

云计算中虚拟机资源自动配置技术的研究*

薛涛, 刘龙

(西安工程大学 计算机科学学院, 西安 710048)

摘要: 针对云资源管理者所面临的负载动态变化以及弹性资源需求等问题, 提出一种虚拟机资源的自动配置管理技术, 把强化学习技术引入云虚拟资源的管理, 将虚拟机的配置管理过程建模为马尔可夫决策模型, 根据系统的运行状态以及输入负载的动态变化自动决策添加或删除虚拟机的行为。实验结果验证了该技术能够根据负载的动态变化完成云虚拟资源的自动配置管理任务, 及时响应终端用户的实时性任务请求, 保证了云资源使用者的SLA需求。

关键词: 云计算; 自动配置; 强化学习; 马尔可夫决策模型

中图分类号: TP393

文献标志码: A

文章编号: 1001-3695(2016)03-759-06

doi:10.3969/j.issn.1001-3695.2016.03.026

Research of automatic configuration technology for virtual machine resources in cloud computing

Xue Tao, Liu Long

(College of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: For the problems cloud resource managers faced, such as the dynamic changes of incoming load and demand elasticity of resources. This paper proposed an automatic configuration management technology for virtual machine resources. It introduced reinforcement learning technique into the cloud virtual resource management, and modeled configuration management process of the virtual machine as a Markov decision model. According to the running state of the application system and the dynamic changes of the input load, this technology could make an automatic decision to add or remove a number of virtual machines. Experimental results show that this technology can complete the tasks of automating configuration of virtual resource management according to the changing load, respond to end user's in a timely manner, and ensure the SLA requirements of cloud users.

Key words: cloud computing; automatic configuration; reinforcement learning; Markov decision model

0 引言

云计算作为一种新型的IT发布模式,催生出大量的新型互联网服务。根据服务类型的不同,云计算通常分为基础设施即服务(infrastructure as a service, IaaS)、平台即服务(platform as a service, PaaS)和软件即服务(software as a service, SaaS)。通过虚拟化技术,虚拟机可以共享物理机资源池,云计算资源提供商可以向云计算应用提供商提供虚拟机资源来部署自己的应用系统。其中,虚拟资源的调度管理技术至关重要,它将直接影响到云资源利用率、服务能力以及SLA。另外,随着云计算资源规模的扩增,手动及人工管理数目庞大的资源集群已经变得非常不现实,需要一种自适应的资源管理技术,以自动响应负载情况变化,并减轻管理人员的负担^[1]。

云计算环境下云计算虚拟资源的自适应资源管理问题是一个非常复杂的课题。目前,国内外的相关研究主要集中在虚拟资源动态伸缩技术和统一资源配置管理技术两个领域。

早期虚拟资源的分配方式大多为固定预分配,实际运行后,不再进行任何调整。这种方式虽然简单,但却不能完美适

应用户负载的动态变化。若申请者按照负载峰值来申请虚拟资源,在负载较低时刻会造成大量浪费;反之,若按照平均负载去申请,用户负载可能超出资源承载能力,造成系统瘫痪^[2,3]。随着技术发展,以亚马逊EC2为代表的弹性云服务采用以虚拟机为单位进行调整的方法,按需增减虚拟机数量,从而达到及时响应负载动态变化的目的^[4]。目前,虚拟资源动态伸缩技术的研究重点在于如何及时、快速地响应用户负载的动态变化^[5-9]。文献[10]提出了一个自治系统,利用自动的资源状态监测来确定虚拟资源的重新分配。文献[11,12]还将虚拟机资源的伸缩问题与控制论问题相结合,通过对未来用户负载进行预测,从而更准确地优化资源分配。但这些技术简单选取每个虚拟机的CPU、内存、网络带宽以及进程数等一种或两种指标,由于所选信息不全面,缺少宏观考虑,并不能有效地对虚拟资源作出精确调整。

云计算环境中的参与主体包括云计算资源提供商、云计算应用提供商以及终端用户^[13]。现有资源配置管理技术大多从云计算资源提供商的角度进行考虑,目的在于降低基础设施运营成本^[14]、最大化利润、提高云数据中心整体资源利用率等,并没有从云计算资源使用者以及终端用户的全局角度进行规

收稿日期: 2014-11-12; 修回日期: 2015-01-06 基金项目: 国家发改委高科技产业化项目(陕发改高技[2009]1365号)

作者简介: 薛涛(1973-),男,副教授,博士,主要研究方向为统一通信、云计算、发布订阅系统(xt73@163.com);刘龙(1988-),男,硕士研究生,主要研究方向为云计算。

划,如降低租用成本、提高使用者的 SLA 等^[15-17]。文献[18]根据云计算应用提供商服务的不同类型,采用不同的资源配置策略管理租用的云资源,但这种方式集中于最大化云计算应用提供商的收益,而忽略了用户的 SLA 需求。目前,已有文献[19,20]研究把强化学习、动态规划模型应用于云计算中的资源管理,把云计算中虚拟资源的配置问题作为学习对象,从控制规划的角度进行研究。文献[20]重点研究每个虚拟机的计算资源配置,并把它们作为学习对象,最终构建了分布式的虚拟机资源自动配置系统。但这种方式单纯地从每个虚拟机资源出发,忽略了虚拟集群的整体资源性能,并没有考虑外部因素的作用,也没有考虑管理人员的参与性。

为了满足云资源使用者的弹性资源需求,为云资源使用者提供快速、可靠的云资源,本文结合现有云资源配置管理技术,提出一种动态自适应性的虚拟机自动配置技术。其与已有工作相比,其创新点主要有以下三方面:

a) 从云应用提供商以及终端用户的角度出发,目的是在云应用供应商与云资源供应商之间建立一个中间层,为云应用提供商提供快速、可靠的云资源。

b) 把强化学习技术应用于云虚拟资源配置管理技术领域,将虚拟机的配置管理过程建模为马尔可夫决策模型,根据系统的运行状态以及输入负载的动态变化自动决策,多方面考虑各种因素,快速响应终端用户的实时性任务请求,及时对负载的动态变化合理配置资源。

c) 本文提出的云资源自动配置管理技术不仅综合考虑了每个虚拟机的指标信息,还选取了能够反映客户端指标的性能参数,如系统吞吐量以及响应时间,在保证系统正常运行的同时,还能满足云资源使用者的 SLA。

1 云计算虚拟机资源自动配置决策模型

1.1 虚拟机自动配置决策问题分析

由于云计算应用提供商可以动态租用云计算资源提供商提供的虚拟机资源,所以本文的目的是为云计算应用提供商提供高效的虚拟机资源配置技术。该技术的应用场景为云计算资源提供商与云计算应用提供商之间建立一个中间层,其应用场景如图 1 所示。

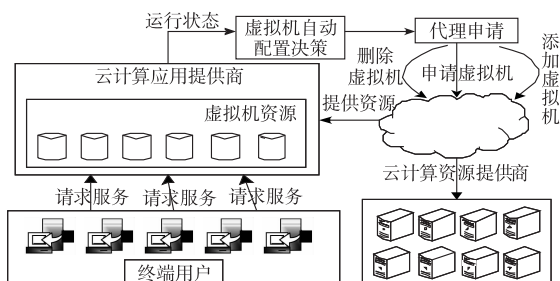


图 1 云计算虚拟机资源自动配置行为场景

由图 1 可知,本技术的应用场景主要在云计算资源提供商与云计算应用提供商之间提供一个中间层,这个中间层主要包括两个模块,即虚拟机自动配置决策模块和代理申请模块。其中,配置决策模块作为强化学习问题中的学习代理,主要把云计算应用提供商的系统状态以及虚拟机资源的运行状态作为输入,根据配置决策算法,把虚拟机资源的配置命令输出给代理申请模块;代理申请模块根据接收到的命令执行添加、删除

虚拟机资源的行为操作。

由上述可知,虚拟机自动配置决策模块为应用场景的核心,因此虚拟机自动配置决策过程为本文重点解决的问题。首先,把自动配置决策过程看成一个强化学习问题,强化学习问题首先要求把自动配置过程建模为马尔可夫决策模型(Markov decision model,MDP),建模过程如 1.2 节所述。

1.2 虚拟机自动配置决策建模

1.2.1 马尔可夫决策模型描述

强化学习技术中首先要对随机的、离散状态、离散时间进行数学建模,在实际应用中,最常用的是马尔可夫决策模型,并且很多强化学习问题基于的一个关键假设就是学习代理与环境交互的过程可以被看成是一个马尔可夫决策过程,因此强化学习的研究主要集中于对 Markov 问题的处理。在强化学习过程中,学习代理通过观察与环境的交互来改善自己的行为。一个马尔可夫决策过程模型由一个五元组组成^[21,22]。

$$\langle S, A, p_{(s,s')}^a, R, \gamma \rangle \quad (1)$$

a) 状态集合 S : 状态是在某一时刻对系统在所处环境中的描述。最一般化的是平铺式表示,即对世界所有可能状态予以标号,以 $s_1, s_2, s_3, \dots$ 这样的方式表示。这种情况下,标号状态的数目也就代表了状态空间的大小。

b) 动作集合 A : 动作会参与改变当前系统的状态。MDP 的一个关键部分便是提供给学习代理的用于作决策的动作集合。当某一动作被执行,系统状态将会发生改变,根据一个已知的概率分布转换为另一状态。

c) 状态转移函数 $p_{(s,s')}^a$: 它描述了系统的动态特性,表示系统在某个状态 s_i 下执行某一动作 a ,得到的是一状态的概率分布 P ,记为 $p_{(s,s')}^a$ 。

d) 回报函数 R : 它表示在状态 s_i 下执行动作 a 所能得到的立即回报,当回报值为正时,表示对系统作出的行为 a 的奖励,值越大,表示奖励越多;当回报值为负时,表示对系统作出的行为 a 的惩罚,它体现了对系统的行为约束。

e) γ : 它是折扣因子,又称衰减系数,在 $[0, 1]$ 间。

MDP 的根本问题就是在当前状态 s , 根据回报函数的约束,求解一个最优策略 π , 根据策略 π 作出一定的动作 a , 使系统状态按照转移函数改变。

1.2.2 状态空间表示

虚拟机自动配置决策模型中的状态主要是云计算中虚拟机资源的相关信息,配置决策模块需要在这个状态空间内求解最优的虚拟机资源配置行为。MDP 要求状态离散化,因此把当前阶段云应用提供商所申请并且处于运行状态的虚拟机资源的数量作为配置决策模型中的状态空间,用 s_i 表示,则配置决策模型的状态空间表示如下:

$$S = \{s_1, s_2, \dots, s_i\} \quad (2)$$

1.2.3 动作空间表示

当用户负载急剧增大时,虚拟机资源虚的 CPU、内存、带宽利用率相应增大,因此,云计算应用供应商的应用系统处理性能会急剧下降,甚至造成应用系统整体瘫痪,不能很好地响应用户的负载请求。此时,为了快速响应用户请求,需要加强云计算应用系统的总体处理能力,因此需要向云计算资源供应商申请更多资源来保证系统的处理能力,满足用户以及云计算应用提供商的 SLA。反之,当用户负载降低时,虚拟机资源的利用

率都很低,在保证应用系统总体处理能力的基础上,可以适当删除部分虚拟机,以此节省云计算应用提供商的租用成本。

通过上述两种情况的自动配置行为,虚拟机资源的数量随着用户负载及应用系统的处理能力动态增减,及时、快速地满足了用户需求,保证了用户及应用提供商的 SLA。因此,虚拟机自动配置决策模型中的动作主要包括添加虚拟机:add 和删除虚拟机:remove 两种行为。另外,自动配置决策模块进行配置决策后,也可以输出无须添加或删除节点的行为,即要求增减虚拟机的数量为 0,称之为空操作 no-action。强化学习过程同样要求行为动作离散化,因此设定配置决策模块输出每一次决策行为时,添加或删除虚拟机的数量为固定值 k 。当应用系统规模很大时, k 值可以设定大些,反之可以设定较小值,本文中设定 $k=3$ 。由此可得虚拟机自动配置决策模型的动作空间如式(3)所示。

$$A = \{\text{add}^{(k)}, \text{remove}^{(k)}, \text{no-action}\} \quad (3)$$

1.2.4 状态转移函数

状态转移函数又称为状态转移概率,在虚拟机资源自动配置模型中,考虑到系统实际运行时的很多不确定性因素,即由于云计算应用提供商与云计算资源提供商的通信不流畅,或者云资源提供商本身资源不充分等原因,系统实际增加或删除虚拟机的数量与自动配置决策模块要求增减虚拟机的数量不一致。在此,引入概率的模式来考虑应用系统从当前状态 s ,执行动作 a 后,转移到下一状态 s' 的概率。因此定义状态转移函数,如式(4)所示。

$$P_{(s,s')}^a = \begin{cases} \frac{n - |n - n'|}{n} & n \neq n' \text{ 且 } n \neq 0 \\ 1 & \text{其他} \end{cases} \quad (4)$$

其中: n 为配置决策模块要求增减虚拟机的数量, n' 为应用系统中虚拟机资源实际的增减数量。 $n=0$ 时的情况即为 1.2.3 节所描述的不增减虚拟机的动作行为。

1.2.5 回报函数表示

回报函数应该对配置决策模块的决策行为作出奖励或惩罚,并且会影响决策模块后来的决策行为,所以在一定程度上体现出系统管理员的控制策略。例如,当应用系统作出添加或删除虚拟机的操作后,虚拟机资源的 CPU、内存、带宽利用率等系统状态会相应变化。此时,如果应用系统的处理能力增加,满足了云计算应用提供商所需的系统处理要求,也满足了用户的实时性负载请求,可以设定大的回报值;反之,经过调整后,应用系统的处理能力没有相应改善,或者更差,没有满足用户及应用提供商的 SLA,可以设定小的回报值。

由此,本文定义回报函数为收益 gain 与成本 cost 的比值,如式(5)所示。

$$R = \frac{\text{gain}}{\text{cost}} \quad (5)$$

其中:收益 $\text{gain} = \frac{\sum_{i=1}^m g(x_i)}{m}$, m 为性能指标个数,在此,本文选取系统吞吐量和响应时间两个参数作为性能指标,因此 $m=2$; $g(x_i)$ 表示每个性能指标的收益。其中

$$g(x_i) = \begin{cases} 1 & \text{如果 } x_i \text{ 满足 SLA 要求} \\ e^{-p * (\left| \frac{x_i - x'_i}{x'_i} \right|)} & \text{其他} \end{cases}$$

其中: p 为控制参数; x'_i 为上一次性能参数值。成本

$$\cos t = 1 + \frac{\sum_{i=1}^z (1 - u_i^k)^{\frac{1}{k}}}{z}$$

其中: k 为控制参数;本文选取 CPU、内存、带宽三个指标作为利用率参数 u_i ,因此 $z=3$, $\bar{u}_i$ 为云计算应用提供商所有申请虚

拟机的利用率参数 u_i 的平均值,即 $\bar{u}_i = \frac{\sum_{i=1}^n u_i}{n}$, n 为所有虚拟机的个数。

2 云计算虚拟机资源自动配置决策过程

2.1 虚拟机资源自动配置决策过程描述

上文对云计算虚拟机资源自动配置决策的学习过程建立了马尔可夫决策模型。通过对文献[19,20]把强化学习引入云计算虚拟资源的配置研究,以及文献[23~26]把强化学习引入甲板调度、自动控制等实际应用场景的综合研究。本文选用 Q-learning 算法来对虚拟机资源自动配置决策进行求解,即在当前云计算应用供应商系统的运行状态下,配置决策模块会从 1.2.3 节中的动作空间根据一定标准来选用一个最优动作,以此来改善系统的状态及运行处理能力。

Q-learning 算法是强化学习算法的一种,与其他强化学习算法相比,Q-learning 算法简单高效,因此更加适合虚拟机资源自动配置决策的应用场景,因为云计算应用系统需要对短时期大量的负载请求作出快速响应。Q-learning 维护着一张由状态—动作值索引的二维查询表,又叫 Q 表,这个表存储的是状态—动作值 $Q(s,a)$ 的估计值。 $Q(s,a)$ 作为最优策略求解的评价标准,其递归形式表示如下:

$$Q(s,\alpha) = R(s) + \gamma \sum_{s' \in S} P_{(s,s')}^a V(s') \quad (6)$$

其中: $V(s') = \max Q(s,a)$; γ 是折扣因子,保证了算法的收敛性。可以看出, $Q(s,a)$ 为系统当前状态的回报值 $R(s)$,与当前状态 S 采取动作 a 后,云计算应用系统以一定概率 $P_{(s,s')}^a$ 转换到新状态 s' 时的值函数之和。

在云应用系统实际运行中,根据用户负载的输入及系统的运行状况,配置决策模块需要不断作出配置决策,选择最优的策略,执行最优的动作。在此期间,要按照式(7)不断更新 Q 表中的 $Q(s,\alpha)$ 估计值,使估计值越来越接近于真实的最大值,从而使决策模块作出更接近于实际情况的最优策略。

$$Q(s,a) = Q(s,a) + \alpha [R(s') + Q(s',a') - Q(s,a)] \quad (7)$$

其中: α 为学习率,指明要给相应的更新多少信任度。

由于寻找虚拟机资源最优配置策略的过程就等同于使 $Q(s,a)$ 的估计值越来越接近真实最大值的过程,所以在执行完每一步动作后都需要对 Q 表中的估计值更新。在此基础上,选用式 $\pi^*(s) = \arg \max_a Q^{\pi^*}(s,a)$ 所定义的贪婪策略,即在当前状态 s ,选用 $Q(s,a)$ 最大的估计值所对应的动作 a 来执行。

在虚拟机资源自动配置决策过程中,正是由于:a)对 Q 表中 $Q(s,a)$ 估计值的不断更新,使之更接近于真实的最大值;b)按照贪婪策略选择使 $Q(s,a)$ 估计值最大的最优策略,这两种行为交替迭代,从而保证了云应用系统的自适应性以及弹性化扩展。

2.2 虚拟机资源自动配置决策算法描述

通过 2.1 节描述可以了解到:a)在每次决策时,决策模块

都需要根据当前状态 s , 在 Q 表中选取 $Q(s, a)$ 估计值最大的动作; b) 每次执行完最优动作后, 系统要不断更新 Q 表中的 $Q(s, a)$ 估计值, 两种行为的交替执行构成了虚拟机资源自动配置决策的过程。由 1.2.2 节中建模过程的状态空间表示可知, 系统状态空间随虚拟机资源数量的增长不断增大, 因此当云应用系统规模很大时, 它们所需的虚拟机资源数相应增大, Q 表的设计将影响配置决策算法中学习性能及决策性能的整体效率。下面结合云计算虚拟机资源的使用场景, 简要描述 Q 表的设计思路。

目前已有大量文献[19, 20, 23 ~ 26]描述了强化学习在具体应用场景中 $Q(s, a)$ 函数值的定义及相关算法, 以及强化学习算法本身的改进及回报函数的定义等研究^[21, 22, 27, 28]。本文在对前人研究的基础上对 Q 表作如下改进:

Q 表是状态 s 、动作 a 以及值函数 Q 构成的二维表。为了提高 Q 表的查找效率, 本文对 Q 表按照相同状态值分组建立一级索引表, 并把状态值作为索引值; 最后根据哈希函数 $\text{hash}()$ 把一级索引值作为 key, 计算出 value 值, 存入内存中。其结构如图 2 所示。

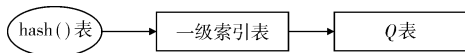


图 2 Q 表结构

下面给出虚拟机资源自动配置决策过程的算法描述。

a) 初始化 Q 表中所有的 $Q(s, a)$ 值为 0;
 b) 初始化 s_i ;
 c) repeat:
 (a) 根据贪婪策略选择状态 s_i 时, $Q(s, a)$ 值最大的动作 a_i ;
 (b) 执行动作 a_i , 收到回报值 R_i 以及下一状态 s_{i+1} ;
 (c) 根据式(6)计算 $Q'(s, a)$:
 if $(|Q'(s, a) - Q(s, a)| < \varepsilon)$
 { break; }
 else
 { $Q(s, a) = Q(s, a) + \alpha(Q'(s, a) - Q(s, a));$ } //更新 Q 值
 d) end

算法中的参数 ε 为给出的置信度区间, 表示 $Q(s, a)$ 估计值与真实值的接近程度。由于不可能计算出系统的真实 Q 值, 只能在系统运行期间通过对 Q 值的更新, 使之无限接近真实值, 所以系统每作出一步决策, 收到一个新的 Q 值 $Q'(s, a)$, 当新的值 $Q'(s, a)$ 与旧的值 $Q(s, a)$ 足够接近时, 说明此时的 Q 值已经足够接近真实的最大值, 此时作出的配置策略就足够接近真实的最优策略, 而这个接近度就取决于 ε 的定义。

决策配置模块每隔一定的时间间隔会调用一次上述的算法, 作出最优决策, 从而动态调整虚拟机数量, 使之满足云计算应用提供商的系统要求以及用户负载的动态变化。

3 实验与分析

3.1 测试方案

如引言所述, 本文的研究重点是为云计算应用提供商提供一个可以进行自动虚拟机资源配置的技术, 根据用户负载情况以及系统运行状况动态的增减从云计算资源提供商申请的虚拟机资源, 从而为用户以及云应用提供商提供一个高可用的、

满足 SLA 的中间层。因此实验的目的主要是测试本技术对于用户负载的动态变化需求, 在能否自动配置虚拟机资源、满足云应用系统可用性的同时, 保证用户的 SLA。

为了更好地观察本自动配置技术的性能, 本文选取文献[1]作对比实验。文献[1]提出了一种云计算虚拟机资源的自适应管理技术, 主要根据一个动态扩展算法来自动扩展虚拟服务器的数量, 该算法先根据云计算环境设定一个阈值, 接着选取每个虚拟机资源中的活动会话数目与阈值进行比较。如果所有虚拟机资源中的会话数目都大于此阈值, 就向云资源提供商申请一个新的虚拟机实例; 如果一些虚拟机资源中的活动会话数目小于阈值, 并且至少一个虚拟机没有活动会话, 就移除这个虚拟机资源。

实验之前先提出两个问题, 再提出对于这两个问题的实验方案, 后续实验主要根据设定的方案来对比研究本文提出的自动配置技术与文献[1]提出的动态扩展技术。

问题 1 本文提出的云计算虚拟机资源自动配置技术能否更加细致地提供虚拟机资源自动配置管理任务? 即能否根据用户负载及云应用系统的动态变化, 实时增加或减少虚拟机的数量?

方案 1 输入三组不同的负载, 观察配置决策模块输出的虚拟机资源调整数量。

问题 2 本自动配置技术能否满足用户和云应用提供商的 SLA? 即能否在保证云应用系统服务性能的同时, 满足用户的实时性访问需求?

方案 2 输入拥有不同幅度的用户负载, 观察云应用系统的响应时间。

3.2 测试环境及实验设定

本实验使用 VMware 公司的虚拟化软件 vSphere 创建 50 台虚拟机资源来模拟云平台, 即云计算资源提供商、云应用提供商在该平台上可以根据需求自动申请或释放虚拟机资源, 借助申请的虚拟机资源构筑自己的应用系统为终端用户提供服务。由于实验条件所限, 同样用虚拟化软件创建 25 台虚拟机运行用户负载程序来模拟用户负载动态访问云应用系统。负载虚拟机与云计算应用虚拟机配置如表 1 所示。

表 1 节点服务器资源配置

配置	应用虚拟机	负载虚拟机
操作系统	Ubuntu Linux 12.04	Ubuntu Linux 12.04
CPU	4 核	双核
内存	8 GB	4 GB
硬盘	50 GB	50 GB
数量	50	25

编写一个程序模拟用户负载, 该程序按一定周期执行, 分时启动线程, 最大可同时启动 200 个线程, 每台 VM 机上同时启动 20 个进程, 同时在 25 台负载虚拟机上运行, 这样就可模拟 100 000 个客户同时访问云应用系统。设置服务器集群初始运行节点数量为 4 个, 后面最多可以增加到 50 个, 最少可以减少到只有 2 台 VM 提供服务^[21]。负载输入率计算公式如式(8)所示。

$$\lambda = \frac{\text{load}}{\text{time}} \quad (8)$$

其中: λ 为负载输入率, 单位为 rea/s (每秒请求次数); load 为

客户访问数量;time 为访问时间。为了体现实验结果的对比度,本文设定用户负载的标准周期 $T=180\text{ min}$ 。

3.3 测试结果与分析

3.3.1 方案1实验结果

云应用系统申请虚拟机资源数量随负载变化情况如图3和4所示。

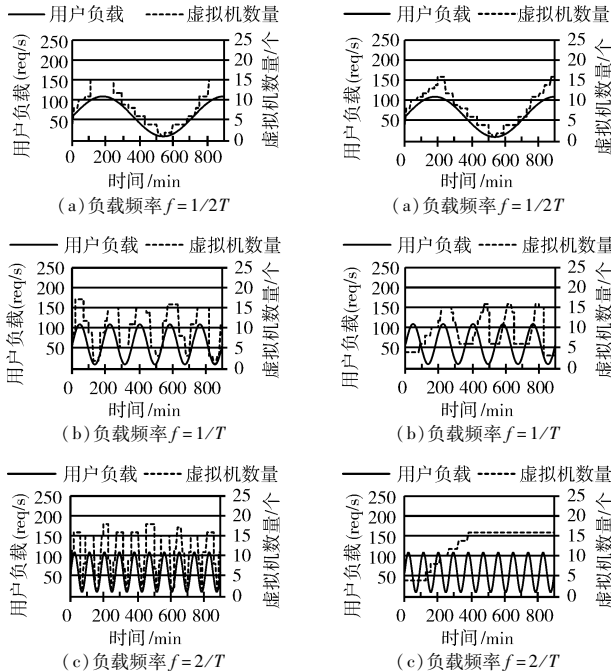


图3 本文自动配置技术在不同负载下的虚拟机数量调整情况

由图3和4可以看到,在两种虚拟机资源配置技术下,虚拟机资源实际的调整情况,其中实线代表用户负载随时间的变化情况,虚线代表虚拟机数量随时间的调整情况。下面按照频率的不同分三种情况加以比较:

a) 用户负载频率 $f=1/2T$

当用户负载为标准频率的一半时,两种技术对虚拟机资源的调整情况大致相同,都能根据用户负载的实时变化,动态改变所申请虚拟机的数量,从而在保证云应用正常运行的同时,满足用户需求。图4(a)显示文献[1]的扩展技术对虚拟机资源的调整更加频繁,这是由于文献[1]是按照所设定的阈值,以数量1为步长来调整虚拟机数量的,而本文配置技术在1.2.3小节中设定虚拟机每次的调整数量为3。

b) 用户负载频率 $f=1/T$

当用户负载为标准频率时,两种技术对虚拟机资源的调整情况有所差异。图3(b)显示,用户负载增加时,本文的自动配置技术依然能够在不同阶段作出精确调整,这是由于本文的配置决策模块基于强化学习,拥有自动学习的能力,能够学习到当前负载的变化情况,并为以后的虚拟机资源调整决策提供支持,从而能够快速根据负载变化作出调整决策。

而从图4(b)可以看出,文献[1]的扩展技术在负载输入的初始阶段有了较大延迟,对于用户负载没有及时添加一定数量的虚拟机资源,随着用户负载的输入以及应用系统的运行,渐渐能够正常调整虚拟机数量,但是仍然有一定程度的滞后性。这是由于在每次调整决策时,它都要求系统中所有虚拟机资源的活动会话数与阈值进行比较,随着虚拟机规模的增加,

这种比较花费大量的时间,并且这种技术每次改变虚拟机的数量为1,当用户负载较快的变化时,这种大量的比较之后在按步长为1进行调整,就会带来延迟。

c) 用户负载频率 $f=2/T$

当用户负载为标准频率的一倍时,面对用户负载的急剧变化,两种技术对虚拟机资源的调整出现了很大的不同。如前所述,由于本技术带有自动学习的功能,能够对负载的变化趋势进行学习,并根据Q-learning算法随时更新Q估计值,使之越来越接近真实最大值,作出满足系统要求的最优决策配置,如图3(c)所示。

而图4(c)显示,由于负载的急剧变化,加之文献[1]的扩展算法对每个虚拟机活动会话与阈值的比较,只有当所有会话都大于阈值后,才作出调整。因此,在负载输入的初始阶段,虚拟机数量的调整就带有大量延迟,一直到 $T=200\text{ min}$ 后,集群才开始逐渐增加虚拟机数量,当虚拟机数量增大到16时,以后基本不再变化。可以看出,面对高输入的用户负载,文献[1]的扩展技术不能很好地对虚拟机资源数量做出配置调整。

由此可见,本实验解决了方案1提出的问题,说明本技术能够完成云应用系统中资源的自动配置管理任务。

3.3.2 方案2实验结果

集群响应时间随负载变化情况如图5所示。

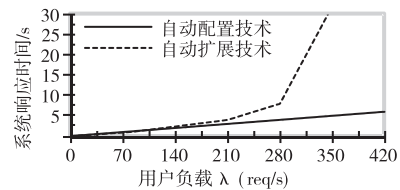


图5 系统响应时间随用户负载变化情况

为了验证两种技术是否满足用户以及云应用提供商的SLA,本文选取系统响应时间作为参考,如图5所示。图5中,实线代表的是本文的虚拟机资源自动配置技术,虚线表示文献[1]中的虚拟机扩展技术。可以看出,本文的自动配置技术使云应用系统的响应时间基本保持稳定,呈缓慢上升趋势,而文献[1]的自动扩展技术在不同阶段,系统的响应时间差异很大。下面分阶段说明:

a) $\lambda \in [0, 100]$

当负载小于 100 req/s 时,两种技术对虚拟机资源的配置情况相似,系统响应时间都维持在 1.5 s 之内,可以为用户提供正常服务,满足SLA的性能要求。

b) $\lambda \in [100, 280]$

当负载 $\lambda \in [100, 280]$ 时,本文虚拟机资源自动配置技术已渐渐显示出了对云资源管理配置的优势,文献[1]技术下系统响应时间的增长速率明显大于本文技术下系统响应时间的增长率。本文技术下系统响应时间维持在 $[1.5 - 3.5]\text{ s}$,而文献[1]技术下的响应时间从 1.5 s 达到了 7.5 s , 7 s 左右的响应时间明显使用户感觉到系统运行得不流畅,用户已不能很好地享受云应用服务。

c) $\lambda \in [280, 420]$

当负载 $\lambda \in [280, 420]$ 时,本文自动配置技术的优势更加明显,随着用户负载不断增大,系统响应时间始终按照一个较低的速率缓慢上升,而文献[1]中仅仅根据阈值来进行决策的自动扩展技术显然已不能适应如此高的用户负载,实际情况可

能是,当大量用户同时访问云应用系统时,文献[1]技术下的云应用系统没有及时作出虚拟机资源的扩展管理,从而使系统崩溃。本文提出的自动配置管理技术由于具有自动学习的功能,能够在回报函数设定的响应时间、吞吐量两个性能参数以及CPU、内存、带宽三个利用率参数的约束下,不断强化系统性能,自动学习适应负载的变化情况,更新学习系统的标准参数 Q 值,从而能够适应负载急剧增长的情况。

由此可见,本实验验证了方案2提出的问题,说明本技术在用户负载不断增大时,能够保证在云应用系统正常运行的同时,满足用户的实时性任务请求,满足SLA。

4 结束语

本文提出了一种云计算中虚拟机资源的自动配置管理技术,技术要点是把强化学习引入云资源的配置管理情景,使系统拥有自动学习的功能。本文首先解决了强化学习要求的马尔可夫决策建模过程,在此基础上解决了虚拟机资源的自动配置决策过程,最后通过实验验证以及与文献[1]对比研究说明了本技术的有效性,能够有效满足云计算中虚拟机资源的自动配置管理任务以及用户和云应用提供商的SLA。同时,本技术也存在一些不足,存在影响自动配置决策速度和粒度的因素,如自动配置模块作出决策调整的速度、云应用系统执行调整行为的时间以及1.2.3节中动作空间要求的每次决策行为允许增减虚拟机的数量等因素,大多数情况下,这都需要根据应用系统和应用程序特定的参数来进行更新和校准后才能实现本技术的鲁棒性,这种不足也为以后进一步的研究提供了方向。

参考文献:

- [1] 王晶,方伟,陈静怡,等. 云计算环境下的自适应资源管理技术综述[J]. 计算机工程与设计,2012,33(6): 2128-2132.
- [2] Nakada H, Hirofuchi T. Toward virtual machine packing optimization based on generic algorithm [C]//Proc of the 10th International Work-Conference On Artificial Neural Networks; Part II: Distributed Computing and Ambient assisted Living, 2009: 651-654.
- [3] Nakada H, Hirofuchi T. Eliminating datacenter idle power with dynamic and intelligent VM relocation [C]//Advances in Intelligent and Soft Computing. 2010: 645-648.
- [4] Skiena S S. The algorithm design manual [M]. Springer, 2008: 595-598.
- [5] De la Vega W F, Lueker G S. Bin packing can be solved within epsilon in linear time [M]. 1981:349-355.
- [6] Hyser C, Mckee B, Gardner R, et al. Autonomic virtual machine placement in the data center, HP Labs Technical Report [R]. 2007.
- [7] Padala P, Zou Kaiyuan, Shin K G, et al. Automated control of multiple virtualized resources [M]. Nuremberg: ACM EuroSys, 2009: 13-26.
- [8] Mylavarapu S, Sukthankar V, Banejee P. An optimized capacity planning approach for virtual infrastructure exhibiting stochastic workload [C]//Proc of ACM Symposium on Applied Computing. 2010: 386-390.
- [9] Zhou Wenyu, Yang Shonbao, Fang Jun, et al. VMC-tune: a load balancing scheme for virtual machine cluster based on dynamic resource allocation [C]//Proc of International Conference on Grid and Cooperative Computing. 2010:81-86.
- [10] Wood T, Shenoy P, Venkataramani A, et al. Sandpiper: Black-box and gray-box resource management for virtual machines [J]. Computer Networks, 2009, 53 (17): 2923-2938.
- [11] Hellerstein J I. Challenges in control engineering of computing systems [C]//Proc of American Control Conference. 2004:1970-1979.
- [12] Karamanolis C, Karlsson M, Zhu Xiaoyun. Designing controllable computer systems [C]//Proc of Conference on Hot Topics in Operating Systems. 2005.
- [13] Mohammed A B, Altmann J, Hwang J. Cloud computing value chains: understanding business and value creation in the cloud [C]//Proc of Economic Models and Algorithms for Distributed Systems. 2009:187-208.
- [14] LEE In. A model for determining the optimal capacity investment for utility computing [C]//Proc of Economic Models and Algorithms for Distributed Systems. 2009: 209-220.
- [15] Allspaw J. The art of capacity planning: scaling Web resources [M]. 2008.
- [16] Menasce D A, Almeida V. Capacity planning for Web services: metrics models and methods [R]. Upper Saddle River, NJ: Prentice Hall PTR, 2001.
- [17] Almedia V. Capacity planning for Web services [R]//Performance Evaluation of Complex Systems: Techniques and Tools performance Tutorial Lectures. London, UK: Springer-Verlag,2002:142-157.
- [18] Lopes R, Brasileiro F, Jr Maciel P D. Business-driven capacity planning of a cloud-based IT infrastructure for the execution of web applications [C]//Proc of IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum. 2010:1-8.
- [19] Tsoumakos D, Konstantinou I, Boumpouka C, et al. Automated, elastic resource provisioning for NoSQL clusters using TIRAMOLA [C]//Proc of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. 2013.
- [20] Rao Jia, Bu Xiangping, Xu Chengzhong, et al. A distributed self-learning approach for elastic provisioning of virtualized cloud resources [C]//Proc of the19st IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems. 2011.
- [21] 黄炳强,曹广益,王占全. 强化学习原理、算法及应用[J]. 河北工业大学学报,2006,35(6): 35-38.
- [22] 刘忠,李海红,刘全. 强化学习算法研究[J]. 计算机工程与设计. 2008,29(22): 5805-5809.
- [23] Tesauro G, Jong N K, Das R, et al. On the use of hybrid reinforcement learning for autonomic resource allocation[J]. Cluster Computing, 2007,10(3):287-299.
- [24] Song Xuan, Zhang Quanshi, Sekimoto Y, et al. Modeling and probabilistic reasoning of population evacuation during large-scale disaster [C]//Proc of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2013:1231-1239.
- [25] 李耀宇,朱一凡,杨峰,等. 基于逆向强化学习的舰载机甲板调度优化方案生成方法[J]. 国防科技大学学报,2013, 35(4): 171-175.
- [26] 徐昕,沈栋,高岩青,等. 基于马氏决策过程模型的动态系统学习控制:研究前沿与展望[J]. 自动化学报,2012, 38(5): 673-687.
- [27] Sutton R S, Barto A G. Reinforcement learning: an introduction [M]. [S. l.]: MIT Press, 1998.
- [28] 金卓军,钱徽,陈沈轶,等. 回报函数学习的学徒学习综述[J]. 智能系统学报,2009, 4(3): 209-212.