

自律计算系统及其关键技术研究

王振东 王慧强 冯光升 吕宏武 陈晓明

(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

摘要 自律计算系统是目前的一个热点研究方向,其目的是采用技术管理技术来降低用户使用系统的复杂性。目前,自律计算理论已广泛应用于网络安全、自动控制等多个领域并取得初步成功。基于学术界已有的研究成果,对自律计算概念、框架结构以及自律计算系统实现的关键技术进行了综述,以为自律计算领域研究人员后续研究提供指导。

关键词 自律计算,自律计算系统,上下文感知,机器学习

中图分类号 TP393 **文献标识码** A

Research on Autonomic Computing System and its Key Technologies

WANG Zhen-dong WANG Hui-qiang FENG Guang-sheng LV Hong-wu CHEN Xiao-ming

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)

Abstract Autonomic computing is an emerging research hotspot, and the aim is to decrease system complexity for users by means of technologies managing technologies. Currently, autonomic theory is used in network security, autonomic control fields for some years and gets initial success. However, there is not mature theory for autonomic computing at home and abroad. This paper summarized the conception, architecture and key technologies for autonomic computing systems, to guide the follow-up studies for researchers on autonomic computing.

Keywords Autonomic computing, Autonomic computing system, Context awareness, Machine learning

1 前言

随着计算机及网络技术的迅猛发展,新技术诸如网格计算、普适计算、并行计算等的实际应用,网络系统及应用环境的日渐复杂,仅仅通过技术人员来管理这些开放、动态、异构的信息系统变得越来越困难。在这样的背景下,以解决日益突出的分布式系统管理复杂性和提高系统可靠性、可用性及容错能力为目标的自律计算技术已成为当前的研究热点。其基本思想是“用技术管理技术”,即把有限的人力资源集中于高层的策略制定上,而将其它诸如系统管理、配置、优化、保护等操作由系统自主完成,以节约成本。自IBM于2001年提出自律计算(Autonomic Computing)概念以来,学术界和产业界都对其进行了大量研究,也取得了一些阶段性的成果,如DARPA的SPS^[1],NASA的ANTS^[1]分别实现了系统的自我愈合功能和基于纳米蜂群的自律系统;Charles. G. Vallerand^[2]等人将自律理论应用于普适环境;Caminero^[3]等人提出一种应用于网格计算的自律网络感知调度结构;Jia Rao^[4]等提出一种让虚拟机自我配置的强化学习方法。国内研究也取得了一些进展,中国科学院计算技术研究所863计划的支持下对自主计算的理论和技术进行了深入研究^[5],并取得发明专利一项。臧铖^[6]等提出一种基于状态的通用自主计算模

型;史忠植^[7]等提出一种自主计算系统的建模方法,实现了一个基于主体的自主计算环境ACE并提出一种服务描述语言SDLSIN。但是,从现有的研究成果看,自律计算理论不够系统和完善,应用范围较为狭窄,缺乏统一通用的框架体系支撑,还没有出现具有突破性的研究成果。因此,对该理论进行持续、深入的研究是必要的。

本文作为自律计算理论及技术研究综述,对自律计算理论研究及实现方法进行了阐述。首先介绍了自律计算的相关概念及框架结构,然后对自律计算系统的两种体系结构进行了详细讨论,最后分析了实现自律计算的相关技术。结论中对自律计算研究目前尚存在的问题进行了说明,并对还需要继续研究的方向进行了展望。

2 自律计算概念及框架结构

2.1 相关概念

自律计算概念由IBM于2001年率先提出,该思想借用了生命智能中自律神经系统原理,目的在于在增加系统效能的同时减少用户使用及管理的复杂程度。关于自律计算的定义,目前还没有准确的描述,但是学术界认为满足自律计算的系统应该具备“四自”属性,即“自配置(self-configuration)”、“自恢复(self-healing)”、“自优化(self-optimization)”和“自保

到稿日期:2012-09-15 返修日期:2013-01-11 本文受国家自然科学基金(60973027),中央高校基本科研业务费专项资金(HEUCF100601, HEUCFZ1213),教育部高等学校博士点基金(20102304120012),黑龙江省自然科学基金(F201037, ZD201102),黑龙江省博士后基金(LBH-210204),哈尔滨市科技攻关及创新人才基金(011AA1CG005, 2011RFQXG007, 2011RFQXG035)资助。

王振东(1982—),男,博士生,主要研究方向为自律计算、认知网络, E-mail: wangzhendong@hrbeu.edu.cn; 王慧强(1960—),男,博士,教授,主要研究方向为自律计算、网络安全。

护(self-protection)”。所谓“自配置”就是系统根据环境的当前状态进行初始配置,以及根据环境的变化进行自我重新配置。“自恢复”是指系统能够对软件故障进行实时监测,做出诊断,在不妨碍系统正常工作的前提下,自动采取相应的动作从常规和意外的灾难中恢复。“自优化”是指系统能够根据当前环境的变化、用户的需求和负载情况进行动态的资源重组和任务分配。“自保护”指系统能够及时发现将会对系统产生危害的事件并做出反应。

采用自律计算理论的系统为自律计算系统,自律计算系统是一种可指导的、状态觉察的、整体自适应的计算机系统^[9]。但这只是一种理想状态,在实际系统中,自律计算的各种特性和功能实现往往是阶段性的,通常在最后阶段才能实现真正意义上的“自律计算”。

2.2 自律计算结构

自律计算单元(Autonomic Computing Element)是自律计算系统的基本组成部分。现阶段对自律计算的研究都基于 IBM2003 年提出的一种自动控制循环模型。它被称为 MAPE-K(Monitor, Analyse, Plan, Execute, Knowledge)循环。在该模型的基础上,考虑到外部策略对自律单元的影响,在 MAPE-K 模型的基础上添加了策略模块,使得 IT 管理人员方便对自律计算单元进行管理。其结构如图 1 所示。

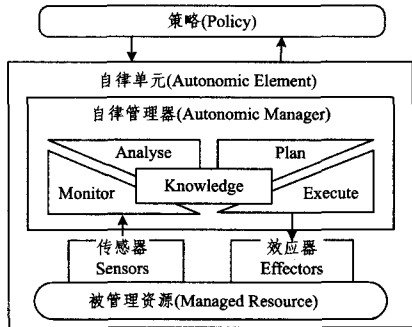


图 1 自律计算单元模型

自律计算单元作为自律计算系统的基本组成模块,能够利用自身的控制结构从环境中收集信息并进行决策。自律计算单元由自律管理器(Autonomic Manager)和被管理资源(Manged Element)两部分构成,自律管理器的作用是从被管理资源中获取信息并通过一系列的操作后,得到相应的动作,交由效应器去执行。自律管理器又包括监视(Monitor)、分析(Analyse)、规划(Plan)、执行(Execute)及知识库(Knowledge)5 部分。其中监视器从被管理资源中收集信息,也就是从外部环境中收集信息。例如,如果被管理资源代表的是 Web 服务器,则信息包括客户请求的应答时间、网络和磁盘特性、CPU 和内存效能等信息。分析组件用于对系统内部环境进行复杂性分析,使得自律管理器能够实时了解系统内部运行状态,以便对未来情况做出预测,及时采取相应策略。规划组件根据监视组件和分析组件从外部环境和内部环境得到的数据信息,以及先前制定的策略产生一个能够达到目标动作的序列,并交由执行组件将动作指令下达给效应器从而调整被管理资源的状态。上述 4 个组件所产生的动作都基于知识库中的知识。因此,如何有效获取知识是一个值得研究的方向。目前知识库中知识的获取主要有两种途径:机器学习和人工

策略注入。在系统运行的过程中,它们必须具备能够及时更新知识使得系统能够应对各种突发情形的能力。

上述模型中的策略模块用于指导自律计算单元的行为。一般策略又分为义务型(obligation)策略和授权(authorization)型策略,分别规定策略的执行者“应该/不应该做什么”和“允许/不允许做什么”。但是在自律计算背景下,策略有更宽松的定义。Kephart 等人从 AI 的角度把自律计算系统中的策略分为 3 种类型:动作策略、目标策略和效用函数策略,要利用策略必须对系统可能出现的各种状态进行形式化定义,根据不同状态采用相应策略。综合各方面的情况提出一组通用的策略并不容易,因此策略必须能够人为地动态添加或删除,还能够对已有策略进行修改,这些可以通过人机界面来实现。

2.3 自律计算系统

自律计算系统由自律计算单元组成,一个自律计算单元必须对周围环境有足够的了解才能够实现自配置、自修复、自优化和自保护的“四自”功能。也就是说,自律计算单元只有在高度自省(self-awareness)的情况下,才能够实现高度自主。

对自律计算系统的研究国内主要集中在网络管理系统、服务器管理系统、Web 服务管理等领域。西北工业大学的刘全中^[10]等将自律计算系统应用于服务器的虚拟化,实现了虚拟服务器资源的智能协调。刘文洁^[11]等提出一种面向服务器集群的自律计算模型,其通过系统部署的自动化及排除操作错误等来削减应用成本,从而实现系统的自我管理。徐云平^[12]构造了一个基于策略的网络故障管理系统,实现了管理任务的动态配置,提高了网络故障管理的灵活性。P. Martin^[13]等人将自律计算应用于 WSDM(Web Services Distributed Management)系统,并评估了实验结果。在前人研究的基础上,从网络体系结构层面,我们抽象出一种自律计算系统的层次模型,其体系结构如图 2 所示。

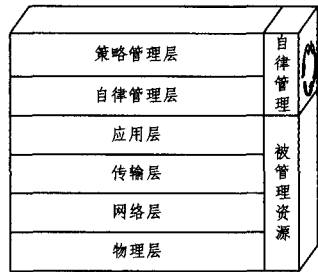


图 2 自律计算系统网络层次结构

从图 2 中可以看出,自律管理层位于网络体系结构的应用层之上,可以在不改变现有网络体系结构的基础上对网络上的各种服务进行自律管理。这种结构的优点是易于实现,方便开发各种应用型自律计算系统。

根据自律计算系统不同的应用方向,其体系结构也有所区别。这里着重说明自律 Agent 计算系统和常规自律计算系统。

2.3.1 自律 Agent 计算系统

自律 Agent 计算系统由多个自律 Agent 组成,目的在于将担负不同任务的自律 Agent 联合起来形成合作或者协作的群体,以实现整个系统的全局自适应,这与传统 MAS(Multi-

Agent System)具有一定相似性。因此,可以采用 MAS 系统的部分理论。与传统 MAS 理论不同,自律 Agent 计算系统除了具备传统 MAS 的组织结构、通讯方式外,还有自己的特点。首先自律 Agent 具有自主决策能力,它们的自律管理器通过对内、外部环境的感知,综合策略及知识库中的知识做出决策,而 MAS 通常具有一个中央处理单元,在对各 Agent 采集来的数据进行汇总分析后,返回 Agent 采取相应动作,其采用分布采集、集中处理、分散执行的运行方式。其次,自律 Agent 计算系统在运行过程中通过整体自适应来达到系统平衡,并且每个自律计算单元都能够得到外部策略的指导。最后,自律 Agent 系统的各 Agent 之间的地位平等,它们之间交流的信息多是高层策略。参照 MAS 的组织结构,抽象出自律 Agent 计算系统组织形式,结构如图 3 所示。

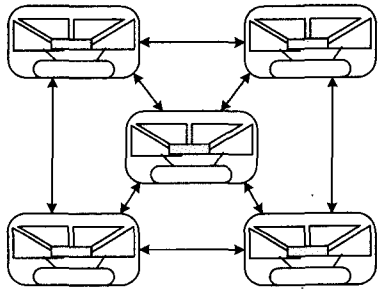


图 3 自律 Agent 计算系统结构

目前,业界对 MAS 的研究较为深入,很多理论也都可以用于自律 Agent 计算系统架构及相关理论研究。在已有研究成果中,Gerald Tesauro^[14]等提出一种基于 Multi-agent 系统的、用于自主计算的分散式体系结构(Unity),实现了许多期望的自主系统行为,包括目标驱动的自我组合、自我修复和实时的自我优化。不过,该文献中没有具体说明策略如何表示,以及如何通过策略来影响效用函数,指导 Agent 的行为。Holger Kasinger^[15]等将 MAS 应用于 OCS(Organic Computing System),该系统的 Self-X 属性可以从系统的业务逻辑模块独立开发,并确立了各种实现的基础,但是系统性能未能得到验证。

自律 Agent 计算系统结构简单,但实现起来有较多困难,特别是 Agent 学习和 Agent 之间通讯、协作、合作、协商等方面,还需深入研究。

2.3.2 常规自律计算系统

与自律 Agent 计算系统不同,常规自律计算系统通常由一个自律管理模块及被管理资源组成。自律管理模块对于被管理的各种资源,如网络资源、服务器资源、操作系统等进行统一管理。这类自律计算系统实现起来比较简单,系统从被管理资源中获取信息,通过自律管理模块的处理分析,发出要执行的动作,从而免去了自律 Agent 计算系统中的 Agent 之间的通讯、协作等一系列的复杂交互。其缺点是系统的开放性较弱,被管理资源较单一,与未来复杂系统的发展方向(各种资源有机交互合作,相互联系)存在差异。常规自律计算系统结构如图 4 所示。

目前对于常规自律计算系统的研究较为深入,其在很多领域都已经得到了实际应用。服务器管理方面,樊星^[16]利用时序模型算法和事件分类匹配算法对服务器系统多个节点的

事件和系统参数进行分析与预测,结合策略库技术,构建了一个服务器系统管理的自律计算模型,但是其有效性未能验证。刘文洁^[11]提出了一种基于自律计算的集群部署和管理模型,它能够对集群服务器中的节点进行快速的应用软件部署,并根据服务器负荷来自动切换节点,实现系统的自我配置。缺点是系统自适应能力不强,不能及时根据环境变化自我调整。在网络管理方面,江国朝^[17]实现了一个基于 Policy 的自律网络资源管理系统,实现了自律计算的基本功能,但是系统的安全性考虑不够,自我保护的能力不足。在操作系统管理方面,Michael Engel^[18]通过在操作系统内核部署动态的 Aspects,实现了操作系统的自我管理。

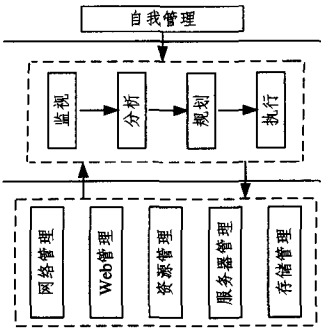


图 4 常规自律计算系统结构

3 关键技术分析

3.1 上下文感知与推理技术

实现自律计算的一个重要手段就是使系统对未来态势具有一定的判断能力和决策能力。采用上下文感知与推理技术能够使系统具有一定的预见性。系统通过对内外环境的感知,将获取的数据进行过滤、融合、推断和演化,其结果用于规划或策略的匹配,能够提高系统的准确性和效率。但是,由于被管资源和环境的分布性和开放性,各类知识在语义上存在异构性,这给推理造成了困难。为解决该问题,可以使用基于本体语言的知识表示和推理。

上下文感知的核心操作为上下文演化。经过上下文演化后,粗糙的底层原始上下文被演化成更有价值的信息。上下文的过滤、推理、融合、存储构成了上下文演化的主要内容。上下文演化的另一个目标是转化为统一的上下文格式以支持上下文的互操作,并允许上下文以统一格式自由传输。上下文演化过程如图 5 所示。

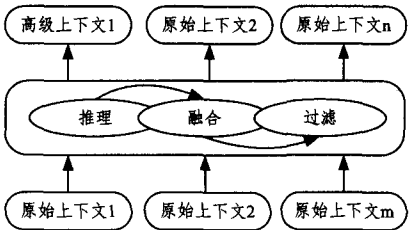


图 5 上下文演化

由于传感器感知的上下文是未经处理的原始上下文,因此信息可能是模糊、动态和不精确的,需要对信息进行过滤以提高上下文信息的精度。融合与推理的目的是解决以下问题:底层上下文提供的信息有限,且可能包含相互冲突的上下

文。上下文的过滤、推理和融合构成了上下文演化的主体,其关键问题是如何甄别有效的上下文,如何利用上下文演绎出有用的结论以及如何联合演绎等。这些问题的产生与所感知的原始上下文所具有的单一、低层、不稳定、不精确等特点有关。目前上下文过滤、推理和融合多采用 Bayesian 网络、模糊推理和人工神经网络等方法解决。相关工作包括李蕊^[19]在其博士学位论文中对上下文感知计算的关键技术的研究;马守明^[20]等提出的基于集对分析的上下文感知不确定性的推理方法,但其存在推理结果偏差大的缺点;潘洪军^[21]使用层次化的模糊 Petri 网来进行知识表示和推理,降低了知识库的复杂性,方便了知识库的优化和管理。

3.2 自律 Agent 系统协同技术

目前学术界对自律 Agent 计算系统 Agent 之间的通讯合作协商理论研究较少,理论上各 Agent 之间能够通过相互协作、合作、协调和协商来达到系统的总体自适应。单个 Agent 具有一定的自主决策能力,能够自我控制状态和行为,在策略及知识库的指导下能够自我调整。但是,由于单个的 Agent 系统在外环境感知、策略制定、数据处理方面还存在局限,不能使整个自律计算系统在更高层面上整体自适应,因此要使得自律 Agent 之间能够协同完成任务,合作解决管理层面上的复杂问题,协商执行某类操作就成为必需。在此,参照 MAS 相关研究成果是一条有效途径。目前,解决 MAS 中协同问题的关键方法主要包括:

(1)组织协同:提供一个对系统具有更广阔视野的 Agent 来解决冲突和确保行为一致,即采取组织的、分层组织的结构。这是最简单的协同技术,经典的主/从,或者 C/S 体系结构就是这样。

(2)为了协同的合约:Agent 之间的任务和资源分配与确定组织结构最著名的协同技术是合同网协议。在这个方法中采取的是市场机制。R. Davis 和 R. G. Smith 最早提出此协议,T. Sandhoim^[22]和 K. Sycara 对合同网协议进行扩充,引入不遵守承诺的惩罚措施和利用金融选择定价理论获得在不确定性环境中的灵活约定方案。

(3)为了协同的规划:更传统的 AI 研究将多 Agent 的协同问题看作一个规划问题。多 Agent 强调避免不一致与冲突情况,这对诸如空中交通管制这样的系统来说至关重要。

(4)有两种类型的多 Agent 规划:集中式和分布式规划。这两种多 Agent 规划都要求节点共享和处理大量的信息,因而多 Agent 规划涉及到更多的计算和通信。

由于自律 Agent 计算系统中 Agent 之间地位平等且智能水平和功能都较 MAS 的 Agent 丰富,因此其解决 MAS 系统协同问题的主/从模式不适合自律 Agent 计算系统,而采用合同网或规划技术应是行之有效的解决方法。

3.3 机器学习技术

自律计算需要系统不仅能够在高层策略的指导下完成动作,还需具备完善的学习能力,以不断获取新的知识并正确应用知识,完善其知识库。其学习能力的强弱直接响到系统的自律管理能力。

将从感知器得到的信息和内部环境得到的信息形式化为符号后,系统可采用基于符号的归纳学习方法。归纳学习旨

在从大量的经验数据中归纳抽取出一般的判定规则和模式,是从特殊情况推导出一般规则的学习方法。归纳学习通常被描述为使用训练实例来引导一般规则的搜索问题。所以它包含两个空间:实例空间和规则空间。其学习就是在规则空间中搜索要求的规则,并从实例空间中选出一些例子来解决规则空间中某些规则的二义性问题。其模型如图 6 所示。

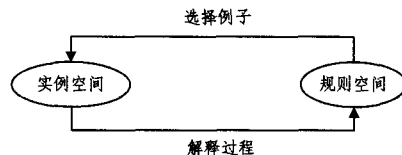


图6 归纳学习的双空间模型

当多个自律管理系统联合起来组成类似于 MAS 的自律计算系统时,各自律计算系统之间的学习成为一个研究重点。由于单个自律管理单元不具备环境知识的完备性,因此它们之间的相互学习就不可避免。一般采用强化学习作为其学习方式。

在自律 Agent 计算系统中,每个 Agent 拥有独立的学习机制,并与其它 Agent 交互,因此被称为交互式强化学习(Interactive RL)。其典型学习算法有瞬时差分方法(Temporal Difference Method)、Q-学习算法(Q-Learning Algorithm)、自适应启发评价算法(Adaptive Heuristic Critic Algorithm)。另外,基于神经网络的强化学习也是 Agent 学习的重要方式。

结束语 自律计算作为新兴的热点研究领域,近年来在概念、理论、技术和应用等层面均取得了一些进展^[23-25]。但是,从已有的研究成果来看,有关自律计算的模型和实现都还不成熟,未能建立完整的自律计算的理论体系和方法体系,目前已有的一些应用实例也只是在某个领域的孤立应用,系统的可扩展性、稳定性都有待提高。本文也仅仅从系统通用框架模型到关键技术做了一般性阐述,对细节性的问题未作深入探讨。未来的研究工作的重点是不断完善自律计算理论及模型,深入研究系统的各关键技术,制定系统性能的评价策略,从而形成一个不仅具备完善的自律计算的理论体系而且具备完备自律能力的产品。

参考文献

- [1] Emminella M, Francescangeli R, Reali G, et al. An enabling Platform for Autonomic Management of the Future Internet [J]. IEEE Network, 2011, 25(6): 24-32
- [2] Charles G V, Bessam A, Sylvain G, et al. Toward Autonomic Pervasive Computing[C]//Proceedings of WAS2008. 2008: 674-676
- [3] Agustn C, Omer R, Blanca C, et al. An Autonomic Network Aware Scheduling Architecture for Grid Computing [C]//Proceeding of the 5th International Workshop on Middleware for Grid Computing, hold at the ACM/IFIP/usENIX 8th International Middlednare Conference. ACM press, 2007: 1-6
- [4] Jia R, Bu X P, Xu C Z, et al. VCONF: A Reinforcement Learning Approach to Virtual Machines Auto configuration [C]//ICAC'09 Proceeding of the 6th International Conference on Autonomic Computing. ACM press, 2009: 137-146

(下转第 53 页)

降低成本,也能够有效地满足用户需求,特别是服务器的托管、教师的移动授课、学生的动态实验环境以及临时性的突发需求。但是,以 IaaS 的模式提供云服务,必须要保证安全性、数据的有效性和可用性。此外,云计算对网络具有极大的依赖性,虚拟化对性能也依然存在一定的损耗,稳定性也依然有待提高,因此在网络环境得不到保障、对单机运算能力要求较高等场合并不适合云计算。

参考文献

- [1] Cloud Computing [EB/OL]. <http://en.wikipedia.org/wiki/>, 2010-12-15
- [2] Armbrust M, Fox A, Griffith R, et al. Above the clouds: A berkeley view of cloud computing[R]. Technical Report No. UCB/EECS-2009-28. UC Berkley, USA, 2009; 1-25
- [3] 林闯,田源,姚敏. 绿色网络和绿色评价: 节能机制、模型和评价[J]. 计算机学报, 2011, 34(4): 593-612
- [4] Rittinghouse J W, Ransome J F. Cloud computing: implementation, management, and security[M]. Boca Raton, FL, USA: CRC Press, 2010
- [5] 金海,等. 计算机系统虚拟化——原理与应用(Virtualization of Computer System——Principal and Application)[M]. 北京: 清华大学出版社, 2009; 54-110
- [6] Adler S. WebOS: say goodbye to desktop applications [J]. Net-worker, 2005, 9(4): 18-26
- [7] Group A. Amazon elastic compute cloud (Amazon EC2) [EB/OL]. [2010-01-01]. <http://aws.amazon.com/ec2/>, 2010-12-15
- [8] 刘鹏. 云计算(Cloud Computing)[M]. 北京: 电子工业出版社, 2010; 69-110
- [9] Tsakalozos K, Killapi H, Sitaridi E. Flexible Use of Cloud Resources through Profit Maximization and Price Discrimination [C]// Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing. USA, 2010; 1423-14435
- [10] 苏兆龙. 排队论基础[M]. 成都: 成都科技大学出版社, 1995; 12-28
- [11] 刘次华. 随机过程(第四版)[M]. 武汉: 华中科技大学出版社, 2008; 42-89
- [12] Calheiros R, Ranjan R, Rose C, et al. Cloudsim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services[R]. GRIDS-TR-2009-1. Grid Computing and Distributed system Laboratory, The University of Melbourne, Australia, 2009; 412-419
- [5] (上接第 18 页)
- [5] 自主计算的理论和技术研究[Z]. 国家科技成果. 中国科学院计算技术研究所, 2005
- [6] 臧铤, 黄忠东, 董金祥. 基于状态的通用自主计算模型[J]. 计算机辅助设计与图形学学报, 2007, 19(11): 1476-1481
- [7] 张海俊, 史忠植. 自主计算环境[J]. 计算机工程, 2006, 32(7): 1-3
- [8] Hella S, Frank O, Wolfgang R. Design and Construction of Organic Computing Systems [C]// CEC 2007. IEEE press, 2007; 4215-4221
- [9] 廖备水, 李石坚, 姚远, 等. 自主计算概念模型与实现方法[J]. 软件学报, 2008, 19(4): 780-795
- [10] 刘全中, 胡正国, 樊星. 自律计算在服务器虚拟化中的应用[J]. 微电子学与计算机, 2005, 22(12): 96-99
- [11] 刘文洁, 李战怀. 一种面向服务器集群的自律计算模型[J]. 计算机应用, 2007, 27(7): 209-301
- [12] 徐云平. 自律计算在网络管理软件中的应用研究[D]. 西安: 西北工业大学, 2005
- [13] Martin P, Powley W, Wilson K. The WSDM of Autonomic Computing, Experiences in Implementing Autonomic Web Services [C]// International Workshop on Software Engineering for Adaptive and Self-Managing Systems. IEEE press, 2007; 4215-4220
- [14] Tesaro G, Chess D M, Walsh W E, et al. A Multi-Agent Systems Approach to Autonomic Computing [C]// Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems. New York: ACM press, 2004; 464-471
- [15] Holger K, Bernhard B. Combining Multi-Agent-System Methodologies for Organic Computing Systems [C]// Proceedings of the 3rd International Workshop on Self-Adaptive and Autonomic Computing System (SAACS 05). IEEE Computer Society, 2005; 160-164
- [16] 樊星, 李战怀, 刘全中. 服务器系统管理的自律计算模型[J]. 计算机应用, 2006, 26(3): 704-705
- [17] 江国朝. 基于 Policy 的自律网络管理系统的研究与实现[D]. 西安: 西北工业大学, 2006
- [18] Michael E, Bernd F. Supporting Autonomic Computing Functionality via Dynamic Operating System Kernel Aspects [C]// Proceeding of the 4th International Conference on Aspect-oriented Software Development. ACM, 2005; 51-62
- [19] 李蕊. 上下文感知计算若干关键技术研究[D]. 长沙: 湖南大学, 2007
- [20] 马守明, 王汝传, 叶宁. 基于集对分析的上下文感知不确定性的推理方法[J]. 南京邮电大学学报: 自然科学版, 2009, 29(1): 35-37
- [21] 潘洪军, 孙吉贵. 层次化模糊 Petri 网: 形式化定义和推理[J]. 吉林大学学报: 工学版, 2007, 37(1): 112-116
- [22] Sandholm T W. Negotiation among self-interested computationally limited agents [D]. Amherst, MA: University of Massachusetts, 1996
- [23] Tsagkaris K, Vlacheas P, Athanasios G, et al. Autonomics in Wireless Network Management: Advances in Standards and Further Challenges [J]. IEEE Network, 2011, 25(6): 41-49
- [24] Zheng Y, Prehofer C. Autonomic Trust Management for a Component-Based Software System [J]. IEEE Transactions on Dependable and Secure Computing, 2011, 8(6): 810-823
- [25] Rubio L J, Merida C C, Serrat J, et al. A Service-Centric Orchestration Protocol for Self-Organizing Autonomic Management Systems [J]. IEEE Network, 2011, 25(6): 16-23