

基于马尔可夫链的校园云平台效用分析

戴 瑶^{1,2} 胥光辉¹ 毛东方² 韩 晶²

(解放军理工大学指挥信息系统学院 南京 210007)¹ (广东省湛江军分区 湛江 524038)²

摘 要 校园网存在能耗高、效率低、浪费多等诸多问题,其可以通过云计算来解决,但云计算所带来的效益增长并没有一个很好的标准。对校园云平台进行了简单描述,对数据中心的成本、用户实例的成本进行了分析,提出了校园云平台成本效用函数。建立了校园云平台马尔可夫链模型来分析负载均衡策略和贪心策略。使用 CloudSim 对校园云平台进行仿真,对模型进行验证分析和比较。通过建立模型和仿真分析得知,建立云平台,采用不同的策略,可以有效提高校园网资源利用率以降低成本,或者提供更好的服务质量。

关键词 马尔可夫链,校园云平台,效用分析

中图分类号 TP302 **文献标识码** A

Utility Analyzing of Campus-wide Cloud Computing Platform Based on Markov Chain

Ji Yao^{1,2} XU Guang-hui¹ MAO Dong-fang² HAN Jing²

(Institute of Command Automation, PLAUST, Nanjing 210007, China)¹

(Military Sub-district in Zhanjiang, Guangdong Province, Zhanjiang 524038, China)²

Abstract There are many problems of high energy consumption, low efficiency, and waste, which can be solved by cloud computing. However, what rate of resource utilization has been improved by cloud computing and what influence should be figured. This paper gave a brief description on the campus-wide cloud computing platform, analyzed the cost of the datacenter and the user instances, put forward a cost-utility function, made a Markov queuing model, researched the resource allocation and scheduling policies, used CloudSim to simulate the Markov queuing model, discussed the optimization strategies. By madding and simulating the model, the rate of resource utilization in the campus network can be improved evidently and the cost can be reduced, or the QoS can be improved by using proper policy.

Keywords Markov queuing model, Campus-wide cloud computing platform, Utility analyzing

随着因特网的发展,数据中心在校园网的建设中发挥着越来越重要的作用。但是,校园网存在科研和实验项目多、学生流动性大、应用场景复杂的特点。随着网络规模的扩大和网络设备的不断更新,能量的利用日益暴露出能耗高、效率低、浪费多等诸多问题,由此产生的经济效益也备受争议。可以采用云计算^[1,2]技术将校园数据中心建成以虚拟机为母体的资源池对外提供服务。然而,对于云计算的使用能够为资源的利用率带来多大的提高,能够对建设校园网的成本带来多大的影响,缺乏量化指标,因此需要对校园云平台进行具体的效用分析^[3]。

本文旨在通过对校园云平台的需求、院校师生对系统的QoS需求和平台的建设维护成本进行分析,建立马尔可夫链模型,对校园云计算平台进行效用分析。通过对投入和产出比的分析,建立对校园云平台的投入、产出以及规模的度量指标,对校园网的建设提出参考意见。

1 校园云平台的成本构成

校园云平台以基础设施即服务^[3] (Infrastructure as a

Service, IaaS)的方式,为全校师生提供虚拟机实例(Instance)的服务^[5]。而每个通过虚拟机托管的服务器、教学和办公用机、学生实验用虚拟机以及其他临时性虚拟机都具备独立的存储空间,镜像主要存储在廉价的共享存储上,而虚拟机的虚拟CPU、虚拟内存、虚拟I/O设备等则运行在宿主主机上^[6],如图1所示。

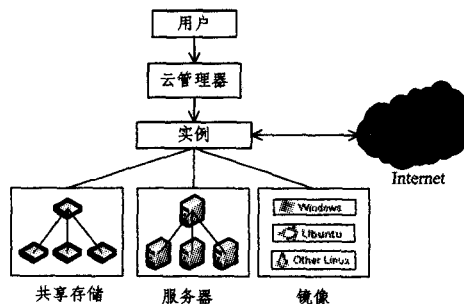


图1 校园云平台架构

1.1 数据中心的成本构成

校园云平台的成本构成和传统数据中心一样,其运营的

到稿日期:2012-09-27 返修日期:2013-01-31 本文受国家重大专项项目(GFZX0404010503),江苏省自然科学基金(BK2010132)资助。

戴 瑶(1985—),男,硕士,主要研究方向为软件工程、云计算, E-mail: trashes@163.com; 胥光辉(1971—),男,博士,副教授,主要研究方向为软件工程、云计算; 毛东方(1973—),男,硕士,主要研究方向为网络安全; 韩 晶(1983—),男,硕士,主要研究方向为无线通信。

整体成本 C_T 可以表示为建设成本 C_B 、购买硬件成本 C_H 、购买软件成本 C_S 、管理维护成本 C_M 和电力成本 C_E 的总和,可表述为:

$$C_T = C_B + C_H + C_S + C_M + C_E$$

1.2 用户实例的成本构成

对于用户(通过虚拟机托管的服务器、教学和办公用机、学生实验用虚拟机等),只需要关心软硬件配置情况,而不需要关心校园云平台的体系结构和技术细节^[7]。在任意时刻,单位实例单位时间的成本可以表示为单位时间的成本除以当前运行的虚拟机实例数 N_V ,可以得出单位实例单位时间的瞬时成本:

$$V_T = (C_B/L_B + C_H/L_H + C_S/L_S + P_M + P_E)/N_V$$

L_B 表示数据中心的使用周期, L_H 表示硬件生命周期, L_S 表示软件的使用周期, P_M 表示管理费用, P_E 表示数据中心的功率, N_V 表示当前的用户数。瞬时成本会随着用户数量、环境等多种因素改变,在一定程度上反映了当前云平台的运行效率。

1.3 成本效用函数

而对任意一个数据中心,建筑成本 C_B 、硬件采购成本 C_H 和软件采购成本 C_S 通常是一次性的投入。维护成本 C_M 可以近似为一个一次线性时间函数 $C_M = P_M t$, P_M 的值相对比较固定,可以看作一个常数,当 t 取整个数据中心的生命周期时, $C_M = P_M t$, C_M 也可以看作一个常数。不考虑建筑成本、软硬件成本和人工管理成本,而是重点从资源利用率的角度去衡量云平台的效用,因此假定在整个生命周期内, $C_B + C_H + C_S + C_M$ 为一个常数 A ,该常数可以通过对数据中心的成本核算求得。在使用云平台的情况下,电力成本根据活动的服务器数量的不同有比较大的变化^[8],可以表示为 $C_E = \int P_E dt$ 。因此,总成本可以表示为:

$$C_T = C_B + C_H + C_S + C_M + C_E = A + \int P_E dt$$

在整个生命周期内,单位实例单位时间的平均成本为:

$$E(C_V) = \frac{C_T}{\sum V_i t_i} = \frac{A + \int P_E dt}{\sum V_i t_i}$$

V_i 表示第 i 个虚拟机资源使用量, t_i 表示第 i 个虚拟机的使用时间。

云平台的成本效用函数可以定义为用户实际使用的资源与总投入的比值,也就是单位实例单位时间的平均成本的倒数,即

$$U = \frac{1}{E(C_V)} = \frac{\sum V_i t_i}{A + \int P_E dt}$$

使用成本效用公式,可以有效地衡量平台的整体经济效益,为投入和产出的评价给出一个可以量化的参考^[9]。

2 马尔可夫排队模型和效用分析

校园云平台可描述为一个典型的生产者——消费者系统^[10],服务器集群提供虚拟机服务,用户按需使用虚拟机。通过建立马尔可夫排队模型,模拟服务器产生虚拟机、用户到达、接受服务、离开服务、结合成本构成,对校园云平台进行效用分析。

2.1 系统描述

在校园云平台中,对服务器、虚拟机的 CPU、内存、磁盘等资源进行统一量化,并进行封装,将虚拟机的资源分配和调度简化为一个背包问题^[11]。

在虚拟机的调度中,由于全虚拟机迁移(迁移的过程中,实例的存储空间、CPU、内存等资源都跨物理机迁移)需要消耗过大的带宽资源并且迁移时间较长,因此只在容灾备份中使用较多。而使用更多的是对存储空间和运行状态进行分离,资源调度的过程中主要是对虚拟机的运行状态(CPU、内存等资源)进行迁移。本文对资源调度的研究,只考虑对运行状态的迁移,不考虑资源的备份以及存储空间的迁移。

当用户使用虚拟机(成功申请并开始使用虚拟机,或者已经成功申请后使虚拟机从休眠状态转为活动状态)时,就需要在可用资源队列中为用户分配封装好的资源(CPU 和内存)。当用户终止虚拟机,或者处于静止状态到达一个阈值,服务结束,虚拟计算中心收回资源(CPU 和内存),将资源重新放回可用队列,如图 2 所示。

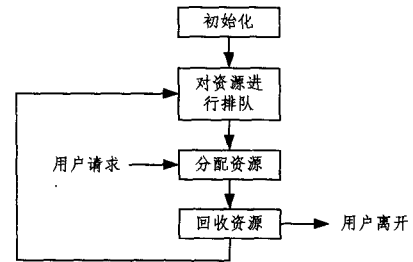


图2 系统运行流程

校园云平台提供虚拟机的服务器是有限的,可将校园云平台看作是一个生灭系统,对其建立排队数学模型: $M/M/c$ 排队系统 $N(t)$,但由于是充分服务,因此实际上它是一个 $M/M/\infty$ 排队系统。用户相继到达发出对虚拟机的申请,时间间隔服从参数为 λ 的指数分布,每个用户使用虚拟机的时间(服务时间)服从参数为 μ 的负指数分布。

假设每个用户申请的虚拟机是相同的,但提供虚拟机的物理服务器的能力不同,可以理解为第 n 服务员可以最多为 n_i 个用户提供服务。排队模型如图 3 所示。

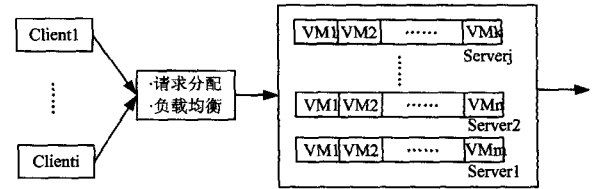


图3 资源分配排队模型

传统的资源调度策略主要有随机策略、负载均衡分配策略和贪心策略。随机策略调度简单,但稳定性差;负载均衡分配策略能在一定程度上保证服务质量,但资源利用率低;贪心策略资源利用率高,但可能出现性能瓶颈。

对简单的随机策略不做讨论,主要分析基于负载均衡的资源分配和调度以及基于贪心策略的资源分配和调度。

2.2 基于负载均衡的资源分配和调度

基于负载均衡的策略的思想是,当有请求到达时,从可用资源最大的服务器上为用户分配虚拟机。平衡状态下,该策略能够避免经过大量的请求和离开后出现拥塞,以保证更好的负载均衡。

负载均衡策略下的服务器队列如图 4 所示。

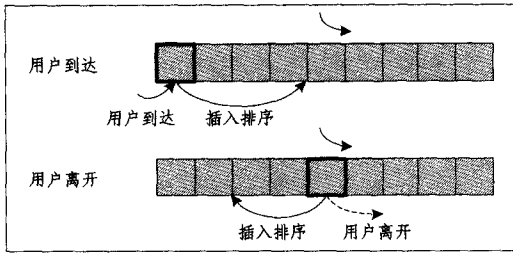


图 4 负载均衡策略下的服务器队列

算法描述为：

步骤 1 将云平台服务器集群中的服务器按照可提供的虚拟机的数量从大到小进行排列；

步骤 2 当有用户到达请求时，从第一台服务器中取出资源，为用户分配虚拟机，回到步骤 1；

步骤 3 当用户使用完虚拟机后，回收资源，回到步骤 1。

在平衡状态，通常是用户申请的虚拟机数量大于服务器的数量，此时所有的服务员都处在忙碌状态，执行负荷 a (任一瞬间忙碌的服务器的平均数定义) 为：

$$a=c$$

其中， c 表示系统总的服务器数。

根据 Little 定理，可求得系统大小 (在平衡状态下用户数) 的均值 L 为：

$$L=\frac{\lambda}{\mu}$$

平均系统时间为：

$$W=\frac{1}{\mu}$$

虚拟机承担的执行负荷称为服务员占有率 ρ ：

$$\rho=\frac{a}{c}=1$$

也即此时每台服务器上都有虚拟机在运行，服务器的利用率为 100%。

而有效利用率为虚拟机使用的资源占总的资源的比例，即：

$$\rho'=\frac{\sum V_i}{\sum P_i}=\frac{L}{\sum P_i}=\frac{\mu}{\lambda \sum P_i}$$

2.3 基于贪心策略的资源分配和调度

贪心策略的基本思想是，接收到用户请求时，虚拟机被分配在已经使用的服务器上，当该服务器的资源耗尽时，才从下一台服务器提供资源。其优点是能够节约能耗，但由于提供资源的节点过于集中，对服务质量会有一定的影响。

贪心策略下的服务器队列如图 5 所示。

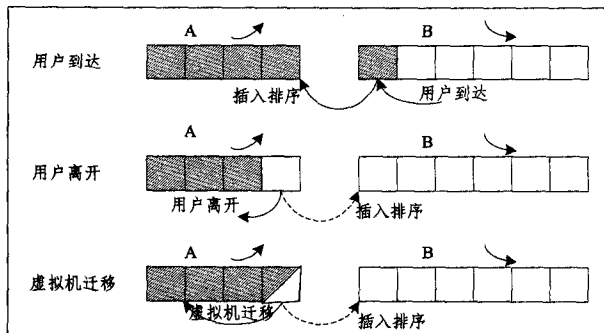


图 5 贪心策略下的服务器队列

算法描述为：

步骤 1 使用队列 A 存放已经为用户分配虚拟机的服务器，按照可用资源从小到大的顺序排列；队列 B 存放没有分配虚拟机的服务器，按照可用资源从大到小的顺序排列；

步骤 2 当用户请求到达时，到 A 中依次查询，若 A 中有满足需求的服务器，则为用户分配资源，队列 A 按照可用资源从小到大的顺序排列；若 A 中没有满足需求的服务器，则从 B 中取第一台服务器分配资源，将该服务器加入队列 A，队列 A 按照可用资源从小到大的顺序排列；

步骤 3 当用户离开时，回收资源，若提供资源的服务器仍在提供服务，则队列 A 按照可用资源从小到大的顺序排列；若该服务器此时不再提供服务，则将该服务器放到队列 B，队列 B 按照可用资源从大到小的顺序排列；

步骤 4 设定阈值 ρ ，当 A 中的有效资源利用率小于 ρ 时，依次将 A 队尾的虚拟机往队头进行迁移。

平衡状态下，假设每个服务器资源都能刚好被虚拟机耗尽，此时的系统大小的均值 L 为：

$$L=\frac{\lambda}{\mu}$$

执行负荷为：

$$a=\frac{\sum V_i}{\sum P_i}c=\frac{L}{\sum P_i}c=\frac{\lambda/\mu}{\sum P_i}c=\frac{\lambda c}{\mu \sum P_i}$$

虚拟机承担的执行负荷称为服务员占有率 ρ ：

$$\rho=\frac{a}{c}=\frac{\lambda}{\mu \sum P_i}$$

在理想情况下，每个在运行的服务器都满负荷运行，因此服务器的有效利用率为：

$$\rho' \approx 100\%$$

3 仿真分析

使用 CloudSim^[12] 云计算仿真工具对校园云平台进行仿真，对负载均衡和贪心策略下的资源分配和调度进行分析。

3.1 CloudSim 简介

CloudSim 是澳大利亚墨尔本大学 Rajkumar Buyya 教授领导的网格实验室开发的云计算仿真工具，是在离散事件模拟包 SimJava 上开发的函数库，在 2.0 以后的版本中将 SimJava 从框架中移除，并继承了 GridSim 的编程模型。其主要目标是在云基础设施上，对不同应用和服务模型调度和分配策略的性能进行量化和比较，达到控制使用云计算资源的目的。CloudSim 提出了一个通用、可扩展的仿真框架，用以支持无缝建模和模拟。

3.2 实验结果分析

使用 CloudSim 构建仿真环境。在仿真环境中，主要关注用户的到达、离开和使用虚拟机的时间，以及云平台的资源利用率。假定总用户数不超过 3000 个，每个服务器可以提供 30 台虚拟机。由于云数据中心的设置必须保证满足所有用户到达时的需求，即保证充分服务，因此数据中心需要 100 台服务器。每个服务器的参数设置为 Linux 系统，使用 Xen 虚拟化技术，每个服务器拥有 4 个 8 核 CPU，32G 内存。在虚拟机的分配中，不考虑存储空间和带宽，因此设置主机的存储空间和带宽为无穷大。每台服务器可以提供 30 台 1 核、1G 内

存的虚拟机。

3.2.1 稳定状态下的效用函数

在稳定状态下,系统大小的均值 L 即为泊松分布的均值,即 $L=\lambda/\mu=2000$ 。图 6 显示了当初始用户为 0、经历了 800min 之后,运行的虚拟机个数与时间的关系。可以看到,在 $T=300(\text{min})$ 左右时,虚拟机的个数趋于稳定并在 2000 个左右波动。

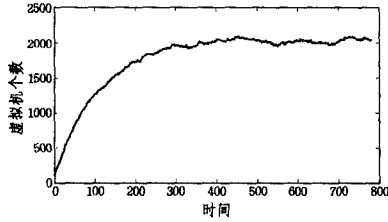


图 6 系统从初始化到稳定的虚拟机个数

在分配过程中,每当有用户请求到达和离开时,服务器资源都要进行一次插入排队,时间复杂度为 $O(\lg n)$ 。提供虚拟机的物理服务器一直是当前可提供最多资源的服务器,因此不需要虚拟机的迁移。

得到校园云平台的成本效用公式:

$$U = \frac{1}{E(C_V)} = \frac{\sum V_i t_i}{A + \int P_E dt}$$

式中, $\sum V_i t_i$ 表示虚拟机数量与使用时间的乘积,也就是虚拟机的总使用量。可知,虚拟机的总使用量可以表示为系统大小的均值与平均系统时间(每个虚拟机的使用时间)的乘积,即:

$$\sum V_i t_i = LW = \frac{\lambda}{\mu} \frac{1}{\mu} = \frac{\lambda}{\mu^2}$$

$\int P_E dt$ 表示云平台使用过程中的总能耗。根据上述研究,云平台的总功率可以近似表示为活动的服务器数量 n 的一个系数为常数的一次线性函数, P_E 可以表示为:

$$P_E = Bn + C$$

可得

$$\int P_E dt = \int (Bn + C) dt = CT + B \int ndt$$

可得效用函数:

$$U = \frac{\sum V_i t_i}{A + \int P_E dt} = \frac{\lambda/\mu^2}{A + CT + B \int ndt}$$

在基于负载均衡的策略中,平衡状态下活动的服务器数为 c (服务器的总数),因此

$$\int P_E dt = CT + B \int ndt = BcT + CT$$

得到效用函数为:

$$U_e = \frac{\sum V_i t_i}{A + \int P_E dt} = \frac{\lambda/\mu^2}{A + (Bc + C)T}$$

3.2.2 负载均衡策略下的效用函数

根据负载均衡策略,在初始用户数为 0、经历时间 $T=100\text{min}$ 的情况下,活动的服务器数与时间的关系如图 7 所示。可知,活动的服务器数瞬间到达 30 个,所有的服务器都

开始并保持运行。

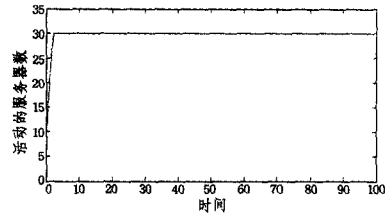


图 7 负载均衡策略下活动的服务器数

3.2.3 贪心策略下的效用函数

在使用贪心策略时,假设每个服务器可以提供 k 个虚拟机,则活动的服务器数的均值为:

$$n = \frac{ck}{\lambda/\mu} = \frac{ck\mu}{\lambda}$$

可得效用函数:

$$U_g = \frac{\sum V_i \cdot t_i}{A + \int P_E dt} = \frac{\lambda/\mu^2}{A + CT + \frac{Bck\mu T}{\lambda}}$$

使用贪心策略时,在初始用户数为 0、经历时间 $T=800\text{min}$ 的情况下,活动的服务器数与时间的关系如图 8 所示。根据上述分析可知,用户数量的均值为 2000,每台服务器可以提供 100 台虚拟机,因此活动的服务器数大约是 20 台。而从仿真的情况来看,活动的服务器数的均值大约是 23 台,略大于理论值 20 台,这是因为用户的频繁到达和离开不会使所有服务器都满负荷运行。但是,跟负载均衡策略相比,贪心均衡策略已经具有较高的资源利用率。

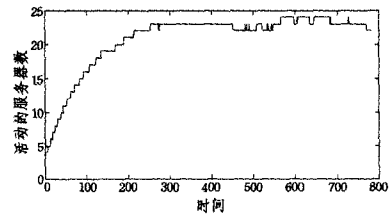


图 8 贪心均衡策略下活动的服务器数

结束语 对上述仿真结果进行比较可知,使用贪心策略能够明显减少活动的服务器数量。

结合对校园云平台效用函数的推导, U_e 表示负载均衡策略下的效用函数, U_g 表示贪心策略下的效用函数,对两种策略下的效用分析结果进行对比可得:

$$\frac{U_e}{U_g} = \frac{\lambda/\mu^2}{A + (Bc + C)T} \div \frac{\lambda/\mu^2}{A + CT + \frac{Bck\mu T}{\lambda}}$$

$$= \frac{A + CT + BcTk/\frac{\lambda}{\mu}}{A + CT + BcT}$$

根据式(6)~式(9)可知,负载均衡策略和贪心策略效用差别在于 $k/\frac{\lambda}{\mu}$,也即每台服务器能够提供的虚拟机数和用户使用的虚拟机数的均值。

在实际应用中,可以采取不同策略,对实际的投入、提供的虚拟机数量和使用时间进行统计分析,并同仿真分析的结果进行对比,以得到更加科学有效的评价依据。

云计算应用在校园云平台中,不仅可以提高资源利用率,

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