

一种新的系统管理技术：自主计算

廖备水

(浙江大学语言与认知研究中心, 浙江 杭州 310028)

摘要: 自主计算是一个新兴的热点研究领域,旨在以自主神经系统的基本工作机制为参照,建立具有自我管理能力的、自适应的计算机系统。在已有工作的基础上,对自主计算的基本概念、实现方法和存在问题进行了初步的归纳和分析。

关键词: 自主计算; 基于策略的管理; 智能主体; 系统管理

中图分类号: TP393

文献标识码: A

文章编号: 1673-9833(2008)01-0031-04

A New System of Management Technology for Autonomic Computing

Liao Beishui

(Study Center for Language and Cognition, Zhejiang University, Hangzhou 310028, China)

Abstract: Autonomic computing is an emerging research hotspot, which aims at establishing self-managable and self-adaptive computing systems, inspired by the basic working mechanism of autonomic nervous system. On the basis of past work, it summarizes and analyzes the fundamental concepts, realization methods, and existing problems of autonomic computing.

Key words: autonomic computing; policy-based management; agent; system management

2001年10月,美国国际商业机器公司(IBM)宣称:阻碍未来IT工业进步的主要障碍是目前隐约可见的软件复杂性危机^[1]。目前,大型分布式信息系统的管理复杂性问题已经成为其发展过程中的瓶颈。为使系统正常运行,IT企业通常要花费高出设备成本4~20倍的管理费用^[2],而且单靠IT专家和技术人员的努力越来越难以驾驭开放、动态和异构的信息系统。在这个背景下,以降低分布式系统管理复杂性问题为目标的“自主计算(Autonomic Computing)(分布式系统自我管理的理论、方法和技术)”^[1,3]正在成为当前研究的热点。作为一个重要的研究领域,自主计算的思想一经提出,就受到了广泛关注,取得了一系列研究成果。如Sterritt(2005)^[4],Hariri(2006)^[5],Tianfield(2004)^[6]等,同时,国内在这方面的研究也有了起步,典型的工作包

括曹建(2004)^[7]、刘涛(2006)^[8]、张海俊(2006)^[9]、王千祥(2004)^[10]等。为了进一步推动自主计算这一研究领域的发展,本文对自主计算的基本概念、实现方法和存在问题进行初步的归纳和分析,以期国内的专家和学者对该研究领域给予更多的关注。

1 自主计算的基本概念

自主计算的思想来源于人体生物学,自主的神经系统监控人的心跳、检查血糖水平、保持身体温度接近于37°C,而不需要有意识的努力。以此为参照,自主计算希望在尽可能少的人为干预之下,自己管理自己,实现整体的自适应。

目前,IBM的研究者将自主计算中的自我管理机

收稿日期: 2007-11-02

基金项目: 国家自然科学基金资助项目(60773177),中国博士后科学基金资助项目(20060400316),浙江大学语言与认知研究中心科研课题基金资助项目(C04)

作者简介: 廖备水(1971-),男,福建古田人,浙江大学助理研究员,博士后,主要研究方向为人工智能,Agent与多Agent系统,自主计算,逻辑学。

制例示为4个方面的能力(特性):自配置、自修复、自优化和自保护^[1]。

自配置——自动地适应于动态变化的环境。自配置组件使用IT专家提供的策略,自动适应IT系统中的变化。这些变化可能包括新组件的部署、现有组件的移除、或者工作负载的急剧增加或减少等。

自修复——发现、诊断和修复故障。自修复组件可以检测系统故障,并启动基于策略的修复活动。

自优化——自动监视和调整资源。自优化组件能够自我调整以满足终端用户或商业的需求。调整活动可能意味着重新分配资源来改善系统的总体效用,或者确保特定的商业交易能够及时完成。

自保护——预见、监测、识别、保护来自各处的侵袭。自保护组件能够在敌意行为发生时监测到它们,并采取行动,使得自己不易受攻击。敌意行为可能包括未经授权的访问和使用、病毒传染与繁殖、以及服务拒绝入侵等。

在自主计算系统中,以上4个自我管理的能力是通过一个智能控制循环来实现的,这个循环从系统收集信息、决策、然后根据需要调整系统,如图1所示^[1]。

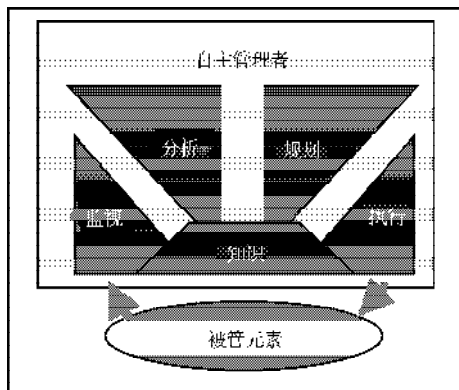


图1 自主元素结构

Fig. 1 The structure of autonomic element

图1中,控制循环(自主元素)由2个主要元素构成:被管元素和自主管理者。被管元素是自主管理者管理的对象,比如服务器、数据库、路由器、Web服务、网格服务等,也可以是其它自主元素。自主管理者是实现特定控制循环的组件,比如Agent。它由5个部分组成:监视、分析、规划、执行和知识。其中,“监视”部分提供机制来搜集、合成、过滤、管理和报告来自被管元素的状态细节;“分析”部分实施情景分析;“规划”部分负责组织动作以实现目标;“执行”部分控制规划的执行;以上4个组件使用的数据作为共享的“知识”存储在知识库中,共享的知识包括拓扑信息、系统日志、性能尺度和策略。其中策略由一组行为约束或偏好组成,它们影响自主管理者的决策。显然,自主管理者的“规划”部分负责解释和翻译策

略细节,而“分析”部分则负责决定自主管理者是否能够遵守策略,自主管理者依据高层策略尽可能有效地完成任务。

自主管理系统由自主元素组成,自主元素之间的关系可以是P2P(Peer-to-Peer)的或层次结构的。比如,在由Agent构成的自主管理系统中,担任自主元素的Agent之间可能通过签订合同构成一对一的契约关系,也可能组成嵌套的Agent联邦,形成层次关系。

在自主计算系统中,自主元素是构造系统的核心要素,它必须具有下列行为特征^[2]:

自主元素必须能够自我管理 自主元素必须负责内部的自我配置,修复内部故障,优化自身的行为,保护自己免受外界攻击。为在总体上减化系统的管理,自主元素在任何可能的时候必须就地处理问题。如,当它发现它所依赖的完成服务的某个自主元素没有遵守协议时,它应试图解决这个问题。它可以通过寻求另外的备用的自主元素来提供这个服务,或者通过发现新的、更适合的元素来提供服务。

自主元素必须能够与其他自主元素建立和维护协作关系 自主元素必须精确描述自己所提供的服务,使其对于其他自主元素来说是可访问的(Accessible)和可理解的(Understandable)。协作关系是建立在协议(合同)的基础之上,所以,自主元素必须理解和遵守协议的条款。另外,自主元素必须具有协商的能力,以便建立协议。

自主元素必须能够依据外部动机来管理自己的行为 自主元素必须通过适当调整或配置其自身的参数,或者通过利用系统中其他自主元素的资源,来管理其行为和关系,以满足协议所规定的义务(Obligations)。与自主元素相关的义务有2种形式:一方面,自主元素必须兑现合同的条款;另一方面,自主元素必须能够接收和遵守策略(反映高层商业目标和策略)。自主元素必须拒绝那些将破坏策略或协议的任何新的服务请求;同样,它也必须拒绝(或者提出反建议)任何新建议的关系(协商时)或者新的策略,如果它们与现有的关系或策略冲突,自主元素必须具备足够的分析能力来支持这些功能的实现。

2 自主计算的主要实现方法

自从IBM提出自主计算的思想之后,如何实现自主计算系统已经成为一个重要的研究领域^[13-23]。作为一个新的领域,自主计算系统的实现问题仍然处于探索的过程之中。为了支持自主计算系统的建立,需要一系列的现有理论和技术的支持。例如,Agent技术、基于策略的管理、分布式计算、控制理论等。不过,Agent技术和基于策略的管理是2种主要的支撑技术。

从目前的研究进展来看,已经有一些学者和专家在这方面取得了进展,下面举几个典型的例子。

2.1 基于 Agent 的方法

Gerald Tesaro等在文献[13]中提出一种基于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。在Unity中,各个应用环境中的自主元素(Agent)依据该应用的服务级函数所说明的信息来计算资源级的效用函数,然后,把来自多个应用环境的资源级的效用函数送到一个资源仲裁元素,最后,该仲裁元素计算所有应用的服务器的全局优化分配方案。Unity系统的组件实现为自主元素——个体Agent,它可以控制资源,并把服务传递给人或其他自主元素。每个自主元素负责自身的内部自主行为,如它所控制的资源的管理,内部操作的管理等;同时,每个自主元素也负责形成和管理自主元素之间的关系。另一方面,每个自主元素的操作受到人制定的高层策略的指导。策略既可以作为效用函数描述的依据,也可以是控制系统操作的更简单的因素。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,并得以指导自主元素(Agent)的行为。

Huaglory Tianfield在文献[16, 18]中阐述的自主计算系统也是基于Multi-agent系统。该系统通过一个分布的、嵌套的“感知-决策”链来实现主要的自主机制。感知-决策链形成多个循环,这些循环可以被嵌套,形成多层的控制体系,每个循环可以由一个或多个Agent来实现。这里的Agent是一个自主实体,它由一组计算实体组成,其中3个实体是标准的,分别用于内部调度、问题求解、以及社交通讯路由,而其它的实体则是可选的。在这个嵌套的Agent系统中,上层的Agent为下层Agent提供指导、控制、目标导向和约束。层与层之间的联系通常是通过离散的事件驱动来实现,某一层可以从相邻的低层获得关于世界的统计的感知,而低层从其相邻的高层接收和执行命令则是随机的。作者将基于该体系结构的系统分别用于协同的医疗诊断和网络管理的自优化。

2.2 基于策略的方法

文献[22]在目前基于策略的模型中加入“行为蕴涵(Behavior Implications)”以支持自主计算。所谓“行为蕴涵”是指对于特定的策略(表示为:事件→如果(条件)→那么(动作)),其动作的执行对系统特性的影响。该方法利用“行为蕴涵”设计了一种自我学习和自动化决策的机制:当指定的目标不能满足时,将产生一个触发信号,决策模块就根据“行为蕴涵”和其他条件进行分析,然后自动决策并调用一条“行为蕴涵”,满足所期望的目标特性的策略。策略执行时,监

视其对系统特性的影响,并以此进行自我学习,不断修正和精化“行为蕴涵”。

文献[23]运用基于策略管理的思想来实现自主的存储分配。该系统使用了一种基于自主计算原理和基于策略管理的管理自动化方法。具体地来说,基于策略的自主存储分配的目标是通过制定相对高层的目标,而不是低层的手工步骤,来实现存储资源的分配、资源性能的监视、热点的消除。该自动化过程允许使用策略规定约束条件,并在此条件的约束下来管理资源。文章描述了基于策略的自主存储分配管理器原型(称为ALOMS-Tango)的设计和实现。

2.3 体系结构方法

文献[12]提出了一种用于自主计算的体系结构方法。文章指出,建立大规模的、能够依据人的高层指导来管理自己的计算系统将面临巨大的挑战。为了克服这种挑战,不仅要依靠发明新的技术,而且也要依靠建立自我管理的体系结构,以适当利用这些技术。正确的体系结构应该能够在系统级提供方法来实现自主的行为,使得系统既能够描述让个体组件自主所需的外部接口和行为,又能够描述如何把这些组件合成起来,实现整个系统的自我管理。为了达到这个目标,文章提出的方法采用面向服务的体系结构,结合面向Agent的系统的思想,通过添加实现自我管理所需要的接口、行为和设计模式,来达到系统自我管理的目标。系统中的主要概念是自主元素,自主元素被定义为能够依据策略负责自我行为管理的组件,并能够与其它自主元素交互,以提供或使用计算服务。这样,自主系统中的每个组件都是一个自主元素,既包括计算资源(如数据库、存储系统或服务器),也包括高层的具有管理权力的元素(如工作负载管理器),还包括一些辅助的管理元素(如策略库、监视器、代理和注册中心等)。

3 存在问题

综合所述,在当前的研究中,虽然Agent技术和基于策略管理的方法能从不同角度为系统提供某些自主管理的功能,不过用这2种技术实现的自主计算系统分别存在以下缺陷:1)单纯基于Agent技术的自主计算系统虽具有自我决策和自我管理的能力^[16,18],但缺乏反映上层管理策略的手段;2)单纯基于策略管理的方法虽能灵活反映动态的商业管理策略,但失去了Agent提供的强大决策和自我管理能力^[22,23]。显然,这2种技术具有互补性,它们的结合将成为建立自主计算系统的理想选择。不过,从目前研究现状看^[16,18,22,23],这2种技术仍没有被很好结合起来用于构建自主计算系统。究其原因,主要是由于目前的Agent模型(如BDI Agent)不能动态接受并组织实施来自管理人员动态制定的策略和偏好。即在传统的Agent模型中,

Agent 体系结构和知识库中的内容, 在系统设计和开发时已确定, 系统目标、商业策略等硬编码在系统中或作为知识存放在知识库中, 不能动态改变。另外, 当 Agent 在决策时, 依据的是环境状态和内部知识, 不能动态接受用户的新动机(要求)。所以, 当 Agent 的结构和性能设计好后, 如果要改变其功能, 往往需要改变底层的软件编码, 灵活性和扩展性均较差。因此, 虽然各种文献从各个侧面提出了实现自主计算的理论和应用, 但没有提供机制来解决这个问题。可见, 如何将 Agent 技术和基于策略管理的方法相结合, 将是自主计算研究领域的一个重要方向。

参考文献:

- [1] Kephart Jeff O, Chess David M. The Vision of Autonomic Computing[J]. IEEE Computer, 2003, 36(1): 41-50.
- [2] Patterson D, Brown A, Broadwell P, et al. Recovery Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies[R]. UCB-CSD-02-1175, USA: University of California Berkeley, 2002.
- [3] Manish Parashar, Salim Hariri. Autonomic Computing: An Overview[C]//Proceedings of UPP 2004, LNCS 3566. Berlin Heidelberg: Springer-Verlag, 2005: 247-259.
- [4] Roy Sterritt. Autonomic computing[J]. Innovations in Systems and Software Engineering, 2005(1): 79-88.
- [5] Salim Hariri, Bithika Khargharia, Houping Chen, et al. The Autonomic Computing Paradigm[J]. Cluster Computing, 2006(9): 5-17.
- [6] Huaglory Tianfield, Rainer Unland. Towards autonomic computing systems[J]. Engineering Applications of Artificial Intelligence, 2004, 17: 689-699.
- [7] Cao Jian, Wang Jie, Zhang Shensheng, et al. A dynamically reconfigurable system based on workflow and service agents [J]. Engineering Applications of Artificial Intelligence, 2004, 17: 771-782.
- [8] 刘涛, 曾国荪, 吴长俊. 异构网格环境下任务分配的自主计算方法[J]. 通信学报, 2006, 27(11): 139-147.
- [9] 张海俊, 史忠植. 自主计算软件工程方法[J]. 小型微型计算机系统, 2006, 27(6): 1077-1082.
- [10] 王千祥, 申峻嵘, 梅宏. 自适应软件初探[J]. 计算机科学, 2004, 31(10): 168-171.
- [11] IBM. Autonomic Computing: IBM's perspective on the State of Information Technology[EB/OL].[2001-不详-不详]. <http://www-1.ibm.com/industries/government/doc/content/bin/auto.pdf>.
- [12] White Steve R, James E Hanson, Whalley Ian, et al. An Architecture Approach to Autonomic Computing[C]//Proceedings of the International Conference on Autonomic Computing (ICAC'04). Washington, DC: IEEE Computer Society, 2004: 2-9.
- [13] Tesauo Gerald, Chess David M, Walsh William 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.
- [14] Konstantinou Alexander V, Towards Autonomic Networks [D]. New York: Columbia University, 2003.
- [15] Dario Bonino, Alessio Bosca, Fulvio Corno. An Agent Based Autonomic Semantic Platform[C]//Proceedings of the International Conference on Autonomic Computing (ICAC'04). Washington, DC: IEEE Computer Society Press, 2004: 189-196.
- [16] Huaglory Tianfield. Multi-Agent Autonomic Architecture and Its Application in E-Medicine[C]//Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT'03). Los Alamitos, California: IEEE Computer Society, 2003: 601-604.
- [17] Huaglory Tianfield, Rainer Unland. Towards autonomic computing systems[J]. Engineering Applications of Artificial Intelligence, 2004, 17: 689-699.
- [18] Huaglory Tianfield. Multi-Agent Based Autonomic Architecture for Network Management[C]//Proceedings of the First Workshop on Autonomic Computing Principles and Architectures (AUCOPA). Banff, Alberta: IEEE Computer Society, 2003: 462-469.
- [19] Li Z, Parashar M. Rudder: A Rule-based Multi-agent Infrastructure for Supporting Autonomic Grid Applications [C]//Proceedings of the International Conference on Autonomic Computing (ICAC'04). Washington, DC: IEEE Computer Society Press, 2004: 278-279.
- [20] Li Z, Liu H, Parashar M. Enabling Autonomic, Self-managing Grid Applications[EB/OL].[2004-不详-不详]. <http://www.cs.unibo.it/self-star/papers/parashar.pdf>.
- [21] Wolf Tom De, Holvoet Tom. Towards Autonomic Computing: Agent-Based Modelling, Dynamical Systems Analysis, and Decentralised Control[C]//Proceedings of the First Workshop on Autonomic Computing Principles and Architectures (AUCOPA). Banff, Alberta: IEEE Computer Society, 2003: 470-479.
- [22] Sandeep Uttamchandani, Talcott Carolyn, Pease David, et al. Eos: An Approach of Using Behavior Implications for Policy-Based Self-Mamanement[C]//Proceedings of the fourteenth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management, LNCS2867. Berlin Heidelberg: Springer-Verlag, 2003: 16-27.
- [23] Devarakonda Murthy, Chess David, Whalley Ian, et al. Policy-Based Autonomic Storage Allocation[C]//Proceedings of the fourteenth IFIP/IEEE International Workshop on Distributed Systems: Operations & Management, LNCS2867. Berlin Heidelberg: Springer-Verlag, 2003: 143-154.

(责任编辑: 张亦静)