代桂平, 侯亚荣. 信任感知的组合服务动态选择方法. *计算机学报*, 2009, 32(8): 1668-1675)
- [16] Li Shao-Zi, Cao Yang, Zhou Chang-Le. Research on key techniques about unstructured P2P information retrieval. *CAAL Transactions on Intelligent Systems*, 2006, 1(2): 60-87(in Chinese)
(李绍滋, 曹阳, 周昌乐. 基于非结构化的 P2P 信息检索关键技术研究. *智能系统学报*, 2006, 1(2): 60-87)



XIE Xiao-Qin, born in 1973, Ph.D., associate professor. Her current research interests include service-oriented computing, knowledge engineering, social network, intelligent information processing and so on.

SONG Chao-Chen, born in 1979, M. S. candidate. His current research interests focus on service-oriented computing.

ZHANG Zhi-Qiang, born in 1973, Ph. D., professor. His current research interests include database and intelligent information processing.

Background

This paper is supported by the National Natural Science Foundation of China under grant Nos. 60803037, 60803036, the National High Technology Research and Development Program(863 Program) of China under grant No. 2009AA01Z143, the Science and Technology Innovation Talents Special Fund of Harbin under grant No. RC2010QN010024, and the Fundamental Research Funds for the Central Universities under

grant Nos. HEUCFZ1010, HEUCF100602.

This project is to make a research on enabling technologies on service-oriented computing, including service discovery, service evaluation, service composition, service community and so on. In recent years, the authors published some results on these fields of research.