

服务格:支持替代 Web 服务检索的服务组织结构

彭敦陆¹, 周傲英²

PENG Dun-lu¹, ZHOU Ao-ying²

1. 上海理工大学 计算机工程学院, 上海 200093

2. 复旦大学 计算机科学与工程系, 上海 200433

1. College of Computer Engineering, University of Shanghai for Sciences and Technology, Shanghai 200093, China

2. Department of Computer Science and Engineering, Fudan University, Shanghai 200433, China

PENG Dun-lu, ZHOU Ao-ying. Service Lattice: service organizational structure supporting retrieval of substitutive Web services. Computer Engineering and Applications, 2007, 43(26): 8-12.

Abstract: Based on deeply studying the substitutability of Web services, a novel structure, called Service Lattice, is proposed for representing the substitutability between Web services. Some fundamental concepts and algorithm which are used to fast retrieve the substitutive services w.r.t a given service on Service Lattice are also presented. The effectiveness of the proposed approach is verified by the experimental results.

Key words: Web services; Service Lattice; substitutive services; information retrieval

摘要:在对 Web 服务可替代性进行深入研究的基础上, 提出一种新型的、能够表达 Web 服务间可替代关系的组织结构——服务格, 并给出服务格上快速检索可替代服务的相关概念及算法。利用所提方法, 可以快速地检索出给定 Web 服务的最优可替代服务集, 通过实验验证了所提方法的有效性。

关键词: Web 服务; 服务格; 替代服务; 信息检索

文章编号: 1002-8331(2007)26-0008-05 文献标识码: A 中图分类号: TP311

1 引言

Web 服务(简称服务)可替代性是指服务间相互替代的能力。为避免服务请求者的应用由于调用的服务不可用(如网络故障、服务过期等)而发生异常, 可以用替代服务来代替发生故障的服务, 以保证服务请求者的应用正常运行。服务可替代性和替代服务的检索对保证基于 Web 服务的松耦合应用稳定性具有重要意义。

图 1 描述了一个可替代服务应用场景。图有 2 个服务请求者 r_1 和 r_2 , 3 个服务 us_1 、 us_2 和 us_3 。 r_1 和 r_2 分别调用了 us_1 的 `getTemperature` 和 `getAirForecast` 方法。在目前的服务实现技术中, 如果 us_1 不可用, 这个异常将传播到 r_1 和 r_2 , 它们的应用也将受到影响而不能正常工作。在图中, 假定具有相同名称的方法提供相同的功能。对服务请求者 r_1 来说, us_2 可以替代 us_1 , 因为 us_2 也提供方法 `getTemperature`; 同理, 对服务请求者 r_2 来说, us_3 是 us_1 的替代服务, 因为 us_3 也提供方法 `getAirForecast`。这样, 当 us_1 不可用而 us_2 、 us_3 可用时, 如果在 r_1 的应用中用 us_2 替换 us_1 , 在 r_2 中 us_3 替代 us_1 , 就可以保证 r_1 和 r_2 的应用不因 us_1 的不可用而受到影响。

实现服务替代过程的关键是高效的替代服务检索技术。目前, 服务提供者在 UDDI 注册中心注册服务时, 根据其商业目

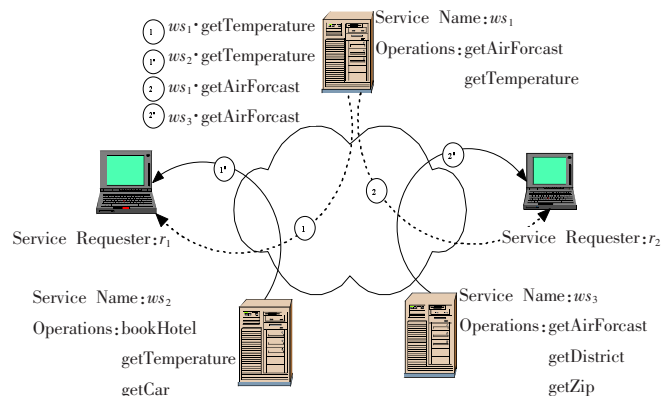


图 1 替代 Web 服务应用场景

的而不是服务的功能将服务进行分类^[1]。在调用服务前, 服务请求者在 UDDI 注册中心再按类型进行查找。这种基于目录分类查找的方法不但不能保证检索效率, 检索准确性也依赖于服务提供者和服务请求者对所调用服务相关知识的共同掌握程度。在某个服务发生异常时, 不能有效地从现有服务集中选择给定服务的替代服务。因此, 有必要研究出一个有效的服务组织方式, 这个组织方式可以反映服务间可替代关系, 并且利用它可以高效地检索替代服务。

目前,研究们从不同角度对服务的检索进行了研究。Wang 等人提出基于文档结构和接口匹配的方法来检索服务^[2];Dong 等通过对服务名及方法参数进行聚类实现了对 Web 服务的相似性查找^[3];DAML-S 对 Web 服务采用基于本体集的语义描述以实现 Web 服务的准确查找^[4]。这些研究都是针对一般性服务检索而进行的,对本文提出的可替代服务检索问题并不适用。

2 替代服务集

首先,给出有关替代服务的一些基本概念。

2.1 替代服务集

服务替代过程对服务请求者来说是透明的,服务管理系统应该具备能够自动检索替代服务能力。根据以前的研究,一个 Web 服务可以抽象地看作是一个能够提供多种功能的方法(operation)集^[5]。因此,服务间的替代关系可以通过方法集之间的包含程度来度量。一个服务要替代另一个服务,它必须能够提供另一个服务能提供的的一个或多个方法。因此,本文对替代服务的检索是在方法层(按各个方法)而不在服务层(将服务项目看作一个整体)上进行的。

检索替代服务时需要考虑以下两点:(1)替代服务与服务被请求的方法有关。本文的目标是在方法层上找到某个服务的替代服务,因此,不需要替代服务包含被替代服务的所有方法,而是部分(或全部)被服务请求者调用的方法;(2)一个 Web 服务,由于有多个服务请求者调用其方法,而调用的方法不一定相同,因此,对不同服务请求者来说,替代服务可以是不一样的。根据上述两点可知,一个服务的替代服务应该是一个服务集合,称之为替代服务集。下面给出替代服务集的形式化定义。

定义 1 (替代服务集) 设 $ws_i(oplist_i)$ 是 Web 服务集 D 中一个服务, r 为 ws_i 中被服务请求者调用的方法集,即 $r \subseteq oplist_i$ 。在 D 中所有满足 $r_{op} = r \cap oplist_j \neq \phi$ 的服务 $ws_j(oplist_j)$ 构成的集合 $S(ws_i, r_{op})$, 称为服务集 D 中 $ws_i(oplist_i)$ 关于方法集 r_{op} 的替代 Web 服务集。

定义 1 表明,如果 ws_j 是 ws_i 关于被调用方法集 r 的替代服务,那么, ws_j 可以提供 r 中的全部或部分方法,而这些方法正是服务请求者所需要的。

表 1 是本文用到的服务实例及其方法集。表 2 是服务 TempService 关于服务请求 r_1, r_2, r_3 和 r_4 的替代服务。为了区别同一服务请求关于不同方法的替代 Web 服务,用括号“{ }”和逗号“,”进行了分隔。如,表 2 的第 2 行第 3 列,即 TempService 关于 r_1 的替代 Web 服务集 $\{(ws_2, ws_4), \{ws_4, ws_3\}\}$ 表示: ws_2 和 ws_4 都支持方法 AF (getAirForecast), ws_4 和 ws_5 都支持方法 DT (getDistrict)。这样, $\{ws_2, ws_4\}$ 和 $\{ws_4, ws_3\}$ 的组合 $\{ws_2, ws_4\}, \{ws_4\}$ 和 $\{ws_4, ws_3\}$, 就是服务 TempService(ws_1) 关于方法集 r_1 的替代服务集。

由表 1 可知,服务 TempService(ws_1) 除被 r_1 调用的两个方法 AF 和 DT 之外,还有方法 TP (getTemperature) 和 ZP (getZip)。相对于服务请求 r_1 来说,TempService 的替代服务能提供后两个方法(TP 和 ZP)不是必要条件。

2.2 最优替代服务集

由定义 1,只要一个服务能够提供一个或一个以上的与被替代服务中被调用的相同方法,那么该服务就是被替代服务

表 1 Web 服务实例及其方法

服务标识	服务名	方法名(缩写)
ws_1	TempService	getAirForecast (AF), getDistrict (DT), getTemperature (TP), getZip (ZP)
ws_2	TravelService	getAirForecast, BookTickets (BT)
ws_3	SkatingService	getTemperature, OrderEquipment (OE)
ws_4	SportService	getAirForecast, getDistrict
ws_5	PublicService	getDistrict, getHospital (HP)
ws_6	TrainService	getTrain (TA), getHospital

注:方法名后面的大写字母为对应方法的缩写

表 2 服务 ws_1 的 4 个替代服务集

服务请求	调用的方法	替代 Web 服务集	最优替代 Web 服务集
r_1	AF, DT	$\{(ws_2, ws_4), (ws_4, ws_3)\}$	$\{ws_4\}$
r_2	AF, DT, TP	$\{(ws_2, ws_4), (ws_4, ws_3), (ws_3)\}$	$\{ws_4, ws_3\}$
r_3	DT	$\{(ws_4, ws_3)\}$	$\{ws_4\}$ 或 $\{ws_3\}$
r_4	ZP, TP	$\{Null, (ws_3)\}$	$\{Null, (ws_3)\}$

关于这些相同方法的替代服务。这个定义是松散的,这种松散性使得最后得到的替代服务集很大。例如,表 1 显示服务 TempService 关于服务请求 r_2 的替代服务集为 $\{(ws_2, ws_4), (ws_4, ws_3), (ws_3), null\}$ 。不难发现, ws_4 提供了方法 AF 和 DT, ws_2 和 ws_3 也分别提供这两个方法中的一个。如果把这个结果集直接发送给服务请求者,服务请求者会根据这些替代服务集的组合 $\{ws_2, ws_4, ws_3, null\}, \{ws_2, ws_5, ws_3, null\}, \{ws_4, ws_3, null\}$ 和 $\{ws_4, ws_5, ws_3, null\}$ 中一个来替换调用的服务 TempService。在这些组合中,除 $\{ws_4, ws_3, null\}$ 只包含两个 Web 服务(ws_4, ws_3)之外,其它组合都包含有 3 个 Web 服务。在请求者的应用程序中请求的服务数越少,复杂度就越低,稳定性就越高。因此,有必要对结果集中的替代服务进行筛选,以选取最优的替代服务集。

定义 2 (最优替代服务集) 设 $S(ws_i, r_{op})$ 是服务集 D 中某个服务 ws_i 关于被请求方法集 r 的替代服务集,如果不存在 $S(ws_i, r_{op})$ 的子集 $S'(ws_i, r_{op})$ 可以提供与 $S(ws_i, r_{op})$ 相同数目的 r 中的方法,那么 $S(ws_i, r_{op})$ 就是 ws_i 在服务集 D 中关于被请求方法集 r 的一个最优替代服务集,记为 $OS(ws_i, r)$ 。

定义 2 表明,一个 Web 服务关于某个特定请求的最优替代服务集也是能够满足服务请求的最小服务集。在求取这个最小替代服务集之前,根据替代服务相对被请求方法的重要程度,将替代服务分为下列 4 类:

第 1 类为必选服务(A_1)。这些服务是在指定方法集上唯一能够替换被替代服务的服务。例如,在表 2 中, ws_1 关于服务请求 r_1 的替代服务集 $\{(ws_2, ws_4), (ws_4, ws_3), (ws_3), null\}$ 中,只有服务 ws_3 提供方法 TP。因此,服务 ws_3 是 ws_1 关于服务请求 r_1 的必选服务;第 2 类为优先服务(A_2),指相比其它服务能够提供较多的被调用方法的替代服务。例如,表 2 中, ws_1 关于服务请求 r_1 的替代服务集 $\{(ws_2, ws_4), (ws_4, ws_3), (ws_3), null\}$ 中, ws_4 包含两个被调用的方法,而其它只有 1 个。因此, ws_4 是 ws_1 关于服务请求 r_1 的优先服务;第 3 类为任选服务(A_3),指那些能够向服务请求者提供相同被调用方法的替代服务。例如,表 2 中, ws_1 关于服务请求 r_3 的替代服务集中有两服务 ws_4 和 ws_5 ,它们都向 r_3 提供方法 DT,因此,它们是任选服务,即在生成最优替代服务集中,可以任选其中之一;第 4 类为可舍弃服务(A_4),除上述 3 类服务之外的服务。一般来说,可舍弃服务所能提供的方法在上述 3 类服务中已经提供了。

图 2 用图示方式说明上述替代服务的分类。该图描述了某个服务 ws 关于服务请求 $r = \{op_1, op_2, op_3, op_4\}$ 的替代服务集。每

个结点(圆圈)表示 ws 中一个被 r 调用的方法,每个边的标签表示提供该边箭头所指向方法的替代服务。根据上述分类可知,只有 ws_6 唯一提供方法 op_4 ,所以是必选服务; ws_1 因为提供了两个方法,相对其它服务来说提供了较多的方法,所以为优先服务; ws_4 和 ws_5 都提供而且都仅提供 op_3 ,因此为任选服务。当选择了替代服务集(图中实线中的服务)后,余下的替代服务就是可舍弃服务(ws_2 和 ws_3)。综上,在求取到某个服务的替代服务集后,按 $A_1 \sim A_4$ 的顺序可选取该服务的最优替代服务集。

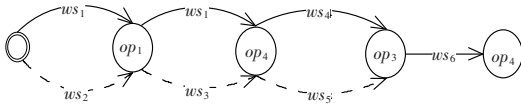


图2 最优替代 Web 服务示意图

3 服务格

在一个服务集中,服务的有关信息通常是以半结构化(如 WSDL 文档)方式进行存放的。如果在检索替代服务时临时对所有服务的有关信息进行比较,检索效率无法得到保证。因此,需要事先将服务按照合理的结构进行组织,以方便替代服务的检索。这里,提出一种新型的服务关系表达结构——服务格(Service Lattice)。

服务格是一种特殊的概念格,它是在指定 Web 服务背景下,通过形式概念分析(Formal Concept Analysis, FCA)得到各个服务间相互关系的描述^[6]。一个服务格与一个服务集相对应,服务格的属性集是由服务集中所有服务所提供的方法组成,对象集是由服务集中所有服务组成。下面,对服务格的一些基本概念进行论述。

定义 3 (可替代服务结点) 给定 Web 服务集 $D, ws_i(oplist_i) \in D$ 为一个 Web 服务。与 D 对应的服务格为 $W \sqsubseteq (\mathcal{V} S, A \circ R)$, 其中 $W = \bigcup_{i=1}^{|W|} ws_i, A \circ = \bigcup_{i=1}^{|W|} oplist_i$, 分别表示服务格 $W \sqsubseteq$ 的对象集和属性集, R 为 $W \in$ 中的服务对 $A \in$ 中的方法提供关系。 $r \subseteq oplist_i$ 是服务 ws_i 的一个请求, $WsC_j(W_j, A_j)$ 是 $W \sqsubseteq (\mathcal{V} S, A \circ R)$ 的一个结点, $W_j \subseteq W \in$ 是 WsC_j 的外延, 记为 $ex(WsC_j) = W_j, A_j \subseteq A \circ$ 为 WsC_j 的内涵, 记为 $in(WsC_j) = A_j$, 且 W_j 中任一服务都能提供 A_j 中所有方法。如果满足 $A_j \cap r \neq \phi$, 则称 WsC_j 是 ws_i 的一个可替代服务结点。

定义 4 (完全可替代服务结点) 给定一个 Web 服务集 $D, ws_i \in D$ 为原子服务。与 D 对应的服务格为 $W \sqsubseteq (\mathcal{V} S, A \circ R)$, $WsC_j(W_j, A_j)$ 是 $W \sqsubseteq (\mathcal{V} S, A \circ R)$ 的一个可替代服务结点。如果满

足 $in(WsC_j) = A_j = oplist_i$, 称 WsC_j 是服务格 $W \sqsubseteq (\mathcal{V} S, A \circ R)$ 中服务 ws_i 的完全可替代服务结点, 记为 $W \sqsubseteq \dots ComSub(ws_i)$ 。

定义 4 表明, 在服务格 $W \sqsubseteq$ 中, 关于某个服务 ws_i 的完全可替代服务结点 WsC_j 的内涵与 ws_i 的方法集相同, 其外延至少中包含服务 ws_i 。如果还包含有其它服务, 则该服务一定在 ws_i 的最优替代服务集中。

定义 5 (关于服务请求的完全可替代服务结点) 设 $r \subseteq oplist_i$ 是服务 $ws_i(oplist_i)$ 一个服务请求, 若服务格 $W \sqsubseteq (\mathcal{V} S, A \circ R)$ 中的某个结点 $WsC_j(W_j, A_j)$ 满足 $r \subseteq in(WsC_j) = A_j$, 则称 WsC_j 是 ws_i 关于服务请求 r_j 的完全可替代服务结点, 记为 $W \sqsubseteq \dots ComSub(ws_i, r)$ 。

例如, 在本文所用的示例服务集所对应的服务格 $W \sqsubseteq$ 中, $in(WsC_{11}) = \{AF, DT, TP, ZP\}$ 和 ws_1 的方法集相同, 所以, WsC_{11} 是在 $W \sqsubseteq$ 中服务 $ws_1(AF, DT, TP, ZP)$ 的完全可替代服务结点, 即 $W \sqsubseteq \dots ComRep(ws_1) = WsC_{11}$ 。又如, $r_1 = \{TP\}$ 是服务 ws_1 的一个请求, 结点 WsC_3 的内涵与 r_1 相等, 因此, WsC_3 是服务 ws_1 关于请求 r_1 的完全可替代服务结点, 即 $W \sqsubseteq \dots ComSub(ws_1, r_1) = WsC_{11}$ 。

4 替代服务检索

在给出服务格上替代服务检索算法之前, 先研究服务格上最优替代服务集(定义 2)的选取方法。考虑上章对替代服务分类, 结合服务格中替代服务结点的特征, 可以这样来确定替代服务结点中的服务类别: (1) 如果某个结点的外延中, 除被替代 Web 服务之外, 只剩 1 个服务, 那么这个就是必选服务; (2) 如果某个结点的外延中, 除被替代 Web 服务之外的服务个数大于 1, 那么这些服务就是任选服务; (3) 所有任选服务中, 出现频率较高的服务为优先服务; (4) 所有除 (1)、(2) 和 (3) 之外的其它可替代服务结点外延中的服务均为可舍弃服务。

下面以图 3 来说明通过服务格来检索最优级替代服务的过程。假设服务 ws_1 的一个服务请求 $r = \{AF, DT, TP\}$, 不难发现, ws_1 关于服务请求 r 的替代服务只可能出现在结点 WsC_2, WsC_3, WsC_4 及 WsC_6 的外延中。从左边的表格可得 $ex(WsC_3) = \{ws_1, ws_3\}$, 而除 ws_1 外, 只有 ws_3 能提供方法 TP , 因此, ws_3 为必选服务。同理, $ex(WsC_6) = \{ws_1, ws_4\}$, 因此, ws_4 也是必选服务。而 $ex(WsC_2) = \{ws_1, ws_2, ws_4\}$ 中的 ws_2 和 ws_4 及 $ex(WsC_4) = \{ws_1, ws_4, ws_5\}$ 中 ws_4 和 ws_5 为任选服务, 但 ws_4 已为必选服务, 所以 ws_2 和 ws_5 为舍弃服务。这样, 在图 3 的服务格中 ws_1 关于服务请求 r 的最优替代服务集为 $\{ws_2, ws_4\}$ 。

结点 ID	外延 ex	内涵 in
WsC_1	$ws_1, ws_2, ws_3, ws_4, ws_5, ws_6$	ϕ
WsC_2	ws_1, ws_2, ws_4	AF
WsC_3	ws_1, ws_3	TP
WsC_4	ws_1, ws_4, ws_5	DT
WsC_5	ws_5, ws_6	HP
WsC_6	ws_1, ws_4	AF, DT
WsC_7	ws_2	AF, BT
WsC_8	ws_3	OE, TP
WsC_9	ws_5	DT, HP
WsC_{10}	ws_6	HP, TA
WsC_{11}	ws_1	AF, DT, TP, ZP
WsC_{12}	ϕ	AF, BT, DT, HP, OE, TA, TP, ZP

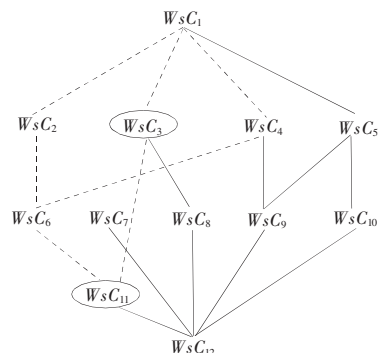


图3 服务格实例

在图 3 中, WsC_2, WsC_3, WsC_4 及 WsC_6 都在经过关于服务请求 $r=\{AF, DT, TP\}$ 的完全可替代服务结点 WsC_{11} 的某条路径上。进一步考察,发现在服务格中,一个服务的替代服务结点必定在经过这个服务的完全可替代服务结点的路径上。因此,在检索替代服务时,不需要遍历整个服务格而只需搜索满足上述条件的路径。利用这一思想,结合第 2 章的基本概念及上述服务格中最优替代服务集的选取方法,下面给出服务格上替代服务检索算法 sub_Retrieving。

算法 sub_Retrieving 从服务格的顶结点开始,以深度优先方式进行查找服务 ws 的最优替代服务集。算法的输入是服务格、被替代 Web 服务及其被调用方法集。算法分 2 步来求得最优替代服务集:第 1 步通过找到替代服务集 S_{rep} ;第 2 步是在第 1 步产生的替代服务集基础上,查找最优替代服务集 S_{optrep}, S_{rep} 和 S_{optrep} 可以用哈希表的方式进行维护。

算法:sub_Retrieving

输入: $W \subseteq L$ 服务格, ws 被替代服务, r 被调用服务方法集;

输出: ws 关于 r 的最优替代服务集 S_{optrep}

算法描述: $S_{rep} \leftarrow \phi; S_{optrep} \leftarrow \phi;$

对于 $W \subseteq L$ 中的每个未被访问的结点 WsC_i

如果 $in(WsC_i) \cap r \neq \phi$ 且 $r \cap in(WsC_i) \neq r$, 那么

$S_{rep} \leftarrow S_{rep} \cup \{ex(WsC_i) - ws\};$

搜索 WsC_i 未被访问过的后代结点 WsC_j , 直到 $r = in(WsC_j) \cap oplist$ 或 $in(WsC_j) \cap oplist = in(WsC_j) \cap oplist$, 则停止访问其它后代结点;

如果 r 中方法能够被 S_{rep} 的服务所提供, 则停止访问其它结点, 否则返回上层结点;

对于 S_{rep} 中的每个服务 ws_i , 如果只有 ws_i 提供 $oplist$ 中的某一方法 op , 那么 $S_{optrep} \leftarrow S_{optrep} \cup ws_i$;

如果 ws_i 相对其它的服务, 能提供更多与 $oplist$ 相同的方法, 则 $S_{optrep} \leftarrow S_{optrep} \cup ws_i$;

如果除 ws_i 外, 还有其它服务(如 ws_j)能提供与 ws_i 相同的方法, 且这些方法在 S_{optrep} 中尚没有其它服务提供, 可以从 ws_i 和 ws_j 中任选一个(如 ws_i)加入结果集, 即 $S_{optrep} \leftarrow S_{optrep} \cup ws_i$;

如果 $oplist$ 中所有方法均被 S_{optrep} 中的服务支持或 S_{rep} 中所有服务都被考察过, 那么返回 S_{optrep} 。

查找替代服务集 S_{rep} 的主要思想如下: 如果一个未被访问过的结点 WsC_i 的内涵与集合 $oplist$ 的交 sop_i 不为空, 那么它的外延 $ex(WsC_i) - \{ws\}$ 中的服务, 就是 ws 关于方法集 sop_i 的替代服务; 如果 sop_i 为空, 那么停止搜索这个结点及其后代结点。这样, 可以保证只搜索经过 ws 的完全可替代服务结点的路径。如果结点 WsC_i 和后代结点 WsC_j 的内涵与被调用方法集的交相等, 那么也停止搜索 WsC_i 的后代结点, 并返回上一层, 继续搜索没有被访问的结点。当 r 中所有方法都能被结果集中的服务所提供时, 停止遍历服务格, 进入最优替代服务的选取。选取最优

替代服务集 S_{optrep} 的主要思想是: 首先, 从 S_{rep} 中选取必选服务, 然后选取优先服务, 最后选取可选服务。

分析上述算法, 在搜索服务格时, 每个结点最多遍历一次, 如果某个结点被标识成被访问过, 就不再对其进行搜索。因此, 搜索服务格的过程实质上是对每个结点查找其邻接点(孩子结点)的过程。其复杂度取决于被替代服务请求集 r 含有方法的数目及所采用的存储策略。如采用二维数组表示邻接矩阵作服务格的存储结构时, 查找每个结点的邻拉点所需时间为 $O(n^2)$, 其中 n 为服务格中结点数。求取最优替代服务的复杂度主要与服务格中可替代结点数及被请求方法数有关。不难分析, 这一过程复杂度为 $O(l * m)$, 其中 l 为含替代服务的结点数, m 被调用的服务方法数。相对 n 来说, l 和 m 要小很多, 故算法 sub_Retrieving 的复杂度为 $O(n^2)$ 。

5 性能评估

在服务格中的替代服务都能被检索到, 因此, 对服务格检索替代服务的准确度是可以得到保证。在此, 通过两组实验对基于服务格的替代服务检索技术进行性能评估。一组用来测试基于服务格的替代服务检索方法的检索效率; 另一组用来测试基于服务格的替代服务检索算法的伸缩性。

实验中的所有程序都是采用 Java (JDK1.4) 编写的。对 WSDL 文档的解析采用的是 IBM 的 WSDL4J 工具包。实验数据集有两种: 一种是从主要 Web 服务注册商 (XMethod、Microsoft 及 Sun) 及通过 Web 搜索引擎 (如 Goolge、Yahoo) 搜索得到的真实服务资料库 (WSDL 文档), 共 2 000 个 WSDL 文档, 4 776 个服务方法; 另一种通过自己开发的 WSDL 文档生成器创建的模拟 Web 服务资料库, 共 2 000 个 WSDL 文档, 4 013 个服务方法。

第一组实验是将本文所提检索方法与基于关键字的检索方法在检索响应时间上进行对比来说明所提方法的检索性能。在这里, 响应时间指的是向系统发出检索请求(某个服务被调用的方法集), 到检索返回最优替代服务集时结束。实验中, 两个服务数从 1 000 变化到 2 000, 每次有 200 个服务加入到服务集中。由于基于服务格的检索方法与关键字检索方法在检索响应时间上差别很大, 因此, 在实验结果图(图 4)中, 对响应时间取了常用对数。从该图不难看出, 无论用真实服务集, 还是在模拟服务集, 基于服务格的替代服务检索效率比关键字检索效率要高几个数量级。这是因为在服务格检索时, 只需查找那么经过被替代服务的完全可替代服务结点的路径中的结点, 而不需检索整个服务格。在关键字检索时, 需要访问所有 Web 服务的信息。这就保证了服务格的检索时具有较高的检索效率。

第二组实验是测试所提方法的伸缩性。与第一组实验类

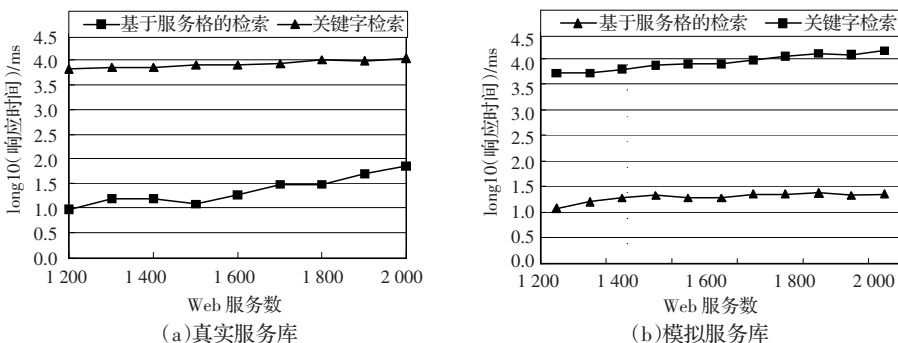


图 4 替代服务检索性能比较

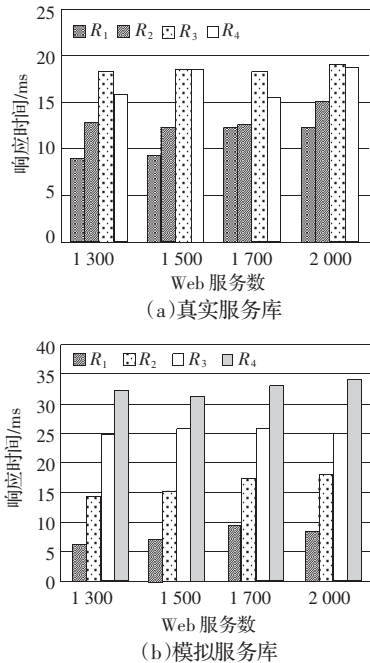


图5 替代服务检索伸缩性测试

似,在真实服务集和模拟服务集上都进行了实验,实验结果见图5。实验测试了在不同服务集的大小的情况下,检索给定服务的最优替代服务集的响应时间变化。在两个数据集中,服务数量都从1300变化到2000。被替代服务中被调用的方法数变化范围为1~4,图中的 R_1, R_2, R_3 和 R_4 ,下标表示有几个方法被调用。

从图中发现,尽管服务数量不断增加,同一个检索的响应时间变化并不大。例如,在真实服务集中,对于检索 R_1 当服务集大小为分别为1300,1500,1700和2000时,检索响应时间分别约为9ms,9ms,12ms和13ms;在模拟服务集中,对于检索 R_1 ,当服务集大小分别为1300,1500,1700和2000时,检索响应时间分别约为6ms,7ms,9ms,9ms。当被调用方法数发生变化时,对于同一检索,在不同大小的服务集中检索响应时间变化也不大。这是因为,尽管Web服务数增加了,替代服务总不一定增加,在服务格中,可替代服务结点也不一定增加。因此,在检索替代服务时,被访问的结点增加的也很少,对检索响应时间影响也不大。由此可见,基于服务格的替代服务检索方法具有好的伸缩性。

(上接7页)

线网络的资源,采取自适应传输调度方法可以更加有效地提高无线系统的整体传输效率。同时,在无线网络资源规模一定的情况下,为提高访问成功率,GIS空间数据的优化也是个关键问题。(收稿日期:2007年6月)

参考文献:

- [1] Meng Ling-kui, Lin Cheng-da, Shi Wen-zhong. Spatial data transfers and storage in distributed wireless GIS[C]//20th ISPRS Congress, Istanbul, 2004: 305-309.
- [2] 林承达,孟令奎,彭溢.无线GIS空间数据传输技术研究[J].计算机与数字工程,2005,33(12):11-15.
- [3] 赵文斌,张登荣.移动计算环境中的地理信息系统[J].地理与地理信息科学,2003,19(2):19-23.
- [4] 孟令奎,史文中,张鹏林.网络地理信息系统原理与技术[M].北京:

6 结论

本文提出一种支持可替代Web服务检索的组织结构——服务格。针对服务格,给出了一些重要概念,包括Web服务可替代结点,可替代服务集及最优可替代服务集等。在此基础上,给出在服务格上进行替代Web服务检索算法sub_Retrieving。通过实验验证了基于服务格的替代Web服务检索有较好检索性能。

本文提出的替代服务检索方法对保持应用程序的不间断运行提供了解决途径。同时,该方法也可用于检索组合服务中的替代子服务。这项技术也可以看作是BPEL4WS^[7]的一个补充。通过服务格将各个服务的可替代服务检索到,并将相关信息与BPEL4WS相结合,可以在很大程度上提高服务系统的可靠性。本文尚未考虑在动态服务集支持替代服务检索的服务格,今后将在这方面做进一步研究。

(收稿日期:2007年1月)

参考文献:

- [1] OASIS.UDDI Version 3.0.2[EB/OL].[2004-10-19].<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
- [2] Wang Y, Stroulia E. Flexible interface matching for Web-service discovery[C]//Proceedings of Fourth International Conference on Web Information Systems Engineering, Roma, Italy, December 10-12 2003.
- [3] Dong X, Halevy A, Madhavan J, et al. Similarity search for Web services[C]//Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.
- [4] Coalition D S. DAML-S: Web service description for the semantic Web[C]//Proceedings of the 1st International Semantic Web Conference (ISWC'02), Sardinia, Italy, 2002.
- [5] Peng D, Huang S, Wang X, et al. Management and retrieval of Web services based on formal concept analysis[C]//The 5th IEEE International Conference on Computer and Information Technology (CIT2005). [S.l.]: IEEE Computer Society, 2005: 269-275.
- [6] Ganter B, Wille R. Formal concept analysis, mathematical foundation [M]. [S.l.]: Springer Verlag, 1999.
- [7] IBM, BEA Systems, Microsoft, et al. Business process execution language for Web services version 1.1[EB/OL].[2007-02-08].<http://www.ibm.com/developerworks/library/specification/ws-bpel/>.

科学出版社, 2005.

- [5] 林承达,孟令奎,彭溢.无线GIS空间数据动态副本管理研究[J].武汉大学学报:信息科学版,2007,32(3):263-265.
- [6] 李德仁.论21世纪遥感与GIS的发展[J].武汉大学学报:信息科学版,2003,28(2):117-1318.
- [7] Noh W, Kim S, Kim D, et al. Adaptive bus model for distributed multimedia stream in mobile computing environments[C]//Proceedings of the 15th International Conference on Information Networking, 2001: 849-854.
- [8] Kogekar S, Neema S, Koutsoukos X. Dynamic software reconfiguration in sensor networks [C]//Proceedings of Systems Communications, 2005: 413-420.
- [9] Denaro G, Mariani L, Pezze M, et al. Adaptive runtime verification for autonomic communication infrastructures[C]//World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, 2005: 553-557.