计算机科学
COMPUTER SCIENCE

面向能源感知的虚拟机深度强化学习调度算法研究

王杨民, 胡成玉, 颜雪松, 曾德泽

引用本文

王杨民, 胡成玉, 颜雪松, 曾德泽. 面向能源感知的虚拟机深度强化学习调度算法研究[J]. 计算机科学, 2024, 51(2): 293-299.

WANG Yangmin, HU Chengyu, YAN Xuesong, ZENG Deze. Study on Deep Reinforcement Learning for Energy-aware Virtual Machine Scheduling [J]. Computer Science, 2024, 51(2): 293-299.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230100031

面向能源感知的虚拟机深度强化学习调度算法研究

王杨民　胡成玉　颜雪松　曾德泽
中国地质大学(武汉)计算机科学与技术学院　武汉 430078
(1397995240@qq.com)

摘　要　随着计算机技术的快速发展,云计算技术成为了解决用户存储、算力需求的最佳方法之一。其中,基于NUMA架构的动态虚拟机调度成为了学术界和工业界关注的热点方向。但是,目前的研究中,基于启发式的算法难以对虚拟机进行实时调度,并且大多数文献没有考虑NUMA架构下虚拟机调度产生的能耗等问题。对此,提出了一种基于深度强化学习的大型移动云中心虚拟机服务迁移框架,构建了NUMA架构下的能耗模型;提出了自适应奖励的分层自适应柔性演员评论家算法(Hierarchical Adaptive Sampling Soft Actor Critic,HASAC);在云计算场景下,将所提算法与3种经典的深度强化学习方法进行实验对比。实验结果表明,所提改进算法在不同场景下可以处理更多的用户请求,且消耗的能源较少。此外,对算法中各种策略进行消融实验,证明了所提策略的有效性。

关键词:NUMA架构;深度学习;强化学习;能源感知;分层缓冲区
中图分类号　TP181

Study on Deep Reinforcement Learning for Energy-aware Virtual Machine Scheduling

WANG Yangmin,HU Chengyu,YAN Xuesong and ZENG Deze
School of Computer Science,China University of Geosciences(Wuhan),Wuhan 430078,China

Abstract　With the rapid development of computer technology,cloud computing technology has become one of the best ways to solve users' storage and computing power demands. Among them,dynamic virtual machine scheduling based on NUMA architecture has become a hot topic in academia and industry. However,in current research,heuristic algorithms are difficult to schedule virtual machines in real time,and most of the literatures do not consider the energy consumption caused by virtual machine scheduling under NUMA architecture. This paper proposes a service migration framework of large-scale mobile cloud center virtual machine based on deep reinforcement learning,and constructs the energy consumption model under NUMA architecture. Hierarchical adaptive sampling soft actor critic(HASAC) is proposed. In the cloud computing scenario,the proposed algorithm is compared with the classical deep reinforcement learning methods. Experiment results show that the improved algorithm proposed in this paper can handle more user requests in different scenarios,and consumes less energy. In addition,experiments on various strategies in the algorithm prove the effectiveness of the proposed strategy.

Keywords　NUMA architecture,Deep learning,Reinforcement learning,Energy perception,Layered buffer

1　引言

云计算(Cloud Computing)是随计算机技术的发展而出现的一种新型计算模式。在该模式下,使用者需要向云服务商发送请求并支付一定费用,云服务商则根据使用者的请求提供满足需求质量的服务[1]。近年来,随着多核并行技术和内存技术的发展,非统一内存访问架构(Non-Uniform Memory Access,NUMA)作为可以扩展的云计算结构被广泛应用和研究。

NUMA架构指一个物理主机有多个NUMA单元,其中每个NUMA单元拥有独立的CPU、内存和I/O等。用户向云服务供应商发送服务请求,将请求交给某一台物理主机进行处理,根据请求的大小调度一个或多个NUMA单元创建虚拟机,最后虚拟机处理完请求将结果返回给用户。调度过程如图1所示。

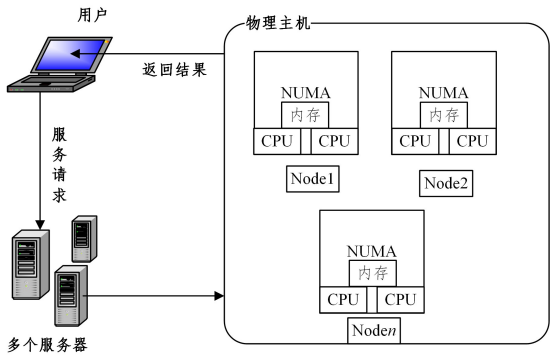


图1　NUMA架构虚拟机调度过程

Fig.1　NUMA architecture virtual machine scheduling process

近年来,学术界和工业界对NUMA架构下的虚拟机

到稿日期:2023-01-06　返修日期:2023-04-12
通信作者:胡成玉(huchengyu@cug.edu.cn)

调度开展了广泛的研究。比如，He 等开展了多核 NUMA 系统的竞争调度研究[2]，Cheng 等开展了在 NUMA 物理服务器上运行多个虚拟机性能的研究[3]。然而，这些研究大多忽略了 NUMA 架构下的能耗问题。NUMA 架构下的能耗降低不仅可以增加供应商的收益，而且能有效地支持社会绿色低碳循环经济的发展[4]。

NUMA 架构下降低数据中心能耗有两种有效手段：一是调节 CPU 的电压或频率，或关闭不必要的服务器；二是提高系统资源利用率，也即从虚拟机在物理机器上的分配和迁移入手，使得虚拟资源被充分利用。如果将大量虚拟机映射到少数几台物理主机上，将导致物理主机迅速发热过载，从而引发宕机；如果虚拟机映射到非常多的物理主机，则又会造成物理资源的浪费。如何在 NUMA 架构下合理分配虚拟机资源，从而降低能耗，是一个亟需解决的难题。

在 NUMA 架构下，虚拟机资源最优调度一般有两类求解算法：一类是传统的启发式算法，另一类则是近年来兴起的深度强化学习类方法。启发式算法可在决策空间内并行搜索可行解，但其是一种对环境变化敏感的并行搜索算法，计算量较大，很难对虚拟机资源进行实时调度[5]。

对于需要实时调度的不确定性复杂情景，强化学习是一类合适的方法。比如，Ren 等[6]为了应对动态变化的环境，提出了一种基于深度强化学习的大型移动云中心虚拟机服务迁移的进化框架。

本文主要贡献包括以下 2 个方面：

1）面向 NUMA 架构的虚拟机资源调度问题，建立了能源感知的 NUMA 虚拟机资源调度模型；

2）提出了分层自适应柔性演员评论家算法（HASAC）。针对轨迹采样率的计算分布，采用分层缓冲区思想，使得每次训练能够更好地掌握调度的全局信息。为了使采样得到的轨迹更有效，同时引入自适应奖励和对轨迹打分的机制。通过大量仿真实验验证了所提算法的有效性。

2 相关工作综述

2.1 统一内存访问架构下的资源调度

为了高效管理云数据中心资源，需要在考虑各种成本并遵守服务水平协议等的基础上采取一些具有鲁棒性的高效策略。

传统上，基于启发式的算法常用来解决数据中心资源调度问题。KRUEKAEW 等使用人工蜂群启发式调度来分配资源[7]，使得负载平衡效果较好；Ragmani 等使用混合蚁群优化算法[8]，有效分配云中负载，缩短了响应时间和处理时间。基于启发式算法对问题求解需要已知虚拟机序列，往往在离线场景中应用，无法达到实时调度的要求。

深度学习在预测方面取得了一些进展，提前预知虚拟机序列再进行调度是研究的一大方向。Tarafdar 等在虚拟机迁移之前预测 CPU 的利用率[9]，可以降低数据中心内的能源消耗，同时降低服务水平协议（Service Level Agreement，SLA）的比率。基于深度学习预测调度方法效果的优劣取决于预测的精度，一旦预测输出与实际相差太大，调度效果也就随之变差。

由于云计算的过程涉及的工作负载、虚拟机建立都是变化的，因此策略也应该是动态学习的。强化学习是在智能体与环境的交互中不断学习，根据奖励值优化策略，找到相对最优解。其由于不需要模型，且环境可以动态变化，因此在无法精确建模的情景下得到广泛研究。近年来，各种各样的强化学习策略被应用于对虚拟机和任务的调度上。比如，Wang 等使用基于 Q-learning 的方法指导多工作流程在基础设施即服务云上的调度[10]。该方法以工作流应用程序和异构虚拟机的数量作为状态输入，在没有专家事先知情的情况下寻求生产跨度和成本标准之间的相关平衡。其虽然有一定成效，但没有考虑可靠性、安全性、负载平衡等，且收集数据耗时长，成本高昂。最近，容器被逐渐用于应用程序部署和云平台资源分配的基本单元。Zhang 等[11]利用 SARSA 算法优化虚拟机在数据中心的调度过程。对比传统启发式算法，这种优化显著降低了能耗，且减少了迁移时违反服务水平协议的虚拟机数量，但未考虑网络带宽等问题。从供应商收益角度出发，Liu 等[12]联合接纳控制和虚拟机配置方案，采用马尔可夫决策过程（Markov Decision Process，MDP）框架，以集成的方式优化虚拟机的准入和安置决策，以使 CP 的运营收入最大化。

以上结构均为统一内存访问（Uniform Memory Access，UMA）架构。统一内存访问架构由于扩展性受限制而无法提供更大规格的虚拟机，这一类型的虚拟机承接的业务范围更小，不符合云供应商的利益需求。

2.2 非统一内存访问架构下的资源调度

现实场景的需求，使得现在大多数云计算中心是基于 NUMA 架构提供云计算、存储等服务。近年来，不少学者和工业界开展了 NUMA 架构下的虚拟机调度研究。比如，Hu 等[13]针对 NUMA 系统中整合虚拟机需要决定托管虚拟机的目标物理机器和 NUMA 单元这一问题，基于真实业务模型推导了系统模型，并在此基础上提出了一种混合启发式群智能优化算法。虽然该算法有效节约了能源，但是作为云服务商来说，应该优先考虑提供的服务次数，这样才能使得利益最大化。Sheng 等[14]将 NUMA 架构虚拟机调度问题建模为组合优化问题，然后使用深度强化学习求解。该研究考虑了完成请求的数量和分配内存利用率等指标，取得了一定的效果，但却没有将能耗这一关键因素计算在内。

NUMA 架构于异构的物理机器内，不同类型的物理机器和能耗之间的映射关系对调度结果有很大影响，因此挑选合适的 NUMA 单元调度虚拟机以完成尽量多的任务，同时耗费尽量少的能耗，是至关重要的。另外，现有的强化学习算法样本利用率低，存在收敛速度慢和奖励稀疏等问题[15]。基于以上分析，提出一种面向能源感知的虚拟机深度强化学习调度算法。

3 问题建模

为了使效益最大化，本章提出能源感知的 NUMA 架构虚拟机调度模型。该模型面向 NUMA 架构，将任务完成数量和能源消耗作为评价指标。3.1 节描述了问题原型，3.2 节把问题转化为马尔可夫决策过程进行求解。

3.1 虚拟机调度模型

在数据中心内，有 N 个异构的主机（$n_1,n_2,n_3,\cdots,n_N$）。每台主机拥有 M 个单元，每个 NUMA 单元有独立的运算单元 cpu 和存储单元 mem。

每一种类型的物理机配置相同，不同型号的物理机在不同的 cpu 利用率下产生的能耗为 pow_i，$i\in(0,10)$。pow 列表具体指的是（$pow_0,pow_1,pow_2,\cdots,pow_{10}$）。其中，$pow_0$ 表示开启机器但没有使用，利用率为 0；pow_{10} 表示物理机的利用率达到 100%。总能耗由式(1)计算：

$$pow=\sum_{l=0}^{L} pow_l\times\Delta t \tag{1}$$

本文将记录虚拟机状态改变时的能耗和时间并将其存放在队列中，l 表示当前记录的下标，L 表示队列的长度，pow_l 则为当前记录每秒产生的能耗，Δt 是改变能耗的间隔。

面对 R 个请求（$req_0,req_1,req_2,\cdots,req_r$），每个请求表示为一个元组（$vmid,cpu,mem,time,type$），$vmid$ 是虚拟机 id，cpu 和 mem 是虚拟机请求（删除）的资源，$time$ 是请求时间。$type$ 取{0,1}，0 表示创建，1 表示删除。整个元组表示 $time$ 时刻 $vmid$ 代表的虚拟机请求（删除）大小为 cpu 的运算单元和 mem 的存储单元。只要数据中心无法处理当前请求，则视为达到终止条件。

理想状态下，虚拟机调度的优化目标是可处理的请求越多越好，并且消耗的能量越少越好，因此该优化问题可表示为一个最大化约束优化问题，如式(2)所示：

$$\max\left(\|req\|+\frac{\lambda}{pow}\right)\quad \text{s.t.}\ \begin{cases}0<mem\leqslant rem_{mem}\\ 0<cpu\leqslant rem_{cpu}\end{cases} \tag{2}$$

其中，$\|req\|$ 表示请求个数；λ 为超参数；req 满足 cpu 和 mem 约束，也即申请的数目小于 rem，其中 rem 表示被选择的 NUMA 单元剩下的资源数目。

3.2 马尔可夫决策过程

基于上述虚拟机调度模型，建立马尔可夫决策过程。马尔可夫决策过程由一个五元组(s,a,p,r,γ)构成，s 表示状态，a 表示动作，p 表示状态转移概率，r 表示回报函数，γ 是折扣因子。代理策略 $\pi:s\to a$ 是从每个状态到动作之间的映射。在每个时间步，智能体基于观察到的状态 s 和学习到的策略 π 执行动作 a。环境状态转换根据 p，它表示在当前状态 s 下经过动作 a 后会转移到其他状态的概率分布情况。智能体会得到一个奖励，并且基于改变的环境继续执行动作，直到过程结束。

智能体的目标是尽快学习到最优策略，也即对于任意的状态 s，都有 $\pi\geqslant\pi'$，此时的 π 就是最优策略，π' 表示其他策略；并且在最优策略下满足 $v_\pi(s)\geqslant v_\pi'(s)$。$v_\pi(s)$ 是满足式(3)的价值函数。

$$v_\pi(s)=\sum_a\pi(a|s)\sum_{s',r}p(s',r|s,a)[r+\gamma v_\pi(s')] \tag{3}$$

其中，s' 是下一时刻的状态，其他参数的具体解释如下。

s（状态）：环境的状态空间，智能体所观察到的信息。在本调度模型中，t 时刻观察的状态 $Obs_t=[(o_t^{1,1},o_t^{1,2},\cdots,o_t^{1,m}),(o_t^{2,1},o_t^{2,2},\cdots,o_t^{2,m}),(o_t^{n,1},o_t^{n,2},\cdots,o_t^{n,m})]$。$o_t^{i,j}$ 表示 t 时刻第 i 台物理主机的第 j 个 NUMA 存储单元观察到的状态信息。本文中 $o_t^{i,j}=(cpu,mem)$，也即该 NUMA 存储单元剩余的 cpu 和 mem。

a（动作）：智能体基于观察到的状态，对环境采取动作。当有任务请求到达时，调度程序需要对到达的任务请求选择物理机或 NUMA 单元创建虚拟机。面对不同的任务类型，调度程序的行为是不同的。面对单 NUMA 任务请求，会选择单一的 NUMA 单元创建虚拟机；面对多 NUMA 任务（大型虚拟机任务 CPU 和内存需求较大，单一 NUMA 单元无法满足）请求，只需要选择单个物理机创建虚拟机。物理主机编号范围是{1,…,N}，虚拟机编号范围是{1,…,M+1}。选择物理主机后，再选择物理机上的 NUMA 单元，其中 M+1 表示该物理机上的所有 NUMA 单元。

p（状态转移概率）：在当前状态 s 下，执行动作 a 后转移到其他状态的概率。本文的状态转移在前一状态和请求下是确定的。t 时刻的状态 s，基于请求 r_t 的类型，相加（减）就能得到 t+1 时刻的状态 s'。

r（奖励函数）：环境对执行的动作给出的反馈信息，表示在状态 s 下执行动作 a 后获得的回报值，回报函数可表示为 $r(s,a)$。回报值是不断与环境交互累计的，RL 的目标是最大化累计奖励。

4 基于缓冲区分层策略的深度强化学习调度算法

现有的强化学习方法通过 TD-error 或 Q-value 这样的指标选择轨迹，然后训练。初始时刻使用 TD-error 训练时，得到的轨迹往往是初始过渡，不会得到最终状态的轨迹；使用 Q-value 训练时，越靠近最终状态的 Q-value 值越高，得到的轨迹往往具有较高的 Q-value。这些评价指标训练得到的轨迹都是基于当前状态且被独立计算，并没有考虑所有轨迹的全局信息。而虚拟机调度是一个过程，从调度第一个虚拟机开始到放置最后一个虚拟机结束，我们需要的不仅仅是基于某一时刻的高 Q-value 或 TD-error，还包含不同物理机状态下的轨迹。

为此，本文提出 HASAC 分层自适应柔性演员评论家算法。HASAC 算法的整体框架如图 2 所示，智能体选择合适的 NUMA 单元，能源感知数据中心收到智能体的决策，创建虚拟机，并返回观测到的状态和奖励。将智能体与环境交互得到的轨迹计算打分后放入分层缓冲区的相应位置，采样时结合轨迹的分数进行。详细过程如图 2 所示。

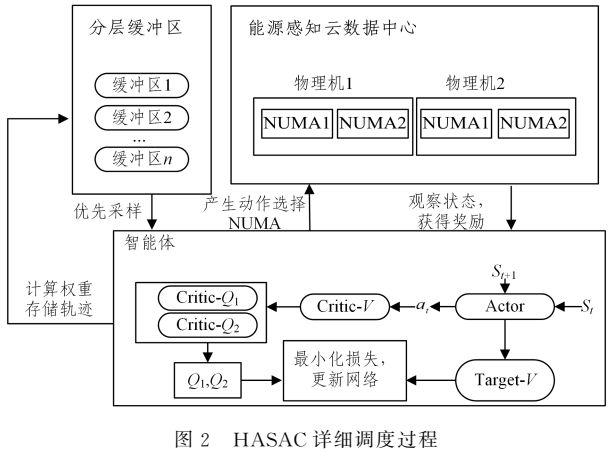


图 2 HASAC 详细调度过程

Fig. 2 Detailed scheduling process of HASAC

4.1 SAC 算法框架

柔性演员评论家算法(Soft Actor Critic)是一种采用异策略形式来优化随机策略的算法。柔性演员评论家算法包含 1 个 Actor 网络和 4 个 Critic 网络。4 个 Critic 网络包含两个当前网络和两个目标网络。在 t 时刻,智能体观察当前环境 s_t 并采取相应的动作 a_t,环境对智能体做出的动作给出反馈 $r(s_t,a_t)$,智能体获得下一时刻的状态 s_{t+1} 以及环境的奖励 r。在这一过程中,*Critic* 网络的损失函数如式(4)所示:

$$L(\phi_i,B)=\underset{(s,a,r,s',score)\sim B}{\mathrm{E}}[(Q_{\phi_i}(s,a)-y(r,s'))^2] \tag{4}$$

其中,ϕ 是 Q 函数的参数,$i\in\{1,2\}$ 表示有两个 Q 网络,B 是缓冲区。目标 y 由式(5)计算得到。

$$y(r,s')=r+\gamma(\min_{j=1,2}Q_{\phi_{\mathrm{targ},j}}(s',\tilde{a}')-\alpha\times\log\pi_\theta(\tilde{a}'|s')),\quad \tilde{a}'\sim\pi_\theta(\cdot|s') \tag{5}$$

其中,下一动作符号为 $\tilde{a}'$ 而不是 a',以凸显下一个动作必须从策略中重新返回。

得到 Q 值和目标值后,通过式(6)得到梯度更新 Critic 网络。

$$\nabla_{\phi_i}\frac{1}{|B|}\sum_{(s,a,r,s',d)\in B}(Q_{\phi_i}(s,a)-y(r,s'))^2,i=1,2 \tag{6}$$

同时,Actor 网络也通过式(7)所示的梯度更新策略网络。

$$\nabla_\theta\frac{1}{|\mathrm{B}|}\sum_{s\in\mathrm{B}}(\min_{i=1,2}Q_{\phi_i}(s,\tilde{a}_\theta(s))-\alpha\log\pi_\theta(\tilde{a}_\theta(s)|s)) \tag{7}$$

其中,$\tilde{a}_\theta(s)$ 是来自 $\pi_\theta(\cdot|s)$ 的样本,α 是超参数。最后通过式(8)更新目标网络,目标网络的更新由当前网络和原目标网络加权和得到。

$$\phi_{t\,\mathrm{arg},i}\leftarrow\rho\phi_{t\,\mathrm{arg},i}+(1-\rho)\phi_i,i=1,2 \tag{8}$$

4.2 分层缓冲区

智能体与环境交互得到(s,a,r,s')四元组轨迹,对该轨迹评判一个分值 *score*,然后将五元组$(s,a,r,s',score)$存储进分层缓冲区。分层缓冲区设置为 n 层,根据物理机的全局 cpu 利用率判断该轨迹应该被划分为哪一区域,随后从这 n 个缓冲区按照 *score* 优先采样得到训练样本,每一层选择样本数量相同,最后将样本交给智能体训练。这样就能确保智能体学习到的信息是全局信息,而不是某一特定状态下的信息。

4.3 自适应奖励

奖励可以设置为每处理一个请求就加 1,但这样会造成奖励稀疏。为了综合考虑全局信息和局部信息,奖励函数设置如下:

$$R=1+r_{\mathrm{local}}+cpu\_util_{\mathrm{global}} \tag{9}$$

其中,数值 1 是完成一个请求的基本奖励,r_{local} 指局部奖励,$cpu\_util_{\mathrm{global}}$ 是 cpu 利用率矫正。欠载会导致单个物理机利用率不高,过多启用不必要的物理主机而造成能源浪费;过载则是单个物理主机利用率过高,会造成同一台运行机器上所有任务效率下降和额外的能耗。为了尽量避免主机欠载和过载,将 r_{local} 定义如下:

$$r_{\mathrm{local}}=\begin{cases}-fabs(now-last)/2, & now<0.2\ \text{or}\ now>0.8\\ fabs(now-last)/10, & \text{else}\end{cases} \tag{10}$$

其中,*now* 表示当前主机的 cpu 利用率,$fabs()$表示对浮点数取绝对值。该公式表示如果欠载或过载,就会产生负的局部奖励,否则给予正值奖励。

cpu 利用率矫正的意义是,由于我们优先考虑的是完成任务的个数,当全局 cpu 利用率过高时,过载是为了完成当前任务,加上这一项就会补偿 r 局部的负值。

HASAC 算法针对在能源感知的数据中心上创建虚拟机形成了良好的处理方案。在能耗方面,用奖励 r 体现了能源感知问题;另一方面,HASAC 的分层缓冲区思想很好地解决了学习信息不全面这一问题。根据 cpu 利用率,直接将轨迹放入缓冲区不同层,旧的轨迹会在缓存区满时被新轨迹更新。每次处理请求,智能体根据状态前向传播计算 *action*,选择创建虚拟机的位置。处理 R 个请求的时间复杂度是 $O(R)$。算法 HASAC 的伪代码如算法 1 所示。

算法 1 HASAC 伪代码

Input:初始化策略参数 θ,Q 函数参数 ϕ_1,ϕ_2,清空缓冲区 D

设置目标网络参数 $\varphi_{\mathrm{targ},1}\leftarrow\phi_1$,$\phi_{\mathrm{targ},2}\leftarrow\phi_2$

1. For i=1,2,3,…,N do
2. 观察状态 s 选择动作 a~π_θ(·|s)然后执行动作
3. 观察下一步状态 s′,获得奖励 r,以及计算分数
4. 将(s,a,r,s′,score)存入缓冲区 D
5. For j=1,2,3,…,N do
6. 根据式(3)计算 Q 函数
7. 式(6)梯度下降更新 Q 函数
8. 式(7)梯度下降更新策略函数
9. 式(8)更新目标网络

5 仿真实验与分析

5.1 实验详细设置

数据集:实验数据集为 HUAWEI Cloud 一个月的真实调度虚拟机数据集。该数据集由位于华东的服务器采集 30 天得来,具体包含 15 种类型的虚拟机,创建请求 125 430 个,删除请求 116 313 个,实验选取其中 cpu 不小于 4 的请求。

实验环境:采用天河 k8s_venus 集群的 pytorch-20 应用进行训练。为了更好地体现算法的性能,使用 VMAgent 平台和 CloudSim 中的模型进行仿真实验。VMAgent 是由华东师范大学和华为云技术有限公司合作开发的云调度平台,调度算法在该平台上仿真模拟。CloudSim 是墨尔本大学的网格实验室和 Gridbus 项目推出的一款云仿真软件。为了更接近真实情况,引入 CloudSim 中 3 种类型的物理主机。不同型号的机器在不同 cpu 利用率时需要的能耗不同,详细信息如表 1 所列。

表 1 不同物理机在不同利用率下的能耗

Table 1 Energy consumption of different physical machines at different utilization rates

(W)

server	ML110G5	DL360G7	DL360Gen9
0%	93.7	54.6	45.0
10%	97.0	87.8	83.7
20%	101.0	98.5	101.0
30%	105.0	107.0	118.0
40%	110.0	115.0	133.0
50%	116.0	123.0	145.0
60%	121.0	131.0	162.0
70%	125.0	140.0	188.0
80%	129.0	153.0	218.0
90%	133.0	165.0	248.0
100%	135.0	178.0	276.0

为了实现复现性，提供超参数设置（如表 2 所列），保持这些值在算法中相同。其中学习率、折扣因子由网格搜索得到。

表 2 超参数配置

Table 2 Super-parameter configuration

参数	值
学习率	5×10^{-3}
折扣因子	0.85
批大小	2048
激活函数	Relu
ε_{start}	1
ε_{end}	0.01
缓冲区总容量	1×10^{6}
优化方法	Adam
隐藏层	MLP(128,256,256)

5.2 对比算法

为了证明所提算法的性能优势，将其与经典启发式算法 FirstFit 以及 3 种经典的深度强化学习算法 DQN，A2C，PPO 进行比较。其中，DQN 为离线算法，实现了 Double Q 和 Dueling Q 。采样开始使用专家指导。A2C 算法中，使用优势函数替代原始 Critic 网络。PPO 算法中实现了 PPO2，解决了 KL 散度求解计算复杂的问题。

5.3 实验结果和分析

每个算法进行 3000 次迭代，每 10 次记录一次结果，求均值，对比改进效果。数据中心采用 5 台 cpu 为 50 个、内存容量为 160GB 的物理主机。其中，ML110G5 两台，DL360G7 两台，DL360Gen9 一台。选取 tot_len 为评价指标，其意义是：在有限的 cpu 和存储容量下，能完成尽量多的虚拟机调度任务，则产生的收益最大。此外，还加入 cpu 利用率、内存利用率、能耗度量指标，使算法尽可能贴近实际调度过程。HASAC 训练时花费 10h，执行时在 3min 以内完成整个调度过程，与其他强化学习算法没有太大差别。实验结果如表 3 所列。

表 3 不同算法及其改进性能指标

Table 3 Different algorithms and their improved performance indicators

算法	完成任务总数	主机 cpu 分配率	主机内存分配率	总能耗/(kw·h)
FirstFit	103	0.832	0.950	0.803
DQN	103	0.832	0.950	0.949
PPO	101.923	0.829	0.946	1.048
A2C	101.942	0.829	0.947	0.971
HASAC	**102.003**	**0.831**	**0.949**	**0.829**

从结果可以看出，在小规模处理任务、并行性要求不高的场景下，深度强化学习似乎不占据太大的优势，比传统启发式算法效果要差。HASAC 无论是完成任务数量还是总能耗，均优于 PPO 和 A2C；完成任务的数量虽然比不上 DQN，但 DQN 平均完成每个任务的能耗为 0.009 kw·h，而 HASAC 为 0.008kw·h，整体每完成一个任务就可以节能 0.001kw·h。总体上看，完成更多的任务，往往主机的 cpu 和内存利用率更高，总能耗也更高。

在实际应用中，除了上述创建删除的一般场景外，大型企业项目或学术项目会在云服务器上一直运行。为了体现算法的泛化性，本文针对这种场景，进行仿真实验，结果如表 4 所列。

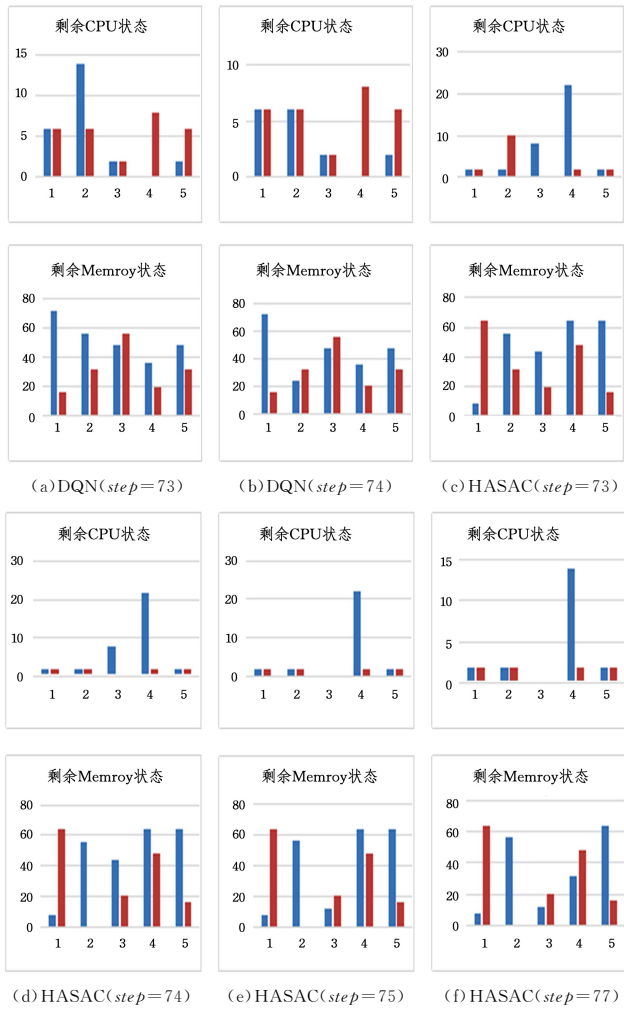
表 4 企业场景下不同算法及其改进性能指标

Table 4 Different algorithms and their improved performance indicators in enterprise scenario

算法	完成任务总数	主机 cpu 分配率	主机内存分配率	总能耗/(kw·h)
FirstFit	74	0.928	0.780	1.315
DQN	74	0.928	0.780	1.310
PPO	74.493	0.948	0.804	1.578
A2C	74.528	0.949	0.805	1.451
HASAC	**74.980**	**0.954**	**0.813**	**1.513**

在企业场景下，由于无法释放资源，更加考验算法对整体调度情况的掌控，强化学习类算法在完成数量上略优。HASAC 算法完成更多的任务，完成任务数对比 DQN，PPO，A2C 分别提升 1.3%，0.65%和 0.6%。4 种算法处理每个云任务的能耗分别为 0.018kw·h，0.021kw·h，0.019kw·h，0.020kw·h，表明 HASAC 算法在提升性能的同时不以耗费额外能耗为代价。本文的改进策略在不同场景下仍然有效，泛化性较好。

为了解释 HASAC 优于其他深度强化学习方法的原因，本文在实际调度过程中挑选一个序列，并在图 3 展示了具体的调度过程。



(a)DQN(*step*=73) (b)DQN(*step*=74) (c)HASAC(*step*=73)

(d)HASAC(*step*=74) (e)HASAC(*step*=75) (f)HASAC(*step*=77)

注：蓝色和深红色分别表示在 NUMA1 和 NUMA2 上分配资源。

图 3 在企业场景中不同步骤集群的状态（电子版为彩图）

Fig. 3 Status of clusters in different steps in enterprise scenario

比如，第 73 步时，接收(cpu,mem)请求(8,32)，DQN

选择在 2 号主机的 NUMA1 上部署，如图 3(a)所示。下一次请求仍然为(8,32)，但是此时集群中已经没有满足条件的物理机，调度结束，如图 3(b)所示。HASAC 在 73 步选择在 2 号主机的 NUMA2 上创建，之后面对下一次请求选择在 3 号主机的 NUMA1 上创建虚拟机。直到第 77 步，HASAC 面对(16,64)的请求没有足够的资源，结束调度，如图 3(f)所示。

5.4 消融实验

为了证明我们的改进都是有效的，分别设置对应取消分层、取消打分、固定奖励的算法 SAC、without_score、reward=1 作为对照，与本文的 HASAC 进行消融实验。实验对比结果如表 5 和表 6 所列。

表 5 一般场景下消去改进与原算法的对比

Table 5 Comparison between elimination improvement and original algorithm in general scenarios

算法	完成任务总数	主机 cpu 分配率	主机内存分配率	总能耗/(kw·h)
SAC	101.889	0.830	0.947	0.971
without_score	102.071	0.832	0.950	1.048
reward=1	101.371	0.819	0.934	0.977
HASAC	**102.003**	**0.831**	**0.949**	**0.829**

表 6 企业场景下消去改进与原算法的对比

Table 6 Comparison between improvement elimination and original algorithm in enterprise scenarios

算法	完成任务总数	主机 cpu 分配率	主机内存分配率	总能耗/(kw·h)
SAC	74.529	0.949	0.807	1.468
without_score	74.896	0.954	0.812	1.564
reward=1	74.104	0.943	0.799	1.427
HASAC	**74.98**	**0.954**	**0.813**	**1.513**

在一般场景下，取消分层和将奖励设置为 1 后，算法完成的任务数量更少了，但是产生的能耗却比 HASAC 更多。在不考虑打分而对分层缓存区随机采样后，完成的任务数量有小幅度上升，但是产生了更多的能耗。这是因为当物理主机处于欠载或过载时，它们获得的分数会下降，这样虽然智能体学习到全局的信息，但可能选择的轨迹更多，从而导致主机处于高能耗，最终产生更多的能耗。

在企业场景下，消去每一种改进算法性能都出现了不同程度的下降。取消打分时下降最明显，算法完成的任务更少却产生了更多的能耗。取消分层和将奖励设置为 1 虽然产生的能耗更少，但是完成任务的数量分别减少了 0.6% 和 1.1%，而平均每个任务产生的能耗却和 HASAC 相差不大。

综合以上两种情况来看，HASAC 的每一项改进都是必要的。分层使得智能体接受到的信息更全局化，给轨迹打分使得智能体需要考虑能耗问题，设置自适应奖励解决了稀疏奖励问题并且为打分提供了依据。

为了研究为什么我们的算法改进能产生更好的结果，对智能体学习过程的 *actor* 损失进行观察，结果如图 4 和图 5 所示。在两种场景下，HASAC 几乎都在固定的迭代次数内从最小的损失值开始到与其他 3 种方法同时收敛，损失值表变化越大说明可以学到的东西更多。所以 HASAC 在全局观察状态下，经过打分挑选得到更好的轨迹，然后学到了更多的信息，调度能力也更强，最终在处理调度问题上效果更好。

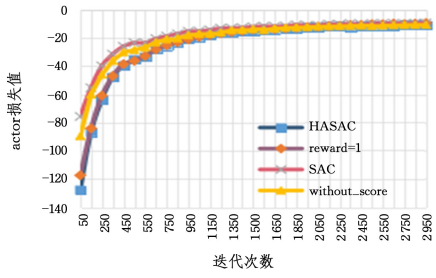


图 4 一般场景下迭代次数与 actor 损失的关系

Fig. 4 Relationship between the number of iterations and actor losses in general scenarios

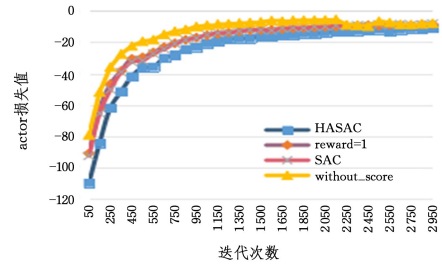


图 5 企业场景下迭代次数与 actor 损失的关系

Fig. 5 Relationship between the number of iterations and actor losses in enterprise scenarios

结束语 随着计算机的发展，云计算研究的价值和意义凸显，过去的研究大部分脱离了实际应用场景，现有的 NUMA 架构没有考虑能耗指标。本文在 NUMA 架构的基础上引入 cpu 利用率能耗模型，符合政府对绿色低碳循环经济的号召。

现有的强化学习方法都是基于当前状态单独计算轨迹，不利于智能体掌握全局信息。本文提出自适应分层缓冲区，并对采用轨迹进行打分。智能体训练时对全局各层次状态根据分数采样，实验结果表明，这种分层缓冲区改进能完成更多的任务且更加节能，应对不同场景仍然有效，鲁棒性较好。

尽管本文在算法层面的改进取得了一定效果，但应用层面针对的是小规模的物理主机调度，未来考虑尝试解决大规模虚拟机调度问题。

参 考 文 献

[1] DUAN W, HU M. Overview of Cloud Computing System Reliability Research[J]. Computer Research and Development, 2020, 57(1): 102-123.

[2] HE D, CHEN L, LIANG J, et al. NUMA-Aware Contention Scheduling on Multicore Systems [C]// 2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE). IEEE, 2021: 452-457.

[3] CHENG Y, CHEN W, WANG Z. Performance-monitoring-based traffic-aware virtual machine deployment on NUMA systems [J]. IEEE Systems Journal, 2015, 11(2): 973-982.

[4] BISWAI N K, BANERJEE S, BISWAS U. An approach towards development of new linear regression prediction model for reduced energy consumption and SLA violation in the domain of green cloud computing[J]. Sustainable Energy Technologies and Assessments, 2021, 45: 101087.

[5] MA O, JIANG X, JIN G. Heuristic dynamic threshold algorithm

for priority inversion suppression of LEDBAT protocol [J]. Computer Research and Development,2020,57(6):1292-1301.

[6] REN H,WANG Y,XU C. Smig-rl:An evolutionary migration framework for cloud services based on deep reinforcement learning[J]. ACM Transactions on Internet Technology(TOIT),2020,20(4):1-18.

[7] KRUEKAEW B,KIMPAN W. Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing[J]. International Journal of Computational Intelligence Systems,2020,13(1):496-510.

[8] RAGMANI A,ELOMRI A,ABGHOUR N,et al. FACO:A hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing[J]. Journal of Ambient Intelligence and Humanized Computing, 2020, 11:3975-3987.

[9] TARAFDAR A,KHATUA S,DAS R K. QoS aware energy efficient VM consolidation techniques for a virtualized data center [C]// 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing(UCC). IEEE,2018:114-123.

[10] WANG Y,LIU H,ZHENG W,et al. Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning[J]. IEEE Access,2019,7:39974-39982.

[11] ZHANG S,WU T,PAN M,et al. A-SARSA:A predictive container auto-scaling algorithm based on reinforcement learning [C]// 2020 IEEE International Conference on Web Services(ICWS). IEEE,2020:489-497.

[12] LIU L,XU J,YU H,et al. Joint Admission Control and Provisioning for Virtual Machines[C] // 2015 IEEE International Conference on Communications(ICC). London,UK,2015:332-337.

[13] HU K,LIN W,HUANG T,et al. Virtual Machine Consolidation for NUMA Systems:A Hybrid Heuristic Grey Wolf Approach [C]// 2020 IEEE 26th International Conference on Parallel and Distributed Systems(ICPADS). IEEE,2020:569-576.

[14] SHENG J,HU Y,ZHOU W. Learning to schedule multi-NUMA virtual machines via reinforcement learning [J]. Pattern Recognition,2022,121:108254.

[15] OH Y,KIMIN L,SHIN J. Learning to Sample with Local and Global Contexts in Experience Replay Buffers [C] // International Conference on Learning Representations(ICLR). 2021.

WANG Yangmin, born in 1999, postgraduate. His main research intrests include reinforcement learning and evolution computation.

HU Chengyu, born in 1978, Ph.D. professor, is a member of CCF (No. 40126S). His main research interests include evolutionary algorithm, reinforcement learning and cloud computing.

(责任编辑:柯颖)