

# 一种面向需求的网格服务发现框架

## A Requirement - Oriented Grid Services - Finding Framework

毛雪涛 薛胜军 (武汉理工大学计算机科学与技术学院 武汉 430070)

**摘要:** 网格服务模型是一个广义的协同工作模型,其固有特征是资源所有者完全拥有资源的支配权。根据这一特征,本文参考市场机制提出了一种面向需求的网格服务发现框架。其基本观念是服务提供方主动查找用户,并提供服务,这与主动服务的观点相吻合。同时可以作为原来需求方查询资源的补充模型。并且资源提供方可以应用该框架进行有针对性的服务开发。文中主要对涉及的相关问题进行了初步的阐述,为今后的研究做铺垫。

**关键词:** 万维网服务 需求描述 需求发现 网格服务

### 1 前言

万维网服务 (Web Service) 框架的一个重要部分就是把服务请求者和合适的服务提供者联系起来的服务发现机制。请求者通过服务发现服务机制获得服务的相关信息。在万维网服务框架中,一个服务提供者首先要通过中介者对外发布自己所能提供的服务,服务请求者获得了该服务的描述文档之后,就可以根据信息和自己的参数组织一个消息,用这个消息调用该远程服务。服务请求者获得自己所要服务的描述的过程称为服务发现。服务发现是服务框架的主要环节,它把分布的服务消费者和服务提供者联系起来。

网格技术把服务扩展到了网格计算环境,提出了网格服务的概念。网格服务仍然采用万维网服务描述文档 (WSDL) 描述调用接口,只是需要对 WSDL 进行扩展。请求者要首先获得相应服务的 WSDL 描述,然后才能访问资源。因为在开放网格服务结构中,所有的资源都是服务,资源发现最终也就归结成了服务发现<sup>[1]</sup>。本文从网格服务的本质特征出发,结合已有的服务发现模型,研究了面向用户需求的服务发现这一问题。对其中可能涉及的问题做了一个初步的探讨。

### 2 网格资源管理中的服务发现

服务发现是把服务请求者和服务提供者联系在一起的服务中介者的主要功能之一。目前主要的服务发现规范有以 UDDI 为代表的集中式服务发现机制和以 WS - Inspection 为代表的分布式服务发现机制两种<sup>[1]</sup>。

UDDI 由注册中心和访问中心的接口 API 组成,用户通过 API 查询中心的信息。UDDI 的缺点是大量访问请求容易造成服务器故障,而且只能发现注册到 UDDI 中心的服务。WS - Inspection 中,服务发布在服务的提供点上,一个 WSIL (Web Services Inspection Language, 万维网服务检查语言) 文档提供相应结点上的服务信息,并可以在其中包含相关服务的链接。这种机制可以发现不在注册中心注册的服务,也克服了集中式发现机制所带来的访问拥挤<sup>[1]</sup>。WS - Inspection 机制能够对单个目标实现有重点的发现。但是,由于其本质上的分散性,如果通信伙伴未知的話,WS - Inspection 无法实现有重点的发现。

目前常用的网格服务是以 UDDI 为主的三层结构,其基本工作框架可以用图 1 描述,其中的标号表示执行步骤。

图中,数字 0 表示资源在事前向服务中介注册其服务描述信息。1 表示用户产生了某个需求。2 表示服务发现在服务中介处执行。3 表示发现相应的服务后通知用户。4 表示用户根据返回的资源信息获取资源。5 表示需求满足后返回结果。可以看到,服务发现所涉及的查询、匹配、更新等操作都在服务中介处执行。服务中介所发现的资源中需要剔除已被占用的资源。步骤 4 中,用户与资源的交互实际上也存在资源在服务前一刻或服务过程中被占用的可能。这一情况是网格固有的,因为资源归资源所有者支配,所以可能随时被收回。文献<sup>[2][3]</sup>提出的基于市场机制的网格

资源管理方法,引用经济学的理论描述了网格服务提供者和用户之间的行为特征。总的来说,资源提供者在市场机制的驱动下,总是希望将富余的资源进行出售的。

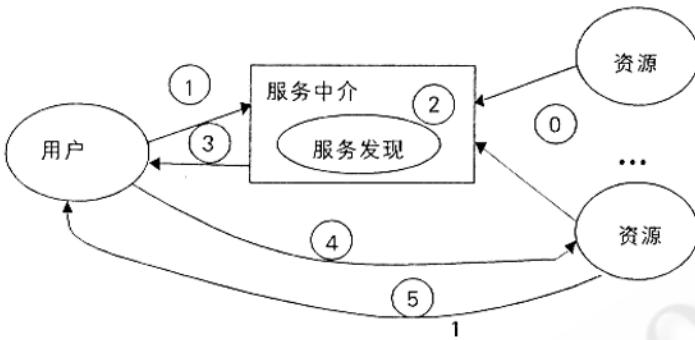


图 1 基于服务发现的网格服务流程

基于网格的这一本质特征,本文提出了面向需求的服务发现(也可称作需求发现)的概念,分析了基于需求发现的网格服务框架,研究了需求发现中的若干问题,并与服务发现框架进行了比较。需求发现可以与服务发现互为补充进行资源管理,能够更好的覆盖各种应用的需要。

### 3 需求发现

#### 3.1 需求发现框架

既然网格资源的所有者支配资源,那么当且仅当资源空闲时,资源提供方主动访问服务中介,查询自己能够为之提供服务的需求描述列表。一旦匹配,则立即提供服务。服务结束后将结果直接返回给用户。其基本工作框架可用图 2 描述。

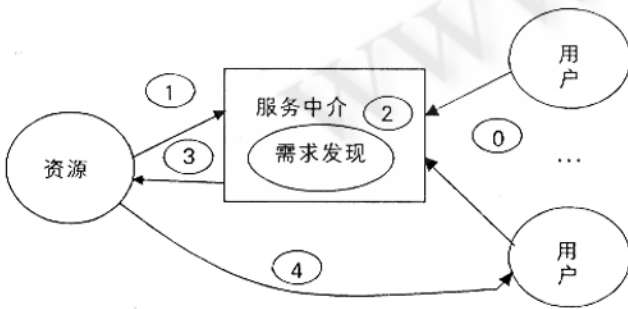


图 2 基于需求发现的网格服务流程

图中数字 0 表示用户在需要的时候向服务中介预定服务。1 表示某资源空闲并愿意主动提供服务,它向

服务中介请求查询可服务的需求。2 表示需求发现在服务中介执行。3 表示发现相应需求后通知资源。4 表示服务结束后结果返回用户。

比较图 1 和图 2 两种服务框架,可以发现以下几点:

(1) 后者较前者减少了一个步骤。图 1 中,用户获得资源信息后,需要定位到资源提供者,然后进行网格服务。而图 2 中,服务获得需求信息后,直接提供服务,并将结果返回给用户。前者适合于实时性较强,在服务过程中需要用户参与的网格应用。后者适合于无交互性要求的服务需求,例如科学计算问题。但是后者并不排斥前者,也就是说,在需要用户参与的时候,服务提供者同样可以实时与用户进行交互。

(2) 对于前一种框架,资源信息数据需要长期驻留在服务中介中,即使某资源总是处于被占用状态。而对于后一种框架则相反,用户需求信息需要及时更新。因为用户的需求一般是包含时间限制的,一旦期限已过,则应该从需求信息数据库中剔除该信息。

(3) 对于网格资源随时可能被收回的问题,后者可以在需求发现阶段通过参数进行预先匹配。资源提供者主动提供服务时,可以估计一个资源可用的时间段参数。而需求描述时也可以给出服务需求的时间段参数。如果在服务中介处的查询无法命中,可以促使资源提供者面向需求进行服务开发。而前者则只能在情况发生时进行资源重新定位。

#### 3.2 需求描述

考虑到需求与服务能够快速匹配,需求描述可以采用与资源描述相同的规范,所以,需求描述实际上也是资源描述(或服务描述)。为了将服务描述规范化,可以采用两种方式:一是服务中介提供能够将用户需求资源进行规范化描述的模块。由于用户的需求形式广泛多样,该方式实现有一定困难。另一种方式是在客户端对客户输入进行规范。该方式可以利用现有的技术来实现,但是对用户需求的提出有一定要求,而且对于各种应用的便利,需要开发多种输入界面。

常用的网格资源描述方法主要有 Globus 设计的资源规范语言 RSL (Resource Specification Language), 万维网联盟提出的 Web 服务描述语言 WSDL (Web Service Description Language), 资源描述框架 RDF (Resource Description Framework) 等<sup>[4]</sup>, 但是目前还没有

一个能够通用的统一规范。有一种方法是对某一个资源所提供的服务(一项或多项)进行多种不同的描述,而查询方可以从同一个服务的多个不同描述中选择适合自己使用的描述进行访问。**WS - Inspection** 规范提供了这种聚集不同类型的服务描述的方法。

基于需求发现的需求描述需要引入一些特定的元素或属性以保证其有效性。例如上面提到的服务时间段和服务时限。采用与服务描述统一的需求描述方法,可以将用户提出的服务需求描述成一个还未实现的虚拟服务 **VS (Virtual Service)**。文献<sup>[6]</sup>指出网格服务相关信息可以表示为一个四元组  $S(P, I, S, E)$ , 其中,  $P$  - 属性;  $I$  - 接口描述;  $S$  - 与服务实例相关的状态信息;  $E$  - 逻辑上的终端信息。而用户提交的请求看作是一个服务的局部文档, 表示为三元组  $R(P', I', S')$ 。事实上, **VS** (包含  $R$ ) 可以是资源所能提供的服务  $S$  的一个子集, 也可以是  $S$  的超集, 即  $R$  和  $S$  可以互换角色。这描述了需求大于单一服务而小于服务组合的情况。需求描述模型的不同决定了服务匹配的策略。如文献<sup>[7]</sup>中提出的 **Web 服务 QoS 需求描述模型**, 决定了基于服务质量感知的服务匹配。

一个需求被满足后, 其描述 **VS** 可以保存为一个新的服务描述。将来如果有类似的需求请求, 可以使用这个服务描述。这样, 可以建立前后服务发现之间的联系, 有效利用历史数据<sup>[1]</sup>。例如: 对于一个“旅行”服务需求, 可以描述为: 旅行 1 (目的地, 交通, 住宿, 用餐, 订票) 多元组。假设服务发现后, 新的描述变为: 旅行 2 (北京, 飞机, 人民宾馆, 人民餐厅, 人民票务)。如果以后又遇到类似的请求, 如: 出差 (北京, 交通, 住宿; 用餐), 则可以使用旅行 2 这个历史建立起来的服务描述。

### 3.3 需求匹配

需求匹配实际上也是服务匹配。服务匹配与所采用的描述模型有关。服务的匹配可以从以下几个方面进行考虑:<sup>[5]</sup>

(1) 上下文感知服务匹配。上下文信息对选择恰当的服务非常重要。上下文信息中最常用的就是位置信息。例如要选择网络打印机打印一份文档, 则位置信息是非常重要的选择依据。如 **SLP** 就可以根据位置信息选择匹配的服务。

(2) 逻辑区域感知服务匹配。现实生活中总有各

种形式的逻辑划分。如一个企业可以分为多个工作部门, 如采购部销售部等。可能销售部的职工只能用销售部的打印机打印文档, 为实现这种要求, 服务匹配技术就必须具有逻辑区域感知能力。**SLP** 也支持逻辑区域感知服务匹配。

(3) 服务质量感知服务匹配。服务质量是进行服务匹配的另一项重要选择依据。有些服务需要一定的服务质量保证, 如多媒体服务等。**INS** 中允许用户指定参数作为选择服务的依据。

目前的资源匹配算法, 大多是查询最优的单机资源或资源组合, 来满足用户应用的需要。其中使用资源组合来提供服务是网格资源管理的主要研究方向之一。这种做法需要各个成员资源之间进行协作, 共同来满足一个较大的应用。

对于可分割的应用, 可以将用户提交的应用需求分解为更细粒度的子需求。如果各个子需求并不一定非要同时被满足的话, 可以将子需求与服务资源进行匹配, 逐个进行服务。既可以减少服务对资源的占用时间, 又可以降低对资源的性能要求。

对于不可分割的应用需求, 只能将服务进行组合, 各服务成员之间进行协同工作来提供对用户需要的服务。一种情况, 当资源提供者发现一个 **VS** 与自身所能提供的服务  $S$  相同或内含 ( $VS \leq S$ ) 时, 则认为查询命中。另一种情况, 如果资源只能覆盖 **VS** 中的一部分 ( $S < VS \ \& \ S \wedge VS \neq \emptyset$ ) 时, 可以仅提供这部分服务, 并且完成后将需求进行更新。

如果匹配结果不命中, 服务器将保存“不命中日志”。资源方可以根据日志开发新的服务, 从资源利用的角度, 这样做可以减少冗余资源和闲置资源。

### 3.4 需求更新

需求是变化的, 在以下两种情况下需要更新需求:

(1) 需求过期: 需求在有效期内无法被服务, 则需要删除或更新有效期。属于主动维护。

(2) 需求变更: 需求变更有用户方变更和资源方变更两种情况。当需求被部分完成时可进行资源方变更。属于被动维护。

对于第一种情况, 可以采取定期检查更新的方式。系统检查出需求描述中要求的服务时限到期后通知用户, 得到回复后决定删除需求或更改时限。对于第二种情况, 则在服务完成后重新描述需求。例如: 修改

XML 文档的元素和属性值。

需求更新可以使用缓存更新的策略。更新策略包括算法和所依据的标准。标准有存贮时间、访问频率、缓存块大小、查询耗费等,或者采用基于多种标准的策略<sup>[10]</sup>。

#### 4 小结

综上所述,本文提出的面向需求的服务发现框架可以分为 3 大模块:需求注册、需求发现和需求更新。其框架的基本结构类似于服务发现框架结构,网格模型为消费者—中介—服务者 3 层模型。需求方先在中介服务器登记需求,服务方登陆中介服务器查询可以为之服务的需求,当查询命中后提供服务。该框架模型基于网格资源归所有者完全支配的特点,由服务提供者主动查询需求描述列表,获取可以为之服务的用户需求,并为之提供完全或局部的服务。其中涉及了需求描述模型、服务与需求的匹配以及需求更新几个问题。该模型运用“不命中日志”可以使资源方有针对性的开发新的服务。基于需求发现的网格服务框架中,各模块基本可以使用现有的技术实现。网格服务是复杂的协同活动,由资源方主动提供服务还需要运用市场机制(如奖惩机制,竞标机制<sup>[8]</sup>等)。下一步需要进一步细化该框架模型,首要的是需求描述的存储

和各种优化方法。

#### 参考文献

- 1 冯百明等,一种面向消费者的服务发现机制,计算机研究与发展,2003.12.
- 2 郭权等,一种动态自适应的计算网格资源管理方法,计算机工程,2004.11.
- 3 曹鸿强等,一种基于市场机制的网格资源分配方法,计算机研究与发展,2002.8.
- 4 朱莹等,网格资源描述技术的比较研究,网络与通信,2006.
- 5 高振国等,服务发现技术设计中需要考虑的问题,计算机工程,2005.10.
- 6 朱焯等,基于两阶段匹配的网格服务发现模型,计算机工程,2005.9.
- 7 牟玉洁等,面向服务质量的交互式 Web 服务选择,计算机集成制造系统,2006.5.
- 8 赵宏等,一种基于竞标机制的通用网格资源管理模型,计算机工程,2006.6.
- 9 徐志伟等,网格计算技术,电子工业出版社,2004.
- 10 Ismail Ari 等,ACME: Adaptive Caching Using Multiple Experts, Workshop on Distributed Data & Struc-