

# 一种基于多 Agent 系统的云服务自组织管理方法<sup>\*</sup>

侯富, 毛新军, 吴伟

(国防科学技术大学 计算机学院 计算机科学与技术系, 湖南 长沙 410073)

通讯作者: 侯富, E-mail: fu.houmail@gmail.com

**摘要:** 在开放动态的互联网环境下的云服务呈现出发散、动态演化、异构等特征,为了适应多样的应用需求以及持续变化的云环境,云需要以灵活、适应的方法来有效地管理和提供云服务.针对这一需求,提出了一种云服务自组织管理方法,该方法利用 agent 的环境感知和自主行为决策的能力,依据它的职责实现对云服务的自主管理,并通过 agent 间的交互以自组织的方式实现对服务资源的有效管理,进而适应云环境和应用需求的变化;提出了支持上述方法和机制的实现技术,包括云服务自组织管理的核心机制和实现框架、云服务汇聚和提供的运行机制及相关实现算法.利用支持语义 Web 服务发现匹配的公共测试数据设计并完成了两组验证实验,实验结果表明:所提出的方法可以在持续变化的云环境下有效地管理和提供云服务,进而满足动态多样化的应用需求.

**关键词:** 云服务;自组织管理;多 agent 系统;服务汇聚;适应性

**中图法分类号:** TP316

中文引用格式: 侯富,毛新军,吴伟.一种基于多 Agent 系统的云服务自组织管理方法.软件学报,2015,26(4):835-848.  
<http://www.jos.org.cn/1000-9825/4760.htm>

英文引用格式: Hou F, Mao XJ, Wu W. Self-Organizing management approach for cloud services based on multi-agent system. Ruan Jian Xue Bao/Journal of Software, 2015, 26(4): 835-848 (in Chinese). <http://www.jos.org.cn/1000-9825/4760.htm>

## Self-Organizing Management Approach for Cloud Services Based on Multi-Agent System

HOU Fu, MAO Xin-Jun, WU Wei

(Department of Computer Science and Technology, College of Computer, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Cloud services in dynamic and open Internet environment are typically decentralized, evolving, and heterogeneous. To satisfy the dynamic application demands and to respond to the continuous changes of the cloud environment, cloud services need to be provided and managed in a flexible and adaptive way. Aiming to address the challenge, this paper proposes an approach for cloud services self-organizing management. It achieves the autonomous and flexible management of cloud services based on MAS that are context-aware and autonomous in behaviors. The agents governing the services interact with each other to implement self-organization in order to respond to the continuous changes of cloud environment and application requirements. The paper presents the mechanisms and implementation techniques for the approach, including cloud service self-organization management mechanisms and implementation framework, cloud services aggregation and provision mechanisms and relevant core algorithms. Experiments based on public test data are performed, and the results show that the proposed approach can effectively provide and manage services under the evolving environments for various requirements of applications.

**Key words:** cloud service; self-organizing management; multi-agent system; service aggregation; adaptive

云计算是一种新的计算范型,它受到越来越多 IT 企业和学者们的关注,当前已有许多 IT 企业部署云平台并对外提供相应的云服务,例如 Amazon、Google、百度、阿里等.从企业界所提出的云结构以及云向应用提供功

\* 基金项目: 国家自然科学基金(61379051, 61133001); 新世纪优秀人才计划(NCET-10-0898)

收稿时间: 2014-08-01; 修改时间: 2014-10-14; 定稿时间: 2014-11-14

能形式来看,云计算可以认作是面向服务计算在商业领域的一种扩展实现,它主要包括 3 个层次:基础设施即服务(IaaS),平台即服务(PaaS),软件即服务(SaaS)。可以说,云计算系统中的一切都是或者可以是服务<sup>[1-4]</sup>。因此,本文中云看作是服务的集合,该集合中包括各种不同类型的资源,例如计算资源、存储资源等,从而使得云计算中的一些问题转变成了面向服务计算的问题,本文将从服务计算的角度来分析和解决相关的问题<sup>[1]</sup>。

随着大量的云平台的部署和使用,云服务资源的数量和类型逐渐增加,而这些服务资源通常分布在不同类型的云上,例如私有云、公共云等。由于云和云服务资源本身类型的不同,因此云服务表现出异构性、分布性等特征<sup>[5]</sup>,而一些云服务间则呈现出功能自相似的特征,这些自相似的服务则可能有着不同的质量状态。为了使云能够提供恰当的服务资源类型,以确保大规模应用动态的资源分配和加载,云需要一种有效的管理方法来管理这些服务资源。这一过程完全由人工的方式管理和控制则是复杂和困难的<sup>[6]</sup>,因为这些云是以发散的方式部署在互联网上的,而互联网环境自身又是动态和持续变化的,这使得每个云所面临的环境变化呈现出局部化的特征,即,每个云面临的变化情况不尽相同。因此,为了实现对所有云服务资源的有效管理,就需要以一种自组织的方式实现对服务资源的有效管理<sup>[7]</sup>。此外,基于互联网的开放云应用的需求具有动态性、多样性、持续变化性等特征,为了满足应用需求并能针对应用需求的变化特征向应用分配、调度和提供云服务资源,同时还确保云资源的有效使用,那么,云就需要一种灵活的、自治的手段来管理这些服务资源。

因此,针对如何管理、分配及提供动态变化环境下的云服务资源来满足多样应用的需求,已经成为工业界和学术界关注的热点问题。多 agent 系统(multi-agent system,简称 MAS)<sup>[8]</sup>被认为是解决具有动态性、自主性等特性的复杂的信息系统的有效手段。在多 agent 系统中,每个行为实体被认为是一个软件 agent,它能够感知其所处的环境上下文,根据其自身状态和环境状态变化信息自主地决策其行为;agent 还可以通过协作的方式,与其他 agent 实现系统的全局性目标<sup>[9]</sup>。由于每个 agent 的自主性以及整个系统的发散式管理和控制,MAS 内的 agent 只能利用局部的交互行为(即,自组织行为)来获取全局性的行为结果(即,涌现现象),进而实现系统目标<sup>[10,11]</sup>。目前,已有一些研究将多 agent 系统引入面向服务计算领域,如解决服务的组合、服务发现、服务管理等问题<sup>[12-14]</sup>。当然,云计算领域也有一部分工作引入 agent 技术,它们主要将 agent 作为代理的形式实现云的功能,如服务的自动协商等<sup>[15]</sup>。

本文提出了一种基于多 agent 系统的云服务自组织管理方法来解决动态变化的云环境下服务资源的有效管理、分配和提供问题,继而为满足应用多样性的需求提供支持。该方法利用 agent 的环境感知能力实现对应用需求信息和云服务环境及其自身状态的感知和监控,而利用 agent 的自主行为决策能力结合感知和监控的信息实现了对服务资源的自主管理,利用 agent 间局部交互能力以自组织方式协作地完成了云服务的组织、分配和管理任务,满足了多样的应用需求。

本文第 1 节介绍相关的研究工作。第 2 节介绍基于多 agent 系统的云服务的自组织管理方法,这部分主要介绍基于多 agent 系统的云服务自组织管理模型以及实现该过程的管理实体模型。第 3 节具体介绍实现云服务自组织管理的相关技术,主要包括云服务自组织管理的实现框架、自组织管理实体的自主管理机制、云服务的汇聚和提供机制以及实现自组织管理过程的核心算法。第 4 节通过两组不同的实验和实验结果说明本文所提出的自组织管理方法的可行性和有效性。最后对全文进行总结并针对下一步工作进行展望。

## 1 相关工作

当前,越来越多的 IT 公司或者互联网企业正在构建或打算构建其自己的云系统,这使得云服务的数量越来越庞大,尤其是计算资源、存储资源等也可以通过服务的形式向用户或应用提供,这使得云服务的类型逐渐增多。然而,这些系统所提供的功能间不可避免地会有重叠,而这些重叠的服务则具有不同的质量状态,因此需要一种方法对这些分散的、异构的云服务资源进行有效的管理。

当前,解决云服务资源的管理问题常采用的技术是自治计算技术,例如:Amanatullah 等人<sup>[1]</sup>提出了一种用于实现云服务管理的云计算体系结构,在该体系结构中,云服务提供商通过合理地规划操作过程来管理他们的云服务资源;Buyya 等人<sup>[16]</sup>提出了一个概念性的框架来解决自治资源的提供问题,同时,他们还提出了一种创新性

的管理方法来支持云上的 SaaS 应用,他们的实验结果证明了云的自治管理方法的优势;Puviani 和 Frei 提出了一种特殊的适应模式,实现了云的自组织管理,同时解决了确保最佳云性能和服务请求响应时间之间的瓶颈问题<sup>[17]</sup>。此外,还有的研究人员从宏观的角度,将应用需求直接映射到具体的运行时的云资源的管理中,典型的工作就是张星等人<sup>[18]</sup>所提出的基于运行时模型的多样化云资源管理方法。

同样,也有一些工作将多 agent 系统引入到云计算中,例如:Mong<sup>[15]</sup>就强调多 agent 系统在自动云服务组合、云商务的复杂协商过程和云服务的搜索方面的潜在应用;Cao 等人则利用多 agent 系统技术来管理云服务,进而使得服务的质量得以保障<sup>[19]</sup>。

总的来说,现有的研究工作要么是从微观或个体的角度利用自治计算的技术实现对单个云资源的管理、分配、调度和提供,要么是从宏观或抽象的角度利用运行时模型实现对多云资源的管理。在这些工作中,也有部分工作将多 agent 系统技术引入到云服务资源的组合、搜索、资源分配等过程中,但是这些工作都没有同时从微观和宏观两个方面来看待云服务的管理问题,即:根据应用的宏观需求来调整微观或个体云资源的提供分配过程,然后通过微观或个体云资源的提供效果或质量来影响宏观的资源需求的满足。本文将同时从宏观和微观两个方面看待云服务管理问题,为了实现对云服务的管理,本文将利用多 agent 系统通过个体层面的自主管理和协作交互,实现在宏观的云服务管理、分配和提供的效率上的提升,即,云服务的自组织管理方法。

## 2 基于多 agent 系统的云服务自组织管理方法

本节利用软件 agent 的环境感知能力对云服务及其运行环境进行感知和监控,根据其感知和监控的信息,软件 agent 能够自主地决策其对特定类型的服务的管理行为。而为了适应动态的应用需求和持续变化的云环境,软件 agent 间则会通过局部的交互行为,以自组织的方式实现对云服务的管理、分配和提供。结合这一思想,本节首先介绍基于多 agent 系统的云服务自组织管理模型,然后介绍实现自组织管理的实体模型。

### 2.1 基于多 agent 系统的云服务自组织管理模型

基于多 agent 系统技术,本文提出了一种云服务自组织管理模型(如图 1 所示),该模型主要由 3 部分组成,分别是客户部分、agent 部分和云服务部分。

- 客户部分(也称应用部分)主要由具体的客户组成(如图 1 左部分所示),这些客户可以是具体的人,也可以是某种应用。当这些客户对某种服务资源有需求时,他们只需要向 agent 部分提出具体的需求描述,然后,agent 部分的 agent 会根据客户的需求描述找到相应的服务资源并向客户提供该服务;
- agent 部分主要由两类 agent 组成(如图 1 中间部分所示),分别是服务代理 agent 和服务管理者 agent,它是实现云服务自组织管理的核心层,其中,
  - 服务管理者 agent(简称服务管理者)实现了对云服务的管理功能,每个服务管理者所管理的服务类型则取决于其职责,而一类服务则可被至少一个管理者所管理,这种由管理特定服务类别的多个管理者所构成的集合被称为服务管理者组。当有应用向某服务管理者请求服务时,该管理者会根据服务当前的服务环境和服务自身状态发现并筛选满足应用需求的服务:如果其所管理的服务中存在应用所需求的服务,那么管理者将向应用提供服务;否则,管理者将与其他在同一服务管理者组内的其他服务管理者协同地提供服务。此外,服务管理者所管理的服务集合根据管理者的管理能力、服务环境变化和应用的请求情况而动态调整,例如新的服务加入或根据负载均衡的需求对管理者所管理的服务集合进行拆分或合并等;
  - 服务代理 agent(简称服务代理)是应用的服务获取代理,它可以解析应用的需求描述并能够根据应用需求获取到其所需求的服务的管理者信息,同时能够根据应用的需求信息和管理者情况选择恰当的服务管理者提供服务,而每个应用则拥有且仅拥有一个服务代理来处理其服务获取的任务和目标;
- 云服务部分(如图 1 中右部分所示)主要是由一组云构成,每个云中都对提供某种或几种类型的服务,而这些服务资源则由 agent 部分的服务管理者根据其职责和能力分类管理。

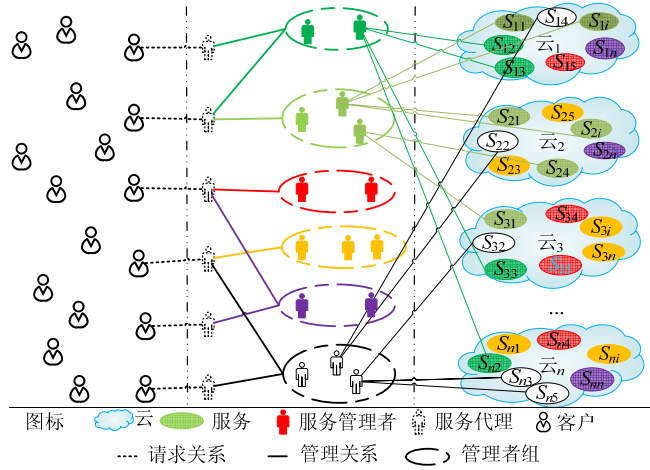


Fig.1 Self-Organizing management model of cloud services based on multi-agent system

图1 基于多 agent 系统的云服务自组织管理模型

通常情况下,云应用的需求是动态多样的.为了向应用提供满足其需求的服务资源,并能够合理而有效地使用服务资源,单一的服务管理者往往是无法满足应用需求的.此外,云系统是分散的并且其所在的互联网环境是动态不确定的,这使得云服务同样具有了一定的不确定性.为了确保服务的分配、提供是相对确定和稳定的,显然,由单一的服务管理者同样是无法完成的.因此,就需要多个服务管理者间通过局部交互的方式来协作地实现对云服务资源的管理、分配和提供,这既提高了服务管理和使用的效率,同时也确保了服务资源的相对稳定性.而自组织技术<sup>[10]</sup>是实现该过程的很好的选择,因为自组织系统是非常健壮的,能够适应环境和需求的变化,而且它也是确保自组织系统内各个组件的生存能力的重要方式<sup>[11]</sup>.而自组织过程通常伴随着涌现现象<sup>[11]</sup>,本文所提出的基于多 agent 系统的云服务自组织管理方法的涌现现象就是云服务资源被分类成若干组,每组内的服务则由若干管理者管理,这些服务组则可以根据需要动态地进行调整(如图 2 所示).

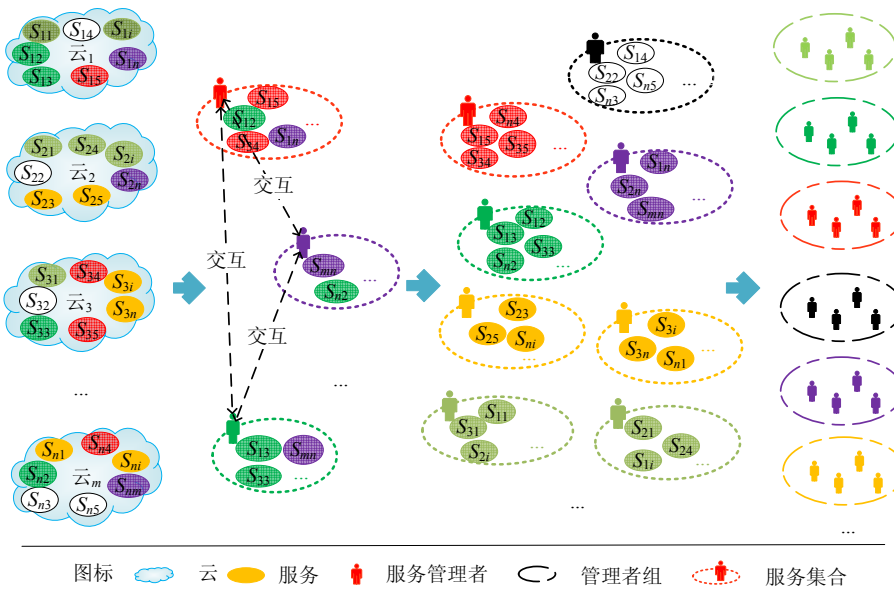


Fig.2 Illustration of cloud services self-organizing management

图2 云服务自组织管理示意图

### 2.2 云服务自组织管理实体模型

在实现云服务自组织管理的过程中,实际的参与者包括两部分:一部分是云服务及其环境,另一部分是服务的管理者.结合服务管理者的功能职责以及为适应持续变化的云环境和动态多样的应用需求,本文提出了一种云服务自组织管理实体模型(如图 3 所示),其中的服务管理者就是一个软件 agent,它的管理功能主要包括 3 个方面:

- 接受应用请求和自主地提供服务的功能;
- 对云服务环境和服务状态监控的功能;
- 云服务管理自主决策的功能.

通常情况下,一个服务如果要被服务管理者管理,首先,服务的提供者要将服务部署到一个云环境下,并在其中配置该服务的信息;然后,云环境会向服务管理者登记该服务的信息;最后,管理者根据其职责和管理能力决定是否管理该服务.

根据云服务自组织管理实体模型中服务管理者的功能,本文设计了一个云服务管理者的内部实现结构(如图 4 所示),在该结构中,主要包含以下功能部件:

- 1) 服务登记部件,负责实现服务与管理者间管理关系的确立过程.当服务与其职责一致时,该模块将服务信息存储到服务池内,并配置态势传感器;否则,它会通知协作单元,通过自组织的方式找到相应的服务管理者来管理该服务;
- 2) 态势传感器,它主要负责监控云服务的运行环境和服务本身的状态信息,并实时地将这些信息的变化情况告知服务管理者,即,存储在服务池内;
- 3) 服务池,存储管理者所管理的服务信息,如功能、访问端口等和该服务及其环境的状态信息,它能够支持行为决策单元和协作单元的查询和访问;
- 4) 行为决策单元,服务管理者的核心组件,它负责实现服务的分配、选择和调用的决策过程.而该决策过程的实现,则依赖于应用请求、服务及其环境的状态;
- 5) 应用请求分析部件,负责接收应用的服务请求描述,根据该描述触发行为决策单元,找到相应的服务资源,并对外提供该服务;
- 6) 协作单元,实现云服务自组织管理的重要模块,它主要负责实现与其他管理者交互协作的功能,即:通过该单元,能够实现管理者间的自组织的管理过程;
- 7) 服务调用器,该模块根据行为决策单元的命令调用相应的服务,并将结果返回给服务请求分析模块.

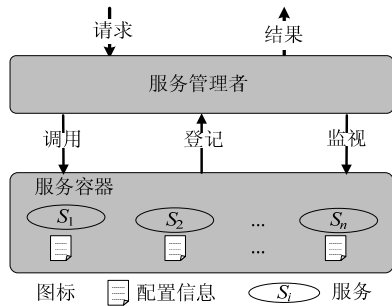


Fig.3 Entity model of cloud services self-organizing management  
图 3 云服务自组织管理实体模型

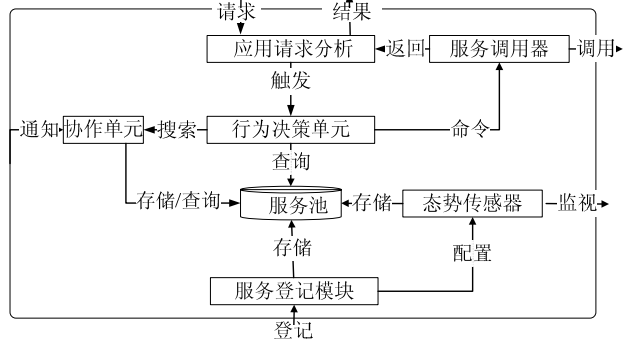


Fig.4 Implementation structure of cloud service manager  
图 4 云服务管理者内部实现结构

### 3 云服务自组织管理方法的实现技术

为了支持基于多 agent 系统的云服务自组织管理方法的实现,同时也为了提高云服务管理的效率,本文基于 SOA 的管理思想设计了一种云服务自组织管理的实现框架(见第 3.1 节);而为了支持云服务的自组织管理过程和适应动态变化的云环境 and 应用需求,本文提出了自组织管理者的自主管理机制(见第 3.2.1 节)、云服务的汇聚和提供机制(见第 3.2.1 节).此外,为了支持云服务的自组织过程的具体实现,本文提出了服务管理者的协作算法和云服务自组织管理算法(见第 3.3 节).

#### 3.1 云服务自组织管理的实现框架

根据前文可知,每个服务管理者都有相应的职责,它定义了服务管理者所管理的服务的功能范围,即规定了哪些服务可以被该管理者管理,因此就需要一种技术,它既能够描述服务管理者的职责,又能够描述服务,因为服务管理者需要理解服务的功能,并根据这种描述清晰、准确地对云服务进行分类查找.本文中,采用语义 Web 服务技术来定义描述云服务和管理者的职责,因为语义文本技术能够更加清晰、准确地定义 Web 服务的功能,可以提高服务发现和分类的效率<sup>[20,21]</sup>.此外,agent 同样支持利用语义技术来定义和描述 agent 的能力,并支持 agent 间复杂的协作功能.在本文中,我们利用 OWL-S<sup>[22]</sup>来描述和定义云服务和管理者的能力,即管理者的职责.

为了支持云服务自组织管理方法的实现和提高服务这种自组织管理的效率,本文设计了一种云服务自组织管理实现框架(如图 5 所示).

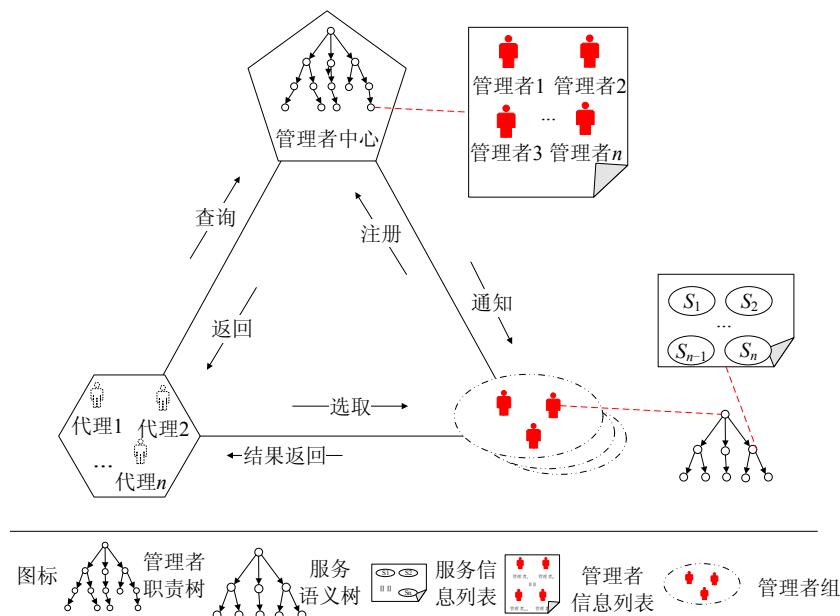


Fig.5 Implementation architecture of cloud services self-organizing management

图 5 云服务自组织管理实现框架

该框架由 3 部分组成:

- 服务管理者组由一组具有相同职责的服务管理者组成,一个服务管理者组中至少包括一个服务管理者,而每个管理者中都有一棵根据其职责所建立的服务语义树,该语义树的每个节点都记录着服务管理者所管理的服务的信息,这些信息主要包括服务的功能性信息、服务的状态信息、服务环境状态信息等.当服务的状态或者服务的环境发生变化时,服务管理者间会通过自组织的方式来应对这些变化.而应用请求的变化,如请求数量、服务质量需求变化等,都会使得服务管理者组以及服务管理者所管理

的服务发生动态的调整和变化.当然,这种变化同样是以自组织的方式实现的;

- 管理者中心是服务管理者的存储中心,在该中心的内部,有一棵存储所有服务管理者职责的职责树,它的每个叶子节点都存储着具有相同或相容职责的服务管理者的信息列表,即一个服务管理者组.该职责树的建立,则是基于语义 Web 服务的本体所建立的.每个服务管理者被创建时都需要向管理者中心注册,然后,管理者中心根据其所注册的管理者的职责将该管理者的信息存储在对应的叶子节点的管理者信息列表中.当有新的云服务加入时,管理者中心会根据服务的语义信息找到相应的管理者,并通知该管理者管理此服务;如果其中没有管理此服务的管理者,那么管理者中心会创建一个相应的服务管理者并更新它的职责树;
- 服务代理正如前面所介绍的,它的主要职能是根据应用的请求向管理者中心查询相关的管理者信息,然后从其中选取一个恰当的服务管理者并获取其所需求的服务.

### 3.2 自组织管理的核心机制

#### 3.2.1 自组织管理实体的自主管理机制

云服务自组织管理实体的自主管理过程主要包括 4 个不同的阶段(如图 6 所示).

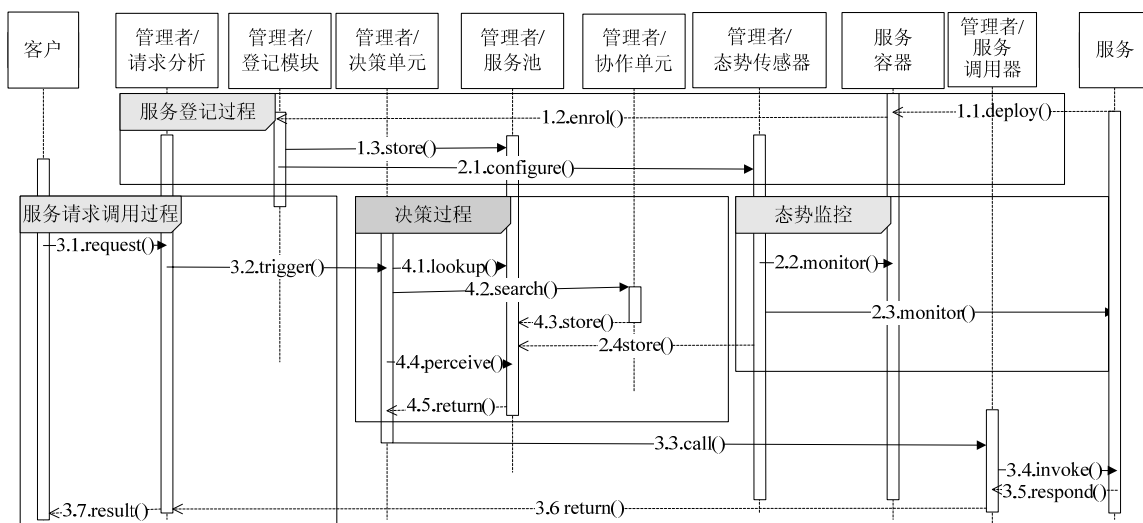


Fig.6 Autonomous management process of self-organizing management entity

图 6 自组织管理实体的自主管理过程

- 服务登记阶段.该阶段,服务首先被服务的提供者部署到服务的运行环境中;之后,服务环境会将该服务信息登记到管理者的服务登记模块;然后,该模块根据管理者的职责和服务的语义信息确立服务的管理关系:如果一致,则将服务信息存储到服务池,并配置态势传感器;否则,通知协作单元借助管理者中心通知相关的管理者来管理该服务;
- 服务监控阶段.当确立了服务的管理关系后,服务登记模块将根据服务的元信息配置服务管理者的态势传感器;然后,传感器将根据服务的配置信息对服务及其环境状态进行监控.当所监视的信息发生变化时,态势传感器会将这些变化的信息存储到服务池中,实现状态信息的更新;
- 自主决策阶段是自组织管理实体实现自主管理的核心过程.当接收到请求分析模块的服务请求时,决策单元将向服务池查询相关的服务,并感知其状态信息:如果找到满足需求的服务,那么服务管理者将提供服务;否则,决策单元将通过协作单元与本服务管理者所在的服务管理组中的其他服务管理者请求协作提供服务;
- 服务请求调用阶段.当一个客户向请求分析单元请求某个服务时,该单元将触发自主决策单元的自主

决策过程,根据自主决策的结果,决策单元将命令服务调用器调用所选的服务,并将通过应用请求分析单元把结果返回给客户.

### 3.2.2 云服务的汇聚和提供机制

为了支持服务管理者之间的服务汇聚和向具有动态多样性需求的客户提供服务,本文提出了关于云服务的汇聚和提供机制.这两个机制是基于多 agent 系统和语义 Web 服务技术实现的.下面加以分别介绍.

#### (1) 云服务汇聚机制(如图 7 所示).

该机制的设立主要是为了使得被服务管理者所管理的服务的自组织涌现变得更加有序,进而使得服务的提供和使用变得更加高效.该机制更加适用于以下几种状况:

- 云服务资源的管理呈现为混乱的或者无序状态;
- 新的服务被部署到云的环境;
- 服务离开或者消失.

汇聚机制的具体工作过程针对不同情况有不同的工作方式,其中,关于服务离开或者消失的这种情况只需要该服务的服务管理者与其所在的同一管理者组的其他管理之间利用自组织的方式就可以实现.本文将重点介绍云服务是混乱或者无序状态和新服务加入云环境状态下的云服务的汇聚过程,这两种状态下的汇聚过程可归结为同一过程,因为无序状态可以被看作是服务加入状态的多次重复,具体的汇聚过程是:

- 首先,服务提供者向某一管理者登记服务信息来建立管理者与服务之间的管理关系,如果该服务与管理者的职责一致,那么按照服务管理者的自主管理机制中的登记和监控过程即可实现服务的管理;否则,该管理者将会向管理者中心注册该服务的信息;
- 然后,根据服务的语义信息,管理者中心将搜索它的职责树,查找管理该服务的管理者组,如果找到这样的管理者组,那么管理者中心将通知该管理者组中的管理者来管理该服务,而该服务的最终管理者将由找到的管理者组通过自组织的方式确立;如果未找到,那么管理者中心将创建一个管理该服务的管理者,并更新它的职责树.

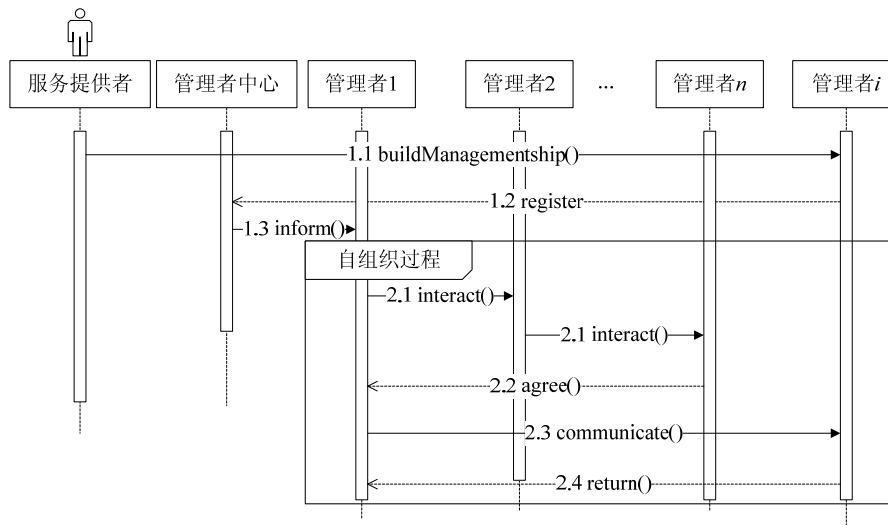


Fig.7 Aggregation process of cloud services

图 7 云服务的汇聚过程

#### (2) 云服务提供机制(如图 8 所示).

该机制利用服务代理、管理者和管理者中心间复杂的交互过程实现对客户的服务提供,具体的服务提供机制包括两种请求方式:正常请求方式和协作请求方式.其中,



- 正常请求方式的工作过程是:首先,客户向服务代理发出服务请求;然后,服务代理解析客户的需求描述,针对客户的需求描述,向管理者中心查询客户需求的服务的管理者信息;然后,从中选取出其恰当的服务管理者;接着,该服务管理者利用其自主管理机制中的服务请求调用过程向客户提供服务;
- 而协作请求过程与正常请求过程的主要区别是,当向所选的管理者请求某一服务时,如果该管理者所管理的服务不能满足客户的需求,则该管理者会通过与其所在的管理者组内的其他管理者以协作方式,即自组织的方式,来向客户提供服务。

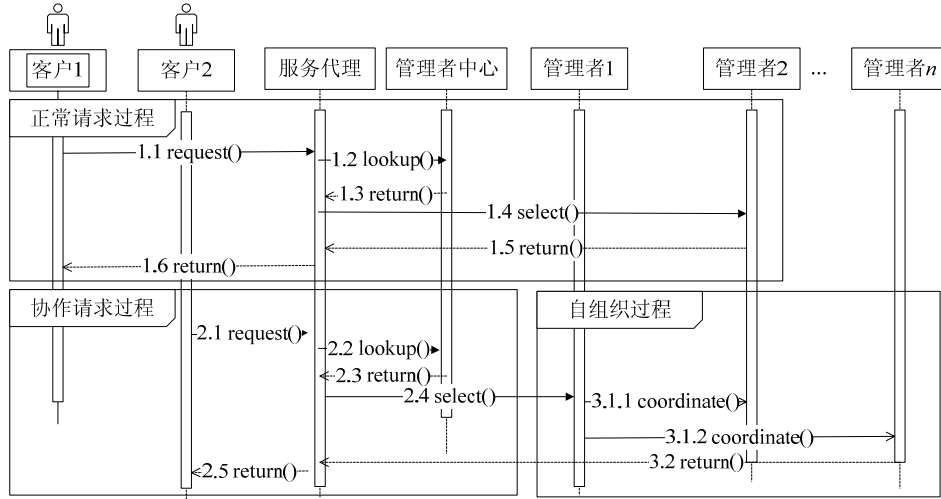


Fig.8 Process of cloud services

图 8 云服务的提供过程

### 3.3 云服务自组织管理过程的核心算法

为了支持本文所提出的云服务自组织管理方法和相关的实现机制,我们设计了两类算法分别是:服务管理者的协作算法和云服务自组织管理算法,它们是实现云服务的自组织管理过程的核心,而服务管理者的协作算法和云服务自组织管理算法就是软件 agent 关于实现云服务自组织管理的核心推理规则和决策依据。

#### (1) 服务管理者的协作算法

该算法描述了单个服务管理者是如何通过协作的方式实现对服务的管理和提供的,该算法主要处理 4 种类型的事件,服务管理者处理这 4 种事件的具体过程为:

- 当服务管理者接收到一个与其职责不一致的服务登记事件时,管理者会通知它的协作单元并将该服务注册到管理者中心,然后等待其他管理者的管理请求(具体见算法 1 的第 3 行、第 4 行);
- 当服务管理者接到其他服务管理者的服务管理请求事件时,它会将该服务返回给请求的管理者(具体见算法 1 的第 5 行、第 6 行);
- 当服务管理者受到服务协作请求事件时,管理者会在其内部找到满足的服务,并将该服务返回给请求者(具体见算法 1 的第 7 行~第 9 行);
- 而当服务管理者接收到服务的请求事件但是服务管理者内没有满足的服务时,服务管理者会利用协作单元搜索服务,以满足应用的需求(具体见算法 1 的第 10 行、第 11 行)。

**算法 1.** 服务管理者的协作算法.

输入:服务管理者  $ma$ ,服务语义树  $st$ ,新部署的服务信息  $si$ ,管理者中心  $MC$ , $MC$  职责树  $rt$ ,外部请求  $Oreq$ ;

输出:协作结果  $result$ .

1. begin
2.  $ssi = GetServiceSemanticInformation(si)$

3. if 接收到与其指责不匹配的服务登记 then
4.      $result=InformCoordinativeUnitRegisterServiceToMC(MC,rt,ssi)$
5. else if  $OReq$  是其他管理者关于服务管理的请求 then
6.      $result=AgreeAndReturnTheService(OReq,ma,si)$
7. else if  $OReq$  是关于协作提供服务的请求 then
8.     服务  $s=FindTheSatisfyingService(OReq,st)$
9.      $result=ReturnServiceToReqManager(OReq,ma,s)$
10. else if  $OReq$  是一个服务提供请求但是  $ma$  中无满足的服务 then
11.      $result=SearchServiceByCoordinativeUnit(OReq)$
12. end if
13. 返回  $result$
14. end

#### (2) 云服务自组织管理算法

该算法主要描述了应对云服务及其环境变化(内部状态)和应用需求的或管理过程变化(外部变化)的自组织管理过程,它以服务管理者组  $MG$ 、对应的语义树  $STree$  和初始管理者  $ima$  作为输入参数,以更新了的服务管理者组  $MG'$ (具体过程如图 10 所示)作为输出:

- 当服务或者其环境发生变化时,服务管理者将会感知到这种变化,而这些感知到变化的管理者间会根据变化类型自主地进行交互协商,根据协商的结果,这些管理者会调整各自所管理的云服务资源(具体见算法 2 的第 2 行~第 8 行);
- 当有外界的应用或服务管理的请求时,服务管理者会根据它所了解的管理者及这些管理者的职能找到相关的服务管理者,通过协商过程实现服务的对外提供或管理(具体见算法 2 的第 9 行~第 14 行).

#### 算法 2. 云服务自组织管理算法.

输入:服务管理者组  $MG=\{ma_1,ma_2,\dots,ma_n\}$ ,服务语义树  $STree=\{st_1,st_2,\dots,st_n\}$ ,内部状态  $IS=\{is_1,is_2,\dots,is_m\}$ ,外部状态  $OS=\{os_1,os_2,\dots,os_k\}$ ,初始管理者  $ima$ ;

输出:已更新的服务管理者组  $MG'$ .

1. begin
2.     if 变化状态  $cs \in IS$  then
3.          $(CMG,CSTree)=FindChangedManager(MG,STree)$
4.          $MG'=add(MG',MG-CMG)$
5.         for 所有  $cma_i$  和  $cst_i$ ,其中, $cma_i \in CMG,cst_i \in CSTree$  do
6.              $InteractWithOthers(cma_i,CSTree,CMG-cma_i)$
7.              $MG'=add(MG',UpdateManager(cma_i,cst_i))$
8.         end for
9.     else if 变化状态  $cs \in OS$  then
10.          $(maj,stj)=InteractWithEachToFindOneManager(MG,STree)$
11.          $MG'=add(MG',MG-cma_i)$
12.         服务  $s=CoordinateAndGetService(maj,ima)$
13.          $MG'=add(MG',UpdateManagerWithNewService(maj,stj,s))$
14.     end if
15. 返回  $MG'$
16. end

## 4 实验及分析

为了验证基于多 agent 系统的云服务自组织管理方法的可行性和有效性,本文基于公共的语义 Web 服务测试数据(数据来源:<http://projects.semwebcentral.org/projects/owls-tc/>)设计实现了两个不同的实验,以分析本文所提方法在支持云服务的管理和提供方面的可行性.同时,分析服务管理者的数量对云服务的管理效率和质量的影响效果.由于本文所提出的管理方法其目的是通过云服务的有效管理、分配及提供,使得在动态变化环境下的云服务能够满足多样应用的需求,也就是说,通过后台的服务管理方法,使得应用在查找、选择、匹配和使用的效率和质量得以提升.因此在具体的实验过程中,我们通过服务发现和匹配的时间消耗来判断我们的云服务自组织管理方法的可行性和有效性.本文利用两个服务相似性匹配的算法来完成实验,它们分别是 extended jaccard 相似匹配算法和 cosine 相似匹配算法<sup>[23]</sup>.而具体的云服务管理平台,即实现云服务的自组织管理平台,则是基于我们项目组的 ServiceAutoManager 平台<sup>[24]</sup>扩展所实现的平台.为了支持对云服务的自组织管理过程,我们扩展了云平台的服务管理者的协同、自组织汇聚和服务请求及提供的功能.

在第 1 个实验中,我们收集了 extended jaccard 和 cosine 两个算法基于原始服务数据集和经过我们的管理方法处理的服务数据集的服务发现和匹配时间.为了说明其可靠性和合理性,我们采用了 10 组不同数量级的数据集进行实验,分别含有 100,200,...,1000,各组之间间隔的服务数量是 100,在处理数据的过程中,我们采用 5 个服务管理者来管理各组实验数据,每组实验重复了 10 次,最后将每组实验的结果取均值.根据实验结果,我们绘制了相应的曲线图,其中,  $x$  轴代表测试的数据集的大小,单位是个数; $y$  轴代表服务发现和匹配所花费的时间,单位是 s.为了说明本文所提方法的普适性,在实验过程中,我们分别针对四轮车价格服务和食物价格服务展开了实验,最终的实验结果曲线图如图 9 所示.

第 2 个实验主要为了说明服务管理者的数量将会影响云服务的管理质量和效率.在第 2 个实验中,我们同样收集服务发现和匹配所花费时间,与第 1 个实验不同的是:

- (1) 第 2 个实验中只收集 cosine 算法发现和匹配服务所花费的时间;
- (2) 只采用两组不同大小的服务的数据集,它们分别包含 500 和 1000 个服务;
- (3) 在第 2 组实验中,我们将利用 4 组不同的数量的服务管理者对服务数据集进行处理,它们的数量从 5~20,间隔为 5 个服务管理者.

针对每组实验,我们同样重复 10 次实验,计算所花费时间的平均值,并记录实验结果,同时绘制出一个柱状图,其中,  $x$  轴代表服务管理者的数量,单位为个数; $y$  轴代表服务发现和匹配所花费的时间,单位是 s.

此外,在第 2 个实验中,为了说明本文所提方法的普适性,在实验过程中,我们也分别针对四轮车价格服务和食物价格服务展开了实验,最终的实验结果柱状图如图 10 所示.

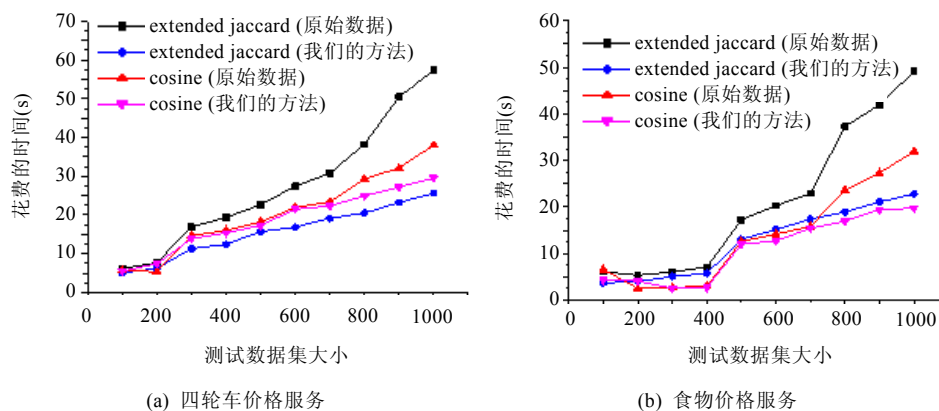


Fig.9 Curves diagram of experiment the 1st result

图 9 实验 1 结果的曲线图

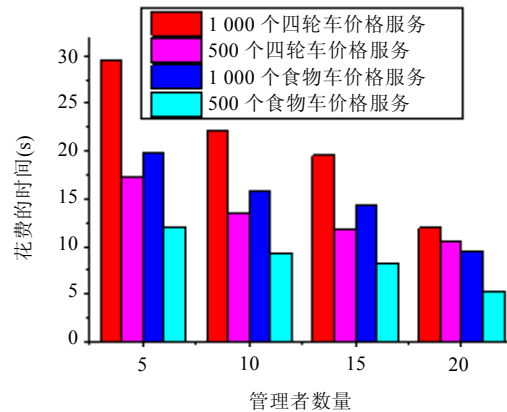


Fig.10 Histogram of experiment the 2nd result

图 10 实验 2 结果的柱状图

从图 9 我们可以看到:用我们的方法对服务进行管理后,服务发现和匹配所花费的时间明显较原始情况下的时间要少很多,而这种现象会随着服务数量的增加变得越来越明显.这一结果不会因为服务发现和匹配的算法的不同而有所不同,这说明,基于多 agent 系统的云服务自组织管理方法提高了服务管理、分配和提供的效率.

而从图 10 我们可以看到:随着服务管理者数量的增加,服务发现和匹配所花费的时间变得越来越少,这一结果不会随着服务数量的变化而有所变化.而根据我们统计的结果,可以发现:无论服务数量是 500 还是 1 000,只要在测试的数据集大小相同的情况下,服务的发现和匹配的数量不会随着管理者数量的变化而发生明显的波动,这说明本文所提出的管理方法不会造成服务的提供质量上的变化.

根据上述两个实验结果我们可以得出结论:基于多 agent 系统的云服务自组织管理方法是可行的和有效的.但是根据实验过程的观察可以发现:我们的方法同样存在着不足,即系统的初始化过程耗费了太多的时间,这将在后续的研究中予以克服.

## 5 结论与下一步工作

云服务具有发散、动态演化、异构等特征,而为了确保云服务能够适应多样的应用需求,并能够应对持续变化的云环境,因此,云需要一种灵活自主的方法来有效地管理、分配和提供这些服务.针对这一需求,本文提出了一种基于多 agent 系统的云服务自组织管理方法,在该方法中,云服务被一种称为服务管理者的软件 agent 所管理,该服务管理者利用 agent 的环境敏感性实现对服务和环境状态信息的感知和监控,根据感知的信息和应用的需求信息,结合自身的状态和它的职责,服务管理者利用 agent 的行为自主决策能力实现对相关的云服务的自主管理.而当单个管理者不能满足云服务管理或应用的需求时,服务管理者间会利用 agent 的交互协作能力,以自组织的方式完成服务管理、分配和提供的任务.

为了支持云服务的自组织管理方法的实现,本文设计并提出了云服务自组织管理的实现技术.基于 SOA 架构中服务的管理思想,本文设计了一种云服务的自组织管理的实现框架;为了适应动态变化的应用需求和云环境,本文提出了云服务自组织管理实体的自主管理机制、云服务的汇聚和提供机制.此外,为了支持云服务自组织管理过程的实现,本文设计并实现了云服务自组织管理过程的核心算法.最后,为了验证自组织管理方法的相关特性,本文基于公共的服务测试数据集,设计并实现了两个不同的实验.实验结果表明:我们的方法可以提高服务管理、分配和提供的效率,该方法是可行的、有效的.

下一步,我们将在现有工作的基础上展开相应的研究工作,主要包括:

- (1) 优化现有的管理方法和算法,提出相关的自组织策略;
- (2) 研究相关的自适应机制,从而使得云服务能够应对其环境和应用的动态变化的需求.

**致谢** 在此,我们向对本文的工作给予支持的同行,尤其要向项目组内的毛新军老师和吴伟同学表示感谢,感谢他们在文章写作和实验验证过程中所提出的宝贵建议。

### References:

- [1] Amanatullah Y, Lim C, Ipung HP, Juliandri A. Toward cloud computing reference architecture: Cloud service management perspective. In: Proc. of the Int'l Conf. on ICT for Smart Society (ICISS). 2013. 1–4. [doi: 10.1109/ICTSS.2013.6588059]
- [2] Tai S, Nimis J, Lenk A, Klems M. Cloud service engineering. In: Proc. of the 2010 ACM/IEEE 32nd Int'l Conf. on Software Engineering. 2010. 475–476. [doi: 10.1145/1810295.1810437]
- [3] Tai S. Cloud service engineering: A service-oriented perspective on cloud computing (invited paper). Towards a Service-Based Internet, 2011,6994(1):191–193. [doi: 10.1007/978-3-642-24755-2\_18]
- [4] Jamil E. What really is SOA: A comparison with cloud computing, Web 2.0, SaaS, WOA, Web services, PaaS and others. White Paper, Soalib Incorporated, 2009.
- [5] Buyya R, Calheiros RN, Li XR. Autonomic cloud computing: Open challenges and architectural elements. In: Proc. of the 2012 3rd Int'l Conf. on Emerging Applications of Information Technology (EAIT). 2012. 3–10. [doi: 10.1109/EAIT.2012.6407847]
- [6] Papazoglou MP, Traverso P, Dustdar S, Leymann F. Service-Oriented computing: A research roadmap. Int'l Journal of Cooperative Information Systems, 2008,17(2):223–255.
- [7] Zhang Q, Cheng L, Boutaba R. Cloud computing: State-of-the-Art and research challenges. Journal of Internet Services and Applications, 2010,1(1):7–18. [doi: 10.1007/s13174-010-0007-6]
- [8] Wooldridge M. An Introduction to Multi agent Systems. 2nd ed., John Wiley & Sons, 2009.
- [9] Mao XJ. Agent Oriented Software Engineering. Beijing: Tsinghua University Press, 2005 (in Chinese).
- [10] Serugendo GDM, Fokua N, Hassas S, Karageorgos A, Mostéfaoui SK, Rana OF, Ulieru M, Valckenaers P, Van Aart C. Self-Organisation: Paradigms and applications. In: Engineering Self-Organising Systems. LNCS 2977, 2004. 1–19. [doi: 10.1007/978-3-540-24701-2\_1]
- [11] Serugendo GDM, Gleizes MP, Karageorgos A. Self-Organisation and emergence in MAS: An overview. Informatica, 2006,30(1): 45–54.
- [12] Liao BS, Jin L, Gao J. PDC-Agent enabled autonomic computing: A theory of autonomous service composition. In: Proc. of the Agent Computing and Multi-Agent Systems. Springer-Verlag, 2006. 596–601. [doi: 10.1007/11802372\_65]
- [13] Sreenath RM, Singh MP. Agent-Based service selection. Web Semantics: Science, Services and Agents on the World Wide Web, 2004,1(3):261–279. [doi: 10.1016/j.websem.2003.11.006]
- [14] Chainbi W, Mezni H, Ghedira K. AFAWS: An agent based framework for autonomic Web services. Multi Agent and Grid Systems, 2012,8(1):45–68. [doi: 10.3233/MGS-2012-0185]
- [15] Sim KM. Agent-Based cloud computing. IEEE Trans. on Services Computing, 2012,5(4):564–577. [doi: 10.1109/TSC.2011.52]
- [16] Buyya R, Calheiros RN, Li XR. Autonomic cloud computing: Open challenges and architectural elements. In: Proc. of the 2012 3rd Int'l Conf. on Emerging Applications of Information Technology (EAIT). 2012. 3–10. [doi: 10.1109/EAIT.2012.6407847]
- [17] Puviani M, Frei R. Self-Management for cloud computing. In: Proc. of the Science and Information Conf. (SAI). 2013. 940–946. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6661855&matchBoolean%3Dtrue%26queryText%3DSelf-Management+for+cloud+computing>
- [18] Chen X, Zhang Y, Zhang XD, Wu YH, Huang G, Mei H. Runtime model based approach to managing diverse cloud resources. Ruan Jian Xue Bao/Journal of Software, 2014,25(7):1476–1491 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4457.htm> [doi: 10.13328/j.cnki.jos.004457]
- [19] Cao BQ, Li B, Xia QM. A service-oriented QoS-assured and multi-agent cloud computing architecture. In: Proc. of the Cloud Computing. Springer-Verlag, 2009. 644–649. [doi: 10.1007/978-3-642-10665-1\_66]
- [20] Paliwal AV, Shafiq B, Vaidya J, Xiong H, Adam N. Semantics-Based automated service discovery. IEEE Trans. on Services Computing, 2012,5(2):260–275. [doi: 10.1109/TSC.2011.19]
- [21] Yang S, Sailer A, Shaikh H. Hierarchical online problem classification for IT support services. IEEE Trans. on Services Computing, 2012,5(3):345–357. [doi: 10.1109/TSC.2011.3]

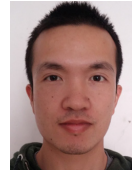
- [22] W3C recommendations OWL-S: Semantic markup for Web services. <http://www.w3.org/Submission/OWL-S/>
- [23] Klusch M, Fries B, Sycara K. OWLS-MX: A hybrid semantic Web service matchmaker for OWL-S services. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2009,7(2):121–133. [doi: 10.1016/j.websem.2008.10.001]
- [24] Hou F, Mao XI, Yin JW, Wu W. An agent-based autonomous management approach to dynamic services. In: Shi Z, Wu Z, Leake D, Sattler U, eds. *Proc. of the Intelligent Information Processing VII, Vol.432*. Berlin, Heidelberg: Springer-Verlag, 2014. 122–132. [doi: 10.1007/978-3-662-44980-6\_14]

#### 附中文参考文献:

- [9] 毛新军.面向主体的软件工程.北京:清华大学出版社,2005.
- [18] 陈星,张颖,张晓东,武义涵,黄罡,梅宏.基于运行时模型的多样化云资源管理方法.软件学报,2014,25(7):1476–1491. <http://www.jos.org.cn/1000-9825/4457.htm> [doi: 10.13328/j.cnki.jos.004457]



侯富(1987—),男,黑龙江林甸人,博士生,CCF 学生会员,主要研究领域为面向服务和云计算,自组织和自适应技术.



吴伟(1989—),男,硕士生,CCF 学生会员,主要研究领域为面向服务计算,面向 agent 软件工程.



毛新军(1970—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为 agent 理论和软件工程技术,自适应,自组织技术.