

集成 Agent 技术和面向服务计算技术:现状和展望^{*}

屈婷婷 毛新军

(国防科学技术大学计算机科学与技术系 长沙 410073)

摘要 Agent 技术和面向服务计算技术是目前计算机科学与技术领域中的二项重要技术。这二种技术都试图为互联网软件系统的开发提供新颖的技术途径,但是它们的关注点、所采用的技术手段、优势和不足各不相同。将这二种技术相互集成有助于发挥各自的优势,同时弥补各自的不足。近年来有关集成 Agent 技术和面向服务计算技术的研究引起了人们的关注和重视,它将为部署在互联网上的动态、开放系统的开发提供新颖、有效的技术途径。分析了集成 Agent 技术和面向服务计算技术的背景和动机,综述了该领域的研究现状和已有的成果,并展望了其面临的问题和挑战。

关键词 Agent 技术,面向服务计算,Web 服务,面向服务体系结构

Integrating Agent Technology with Service-Oriented Computing: State-of-the-Art and Prospect

QU Ting-ting MAO Xin-jun

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

Abstract Agent technology and service-oriented computing are two important techniques in the literature of computer science. Both of them attempt to provide effective ways to develop software systems deployed in Internet. However, the problems that they attempt to deal with, technology solutions, richness and weakness are actually various and different. Recently, many attentions are put on the integration of agent technology with service-oriented computing in order to deal with the open and dynamic applications in Internet. The background and motivation of such integration are analyzed and investigated, the state-of-the-art research is surveyed, the problems and challenges are discussed.

Keywords Agent technology, Service-oriented computing, Web service, Service-oriented architecture

1 引言

自上世纪 90 年代初开始, Internet 迅速而富有戏剧性地变成一个十分重要和强大的通讯、研究和商业的中介^[1]。随着应用的不断深入,海量信息涌入 Internet,这些部署在 Internet 上的应用和资源呈现出两个十分重要的趋势:

- 跨组织、技术多样性、异构性,需要进行充分的共享和综合利用;
- 复杂,但不仅仅是规模问题,动态、开放、不确定性、自适应和自组织等。

这些应用和资源的特点使得互联网软件系统的开发迫切需要新颖、有效的方法提供技术途径。当今备受 IT 界关注的面向服务计算 (Service-Oriented Computing, SOC) 技术可以有效地适应 Internet 上资源的其中一个特点。面向服务计算是一种新的软件架构体系,它的核心理念是在交互的软件成分之间,构建松耦合的协同软件体系。SOC 提供了一种方法来创建一种新的体系结构 SOA (Service-Oriented Architecture),它是一种用于构建分布式系统的方法,也是一种松散耦合、位置透明、协议独立的系统架构^[2]。从 SOC 的这些特性可以看出这项技术很好地适应了资源的跨组织、技术多样性、异构性等特性,并为之提供技术途径。

我们已经看到了 SOC 具有种种优点,但是也要看到它的不足之处:它无法有效地应对互联网环境的动态性和开放性。服务缺乏自主性,不具有自适应和自组织的特性,缺乏相互通

信的“理解力”。因此服务自身的这些缺点使它很难适应互联网的复杂、动态、开放、不确定等特点。而近年来一种新兴的技术——Agent 技术很好地为互联网的以上特性提供了有效的技术途径。

Agent 是分布式计算和人工智能结合的产物,是能够独立自主地感知环境、具有一定的智能性并可以在网络上进行自主移动的软件实体^[3],它横跨了一个特定的技术领域并且可以在动态、开放的环境下处理进行交互的算法^[1]。Agent 的自主性、感知性、交互性、主动性等特征无疑能有效地解决和应对系统的动态性、开放性、不确定性等特点。但是在如何有效地适应互联网的跨组织、技术多样性、异构性等特点方面还有待进一步的提高。

由此可见,这两项技术的优势具有互补性。Agent 的自主性等特点可以弥补 SOC 技术的不足,而 Agent 需要在异构平台间实现互操作以及松耦合等特性都是 SOC 最擅长的。在此背景下,将这两种技术进行集成是十分必要的,不仅是为了利用各个技术的优势,弥补其不足,而且因为集成之后充分地适应了 Internet 的发展趋势,为复杂的互联网软件系统的开发提供了新颖和有效的技术途径。

当前,在新的应用环境中,随着 Agent 技术的不断成熟和在业务需求驱动的促使下,将 Agent 技术和面向服务技术进行集成已成为面向服务领域一个重要的研究方向。在这十多年来在多 Agent 系统研究中的许多想法都被面向服务系统实现了,同时多 Agent 系统的进一步发展也成功地完善了面向

^{*} 本文得到国家自然科学基金(60373022)和霍英东优选资助课题(104029)的资助。

服务计算方法,两者结合得如此之深^[4]。

本文首先概述了集成 SOC 技术和 Agent 技术的背景和动机,分析了该方面研究的现状和成果。讨论了现有研究存在的问题和挑战,最后总结全文并阐述了进一步的研究工作。

2 研究现状

如今,面向服务的计算(SOC)已成为软件领域最热门的研究内容之一。关于 SOC 的研究涉及很多方面,Agent 技术可以对 SOC 中多个方面进行集成,本文将要讨论的集成的方面如表 1 所示。

表 1 Agent 技术与 SOC 技术集成内容表

研究对象	SOC 或 Agent 技术存在的问题/集成的原因	集成 Agent 与 SOC 技术的目的和动机	代表性工作
服务通信	1. SOAP 的性能成为 SOAP 的应用瓶颈。 2. 原 Agent 通信语言无法实现 Agent 之间的跨平台通信。	1. 利用 Agent 的交互性对 SOAP 进行有效的改善。 2. 利用 SOAP 具有的跨硬件平台,操作系统,编程语言和网络平台的高度互操作性。	如马建红的工作 ^[5]
服务发布和发现	1. 系统在运行时只能使用固定服务商提供的服务。 2. 服务客户端的开发者而不是请求者手动地进行查询注册例如 UDDI。 3. 发布的服务描述和服务发现的匹配是基于类似服务分类等传统方法。 4. UDDI 查找的效率和准确度有待提高。	1. 利用 Agent 的灵活性和自主性使系统在使用期间能灵活更换服务。 2. 从已发布的语义 Web 语言描述的服务里来寻找合适的服务。 3. 利用 Agent 技术提供更灵活的服务的选择和比较。 4. 利用 Agent 的智能决策能力优化 UDDI	如 Alberto Fern'andez 的工作 ^[6] 及 Fang Liu 的工作 ^[7]
服务组合	1. 组合服务的通讯方式是集中式的结构,服务间通信负担过重。 2. 服务组合需要合适的方法来描述服务调用的约束。 3. 自动服务组合中需要用到 Agent 技术进行推理。	1. 利用软件 Agent 的迁移、自主工作的特征实现分布式的 Web 服务组合。 2. 利用 Agent 灵活性和适应性来进行服务组合合法性验证。 3. 利用 Agent 的智能特征和问题求解能力。	如 Hui Wang 的工作 ^[8] 及 Mathieu Vall'ee 的工作 ^[9]
服务语义	Agent 需要互相理解的能力 ^[10] 。	通过语义 Web 服务的描述语言,Agent 程序可以更好地识别与理解信息源,并在 Agent 程序之间实现基于语义的互操作。	如 Michael Huhns 的工作 ^[11] 及 Paul A. Buhlerl 的工作 ^[12]
服务网络	资源的查找不智能,资源利用率有待提高。	利用 Agent 的自主性使资源的注册和查找更加智能化。利用 Agent 对环境的感知性提高资源利用率。	杨欣、语义 Web 服务安全研究 ^[13]
服务安全	对服务的认证和服务的运行监控存在问题。	利用 Agent 的自主性,提高服务的安全性。	

下面就服务通信、服务发布和发现、服务组合、服务语义这 4 个方面展开具体分析。

2.1 服务通讯

Agent 技术与面向服务技术两个不同领域的连接,首先可以考虑两个领域的通信方面是否能进行技术的互补,不少研究者针对 Agent 和 Web 服务集成的通信方面进行了深入的研究,包括通信机制,通信语言以及信息建模等等。

在通讯机制方面,Web Service 支持的简单对象访问协议 SOAP(Simple Object Access Protocol)具有跨硬件平台,操作系统、编程语言和网络平台的高度互操作性。但是 SOAP 消息处理与消息传输的性能问题成了 SOAP 的应用瓶颈。在

这方面,我们可以考虑在 SOAP 机制上集成 Agent 技术来改善 SOAP 的应用性能。文献^[14]就是把这两项技术进行了结合,将 SOAP 应用中的服务发现、消息处理和消息传输 3 个阶段分别赋予不同的 SOAP Agent 去实现。通过 SOAP Agent 之间的协作,使用更好的技术方法,改进了 SOAP 的性能。这样的集成也为 SOAP 在可靠性、安全性等方面的扩展提供了灵活的机制。

在通信语言方面,主要是对 Agent 通信语言(ACL)进行 Web 服务协议封装。Agent 通信语言是实现多 Agent 系统交互与合作的基础。SOAP 作为一种新的与平台无关的通信协议,可以实现异构多信息的交换,使得各孤立系统的通信变得容易。采用 SOAP 协议来实现 Agent 之间的 KQML 通信,解决了跨平台的多 Agent 系统互连的接口问题。这种方法在实际系统应用中也取得了显著效果。文献^[5]中结合 KQML 语言提出的基于 SOAP 协议的 Agent 通信语言,比较清楚地表达了这一实现思想。基于 SOAP 协议的 KQML 通信语言的消息结构由 3 层组成:SOAP 层、KQML 层和内容层,如图 1 所示。

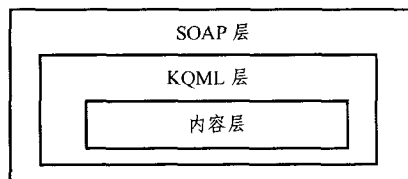


图 1 基于 SOAP 协议的 KQML 消息层次结构图

2.2 服务发布和发现

传统的服务发布和发现模型基于 Web Service 体系结构,该结构主要包含 UDDI, SOAP, WSDL 这 3 个部分。如今,Web 服务存在一些问题如:在构建系统的过程中绑定了大量固有的服务,致使系统在运行时不能灵活更换服务,UDDI 的作用没有充分发挥。在这种情况下,一旦某一服务中断就可能造成整个系统的停顿,如果使用其他服务商提供的服务代替,则需对原系统进行修改,造成系统的不稳定。

为了解决这个问题,使发现的服务更加灵活地适应用户的需求,文献^[15]构建了一种 Agent 的组件,可以达到以下目的:

(1) 基于 SOA 方式构建的系统在使用期间能灵活更换目前所使用的服务,使更多的服务能够参与竞争,提高系统的服务质量并扩展系统。

(2) 采用改进的服务机制后,系统的构建、使用方式不发生改变,提高了系统的可靠性稳定性。

类似这样一类 Agent 组件可以直接在 UDDI 中驻留或者独立构建。问题在于:在 UDDI 中驻留 Agent 组件方案中,请求者作为第三方并不能直接干预服务的选择,以致当用户有特殊需求例如安全问题、信任问题时不能选择令人满意的服务^[15]。因此该 Agent 组件的性能和满足用户需求的能力有待进一步的完善。

UDDI 存在的问题还有:是服务客户端的开发者而不是请求者手动地进行查询注册;当前的 Web 服务标准使得发布服务描述和服务发现的匹配是基于类似方法的署名或服务分类等传统方法,不能提供更灵活的服务的选择和比较;在库里的服务是典型无组织的,致使 UDDI 作用不能充分发挥等等。对于前两个问题,一部分研究者都尝试着做出解决:通过语义

网匹配处理请求,从用语义 Web 语言描述的已发布的服务里来找合适的服务,这是一种很好的解决办法。对于未解决的遗留问题,我们可以试着从其它的角度去解决。

一些研究者热衷于对 UDDI 做优化,试图向用户提供动态和更加有效的服务。例如可以依据 Agent 的智能决策能力来对 UDDI 的查询结果做进一步的处理,经过筛选后的 Web Service 的数量大大减少,UDDI 查找的效率和准确度得到了提高。

另一方面,随着 Web 上 Agent 应用的增多,Agent 体系结构也呈现出许多与 Web 服务相同的特征^[16]。因此把 Agent 服务描述成 Web 服务,在 Agent 体系结构上利用 Web 服务的一系列服务发布和发现协议,可以很好地解决跨站点 Agent 之间的信息资源共享问题。

2.3 服务组合

在面向服务的方法里,服务有两个抽象级别。一方面,基本的单个服务是一些基础的功能,而另一方面,组合服务累积了一些功能集合作为高级应用^[1]。对于单个服务而言提供的功能是有限的,因此有必要将共享的服务组合起来解决更复杂的问题,满足用户的实际需要。

服务组合从组合的生成方式来分有两大类:静态组合和动态组合。静态组合是指请求者在组合计划实施前创建一个服务组合流程来完成的任务,而动态组合意味程序根据用户在高层上给出的任务描述自动地选择、绑定和组装适当的服务,自动地创建服务流程。由此可见在实际应用中要求服务的执行是具有动态性的,并且能够达到自动化,同时服务的动态组合也为解决信息共享和应用协作的瓶颈问题带来了新的方案。

服务组合在 Web 服务里广泛地执行。现在用于描述 Web 服务的是 WSDL,它并不能很好地表达 Web 服务的语义信息,并且所描述的是静态的 Web 服务。于是语义 Web 的出现就变得十分重要,它使得 Web 服务的动态组合变成可能。关于基于语义的 Web 服务的动态组合,许多研究者都提出了不同的方案,实现了一些动态组合的系统原型,在这里我们就不做详细的分析。而语义 Web 的实现很需要 Agent 提供基础技术上的支持,这一部分在服务语义部分会进行讨论。

从基本的服务描述开始,人们提出诸多标准用于执行 Web 服务的组合。这些标准有 BPEIAWS,WSFL 等。这些标准提供语法用于确定一个服务如何同其他服务协同工作,但是它们实际导致了一种集中式的通信结构。中心必须自始至终地参与服务组合执行的全过程,并且所有数据都要经过中心实体,给网络传输带来了较大的负担^[17]。为解决这个问题,文献^[17]提出的基于移动 Agent 的服务组合模型提供了一种可行的解决方案,它利用软件 Agent 迁移、自主工作的特征实现了分布式的 Web 服务组合。方法是让软件 Agent 携带数据迁移,负担了服务间通信,降低了数据通信量,实现的是 P2P 的通信方式而非集中式的结构。

在基于 Agent 的服务组合研究中,研究人员大部分是在服务组合流程的建模过程中集成了 Agent 技术,有些则利用 Agent 来进行辅助的工作,例如用描述 Agent 行为的文件来描述服务调用的约束以此来服务组合流程的合法性验证等等。根据 Agent 的特点提出基于 Agent 的 Web 服务组合的设计使得整个 Web 服务更加具有动态性和适应性^[18]。

2.4 服务语义

如上所述,WSDL 不能很好地表达 Web 服务的语义信

息,因此我们会用语义 Web 来进行动态的服务发现和组合。基于语义服务描述是当前服务描述的一种扩展,其中信息被注以明确定义的含义,有助于计算机之间及其与人之间的协同工作。

语义 Web 服务的描述语言 OWL-S(Web Ontology Language for Services)是一种描述 Web 服务本体的语言,也是具有显式语义和机器可理解的标记语言(markup language),用来描述 Web Service 的属性和功能。OWL-S 由 3 部分组成:Service Profile,Service Model 和 Service Grounding,其中前两部分用于 Agent 完成服务的语义注册、查询和组合。另外,由美国国防部 DAPRA 机构主持开发的 Web Agent 标注语言 DAML,其目标是开发一个以机器可读的方式表示语义关系和与技术相容的语言,另外还开发出一套工具与技术,使得 Agent 程序可以识别与理解信息源,并在 Agent 程序之间实现基于语义的互操作。通过 DAML 语言提供更多的表达方式来通过本体对服务进行描述。

由此可见,语义 Web 提供了一种公共的语言作为软件 Agent 之间语义信息转换的公共语言^[1]。采用语义 Web 服务作为详细说明 Agent 和 MAS 行为的方法也在 P. Buhler 和 J. M. Vidal 的两篇文章中提到^[19]。不光如此,对实现语义 Web 的迫切需求理由还有很多:Web 服务的丰富的描述可以改进发现和组合;公共的接口可以简化异构系统的集成等等^[1]。

显然,语义 Web 也需要 Agent 领域的包含和促进,这两个领域是密切联系的。语义 Web 为两个领域以后的基础研究和基础研究之上的 Agent 的广泛应用提供了丰富的滋生土壤^[1]。

2.5 其他

Agent 技术与 SOC 技术集成方面还有很多,例如 Agent 技术对服务安全和服务网络的改进等。

在 SOC 为客户端提供服务时必须考虑通信实体在通信过程中所面临的典型安全问题,包括鉴别、授权、机密性、数据完整性、不可抵赖性和可用性。通常 Web 服务存在的安全问题,SOC 也同样存在。例如,恶意内容攻击试图强制一个 SOAP 节点(服务器)去做其它的事情等等。

对此,文献^[13]提出将 Agent 安全构件注入到服务之中来实现服务的安全性。Agent 安全构件不仅具有通讯、交互能力,而且还有安全通信能力(比如拥有一些凭证或者能够使用一些安全协议)。利用 Agent 安全构件,可以在不改变原有服务的基础之上提供在服务动态交互情形下的服务安全保障。以下是基于 Agent 安全构件的服务通讯图。

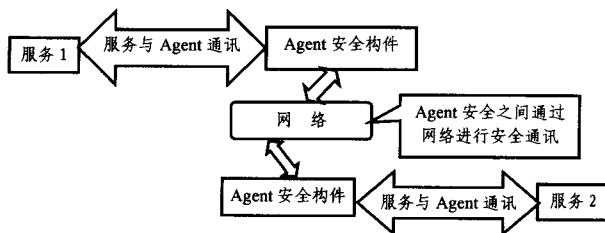


图 2 基于 Agent 安全构件的服务通讯图

服务网络是一个动态交互过程,它提倡非集中控制资源、自治性、开放统一的标准,因此非常适合商业应用的需求。但是网格服务在现有的企业使用之中,还无法做到“像提供水、电等公用设施一样提供运算力”,因为要调用和集成网格服务非常复杂。服务网络存在的这些缺陷和不足之处正是 Agent

技术擅长的,可以利用 Agent 的智能性、自治性和并行处理能力,提出运用 Agent 技术集成网格服务,以实现服务调用的自动化和灵活性。

3 问题和挑战

3.1 技术方面

Web 服务和 Agent 技术都是独立发展起来的技术,因此集成的主要障碍是没有一个 Agent 技术提供的支持是当前 SOC 设计者正好完全想要的,如果想要利用某些技术,则必须作一些改造。但是集成过程中必须考虑到 Agent 的性能,Agent 推理和交互有时被认为消耗资源且在普遍约束的环境下不合适^[9]。

SOA 发展的一个制约是技术的不足:还无法支持事务的最高可靠性,跨业务域的安全如何保障? 服务如何高效组合? 语义方面,如何定义事务和数据的业务含义? 这种分布式的处理模式是否能满足性能上的需求? 研究人员通过各种方式集成 Agent 技术,使一部分问题得到了相应的改善,而另一些则没有很好地解决。因此这些技术的不足在集成之后照样存在并且影响了集成的发展。

技术上来说,从两项技术单方面的优势可以看出,集成还有更广阔的空间。当然在具体的集成过程中还遇到了许多问题:

服务在互操作方面会遇到两个障碍:不协调的信息模式和不同服务提供者交互协议的错配。基于语义 Web 上发布的共享服务能力和利用协议的动态可接受语义描述为解决这个障碍提供一些可能的方法^[11]。

从基本的服务描述开始,人们提出诸多标准用于执行 Web 服务的组合,例如 BPEIAWS 等。这些标准提供语法用于确定服务如何同其他服务协同工作,但是它们实际导致了一种集中式的通信结构。中心必须自始至终地参与服务组合执行的全过程,这样给网络传输带来了较大的负担。为解决这个问题,文献^[17]提出的服务代理提供了一种可行的解决方案。

在基于 Agent 的 Web 服务中 Web 服务的相容匹配机制十分重要。Web 服务的供需方在描述 Web 服务时往往是相互独立的,采用不同的 Agent 通信语言,参照不同的应用域本体论,有时采用的术语也会有差异。因此搜索满足需求的 Web 服务只能依赖相容匹配机制。因此为实现相容匹配需解决以下 3 个问题:消除 Agent 通信语言间的差异,比较应用域本体论间的相似性和应用域术语间的相容性。在这方面,有些研究者提出了基于本体论的相容匹配策略,在一定程度上提高了相容匹配机制的精确性。

把 SOC 系统工程化,它就是一个发现和匹配合适的服务的过程,计划服务的交互,并在运行时组合服务^[10]。由于 WSDL 缺乏语义信息,即使可以提供足够的信息来调用一个服务,但如果使用 WSDL 描述 Agent 服务,那么该服务如何工作以及服务状态都不会很清楚。UDDI 试图通过预定义服务模型来询问服务提供者,对所提供的服务进行分类来解决该问题,但 WSDL 和 UDDI 的结合仍然不能有效地进行服务发现。同时 UDDI 也存在许多没有很好解决的问题,如:是服务客户端的开发者而不是请求者手动地进行查询注册;在库里的服务是典型无组织的,致使 UDDI 作用不能充分发挥等等。

在自动服务组合中需要用到 Agent 工具进行推理,因为

这样有助于为服务组合流程建模。但是,在实际中自动定义一个目标还是很困难的,所以这种组合方式还在研究中。

3.2 应用方面

应用瓶颈是 SOA 发展的另一个制约,当然也在一定方面制约了两项技术的集成的发展。把两项技术集成之后的应用需求是否得到扩大,这还很难说。因为现在的商业应用整体上还处于数据和应用的整合阶段,离按需的企业间协作尚有一定的距离,从而导致难以找到具有实际意义的应用场景大量需求的案例。

结束语 目前,SOC 的重点还是停留在单个服务的执行上,而没有停留在服务如何选择和服务如何协作来完成更高水平的功能的重要问题上。因此,把工作重心放到运用 Agent 技术良好的交互性和协作性使之在 SOC 技术中有所体现是很有意义的^[20]。

当前多 Agent 和面向服务的系统都根据它们的组件之间固定的协议来操作,但是在许多应用领域中这是不确定的。因此需要在商议和交互协议上做些额外的工作,如系统必须对用户显示出交互的规范并且提供可发现的交互和协议等。我们必须发展协作,发展建模和指定交互目标的新方法;理解团队工作和解决协作问题以及当面对各种各样的异常时如何建造一个更加具有鲁棒性的面向服务系统;监控具有自己的目标和对对象的异构组件组成的大规模系统的行为^[4]。另外,面对日益复杂的 Web Service 协议族,我们能采取什么措施进行改善也是值得思考的问题。

本文对 Agent 技术和面向服务计算技术集成研究进行了综述和分析,并对关键问题提出了质疑和挑战。毋庸置疑,未来软件将会朝着智能化和分布化的方向发展。将 Agent 技术具有的智能性与服务具有的完全开放、松散耦合、标准协议规范和高度可集成能力相结合,不仅推动面向服务技术向一个新的方向发展,同时也是 Agent 应用的一个发展方向。虽然两个技术的集成方面还有许多没有解决的问题需要进一步的研究和探讨,但是通过对现状的分析我们已经看到了一些成果。现在 Agent 技术与面向服务技术集成的研究还处于初步阶段,从两项技术单方面的优势可以看出,集成还有更广阔的空间。同时,希望看到在实际的企业业务中集成技术会有所应用,毕竟在业务需求驱动的促使下,集成技术会更加成熟和发展。

参 考 文 献

- [1] Luck M, McBurney P, Shehory O, et al. Agent Technology Roadmap -A Roadmap for Agent Based Computing. AgentLink III, 2005
- [2] Cabri G, Leonardi L, Puviani M. Service-Oriented Agent Methodologies
- [3] 毛新军. 面向文体的软件开发[M]. 北京:清华大学出版社, 2005:11-15
- [4] Huhns M, Munindar, Singh P. Research Direction for Service-Oriented Multi-agent System. IEEE INTERNET COMPUTING, 2005:11-12
- [5] 马建红,王万森,卫权岗. 基于 SOAP 协议的 Agent 通信语言的研究[J]. 微机发展, 2005, 15(6):39-41
- [6] Fernandez A, Vasirani M, Caceres C', et al. Artificial Intelligence Group. A Role-Based Support Mechanism for Service Description and Discovery

(下转第 42 页)

表4 $r=1$ 实验结果

Total transaction times	1000	5000	10000	15000	20000
Transaction failing times	199	904	1095	1132	1133
Transaction success rate(%)	80.1	81.92	89.05	92.45	94.34

由上述实验结果可知,当总交易数达到1000,即平均每个节点完成一次下载时,交易失败的次数接近恶意节点数与其行骗比率 r 的乘积。随着总交易数的增加,交易失败次数的增加量越来越接近零。同时,随着总交易数的增加,交易成功率也不断的增加。表4显示了即使恶意节点以 $r=1$ 的比率行骗,当交易量达到20000,即平均每个节点完成20次下载时,交易成功率就已经达到了将近95%。

此外,恶意节点表现恶意行为的比率 r 越大,在起初阶段的交易成功率越低,但是随着总交易数的增加,交易成功率增加得越快,发现恶意节点的速度越快,交易失败次数增加得越慢。

可见,本实验不仅仅显示了本模型的交易成功率的增加幅度,从交易失败的次数可以看出本模型对恶意行为有很强地抑制作用。交易量越大,恶意节点行骗的机会越少。

5.2 孤立恶意节点的有效性

本节将通过实验来证明本信任模型对恶意节点的孤立作用。在本实验中,每个节点都以相同的频率 q 发送请求,当总交易数达到20000时,实验结束。假定恶意节点每次的行为都是恶意的,即 $r=1$,其它参数值为表1中给定的值。我们来分析一下恶意节点成功下载次数和总交易次数的变化关系。

表5 实验结果

Total transaction times	1000	5000	10000	15000	20000
malicious peers' download times	206	974	1486	1760	1846
The rate of malicious peers' download times to total transaction times (%)	20.6	19.48	14.86	11.73	9.23

由表5可知,当总交易数为1000次,即每个节点平均完成一次下载时,恶意节点的下载次数占总交易数的20.6%,与网络中恶意节点数占节点总数的比值20%相近。当交易总量达到20000次时,恶意节点的下载次数占总交易数的比值不到10%,并且在最后的5000次交易中,恶意节点的下载次数仅有86。可见随着交易量的增加,恶意节点的身份逐渐

被暴露,导致其下载文件越来越艰难,同时由上一节实验分析知总交易数越大,恶意节点越难行骗,所以恶意节点将会被快速地孤立起来。

结束语 在P2P网络中,文件共享系统得到了广泛的应用。在本文,针对文件共享系统中节点的欺骗和自私行为,我们提出了一种基于访问控制的信任模型。通过实验结果及分析,这种信任模型在下载成功率、对恶意节点的孤立以及激励文件共享等方面都有很大的优势。

参考文献

- [1] Park J S, Hwang J. Role-based Access Control for Collaborative Enterprise in P2P Computing Environment. SACMAT, Jun. 2003
- [2] Resnick P, Zeckhauser R. Trust among strangers in Internet transactions; Empirical analysis of eBay's reputation system // NBER Workshop on Empirical Studies of Electronic Commerce. California, 2000
- [3] Cornelli F. Choosing reputable servants in a P2P network // Proceedings of the Int'l World Wide Web Conference. Hawaii, ACM Press, 2002; 441-449
- [4] Khambatti M, Dasgupta P, Ryu K D. A role-based trust model for Peer-to-Peer communities and dynamic coalitions // Cole JL, Wolthusen SD, eds. Proc. of the 2nd IEEE Int'l Information Assurance Workshop. New York, IEEE Press, 2004; 141-154
- [5] Wang Yao, Julita V. Bayesian network trust model in peer-to-peer networks // Moro G, ed. Proc. of the 2nd Int'l Workshop on Agents and Peer-to-Peer Computing. Berlin, Springer-Verlag, 2004; 23-34
- [6] Zhang Zhen, Wang Xiao-Ming, Wang Yun-Xiao. A P2P global trust model based on recommendation // Proceedings of the 4th International Conference on Machine Learning and Cybernetics. Guangzhou, 2005; 3975-3980
- [7] Kamvar S D, Schlosser M, Garcia-Molina H. Eigenrep: Reputation management in p2p networks // Lawrence S, ed. Proc. of the 12th Int'l World Wide Web Conf. Budapest. ACM press; 123-134
- [8] Xiong Li, Liu Ling. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities // Proc. IEEE Trans. Knowledge and Data Engineering. 2004, 16(7): 843-857
- [9] Liu Fang, Yao Li, Zhang Weiming, et al. A Conceptual Model of Agent Mediated Web Service [C] // 2004 IEEE International Conference on Services Computing, 2004. (SCC 2004). Proceedings, 2004, 9: 638-642
- [10] Hui Wang, Yuhui Zhao, Deguo Yang, et al. An Agent-Based Services Composition Framework for Ubiquitous Media
- [11] Vall'ee M, Ramparany F, Vercouter L. A Multi-Agent System for Dynamic Service Composition in Ambient Intelligence Environments. IEEE INTERNET COMPUTING, 2005; 11-12
- [12] Huhns M N. Software Development with Objects, Agents, and Services
- [13] Huhns M, Munindar, Singh P. A Semantic Web Services Architecture. IEEE INTERNET COMPUTING, 2005; 9-10
- [14] Paul A, Buhler1, José M. Vidal2. Semantic Web Services as Agent Behaviors
- [15] 杨欣, 沈建京. 语义 Web 服务安全研究. 计算机科学, 2007, 34
- [16] 马长东, 徐伟, 李京. 一个基于 Agent 的 SOAP 应用框架. 计算机工程与应用, 2004(1): 148
- [17] 白伟华, 苏卓夫, 李吉桂. 服务代理在面向服务的体系结构中的应用. 计算机应用与软件, 2006, 23(11)
- [18] Huhns M N. Agents as Web Services. IEEE INTERNET COMPUTING, 2002; 7-8
- [19] 戚玉松, 钱柱中, 是湘全. 基于 Agent 的 Web 服务组合研究. 南京理工大学学报, 2006, 30(3)
- [20] Lu Hongen, Chhabra M. A Methodology for Agent Oriented Web Service Engineering
- [21] Bozzo L, Mascardi V, Ancona D. COOWS: ADAPTIVE BDI AGENTS MEET SERVICE-ORIENTED COMPUTING
- [22] Huhns M, Munindar, Singh P. Service-Oriented Computing: Key Concepts and Principles. IEEE INTERNET COMPUTING, 2005; 1-2

(上接第8页)