Journal of System Simulation

Volume 34 | Issue 2 Article 8

2-23-2022

Job Scheduling and Simulation in Cloud Based on Deep Reinforcement Learning

Qirui Li
1.College of Computer Science, Guangdong University of Petrochemical Technology, Maoming 525000, China;, liqirui@gdupt.edu.cn

Xinyi Peng
2.School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China;, 1742043887@qq.com

Job Scheduling and Simulation in Cloud Based on Deep Reinforcement Learning

Abstract

Abstract: To solve the difficulty in job scheduling in the complex and transient multi-user, multi-queue, and multi-data-center cloud computing environment, this paper proposed a job scheduling method based on deep reinforcement learning. *A system model of cloud job scheduling and its mathematical model were built, and an optimization goal consisting of transmission time, waiting time, and execution time was obtained. A job scheduling algorithm based on deep reinforcement learning was designed, and its state space, action space, and reward function were given. A simulated cloud job scheduler was designed and developed, and simulated scheduling experiments were conducted on it.* The results show that compared with benchmark algorithms such as random scheduling, round-robin scheduling, firstfit, and optimal fit, the proposed algorithm could effectively reduce the overall makespan of the jobs.

Keywords

cloud computing, job scheduling, deep reinforcement learning, makespan, multi-user, multi-queue, multi-data-center

Recommended Citation

Qirui Li, Xinyi Peng. Job Scheduling and Simulation in Cloud Based on Deep Reinforcement Learning[J]. Journal of System Simulation, 2022, 34(2): 258-268.

第 34 卷第 2 期 系统仿真学报© Vol. 34 No. 2
2022 年 2 月 Journal of System Simulation Feb. 2022

基于深度强化学习的云作业调度及仿真研究

李启锐[1]，彭心怡[2*]
(1. 广东石油化工学院 计算机学院，广东 茂名 525000；2. 华南师范大学 数学科学学院，广东 广州 510631)

摘要：针对复杂瞬变的多用户多队列多数据中心云计算环境中作业调度困难的问题，提出一种基于深度强化学习的作业调度方法。*建立了云作业调度系统模型及其数学模型，并建立了由传输时间、等待时间和执行时间三部分构成的优化目标。基于深度强化学习设计了作业调度算法，给出了算法的状态空间、动作空间和奖赏函数。设计与开发了云作业仿真调度器，完成作业的仿真调度。*仿真结果表明，相比随机调度、轮转调度、首次适应、最佳适应等基准算法，提出的算法能够有效降低作业的整体完工时间。

关键词：云计算；作业调度；深度强化学习；完工时间；多用户；多队列；多数据中心
中图分类号：TP391.9 文献标志码：A 文章编号：1004-731X(2022)02-0258-11
DOI: 10.16182/j.issn1004731x.joss.21-0337

Job Scheduling and Simulation in Cloud Based on Deep Reinforcement Learning

Li Qirui[1], *Peng Xinyi*[2*]
(1. College of Computer Science, Guangdong University of Petrochemical Technology, Maoming 525000, China;
2. School of Mathematical Sciences, South China Normal University, Guangzhou 510631, China)

Abstract: To solve the difficulty in job scheduling in the complex and transient multi-user, multi-queue, and multi-data-center cloud computing environment, this paper proposed a job scheduling method based on deep reinforcement learning. *A system model of cloud job scheduling and its mathematical model were built, and an optimization goal consisting of transmission time, waiting time, and execution time was obtained. A job scheduling algorithm based on deep reinforcement learning was designed, and its state space, action space, and reward function were given. A simulated cloud job scheduler was designed and developed, and simulated scheduling experiments were conducted on it.* The results show that compared with benchmark algorithms such as random scheduling, round-robin scheduling, firstfit, and optimal fit, the proposed algorithm could effectively reduce the overall makespan of the jobs.

Keywords: cloud computing; job scheduling; deep reinforcement learning; makespan; multi-user; multi-queue; multi-data-center

引言

“云计算”是近年来兴起的一种新商业模式，它通过构建拥有强大计算能力的计算资源池，为企业、组织或个人提供弹性计算、带宽等资源服务，能够满足不同用户的需求[1]。根据对外提供服务的不同，云计算体系结构被分为IaaS(infrastructure as a service)、PaaS(platform as a service)和SaaS (software as a service)3种不同的服务模式，其中IaaS应用最为成熟和广泛[2]。在IaaS服务模式中，

收稿日期：2021-04-20 修回日期：2021-07-01
基金项目：国家自然科学基金资助项目(61772145)；广东省自然科学基金资助项目(2020A1515010727，2021A1515012252)；广东省科技专项资金资助项目(mmkj2020008)
第一作者：李启锐(1982-)，男，硕士，副教授，研究方向云计算资源调度。E-mail：liqirui@gdupt.edu.cn
通讯作者：彭心怡(2000-)，女，本科生，研究方向为人工智能。E-mail：1742043887@qq.com

由于云计算平台中资源的异构性、作业的多样性、服务质量的差异性以及用户数量巨大性，使得云计算系统要处理大量的作业和数据。在这种情况下，尤其是业务较多的集团公司，单独部署一台虚拟机容易使虚拟机服务器过载，造成作业响应过慢，加大SLA(service level agreement)违规的风险。为此，云服务提供商和用户倾向于使用多用户多队列多虚拟机集群的服务模式。同时，大规模数据中心是当今企业级互联网应用和云计算系统的关键支撑[3]。在这种数据中心架构中，为了提高数据处理的灵活性与可靠性，用户通常将虚拟机部署在不同的数据中心。按照这种部署模式，不同的作业调度算法会带来不同的作业完成效果，例如作业的响应时间、优先级保证等。因此，如何选择最优的虚拟机来部署用户的作业，成为该模式下作业调度时要解决的重要问题。

针对云计算的作业调度优化问题，众多学者以及机构展开了多方面的研究。Verma等[4]提出了一种基于非优势排序的混合粒子群优化算法，以处理IaaS云上具有多个相互冲突目标函数的云作业工作流调度算法，解决了IaaS云科学工作流调度的多目标优化问题。Duan等[5]提出了一种称为PreAntPolicy的虚拟机调度方法，该方法由基于分形数学的预测模型和基于改进蚁群算法的调度器组成。预测模型通过预测负载趋势来协助调度器进行更加合理的调度。Srichandan等[6]提出一种结合遗传算法和细菌觅食算法的作业调度算法，在保证满足SLA前提下实现高效的作业调度。李强等[7]将光能量函数作为植物生长的动力来提升模拟植物生长算法的性能，提出一种可变生长速度的植物模拟算法来实现云作业的调度策略，获得比蚁群算法、粒子群算法等经典云作业调度算法更好的调度效率。但是传统的启发式算法需要在特定的条件下才能获得最优解。面对复杂多变的云环境，其通用性不强，而且在多目标优化问题的求解过程中容易陷入局部最优解，而无法得到全局最优解。

因此，有研究人员采用强化学习方法来解决云作业调度问题。强化学习(reinforcement learning, RL)作为一种无模型的学习方法，具有强大的决策能力，其通过不断试错机制来探索解决问题的最优解，是解决多约束多目标优化问题的有效手段[8]。Peng等[9]采用强化学习Q算法和队列理论来解决复杂云环境下的作业调度和资源配置问题。他们将调度问题转化为序列决策问题，然后采用RL的试错机制，探索最优的调度策略。Cui等[10]提出一种基于强化学习的新型作业调度方案，采用多Agent技术与并行技术来平衡学习过程中的探索和利用，实现了在虚拟机资源和作业期限约束下最大程度地缩短作业的完工时间和平均等待时间。袁景凌等[11]针对异构云环境多目标优化调度问题，设计了一种AHP(analytic hierarchy process)定权的多目标强化学习作业调度方法，较好地优化了作业执行效率和保障用户及服务提供商的利益。虽然强化学习算法能够通过不断试错的机制来获取云作业调度优化问题的最优解，但在面对大规模的状态空间的情况下，强化学习算法容易出现收敛速度慢，或是不收敛的情况。深度神经网络具有强大的感知能力，能够有效应对大规模状态空间，可以很好地弥补强化学习这方面的不足。

Lin等[12]提出了一种多智能体两阶段云作业调度与资源分配框架，实现了作业与资源之间的协同调度，其中作业调度阶段使用异构分布式深度学习模型将多个作业调度到多个数据中心。郭玉栋等[13]通过分析影响云作业调度相关资源的特点，建立基于综合资源利用的特征模型，然后基于HNN(hopfield neural network)技术设计和实现了云作业调度算法。Rangra等[14]提出了一种基于多任务卷积神经网络的云作业调度算法，实现了作业执行时间与执行成本之间的平衡。深度学习作为一种监督学习，通过大量的训练能够学习到根据作业的特征进行优化调度。但是，这种调度策略本质上是离线或静态的调度，比较适用于批量处理的作业提交方式。而云计算环境瞬息万变，面对更多的是在线或动态作业，在这种情况下，深度学习因其缺乏动态

DOI: 10.16182/j.issn1004731x.joss.21-0337

决策能力而无法进行有效应对。

强化学习具有强大的决策能力，而深度学习具有强大的特征获取能力，有学者尝试将它们结合起来进行优势互补，形成了深度强化学习，用来解决云环境下作业的在线调度[15-16]。Guo 等[17]提出了一个名为 DeepRM_Plus 的云资源管理方案，使用卷积神经网络来捕获资源管理模型，并在强化过程中利用模仿学习来减少最优策略的学习时间，提高了算法的收敛速度，减少了平均循环时间和平均加权周转时间。Peng 等[18]提出一个基于深度强化学习的云作业与资源调度框架，该框架协同考虑了用户与云服务提供商双方的利益均衡，并且可以通过调整相应的优化权重实现对不同目标的优化。Lin 等[19]充分利用卷积神经网络的感知力和强化学习的决策能力，提出基于深度卷积网络强化学习模型的云资源调度模型，该模型将云系统的资源和任务资源抽象成图像的形式，作为卷积网络的输入，输出调度策略，实现云系统多资源云作业调度。

在上述有关深度强化学习的研究当中，主要从多目标优化的角度进行云作业调度算法的设计与改进。但是云环境中作业的种类、用户数量、调度的批量、计算资源使用情况等均是变化的，需要根据这些变化采用不同的调度策略。为此，本文提出一种基于深度强化学习的云作业调度算法，实现多用户多队列多数据中心下作业的优化调度。

本文共分为 5 个部分：第 1 部分介绍多用户多队列多数据中心的系统模型，并建立数学模型及优化目标；第 2 部分介绍基于深度强化学习的云作业调度算法的状态空间、动作空间、奖赏函数，并设计相应的作业调度算法；第 3 部分为仿真平台的设计；第 4 部分为仿真实验及结果分析；第 5 部分为总结和展望。

1 系统模型

假设某大型公司计划组建一个虚拟机计算集群，为了提高集群的稳定性和灵活性，避免出现单一数据中心带来的可靠性风险，将组成计算集群的虚拟机分别部署到若干数据中心。当公司的虚拟机服务器部署好之后，公司的各个用户就可以将作业提交到虚拟机服务器上面进行处理。为方便问题描述，将系统模型进行细化，如图 1 所示。

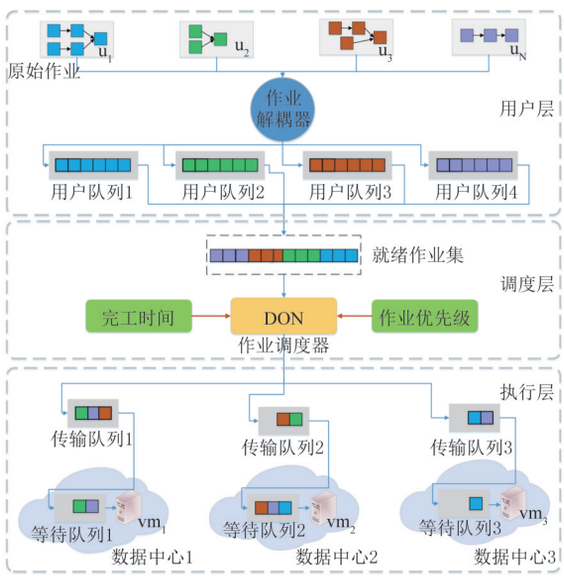

图 1 系统模型
Fig. 1 System model

图 1 所示为一个拥有 4 个具有不同优先级的用户作业队列的作业提交系统模型。用户将种类各异的作业提交到虚拟机服务器进行部署运行。用户作业中不仅包含原子作业，也包含拥有多个存在依赖关系子作业。云系统接收到用户提交的作业后，首先需要对作业进行子作业解耦，按照子作业之间的依赖性和优先级组织成多个作业队列。作业调度器负责将不同队列中的作业部署到虚拟机中，尽可能充分利用可用虚拟机资源对作业进行处理，提高服务质量，如最小化作业完工时间和保证作业优先级等。在调度时刻，各个作业队列中按照先来先服务的规则，取出批量作业来组成就绪作业集，作为调度器的调度单位。在此模型中，作业完工时间主要由执行时间、等待时间和传输时间构成。

假设有 N 个用户需要将作业要提交到数据中心进行处理，这些有计算任务的用户用集合 $\{u_1, u_2, \cdots, u_N\}$ 表示。用户 u_k 提交作业的数量用

$\phi(u_k)$表示，其第i个作业用二元组$J_i^k=(D^k(i), L^k(i))$来表示，其中$D^k(i)$表示J_i^k需要传输的数据量，$L^k(i)$为J_i^k的长度。本文设置$D^k(i)$是一个随机变量并服从均匀分布，即$D^k(i)\sim(D_{\min}, D_{\max})$，其中$D_{\min}$和$D_{\max}$分别表示作业数据量的最小值和最大值。另外，假定每个作业长度与作业数据量呈线性相关[20]，即

$$L^k(i)=\mu\cdot D^k(i), \tag{1}$$

式中：μ表示计算力与数据量的比率(computation to data ratio, CDR)，其取值取决于作业的类型，不同类型的作业有不同的CDR。

由于虚拟机的CPU和带宽是影响作业响应时间的最主要因素，为了简单起见，我们这里只考虑这两种资源。假设用户在S个数据中心中部署有S台虚拟机，用集合$\{vm_1, vm_2, \cdots, vm_S\}$来表示，其第$s$台虚拟机用二元组$vm_s=(R_s^{\text{cpu}}, R_s^{\text{bw}})$表示，其中$R_s^{\text{cpu}}$表示该虚拟机的计算能力，通常用MIPS (million instructions per second)来表示，R_s^{bw}表示该虚拟机的带宽。在获得作业和虚拟机的二元组后，便可以建立多用户多数据中心场景下用户作业响应时间的计算模型。

假设作业J_i^k被调度到虚拟机vm_s执行，则J_i^k的执行时间、传输时间和等待时间计算方式如下:

(1) 作业执行时间

假设在时间步t，vm_s上共有M_t^s个作业同时在执行，采用平均分配的原则将vm_s的MIPS分配给此M_t^s个作业。令$C_t^{J_i^k}(s)$为作业J_i^k在时间步t所获得的MIPS，则

$$C_t^{J_i^k}(s)=\frac{R_s^{\text{cpu}}}{M_t^s} \tag{2}$$

令J_i^k的执行时间记为$t_{i,e}^k$，则

$$t_{i,e}^k=\sum_{t=t_s}^{t_e}\frac{L^k(i)}{C_t^{J_i^k}(s)} \tag{3}$$

式中：t_s为作业开始执行的时间步; t_e为作业结束执行的时间步。

(2) 作业传输时间

在作业传输过程中，虚拟机的带宽资源同样采用均等分配策略，将带宽资源平均分配给当前各个提交作业的用户。假设在时间步t，共有N_t^s个作业正在向虚拟机vm_s传输作业，令$B_t^{J_i^k}(s)$为作业J_i^k在时间步t所获得的带宽资源，则

$$B_t^{J_i^k}(s)=\frac{R_s^{bw}}{N_t^s} \tag{4}$$

作业J_i^k向虚拟机传输数据所需要的时间记为$t_{i,t}^k$，则

$$t_{i,t}^k=\sum_{t=t_s}^{t_e}\frac{D^k(i)}{B_t^{J_i^k}(s)} \tag{5}$$

式中：t_s为作业开始执行的时间步；t_e为作业结束执行的时间步。

(3) 作业等待时间

当虚拟机计算能力不足时，提交的作业将进入虚拟机的等待队列，令$t_{i,w}^k$为作业的排队时间，表示在J_i^k之前进入等待队列正在等待执行的作业的执行时间总和，则

$$t_{i,w}^k=\sum_{J_i\in Q}t_{j,e} \tag{6}$$

式中：Q表示在J_i^k之前进入队列并正在等待执行的作业集合。

(4) 作业完工时间

记J_i^k的完工时间为T_i^k，则

$$T_i^k=T_{i,e}^k+T_{i,t}^k+T_{i,w}^k \tag{7}$$

由式(3)，式(5)和式(6)即可计算式(7)。

(5) 优化目标

将作业合理地提交到不同数据中心的虚拟机服务器，就用户而言，要尽可能减少作业的完工时间，提高作业的响应速度。令D表示为由作业集J所有调度策略组成的集合，T表示为所有时间步的集合，则本文研究问题的优化目标可以形式化定义为

$$\begin{aligned}&\min_{d\in D}\sum_{J_k^k\in J}T_i^k\\&s.t.\\&\forall t\in T,\ \sum C_t^{J_k^k}=R_s^{\text{cpu}},\\&\forall t\in T,\ \sum B_t^{J_k^k}=R_s^{\text{bw}},\\&1\leqslant k\leqslant N, 1\leqslant i\leqslant N^k\end{aligned} \tag{8}$$

这是一个多约束条件下目标优化问题，而且每个时间步云系统状态是动态变化的，求解变得

十分困难。下面研究如何对该优化问题进行求解。

2 作业调度

当在数据中心中部署好虚拟机之后，公司的各个用户就可以源源不断地将作业提交到虚拟机进行处理。作业调度器在每个调度时刻，将就绪作业集中的作业发送给不同数据中心的虚拟机服务器执行。

在分布式数据中心中部署作业的问题是NP难问题。加上用户作业种类、数量以及虚拟机的运行状态均为不断变化，问题求解更为复杂。深度学习是目前人工智能研究的一个热门领域。深度学习是数据驱动的机器学习方法，它根据已有历史数据来推测将来某一事件发生的概率，相对机械和静止，不太适用于云计算环境中动态的用户作业调度。强化学习则是根据本时刻与上一时刻的状态和动作，推断下一时刻某动作发生的概率，是不断变化连续的过程，能够进行作业动态调度决策。但是，由于云计算环境复杂以及状态连续变化，离散化后状态空间集合也很大，此时传统的强化学习方法，例如Q-Learning，难以在内存中维护庞大的Q表。深度强化学习使用神经网络代替Q表以及经验回放机制解决训练样本问题，结合了强化学习的决策能力和深度学习的感知能力，是解决复杂感知决策问题的有效办法，目前被广泛用在游戏、机器视觉等领域[18]。DQN(deep Q network)是最常用的深度强化学习框架，我们将使用DQN来解决本文问题。下面给出DQN算法的状态空间、动作空间、奖赏函数的表示方法以及算法具体过程。

(1) 状态空间

本文问题的目标是最小化用户作业的完工时间，同时考虑到将作业调度到不同的虚拟机会引起虚拟机状态的变化，并影响着作业的完工时间，因此，将虚拟机的状态形式化表示为环境的状态，主要由虚拟机可用CPU核心数量、传输队列数列和等待队列数量3部分组成。具体表示为

$$s_t=(vm_1^{\mathrm{rpe}},-vm_1^{\mathrm{wqu}},-vm_1^{\mathrm{tqu}},vm_2^{\mathrm{rpe}},-vm_2^{\mathrm{wqu}},-vm_2^{\mathrm{tqu}},\cdots,vm_m^{\mathrm{rpe}},-vm_m^{\mathrm{wqu}},-vm_m^{\mathrm{tqu}}) \tag{9}$$

式中：vm_i^{rpe}为第i个虚拟机剩余可用的CPU核心数量；vm_i^{wqu}为第i个虚拟机等待队列数量；vm_i^{tqu}为第i个虚拟机传输队列数量。

(2) 动作空间

作业调度器的任务是为就绪作业选择合适的虚拟机部署执行，假设虚拟机的个数为m，则动作空间表示为$A=\{a_1,a_2,\cdots,a_m\}$，表示本批作业提交到哪台虚拟机处理。采用二进制one-hot的形式表示，例如$a_t=(0,0,1,0)$表示本批作业选择部署到第3台虚拟机；$a_t=(1,0,0,0)$表示本批作业部署到第1台虚拟机。

(3) 奖赏函数

回报函数的设计在深度强化学习中是极其重要的一环，通过将任务目标具体化和数值化，引导Agent通过探索生成动作策略。回报函数的设计是否符合目标需求将决定Agent能否学到期望的策略，并接影响算法的收敛速度和最终性能。鉴于本阶段的优化目标为最小化作业调度的整体完工时间，在虚拟机性能相差不大的情况下，如果调度到作业较多的传输队列和等待队列中，势必需要更长的传输时间和等待时间，从而影响作业的整体完工时间。因此，将奖赏函数定义为

$$R=-(\zeta Q^w+\xi Q^t) \tag{10}$$

式中：Q^w为等待队列作业数量；Q^t为传输队列作业数量；ζ为等待时间系数；ξ为传输时间系数。

(4) 作业调度算法

深度强化学习是一种试错学习机制。通过在云环境中不断探索，Agent学习到好的调度策略。通常将所有作业执行完毕称为一个回合。在经过若干回合学习训练后，回报函数应该趋于收敛。每个回合中，Agent学习训练的过程如下：

步骤1：重置云环境，包括初始化虚拟机、各用户队列、更新就绪作业集等，获得当前环境状态；

步骤2：若系统资源利用率尚未达到规定的阈值，则生成一个选择动作；

步骤3：从就绪作业集中调度一个批量的作业进入步骤2选择动作对应虚拟机的传输队列；

步骤4：计算本次调度获得的回报值；

步骤5：更新全局就绪作业集；

步骤6：更新每台虚拟机的运行情况，若系统资源利用率尚未达到规定的阈值，则进入下一步骤，否则时间步向前走1步，继续执行步骤6；

步骤7：更新环境状态；

步骤8：判断所有作业是否已经完成，并记录在完成标记中；

步骤9：将当前状态、回报值和完成标记保存到记忆池；

步骤10：累计回报值，若已经完成，则转步骤11，否则转步骤2；

步骤11：若记忆池中样本数量已经达到规定的阈值，则训练网络；

步骤12：记录当前回合的回报值，并统计每个作业的传输时间、等待时间和执行时间。

3 仿真实验平台设计

根据系统模型的工作流程以及作业调度算法，基于Python和TensorFlow设计一个基于DQN的仿真实验平台进行实验验证，实验平台主要由以下几个模块组成：

(1) 系统参数模块：主要功能为设置学习率、虚拟机数量及配置、调度作业批量大小、记忆池大小、训练回合数等所需要的环境参数、算法参数等。

(2) 云环境模块：主要功能包括创建与管理虚拟机集合、创建与管理用户队列、创建与管理全局就绪作业集、更新状态空间、重置云环境、每个时间步执行、计算回报值、统计结果等。

(3) DQNAgent模块：主要功能包括创建DQN网络、选择动作、学习训练、样本保存等。

(4) 虚拟机模块：主要功能包括重置虚拟机、作业执行、虚拟机状态管理、运行情况统计等。

(5) 云作业模块：主要功能是根据作业类型生成相关的作业集。

(6) 工具模块：主要功能是绘制实验结果图表，实现实验结果可视化。

各个模块之间的关系如图2所示。

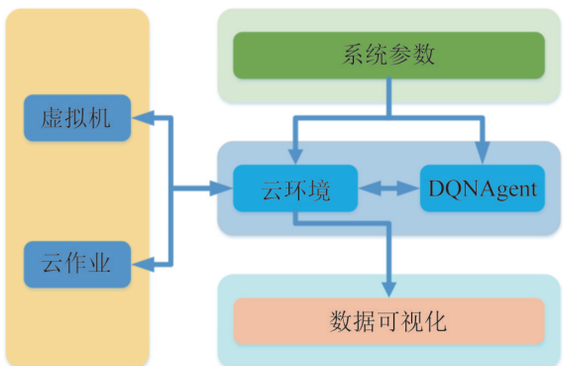

图2 仿真平台各模块之间关系
Fig. 2 Relationships among modules of simulation platform

4 仿真实验及结果分析

实验平台的具体参数为：用户数量为4，用户作业队列数量为4，资源利用率阈值为0.9。实验用的作业集中包括4种作业类型，作业的数据传输量与计算量的比值包括有330 Cycle/MB、1 300 Cycle/MB、1 900 Cycle/MB和2 100 Cycle/MB。实验时，生成作业的类型均匀地从4种作业类型中随机获取[20]。

作业数据量最小值D_{min}取值为10，作业数据量最大值D_{max}取值为20，子作业间依赖性随机生成，总作业数为200。在6个数据中心部署6台虚拟机，其计算能力、计算核心数和带宽如表1所示。

深度强化学习网络模型的参数包括有模型参数和超参数2种。模型参数通常是由数据来驱动调整，例如卷积核的具体核参数、神经网络的权重等。超参数则不需要数据来驱动，而是在训练前或者训练中人为的进行调整，例如学习率、折扣因子等。超参数通常先按经验设定初始值，然后根据训练效果进行调优。

DOI: 10.16182/j.issn1004731x.joss.21-0337

表1 虚拟机配置表
Table 1 Configuration of VMs

序号	计算能力(cycles)	核心数(个)	带宽/Mbps
1	650	4	200
2	1 850	8	300
3	2 500	12	500
4	700	6	250
5	2 050	10	400
6	1 500	8	200

设计了一个有2层隐藏层的DQN网络，第1层隐藏层的神经元数量为30，第2层隐藏层的神经元数量为10。因为学习率直接控制着训练中网络梯度更新的量级，直接影响着模型的有效容限能力，因此在所有的超参数中，学习率最为重要。图3给出了DQN算法在不同学习率(α)情况下的回报函数。

从图3可以看出，当$\alpha=0.1$和$\alpha=0.05$时，回报值并没有随着训练的深入递增并最终趋于收敛。当$\alpha=0.01$时算法的收敛效果最好。当$\alpha=0.001$时，虽然算法也能收敛，但是收敛速度没有$\alpha=0.01$时快。

折扣因子也是深度强化学习中一个重要的超参数。图4给出了DQN算法在不同折扣因子(β)情况下的回报函数。

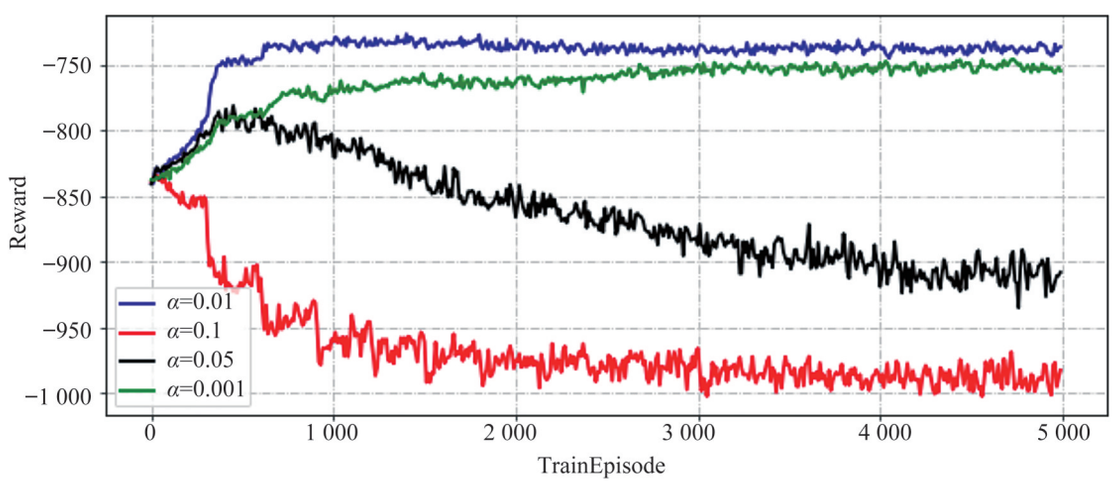

图3 不同学习率情况下算法的收敛性
Fig. 3 Algorithm convergence versus learning rate

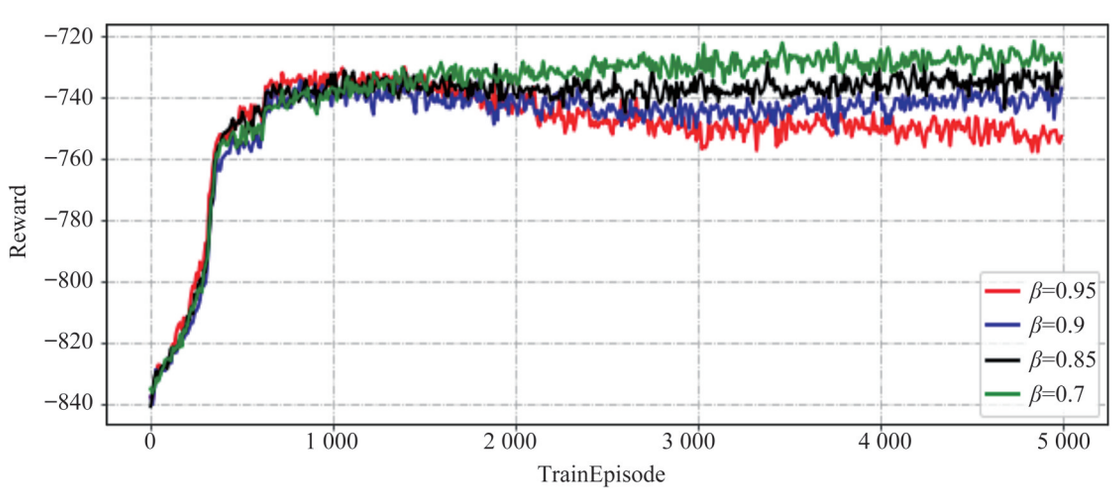

图4 不同折扣因子情况下算法的收敛性
Fig. 4 Algorithm convergence versus discount factor

从图4可以看出，折扣因子的变化对回报函数的影响并没有学习率那么明显。但是也可以看出，当$\beta=0.9$时，回报函数更为平稳，能够较早收敛。

经过参数调优，提出的DQN网络的关键超参数具体如表2所示。

表2 DQN网络超参数表
Table 2 Hyperparameters of DQN network

参数	值	参数	值
训练回合数量	5 000	目标网络更新频率	300
学习率α	0.01	初始ε值	0.2
折扣因子β	0.9	最大ε值	0.9
样本池规模	500	ε每回合增幅	0.002
批样本数	64	等待时间系数ζ	0.5
隐藏层数	2	传输时间系数ξ	1

(1) 算法的收敛性验证

首先考察训练过程中的虚拟机数量变化对算法收敛性以及收敛速度的影响。本次实验中，每个时间步从用户作业队列进入就绪作业集的批量设置为9，虚拟机数量分别取值3，4，5，6，算法奖赏值的变化情况如图5所示。

从图5可以看出，随着训练的深入，Agent从环境中获得的总回报值不断增大，大约经过1 000回合后开始趋于收敛。说明模型通过不断的训练，学习到可实现目标优化的策略。而且可以看出，随着虚拟机数量的增加，算法获得的奖赏值也相应增加，这是因为随着虚拟机数量的增加，用户作业获得的计算资源和网络资源将会增加，从而降低作业的等待时间和传输时间。

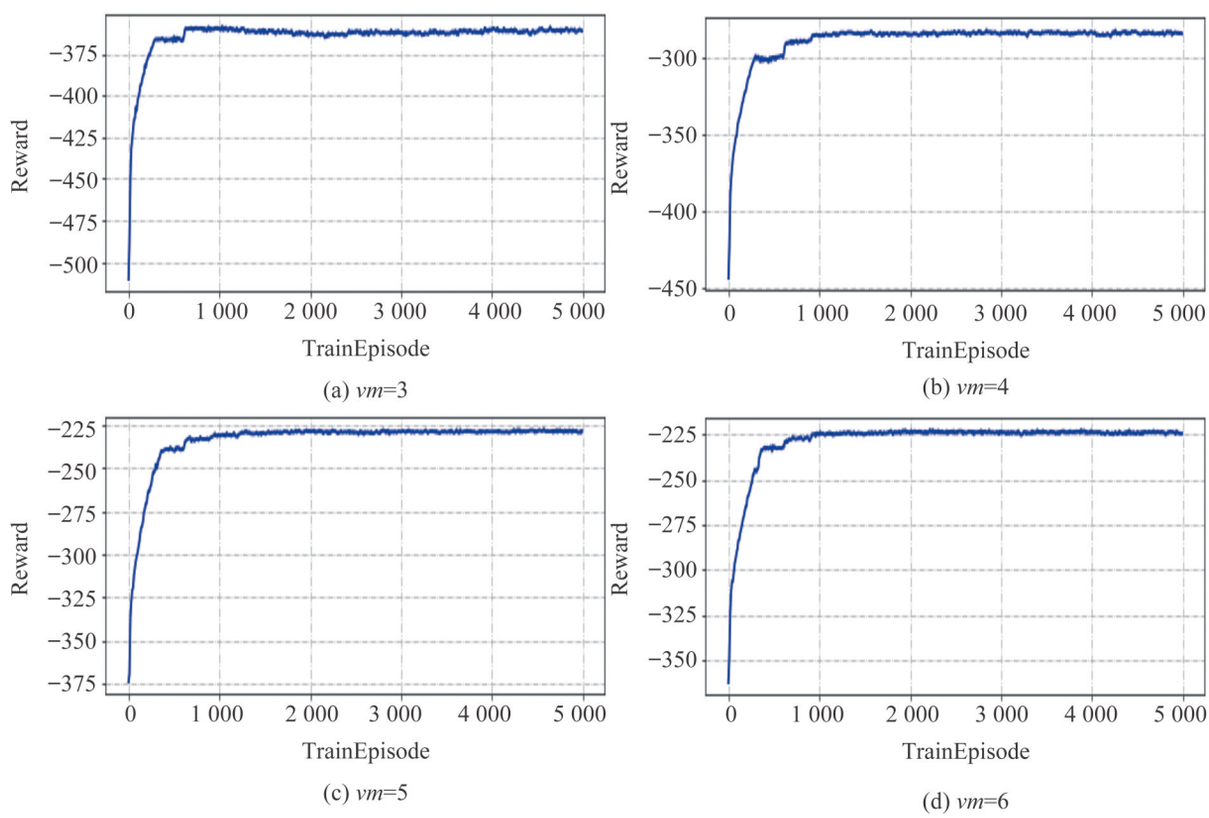

图5 不同虚拟机数量情况下算法的收敛性
Fig. 5 Algorithm convergence versus number of VMs

然后考察调度作业批量变化对算法收敛性以及收敛速度的影响。本次实验中，虚拟机数量设置为4，每个时间步从用户作业队列进入就绪作业集的作业批量分别取值6，8，10和12，算法回报值的变化情况如图6所示。

从图6同样可以看出，大约经过1 000回合后

https://dc-china-simulation.researchcommons.org/journal/vol34/iss2/8
DOI: 10.16182/j.issn1004731x.joss.21-0337

算法开始趋于收敛。而且可以看出，随着批量的增加，算法获得的奖赏值有所降低，这是在计算资源和网络资源固定的情况下，批量的增加意味调度到某虚拟机上面的作业将会增加，在平均分配资源的前提下各作业获得的计算资源和网络资源将会减少，从而增加等待时间和传输时间。此外，从图6中也可以看出，算法收敛曲线的波动性随着批量的增大而有所加大，这是由于作业以批量调度到某数据中心的服务器中，批量的增大意味着作业调度灵活性的降低，从而，收敛曲线的波动性有所增加。

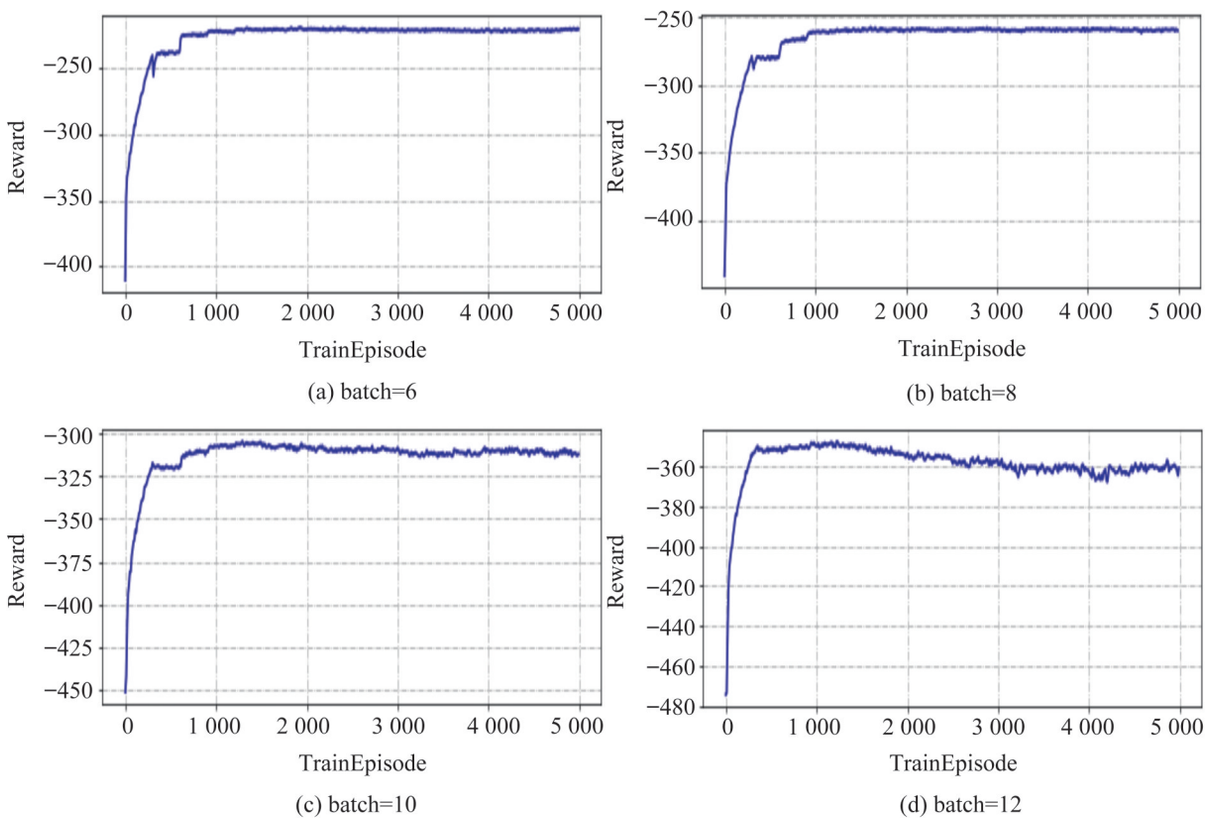

图6 不同批量情况下算法的收敛性

Fig. 6 Algorithm convergence versus number of batches

(2) 算法的比较性验证

接下来，对本文算法与其他算法在全局完工时间方面的优化效果进行对比验证。采用的基准算法有随机算法(Rnd)、轮询算法(RR)、首次适应算法(FF)和最佳适应算法(BF)。随机算法选择动作时从可用虚拟机中随机选择一台进行作业部署。循环算法则按可用虚拟机顺序，依次循环部署。首次适应算法首先计算虚拟机剩余可用的核心数，然后按顺序查找剩余核心满足作业要求的虚拟机，并将作业部署到第一台满足条件的虚拟机上。最佳适应算法同样首先计算虚拟机剩余可用的核心数，然后将作业部署到剩余核心满足作业要求并且数量最多的虚拟机上。

首先考察在不同虚拟机数量相同批量条件下不同算法在作业完工时间方面的性能比较。本次实验中批量设置为9，虚拟机数量分别设置为3，4，5和6，实验结果如图7所示。

图7表示的是各个算法在第3 000~5 000回合的平均结果，其中DQN表示本文提出的算法。可以看出，在批量相同的情况下，随着虚拟机数量的增加，作业的总体完工时间相应减少，说明增加虚拟机资源可以有效降低作业的响应时间。也可以看

出，上述几种虚拟机数量情况下，本文提出算法的作业完工时间均小于其他基准算法，分别比随机算法、轮询算法、首次适应算法和最佳适应算法平均降低了41.37%，28.68%，12.37%和9.04%。以上结果证明在云资源竞争较大的情况下，提出的算法能够根据作业属性和系统资源状态来动态制定作业的调度策略，从而减少全局作业完工时间。

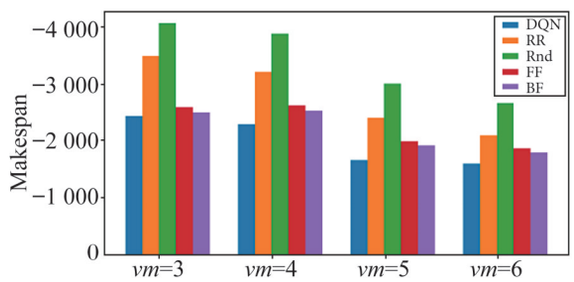

图7　不同虚拟机数量情况下算法的性能比较
Fig. 7　Algorithm performance versus number of VMs

接下来考察相同虚拟机数量不同批量条件下各个算法在作业完工时间方面的性能对比情况。本次实验中虚拟机数量设置为4，批量分别取值6，8，10和12，实验结果如图8所示。

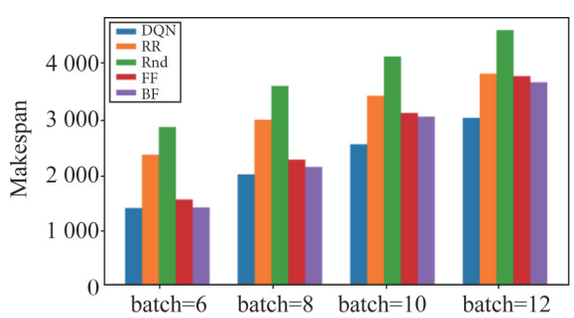

图8　不同批量情况下算法的性能比较
Fig. 8　Algorithm performance versus number of batches

图8表示的也是各个算法第3 000~5 000回合的平均结果。同样可以看出，在虚拟机数量相同的情况下，随着批量的增加，作业的整体完工时间在也在增加，这是因为调度批量越大，每个作业平均得到的计算资源和网络资源将会越少，从而增加作业的等待和传输时间，进而导致作业完工时间的增加。但是不管是何种情况，本文提出算法的作业整体完工时间均小于其他基准算法，分别比随机算法、轮询算法、首次适应算法和最佳适应算法平均降低了42.30%，30.28%，15.15%和10.18%。

5　结论

本文针对如何在分布式数据中心的虚拟机集群中选择最优虚拟机进行作业部署的问题，提出了基于深度强化学习的动态作业调度算法。算法通过深度强化学习模型感知虚拟机的运行状态，求得相应的作业优化调度策略，解决了由于用户作业类型、大小、虚拟机状态等动态变化导致用户作业动态调度困难的问题，取得比随机算法、轮询算法、首次适应算法和最佳适应算法更好的优化效果。

自从以深度强化学习技术为核心的AlphaGo在2016年击败了人类高级围棋选手之后，深度强化学习受到人们的广泛关注和研究，目前在自动驾驶、控制论、理解机器学习、智能推荐等人工智能领域有着广泛应用。这些应用通常需要云计算和大数据的支持，因此数据中心作业的高效调度就显得尤为重要。目前的研究工作只是针对多用户多队列多数据中心环境的在线作业调度，如何实现在线作业和离线作业混合调度是接下来的主要研究工作。

参考文献

[1] Xu Z, Liang W, Xia Q. Efficient Embedding of Virtual Networks to Distributed Clouds via Exploring Periodic Resource Demands[J]. IEEE Transactions on Cloud Computing(S2168-7161), 2018, 6(3): 694-707.

[2] 李成辉, 李仁旺, 杨强光, 等. 基于改进萤火虫算法的云计算任务调度算法[J]. 浙江理工大学学报(自然科学版), 2019, 41(3): 354-359.
Li Chenghui, Li Renwang, Yang Qiangguang, et al. Cloud Computing Task Scheduling Algorithm Based on Improved Firefly Algorithm[J]. Journal of Zhejiang Sci-Tech University(Natural Sciences Edition), 2019, 41(3): 354-359.

[3] 王康瑾, 贾统, 李影. 在离线混部作业调度与资源管理技术研究综述[J]. 软件学报, 2020, 31(10): 3100-3119.
Wang Kangjin, Jia Tong, Li Ying. State-of-the-art Survey

https://dc-china-simulation.researchcommons.org/journal/vol34/iss2/8
DOI: 10.16182/j.issn1004731x.joss.21-0337

of Scheduling and Resource Management Technology for Colocation Jobs[J]. Journal of Software, 2020, 31(10): 3100-3119.

[4] Verma A, Kaushal S. A hybrid Multi-objective Particle Swarm Optimization for Scientific Workflow Scheduling [J]. Parallel Computing(S0167-8191), 2017, 62: 1-19.

[5] Duan H, Chen C, Min G, et al. Energy-aware Scheduling of Virtual Machines in Heterogeneous Cloud Computing Systems[J]. Future Generation Computer Systems(S0167-739X), 2017, 74: 142-150.

[6] Srichandan S, Turuk A K S. Task Scheduling for Cloud Computing Using Multi-objective Hybrid Bacteria Foraging Algorithm[J]. Future Computing and Informatics Journal(S2314-7288), 2018, 3(2): 210-23.

[7] 李强, 刘晓峰. 基于模拟植物生长算法的云作业调度模型[J]. 系统仿真学报, 2018, 30(12): 4649-4658.
Li Qiang, Liu Xiaofeng. Cloud Job Scheduling Model Based on Improved Plant Growth Algorithm[J]. Journal of System Simulation, 2018, 30(12): 4649-4658.

[8] 殷昌盛, 杨若鹏, 朱巍, 等. 多智能体分层强化学习综述[J]. 智能系统学报, 2020, 15(4): 646-655.
Yin Changsheng, Yang Ruopeng, Zhu Wei, et al. A Survey on Multi-agent Hierarchical Reinforcement Learning[J]. CAAI Transactions on Intelligent Systems, 2020, 15(4): 646-655.

[9] Peng Z, Cui D, Zuo J, et al. Random Task Scheduling Scheme Based on Reinforcement Learning in Cloud Computing[J]. Cluster Computing(S1386-7857), 2015, 18: 1595-1607.

[10] Cui D, Peng Z, Xiong J, et al. A Reinforcement Learning-Based Mixed Job Scheduler Scheme for Grid or IaaS Cloud[J]. IEEE Transactions on Cloud Computing (S2168-7161), 2020, 4: 1030-1039.

[11] 袁景凌, 陈旻骋, 江涛, 等. 异构云环境下 AHP 定权的多目标强化学习作业调度方法[J/OL].(2021-01-05) 控制与决策, 2021: 1-8. https://doi. org/10.13195/j. kzyjc. 2020.0911.
Yuan Jingling, Chen Minchi, Jiang Tao, et al. Multi-Objective Reinforcement Learning Job Scheduling Method using AHP Fixed Weight in Heterogeneous Cloud Environment[J/OL]. (2021-01-05) Control and Decision, 2021:1-8. https://doi.org/10.13195/j.kzyjc.2020.0911.

[12] Lin J, Cui D, Peng Z, et al. A Two-Stage Framework for the Multi-User Multi-Data Center Job Scheduling and Resource Allocation[J]. IEEE Access(S2169-3536), 2020, 8: 197863-197874.

[13] 郭玉栋, 左金平. 基于霍普菲尔德网络的云作业调度算法[J]. 系统仿真学报, 2019, 31(12): 2859-2867.
Guo Yudong, Zuo Jinping. The Scheduling Algorithm of Cloud Job Based on Hopfield Neural Network[J]. Journal of System Simulation, 2019, 31(12): 2859-2867.

[14] Rangra A, Sehgal V K, Shukla S. A Novel Approach of Cloud Based Scheduling Using Deep-Learning Approach in E-Commerce Domain[J]. International Journal of Information System Modeling and Design(S1947-8186), 2019, 10(3): 59-75.

[15] 李凯文, 张涛, 王锐, 等. 基于深度强化学习的组合优化研究进展[J]. 自动化学报, 2021, 47(11): 2521-2537.
Li Kaiwen, Zhang Tao, Wang Rui, et al. Research Reviews of Combinatorial Optimization Methods Based on Deep Reinforcement Learning[J]. Acta Automatica Sinica, 2021, 47(11): 2521-2537.

[16] 朱斐, 吴文, 伏玉琛, 等. 基于双深度网络的安全深度强化学习方法[J]. 计算机学报, 2019, 42(8): 1812-1826.
Zhu Fei, Wu Wen, Fu Yushen, et al. A Dual Deep Network Based Secure Deep Reinforcement Learning Method[J]. Chinese Journal of Computers, 2019, 42(8): 1812-1826.

[17] Guo W, Tian W, Ye Y, et al. Cloud Resource Scheduling With Deep Reinforcement Learning and Imitation Learning[J]. IEEE Internet of Things Journal(S2327-4662), 2021, 8(5): 3576-3586.

[18] Peng Z, Lin J, Cui D, et al. A Multi-objective Trade-off Framework for Cloud Resource Scheduling Based on the Deep Q-network Algorithm[J]. Cluster Computing (S1386-7857), 2020, 23(4): 2753-2767.

[19] Lin J, Peng Z, Cui D. Deep Reinforcement Learning for Multi-resource Cloud Job Scheduling[C]// 2018 25th International Conference on Neural Information Processing. Berlin: Springer, 2018: 289-302.

[20] Miettinen A, Nurminen J. Energy Efficiency of Mobile Clients in Cloud Computing[C]// Boston: USENIX Association, 2010: 1-7.