

一种基于语义的分布式云服务发现方法^{*}

王文丰^{1,2}, 韩龙哲^{1,2}, 李沛武^{1,2}, 李岚³, 刘天元²

- (1. 江西省水信息协同感知与智能处理重点实验室//南昌工程学院, 江西 南昌 330099;
2. 南昌工程学院信息工程学院, 江西 南昌 330099;
3. 南昌大学软件学院, 江西 南昌 330047)

摘 要: 随着云服务的迅猛发展, 传统集中式注册的云发现技术容易出现单点失效和性能瓶颈问题。因此, 提出了一种基于语义的分布式云服务发现方法。该方法采用多层云本体的方式表示云服务, 把服务描述映射到云本体结构中, 服务描述中的对等点根据它们的语义相似度组成集群。同时, 在高相似度的节点之间建立语义边, 从而使查询只需在相关的节点中进行, 以提高发现效率。实验结果表明: 本文提出的方法能够提高云服务发现效率和降低查询时间。

关键词: P2P 网络; 语义; 云服务; 本体

中图分类号: TP393 **文献标志码:** A **文章编号:** 0529-6579 (2019) 03-0145-08

A semantic decentralized cloud service discovery approach

WANG Wenfeng^{1,2}, HAN Longzhe^{1,2}, LI Peiwu^{1,2}, LI Lan³, LIU Tianyuan²

- (1. Jiangxi Province Key Lab of Information Cooperative Sensing and Intelligent Processing//
Nanchang Institute of Technology of China, Nanchang 330099, China;
2. School of Information Engineering, Nanchang Institute of Technology of China,
Nanchang 330099, China;
3. School of Software, Nanchang University of China, Nanchang 330047, China)

Abstract: With the rapid development of the cloud services conventional service discovery techniques using centralized registries stumble upon performance bottleneck and single point of failures. To address these issues, this paper proposes a semantic decentralized service discovery approach. This paper also introduces a multi-layered cloud ontology which closes the gap to represent cloud services in a standardized way. Here, service descriptions are mapped to a well-structured cloud ontology, which captures vital elements in different service description approaches. Peers are organized into clusters based on the semantic similarity of the service descriptions. To bolster the service discovery efficacy, semantic links are established between peers having higher similarity so that query forwarding is done only to relevant peers. Simulation results show that the proposed approach effectively enhances the query success rate and drops of the execution time.

Key words: P2P network; semantic; cloud service; ontology

云服务是面向云计算提供的一系列服务程序, 它可以为云计算用户解决其想要解决的问题。随着

云服务的普及, 云服务提供者便应运而生。云服务提供者能够让用户挑选出自己满意的云服务, 但云

* 收稿日期: 2018-06-11

基金项目: 国家自然科学基金(61402217)

作者简介: 王文丰(1983年生), 男; 研究方向: 分布式计算、分布式存储; E-mail: Wangwf@nit.edu.cn

服务的挑选过程却是相当费时和费力的^[1]。因云服务规范是异构和非标准化的,导致了语义互操作性问题。此外,用户对云服务的 QoS 保证要求更加严格。因此,云服务技术的难点在于如何发布最有效且具有低成本的服务,而又能保证最佳的服务质量。向云用户提供令人满意的服务,是高可扩展的服务发现技术的根本。对于云服务发现,一个难点就是服务消费者的定位服务,尤其是当该服务独立发布的时候^[2]。

在这种情况下,服务发现成功的关键是可用性、可访问性的有效服务注册。由于集中服务注册方法会存在单点失效、网络拥堵和性能瓶颈等问题,P2P 网络必须探索一种有效的服务发现机制^[3]。P2P 的容错性和可扩展性解决了云环境中服务发现问题;同时,随着云服务的发展,分布式服务注册也促进了 P2P 的可扩展性和容错性发展。在 P2P 系统中,研究人员已经对目前主流的服务发现技术进行了一定的研究^[4]。研究发现,在基于分布式的结构化 P2P 覆盖网络中发布和查询服务可以解决集中式注册的问题^[5],这是因为在结构化 P2P 覆盖网络中存在一个分布式哈希表,它能够支持关键字的存储和检索,如 put (key, value) 和 get (key) 操作。因此,结构化 P2P 覆盖网中的用户能够更加有效地发现服务,并且发现服务的时间也可以减少到 $O(\log n)$ 。然而结构化 P2P 覆盖网络也存在一个缺点,即它只支持精确匹配的查询,而且维护其拓扑结构也是需要相当大的开销。如对等点和服务经常会频繁地加入和退出,占据了系统很大的开销^[6]。另外,由于云服务描述没有任何标准格式,所以精确匹配的查询方式可能无法查询到相关的所有服务。与结构化 P2P 相比,非结构化 P2P 模型可以更好地模拟现实世界的问题。在非结构化 P2P 系统中,对等点加入时不需要事先了解它的拓扑结构^[5]。非结构化 P2P 系统的拓扑结构存在一定的随机性^[7],因此非结构化 P2P 系统的搜索方法不能采用 Hash 函数,但它可以采用灵活性的数据定位搜索方法并支持非精确搜索。泛洪技术及其变种^[8-10]就是非结构化 P2P 系统中一种代表性的搜索方法,它被广泛用于处理非结构化 P2P 系统中的消息路由。然而非结构化 P2P 系统也存在一个主要的缺点,即如果对等点之间的联系经常改变,在本地索引存储的统计数据就会变得无效。为了克服结构化和非结构化 P2P 系统中的缺点,语义覆盖网络便应运而生。在语义覆盖网络中,相似的对等点之间建立语义边,

以提高搜索精度,改善搜索结果^[11]。语义覆盖网络是一种语义相关对等点相连的虚拟网络,通过集成语义,可以提供更有效的多维查询^[12-15]。

语义覆盖网络 (SONs)^[11]提高了 P2P 系统的搜索质量,在语义覆盖网络中,把服务描述相似的对等点组织在一起,提供一组服务描述,以确保服务发现的成功。因此,在搜索时查询只需转发到相应的对等点即可,从而减少了通信开销,提高了成功率。本文旨在探讨云服务发现方法,通过探索云本体 (cloud ontology, CO) 来描述云服务,提高语义覆盖网络的服务发现成功率。同时,本文也研究了使用语义覆盖网络在非结构化 P2P 网络中高效转发查询机制。本文提出了基于语义覆盖网络的服务发现方法 (SD-CSR),该具有良好的可扩展性和适应性。由于它的分布式特性,所以在服务发现过程中,对等点只考虑其本地信息和它的邻居信息。并根据对等点之间的相似性,利用语义描述的服务来计算这种相似性。

1 基于语义的分布式云服务发现方法

一般地,云本体是利用网络中的所有对等点来存储和检索的云服务描述。云本体 CO 由云概念 $C = \{C_1, C_2, C_3, \dots, C_n\}$ 组成,其中包括 n 个概念和与这些概念有关的联系。服务注册根据不同的服务类别分散到各个区域中,每个区域保持本地本体的副本。用户查询被转发到最相关的区域,由能力表添加到每个区域。能力表中保存了服务类别的兴趣点。

1.1 对等点聚类

为了减少查询处理时间和网络流量,提出了聚类服务的概念,它产生于基于语义描述的每个区域^[11]。在本文中,用本体论的概念作为对等聚类的基础。具有语义描述服务的对等点,最初聚集到一些代表不同层次服务集的固定簇数 C_i ($1 \leq i \leq m$) 中。随后,每个区域都存在对等点的簇。一个服务描述 D 由以下部分组成:类别、服务描述属性、供应商信息和 QoS,可以表示为 $D = \{CC_i, SD_Attributes, Provider_info, QoS_facets\}$ 。其中, $CC_i \in Service_category_class$ 。每个集群管理节点中存在一个集群信息表 $\{Peer_id, current_status, last_update_info\}$,它表示集群注册的云服务,并构建集群内和集群之间的路由。集群管理节点还保存另外一个表,称为邻居集群表。它提供了另一个相关集群中对等点的信息,该信息与该集群中的对等点具有语义相似性。该表用一个三元组

$\{neighbor_cluster_manager_id, neighbor_peer_id, stauts\}$ 表示, 它保留了一组集群描述 (CD), CD 包括了簇标识符 C_i 、属性向量 A_i , 属于集群的对等点 $\{P\}$ 和集群 M , 例如: $CD_i = (C_i, A_i, \{P\}, M)$ 。而, M 中的对等点作为每个集群的质心。

1.2 语义覆盖网络的创建与维护

在集群中, 可以通过查找语义相关的对等点来降低查询成本, 并提高服务发现的质量。一般情况下, 其他集群的对等点可以提供相应的响应服务请求。由于其他集群的对等节点能够提供相应的响应服务请求, 因此在语义覆盖网络中, 可以很方便地查询所有相关的对等点, 并保证簇间的连通性, 从而实现每个对等点获得其邻居的信息。对等点保持两种与邻居相连的语义边: ① 内部节点的语义边: 连接属于同一个群集内的对等点; ② 外部节点的语义边: 连接属于不同集群的对等点, 实现语义可达节点之间的语义联系。特别地, 内部对等语义边涉及在同一集群上发布的一对类似服务。内部对等边定义为元组 $\{peer_id, semantic_similarity, status, time\}$ 。外部对等边定义为元组 $\{cluster_id, neighbor_peer, semantic_similarity, status, time\}$ 。这里, 链接的对等点被称为语义邻居。一个对等的

语义边连接着一个本地服务与对等的语义邻居上发布的服务。一个对等点 P_i 是对等点 P_j 面向服务描述 S_i 的语义邻居, 如果对等点 P_j 有一个服务描述 S_j , 则 S_i 和 S_j 通过对等点间的语义连接。在任意两个对等点间存在以下四种关系:

- (1) 精确: 两个对等点提供相同的功能。如, $sim(P_x, P_y) = 1$ 。
- (2) 延伸: 一个对等点提供额外的功能给其他对等点。如, $sim(P_x, P_y) \in (0, 1)$, 如果 S_1 提供了额外的功能给 S_2 , 则 S_1 延伸 S_2 。
- (3) 相交: 在对等点提供的功能之间存在一个非空的交集 (重叠)。如, $sim(P_x, P_y) \in (0, 1)$ 。
- (4) 失配: 没有共同之处。如, $sim(P_x, P_y) = 0$ 。每个区域都可以表示为一个元组 (C, P, LL, NL) , 其中 $C = \{c_1, \dots, c_n\}$ 是一组有限的集群, 每个集群是一组有限的自治节点 P 。 $LL \subseteq P \times P$ 是一组本地边的集合, 每条边 $(P_i, P_j) \in LL$ 表示同一集群内的对等点 P_i 和 P_j 之间存在直接连接。 $NL \subseteq P \times P$ 是一组邻居边的集合, 其中每条边 $(P_i, P_j) \in NL$ 表示不同集群内的对等点 P_i 和 P_j 之间存在间接连接。图 1 给出了一个区域的元素。

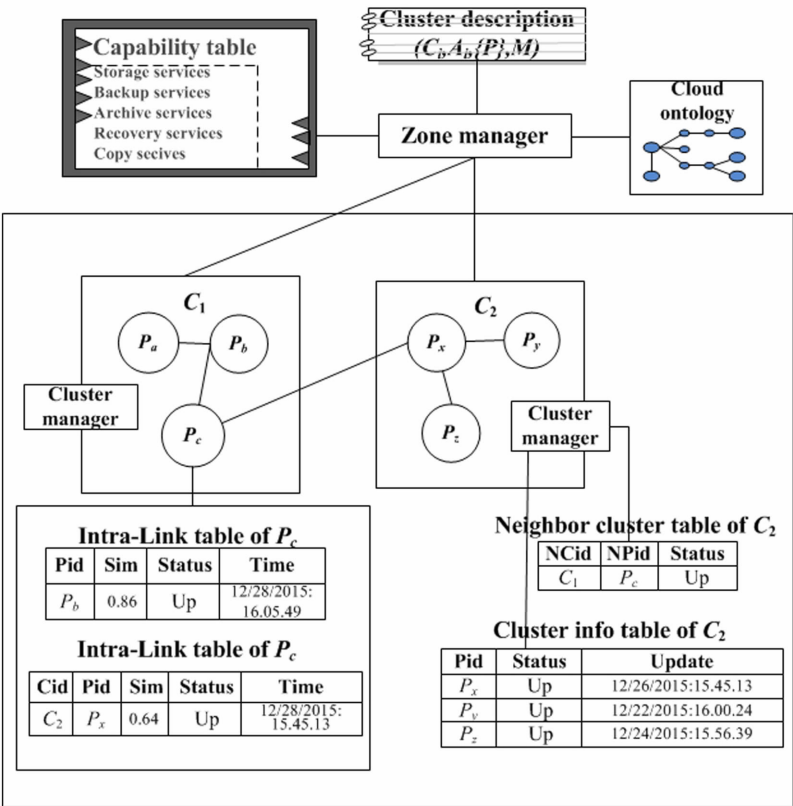


图 1 区域的元素

Fig. 1 Elements of zone

外部语义边根据探测请求的答复情况进行连接，如果对等点 P_x 发送到 P_y 一定数量的探测请求没有应答，则认为语义邻居 P_y 是不可达的。在这种情况下，移除连接 P_x 和 P_y 的外部边。发布/取

消服务就是采用类似的方式进行管理的。表 1 展示了对等点内部边的建立过程，表 2 则显示了对等点外部边的建立过程。

表 1 对等点内部边的发现算法
Table 1 Algorithm of intra-peer link discovery of peer

算法 1: Intra-peer link discovery of peer $P_x \in$ cluster C_x	
1	flood probe message request $m = \langle peer_id, service_desc, threshold_limit \rangle$ of peer $P_x \in C_x$ to $\forall$ peer $P_y \in C_x$
2	if $\exists P_y \in C_x \mid sim(P_y < service_desc >; m < service_desc >) \geq m < threshold_limit >$ then
3	add P_y as intra-peer link of P_x : $(P_y, sim_value, status_time)$
4	similarly, add P_x in the intra-peer link table of P_y

表 2 对等点外部边的发现算法
Table 2 Algorithm of inter-peer link discovery of peer

算法 2: Inter-peer link (semantic neighbor) discovery	
1	peer P_x of cluster C_x sends probe message request $m = \langle peer_id, cluster_id, service_desc, threshold_limit \rangle$ to zone manager ZM_i // $m \in$ 区域管理 ZM_j
2	ZM_i forwards m to M_i where M_i is the representative of cluster C_i , $\forall C_i \in Z_i$
3	if $sim(M_i < service_desc >, m < service_desc >) \geq m < threshold_limit >$ then
4	flood m to peer P_i , $\forall$ peer $P_i \in C_i$
5	if $sim(P_i < service_desc >, m < service_desc >) \geq m < threshold_limit >$ $\forall$ peer $P_i \in C_i$ then
6	send neighbor-link-creation, inter-peer-link $(C_i, P_i, sim_value, status, time)$ to P_x
7	add P_x in the inter-peer link table of P_i
8	update neighbor-cluster table if P_x is not already a semantic neighbor of C_i
9	update neighbor-cluster table of C_x if P_i is not already its semantic neighbor
10	endif
11	else
12	discard m
13	endif

在语义覆盖网络有效地创建后，主要关心如何利用他们在规定时间内提高搜索性能。获取满足用户请求的一组服务的过程称为服务发现。服务请求被发送到区域管理器能力表上。它是对发布在集群上的语义服务接口进行匹配和转发给潜在的集群 C_x 的服务描述进行查询。内部节点的语义边在对等点 $P_x \in C_x$ 上匹配服务描述的请求，用来修剪相关服务集和减轻服务比较的数量。外部对等语义边用于转发请求给远程相关的语义邻居对等点 P_x 。在这种方式中，只有相关的节点进行并行搜索，从而减少网络拥堵和寻找的服务时间。表 3 和表 4 分

别显示了对等点内部语义边和外部语义边的建立过程。

在表 5 的算法中，当发现满足请求的服务后才停止搜索，而请求转发到语义邻居实现相应的服务。重复这一过程直到：1) 没有其他对等点的请求可以被传播；2) 请求消息的生存时间（Time-To-Live）下降为零。区域管理器为每个服务请求分配一个唯一标识符和时间。为了避免重复请求的泛滥，每当对等点收到请求时，它会将服务标识符存储在指定的生存时间内。已被匹配的群集管理器在 TTL 时间周期内处于封闭状态，并将所有相关

服务的列表转发给区域管理器。每个区域管理器有一个 QoS 过滤器, 以更进一步过滤掉不相关的服务。当来自不同群集的一组服务到达区域管理器时, 它将根据 QoS 和信任程度对服务描述和用户请求进行第二级筛选。将不同的列表组合成一个列表, 其中包含降序排序的服务, 并将结果提交给用户。

表 3 对等点内部语义边的维护算法
Table 3 Algorithm of intra semantic link maintaince of peer

算法 3: Intra semantic link maintenance of P_x	
1	repeat with time period T
2	for $\forall$ peer $P_j \in$ intra-link table of P_x do
3	send $message_alive(P_x, current_time)$
4	if P_j not updated between $previous_message_alive_time$ and $current_time$ then
5	send $reply_alive(P_j, current_time)$
6	else
7	send $service_update(P_j, current_time)$
8	endif
9	if P_x not received ($reply_alive$ or $service_update$) from P_j for 3 consecutive probe request then
10	mark status of P_j as inactive in intra-peer link table of P_x
11	update the status of P_j as inactive in cluster manager node of C_j
12	endif

表 4 对等点外部语义边的维护算法
Table 4 Algorithm of inter semantic link maintaince of peer

算法 4: Inter semantic link maintenance of P_x	
1	repeat with time period T
2	for $\forall$ peer $P_i \in$ $semantic_neighbor_table$ of P_x do
3	send $message_alive(P_x, C_x, current_time)$
4	if P_i not updated between $previous_message_alive_time$ and $current_time$ then
5	send $reply_alive(P_i, C_i, current_time)$
6	else
7	send $service_update(P_j, C_i, current_time)$
8	endif
9	if P_x not received ($reply_alive$ or $service_update$) from P_i for 3 consecutive probe request then
10	mark status of P_i as inactive in intra-peer link table of P_x
11	update the status of P_i as inactive in $neighbor_cluster$ table of C_x
12	endif

表 5 服务发现算法

Table 5 Algorithm of discovery of services

算法 5: Discovery of services at cluster C_i

Input: service id S_{id} , service request S_r , threshold value θ , TTL T_0

Output: L-List of services with similarity value

```
1  L ← Φ
2  for ∀ peer  $P_i \in C_i$  do
3      if ( $S_{id} \cup P_{i\_cache}$  and  $T_0 > 0$ ) then
4           $T_0 \leftarrow T_0 - 1$ 
5          if  $\text{sim}(S_r, S_x) \geq \theta$ , where  $S_x$  is the service description in  $P_x$  then
6               $L \leftarrow L \cup \langle S_x, \text{sim}(S_r, S_x) \rangle$ 
7               $P_{i\_cache} \leftarrow P_{i\_cache} \cup S_{id}$ 
8              forward request ( $S_{id}, S_r, \theta, T_0$ ) to ∀ peer  $P_y \in$  intra-link table of  $P_i$ 
9              forward request ( $S_{id}, S_r, \theta, T_0$ ) to ∀ peer  $P_z \in$  inter-link table of  $P_i$ 
10         endif
11     endif
12 return L
```

2 实验结果与分析

本节通过实验分析 SD-CSR 方法的基本性能。所有的模拟实验在模拟器 Neurogrid^[16] 进行，该模拟器随机生成一个了 N 个对等点和 NS 个服务的 P2P 网络。服务 S_1 和 S_2 之间的匹配根据以下概率分布进行设置： $P(S_1 \text{ 精确 } S_2) = 5\%$ ， $P(S_1 \text{ 不匹配 } S_2) = 45\%$ ， $P(S_1 \text{ 扩展 } S_2) = 15\%$ ， $P(S_2 \text{ 扩展 } S_1) = 15\%$ 和 $P(S_1 \text{ 相交 } S_2) = 20\%$ 。为证明本文 SD-CSR 方法的有效性，我们与 Gnutella 协议^[17] 进行了实验对比。Gnutella 是一个用于分布式搜索和数字资源共享的协议，尽管支持传统的客户机/服务器架构，但 Gnutella 协议是点对点、非中心的模型。为了体现 Gnutella 协议的简单性，采用了一种盲目的搜索机制—洪泛。在该机制中，对等点在 TTL 时间半径内进行洪泛查询。

TTL 决定了一个查询可以被转发到网络的最大次数。TTL 值越高，响应请求的对等点的数量就越高。在模拟实验中，TTL 被设置为 1 和 15 之间。设 $T \in N^+$ 为 TTL 初始值。如果所有潜在的对等

点，提供相关的服务在 TTL 时间内不能被导航，结果就可能不完整。在具有 N 个对等点的非结构化 P2P 语义网络中，所有相关服务匹配的查询请求能够通过对等点集 P 在 T 内被导航。设 v 是被转发到对等点集 P 中一个对等点的请求概率 $|P| \leq T$ ，则在 T 内到所有相关节点的概率 (P_i) ^[18] 为：

$$P_r = \sum_{(j=|P|+1)}^T \binom{j-1}{|P|-1} \cdot V^{|P|} \cdot (1-v)^{j-|P|} + V|P|; \quad \text{otherwise, } P_r = 0 \quad (1)$$

在最坏情况下， $|P| = T$ ，且对于任意的 $p \in P$ ，都有一个相关的服务匹配查询，这将产生一个通用下界 $v(|P|)$ 。命中率是由分布式搜索策略检索到的服务数量的比率，如果所有的服务都在一个对等节点上可得。当命中率较高时，分布式搜索将更加成功。命中率取决于 TTL 值，命中率的结果如图 2 所示。通过设置 TTL 参数的不同值，可以改善命中率；当 TTL 增大时，命中率值会合理上升。从图 2 可以看出，即使在低的 TTL 值，SD-CSR 方法也要优于 Gnutella。这是因为与对等点的

语义联系有助于低数量的请求转发到潜在的对等点。

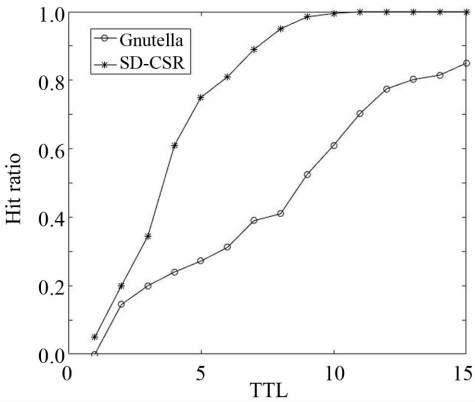


图 2 命中率与 TTL
Fig. 2 Hit ratio and TTL

召回是现有相关结果数和相关结果总数的比率。图 3 显示了一个召回与不同数量的服务返回时的情形。与 TTL 类似, SD-CSR 的召回率明显高于 Gnutella, 因为所有潜在的对等点都通过语义边聚集在同一个语义覆盖网络中。

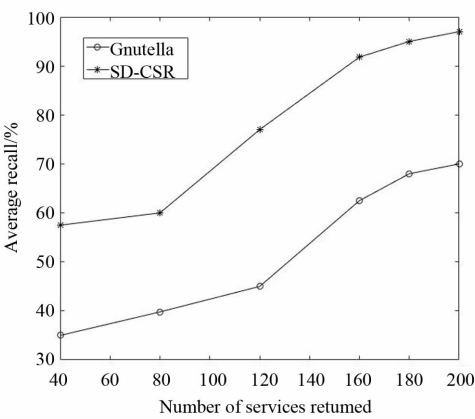


图 3 召回率分析
Fig. 3 Analysis of recall ratio

从图 4 可以看出, 召回率随着跳数的增加而上升。当 TTL 大于 7, 对 SD-CSR 系统的召回率均在 90% 以上, 满足要求。

执行时间是用户接收到请求服务并进行应答所需的时间。分析表明: SD-CSR 与 Gnutella 协议相比, 在执行时间上明显减少; 且, Gnutella 协议执行时间随着对等点的数量呈线性增长。这是因为 Gnutella 协议利用的是基本的洪泛算法, 执行时间较大。如图 5 所示, SD-CSR 的执行时间较小。这

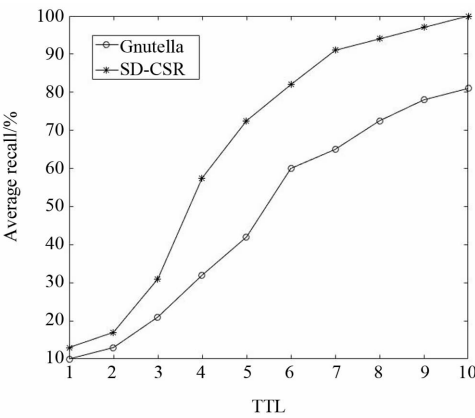


图 4 召回率与 TTL
Fig. 4 Recall ratio and TTL

是由于语义匹配技术大大减少了执行时间。SD-CSR 和 Gnutella 的效率, 如图 6 所示。因对等点基于语义链接被聚集到一组中, 这样服务匹配可以限制到相关组中, 所以 SD-CSR 可以减少大量的发现时间。

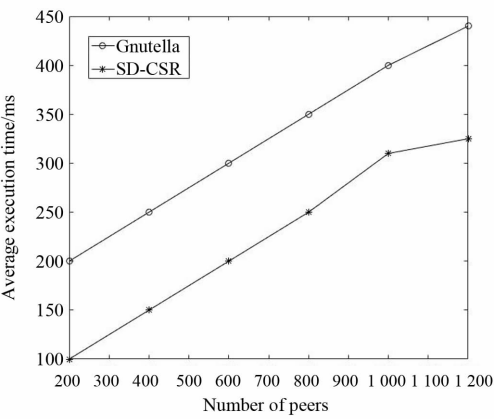


图 5 执行时间比较
Fig. 5 Comparison of execution time

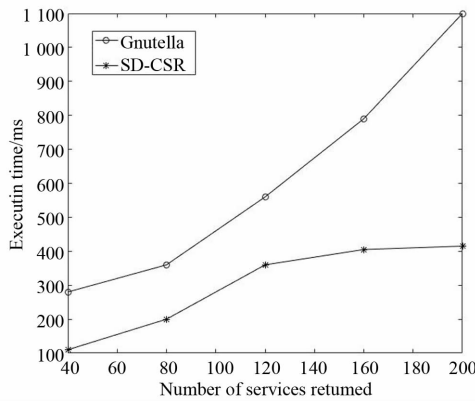


图 6 执行时间和返回服务的数量
Fig. 6 Execution time and number of sevice returned

3 结 论

如何在分布式和异构环境中发现云服务是解决云计算的主要挑战之一。本文采用语义覆盖网络实现了云服务发现的探索性研究,通过分布式服务注册表,把服务定制在基于语义概念和本体共享边的覆盖网络中,并在对等点间建立语义链接,由对等

点提供语义丰富的服务和对等点之间持有类似服务的链接,保证快速探索语义覆盖网络的目标和合法传播查询给合适的对等点。最后,利用基于本体的查询策略,以提高发现结果。实验结果表明:与 Gnutella 相比,无论是在发现过程的性能和效率方面,SD-CSR 都具有明显的优势。

参考文献:

- [1] BROCK M, GOSCINSKI A. Toward ease of discovery, selection and use of clusters within a cloud [C]//IEEE, International Conference on Cloud Computing. IEEE, 2010:289–296.
- [2] YASMINE M A, IBRAHIM F M, NAGWA L B, et al. Cloud services discovery and selection: survey and new semantic-based system [J]. Intelligent Systems Reference Library, 2014, 70:449–477.
- [3] HE Q, YAN J, YANG Y, et al. A decentralized service discovery approach on peer-to-peer networks [J]. IEEE Transactions on Services Computing, 2013, 6(1):64–75.
- [4] ARTURO C, HECTOR G M. Semantic overlay networks for P2P systems [J]. Lecture Notes in Computer Science, 2005, 3601:1–13.
- [5] NAVIMPOUR N J, MILANI F S. A comprehensive study of the resource discovery techniques in Peer-to-Peer networks [J]. Peer-to-Peer Networking and Applications, 2015, 8(3):474–492.
- [6] PARK H, YANG J, PARK J, et al. A survey on peer-to-peer overlay network schemes [C]//IEEE, International Conference on Advanced Communication Technology. IEEE, 2008:986–988.
- [7] PIRRO G, TALIA D, TRUNFIO P. A DHT-based semantic overlay network for service discovery [J]. Future Generation Computer Systems, 2012, 28(4):689–707.
- [8] THAMPI S M, CHANDRA S K. Survey of search and replication schemes in unstructured P2P networks [J]. Network Protocols & Algorithms, 2010, 2(1):93–131.
- [9] DORRIGIV R, ALEJANDRO L. Search algorithms for unstructured peer-to-peer networks [C]//IEEE Conference on Local Computer Networks. IEEE, 2007:343–352.
- [10] FLETCHER G H L, SHETH H A. Unstructured peer-to-peer networks: topological properties and search performance [C]//International Conference on Agents and Peer-To-Peer Computing. Springer-Verlag, 2004:14–27.
- [11] LV Q, CAO P, COHEN E, et al. Search and replication in unstructured peer-to-peer networks [J]. Acm SIGMETRICS Performance Evaluation Review, 2002, 30(1):258–259.
- [12] BISIGNANO M, MODICA G D, TOMARCHIO O. A semantic P2P overlay network for web service discovery [C]//World Conference on Services-i. IEEE, 2009:438–445.
- [13] FRAGOPOULOU P, MASTROIANNI C, MONTERO R, et al. Self-adaptive mechanisms for large scale distributed systems [J]. Grids, P2P and Services Computing. Springer US, 2010:147–156.
- [14] GHARZOULI M, BOUFAIDA M. A distributed P2P-based architecture for semantic web services discovery and composition [C]//International Conference on New Technologies of Distributed Systems. IEEE, 2010:315–320.
- [15] WANG X, LIU C, YANG Z. An efficient semantic web service discovery algorithm in DHT-based P2P network [C]//International Conference on Future Information Networks. IEEE, 2009:188–193.
- [16] 霍英, 陈志刚, 施宜. P2P 系统模拟器的分析与比较 [J]. 计算机应用研究, 2007, 24(11):37–39.
HUO Y, CHEN Z G, SHI Y. Analysis and contrast of P2P simulators [J]. Application Research of Computers, 2007, 24(11):37–39.
- [17] 王先兵, 张荣, 胡建光. Gnutella 协议的研究 [J]. 计算机工程, 2001, 27(11):56–57.
WANG X B, ZHANG R, HU J G. The study of Gnutella protocol [J]. Computer Engineering, 2001, 27(11):56–57.
- [18] CAO X, KAPAHNKE P, KLUSCH M. SPSC: efficient composition of semantic services in unstructured P2P networks [C]//European Semantic Web Conference on the Semantic Web. Latest Advances and New Domains. Springer-Verlag New York, Inc. 2015:455–470