计算机科学
COMPUTER SCIENCE

基于决策树和由均匀分布改进Q学习的虚拟机整合算法

师亮, 温亮明, 雷声, 黎建辉

引用本文

师亮, 温亮明, 雷声, 黎建辉.基于决策树和由均匀分布改进Q学习的虚拟机整合算法[J]. 计算机科学, 2023, 50(6): 36-44.

SHI Liang, WEN Liangming, LEI Sheng, LI Jianhui. Virtual Machine Consolidation Algorithm Based on Decision Tree and Improved Q-learning by Uniform Distribution [J]. Computer Science, 2023, 50(6): 36-44.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220300192

基于决策树和由均匀分布改进Q学习的虚拟机整合算法

师　亮[1,2]　温亮明[1,2]　雷　声[1,2]　黎建辉[1]
1 中国科学院计算机网络信息中心　北京 100090
2 中国科学院大学　北京 100049
(shiliang6402@foxmail.com)

摘　要　随着云数据中心规模的不断扩大,次优虚拟机整合算法所引起的高能耗、低资源利用率和用户服务质量下降等问题逐渐凸显。为此,提出了一种基于决策树和由均匀分布改进Q学习的虚拟机整合算法(DTQL-UD)。该算法采用决策树实现状态表征,并在评估下一时刻状态-动作价值时采用均匀分布选取下一时刻动作,可直接从云数据中心状态到虚拟机迁移的过程中通过实时反馈来不断优化决策。此外,针对强化学习中模拟器与真实场景中的差异问题,基于大量真实云数据中心负载跟踪数据,使用监督学习模型训练模拟器以增加模拟器的仿真度。仿真实验结果表明,DTQL-UD在能耗、资源利用率、用户服务质量、虚拟机迁移次数和剩余活跃主机数量方面分别优化了14%,12%,21%,40%和10%。同时,得益于决策树在表格型数据上更强的特征提取能力,DTQL-UD相比其他现有的深度强化学习方法可学到更优的整合策略,并且在本实验中随着云数据中心规模的增大,可将传统强化学习模型的训练耗时逐步减少60%～92%。

关键词:云资源调度;虚拟机整合算法;强化学习;决策树
中图法分类号　TP393

Virtual Machine Consolidation Algorithm Based on Decision Tree and Improved Q-learning by Uniform Distribution

SHI Liang[1,2],WEN Liangming[1,2],LEI Sheng[1,2] and LI Jianhui[1]
1 Computer Network Information Center,Chinese Academy of Sciences,Beijing 100090,China
2 University of Chinese Academy of Sciences,Beijing 100049,China

Abstract　As the scale of cloud data centers expands,problems such as high energy consumption,low resource utilization,and reduced quality of service caused by sub-optimal virtual machine consolidation algorithm becomes increasingly prominent. Therefore,this paper proposes DTQL-UD,a virtual machine consolidation algorithm based on decision tree and improved Q-learning by uniform distribution. It uses the decision tree to characterize the states and selects the next action by uniform distribution when evaluating the next state-action value. At the same time,it can optimize decision-making with real-time feedback directly from the state of the cloud data center to the virtual machine migration process. Besides,aiming at the difference between the simulator and real world in reinforcement learning,we train the simulator by supervised learning model based on a large amount of real cluster load tracking data to enhance the degree of the simulator. Compared with the existing heuristic methods,experiment results show that DTQL-UD can optimize energy consumption,resource utilization,quality of service,number of virtual machine migrations,and remaining active hosts,by 14%,12%,21%,40%,and 10%,respectively. Meanwhile,due to the stronger feature extraction capability of decision tree on tabular data,DTQL-UD can learn better scheduling strategy than other existing deep reinforcement learning(DRL)methods. And in our experiments,as the cluster size increases,the proposed algorithm can gradually reduce the training time of traditional reinforcement learning models by 60% to 92%.

Keywords　Cloud resource scheduling,Virtual machine consolidation algorithm,Reinforcement learning,Decision tree

1　引言

近年来,云数据中心在规模扩大的同时也面临着管理成本增高、能耗加剧、资源利用率降低、服务质量下降等问题。有研究表明,空载状态下的主机平均能耗为满载状态的60%,而企业云数据中心的CPU平均利用率仅为10%左右[1]。对于大型公司而言,节省3%的能耗即可减少数百万美元的开销[2]。当前,降低云数据中心能耗的主流方式为采用虚拟整合技术。然而,采用次优的虚拟机整合策略会增加过多的迁移次数,从而导致用户服务质量下降[3]。因此,研究

到稿日期:2022-03-21　返修日期:2022-09-22
基金项目:国家重点研发计划(2021YFE0111500);中国科学院国际大科学计划培育专项(241711KYSB20200023)
This work was supported by the National Key R&D Program of China(2021YFE0111500) and International Mega-science Programs of the Chinese Academy of Sciences(241711KYSB20200023).
通信作者:黎建辉(lijh@cnic.cn)

同时兼顾能耗、资源利用率和用户服务质量等多目标提升的虚拟机整合(Virtual Machine Consolidation，VMC)策略具有重要意义[4]。

目前，多目标优化的云资源调度问题通常被分解为主机过载检测、主机欠载检测、虚拟机选择和虚拟机重放置 4 个子问题[5]，且主要通过整合虚拟机资源来求解[6]。但在大部分研究中，针对以上 4 个子问题的解决办法都是解耦后进行单独建模，并未充分考虑不同子问题间的依赖关系，这在动态变化的云环境中容易产生次优的局部最优解。此外，现有工作大多聚焦于同构云场景下的资源调度方案[7]。而这些方案往往不具备异构云环境的自适应能力。

针对以上问题，需要一种动态调整并可应对异构化云场景的多目标优化虚拟机整合方法。近年来不断发展成熟的强化学习(Reinforcement Learning，RL)算法为解决上述问题提供了新的思路[8]。强化学习智能体在生物学中和人类的行为相似，其本质均为“试错”学习的过程，智能体在动态变化的环境中通过实时交互来不断优化自身的决策行为[9]。因此，本文采用强化学习的思想来解决虚拟机整合问题。具体而言，本文提出了一种基于决策树和由均匀分布改进 Q 学习的虚拟机整合算法(Virtual Machine Consolidation Algorithm Based on Decision Tree and Improved Q-learning by Uniform Distribution，DTQL-UD)。该算法基于多目标奖励机制并采用决策树实现状态表征，在评估下一时刻状态-动作价值时使用均匀分布选取下一时刻动作。相比采用主流双学习机制来减小由传统 Q 学习更新导致的最大化偏差问题(Maximization Bias)[10]，此策略可在不影响准确度的前提下大幅缩短训练耗时，并可在动态变化的负载下采取较优的自适应策略。此外，本文基于大量真实云数据中心负载跟踪数据并使用监督学习模型(决策树)来训练模拟器[11]，以增大模拟器的仿真度。

本文的主要贡献包括：

(1)提出了一种适用于虚拟机整合问题的强化学习算法，设计状态表示、低维动作表示和多目标即时奖励。同时，针对深度神经网络在表格型数据中表征能力不足的问题，采用决策树代替深度神经网络进行状态表征。

(2)考虑到真实异构云环境以及强化学习中模拟器与真实场景中的差异问题，本文基于大量真实云数据中心负载跟踪数据，使用监督学习模型训练模拟器以增大模拟器的仿真度。

(3)针对 Q 学习方法簇[10,12-14]中的 Q 值更新公式进行改进，采用均匀分布取代 argmax 算子和双学习机制来选取下一时刻动作，以在不降低性能的前提下减少训练耗时。

2　相关工作

2.1　虚拟机整合方法

考虑到虚拟机整合问题的复杂性，目前的工作大都采用启发式或元启发式方法来实现虚拟机的资源管理[7]，如 OpenStack 和 CloudStack 平台采用首次适应(First-Fit，FF)和降序首次适应(First-Fit Decreasing，FFD)的方法进行虚拟机重放置。在通用启发式方法的基础上，Abdullah 等[15]提出了降序快速最佳适应算法，该方法基于动态利用率进行虚拟机智能整合；Beloglazov 等[16]则提出了基于功耗感知的最佳适应降序算法，该算法基于虚拟机的历史资源利用数据进行整合。以上两种方法虽然都能有效降低云数据中心能耗，但难以针对实际场景中的多维资源负载进行建模。在元启发式方面，Farahnakian 等[17]采用基于蚁群算法的虚拟机整合方法来有效降低云数据中心的能耗；Singh 等[18]提出了一种基于超立方体的遗传算法，该方法虽然能有效降低能耗，但寻优速度较慢，不适用于大规模云场景。

此外，为了尽可能实现虚拟机整合的全局最优目标，一些学者开始对虚拟机整合问题进行理论研究，如采用线性规划[19]、随机线性规划[20]、整数规划[21]、蚁群系统算法[22]、多目标粒子群优化算法[23]等方法对虚拟机整合问题进行建模，并分析了当前云数据中心状态下的最优解。以上工作虽能有效提高各项性能指标，但在动态化的云数据中心负载环境中无法通过实时反馈来不断调整策略。

2.2　强化学习在虚拟机整合上的应用

由于强化学习方法的本质是根据实时反馈来自适应调整策略，因此一些学者提出了基于多智能体[24]、基于性能功率比[25]、融合模糊逻辑[26]、元学习[27]、随机加权三重 Q 学习[28]的强化学习算法。以上工作均可根据不同指标的实时反馈来动态调整虚拟机迁移策略，但存在决策实时性或模型准确度较低，或者不支持真实的云基础设施，难以在异构云场景中落地的问题。为此，Masoumzadeh 等[29]采用模糊 Q-learning(FQL)来有效整合云数据中心的虚拟机资源，该方法分别从主机过载检测和虚拟机选择两个子问题出发，提出了一种基于 FQL 的自适应主机过载检测机制和基于 FQL 的虚拟机选择机制。FQL 虽然可以实现一定程度的性能平衡，但无法解决真实环境中的多维资源建模问题。由于真实矢量装箱问题中的维度较高，需要通过函数逼近来实现，因此 Yu 等[6]提出了一种基于深度强化学习的自适应虚拟机整合方法，该方案在深度确定性策略梯度(DDPG)的基础上设计张量化状态表示、确定性动作输出和加权奖励机制并引入反向梯度限定机制，同时构建了适用于虚拟机整合问题的端到端模型，从而大幅度提高了用户服务质量。

然而，真实场景下的主机负载必定具有波动性，对状态的张量化建模容易因为状态表征过于稀疏而导致在神经网络训练中出现欠拟合，从而影响模型的训练效果。因此，本文综合考虑了能耗、资源利用率和用户服务质量等多个优化目标，并同时从模型训练和异构云场景两方面全局考虑，进行建模，以更好地提升异构云场景下的各项性能指标。

3　问题描述

3.1　能耗定义

已知计算机功率与 CPU 利用率之间存在如下线性关系[30]：

$$p(u)=k\,P_{\max}+(1-k)P_{\max}u \tag{1}$$

其中，$P_{\max}$表示主机的工作负载处于峰值时所消耗的最大功率，k 表示空闲主机功率占$P_{\max}$的百分比，u 为主机 CPU 的利用率。由于主机的工作负载随时间而变化，因此主机从t_0时刻开始，经过时间 t 后的能耗为：

$$E_t=\int_{t_0}^{t_0+t}p(u(t))\mathrm{d}t \tag{2}$$

3.2 云数据中心资源利用率定义

云数据中心上的资源通常具有多维性(如 CPU 和 RAM 等),需要先分别定义多维资源容量和多维资源占用量,再定义云数据中心资源利用率。

定义 1(资源容量) 假设云数据中心的主机数量为 N,则第 i 台主机的资源容量为:

$$R_i=\{r_i^1,\cdots,r_i^k,\cdots,r_i^K\},i\in[1,N],k\in[1,K]$$

其中,K 表示主机的资源维度数,r_i^k 表示第 i 台主机对应第 k 维度上的资源满载容量。

定义 2(资源占用量) 第 i 台主机在 t 时刻上的资源实际占用量为:

$$O_{i,t}=\{o_{i,t}^1,\cdots,o_{i,t}^k,\cdots,o_{i,t}^K\},i\in[1,N],k\in[1,K]$$

其中,$o_{i,t}^k$ 表示 t 时刻第 i 台主机对应第 k 维度上的资源实际占用量。

定义 3(云数据中心资源利用率) t 时刻云数据中心的资源利用率为:

$$U_t=\frac{\sum_{i=1}^{N}\prod_{k=1}^{K}o_{i,t}^k}{\sum_{i=1}^{N}\prod_{k=1}^{K}r_i^k} \tag{3}$$

式(3)先将主机中不同维度的资源作相乘处理,再对相乘结果求和得到云数据中心总资源容量和 t 时刻总占用量,据此求得云数据中心资源利用率 U_t。

3.3 服务质量定义

在云数据中心系统中,服务质量(Quality of Service, QoS)的好坏在不同种类应用中有不同的定义。此处,本文参考文献[6],采用服务等级协议的违反程度(Service Level Agreement Violation, SLAV),表示如下:

$$SLAV=\frac{\sum_{i=1}^{M}u_{\mathrm{req}_i}-\sum_{i=1}^{M}u_{\mathrm{allo}_i}}{\sum_{i=1}^{M}u_{\mathrm{req}_i}} \tag{4}$$

其中,M 表示虚拟机总数,u_{req_i} 表示第 i 个虚拟机请求的资源量,u_{allo_i} 表示真实分配给第 i 个虚拟机的资源量。因此,该式的物理意义为请求但未分配给用户的资源与总请求资源的占比,且 SLAV 越低服务质量越好。

3.4 多目标组合优化函数定义

为了在保证用户服务质量的前提下降低能耗并提升云数据中心资源利用率,本文通过定义最优整合策略引出多目标优化函数的形式化描述。

定义 4(最优整合策略) 假设 t 时刻存在一个最优整合策略 π_t^*:

$$\begin{aligned}&\pi_t^*=\underset{\pi}{\operatorname{argmin}}\ w_eE_t^\pi-w_uU_t^\pi\\&\text{s. t.}\ \ QoS_t^\pi\geqslant\hat{\mathbb{E}}[QoS]\end{aligned} \tag{5}$$

其中,w_e 和 w_u 分别表示能耗和云数据中心资源利用率的目标权重,约束条件则为当前 t 时刻采用策略 π 得到的实际服务质量不低于预估期望服务质量。

4 多目标虚拟机整合方法

4.1 整体框架

图 1 给出了采用 DTQL-UD 整合算法的云数据中心在主机触发过载响应时的调度框架。

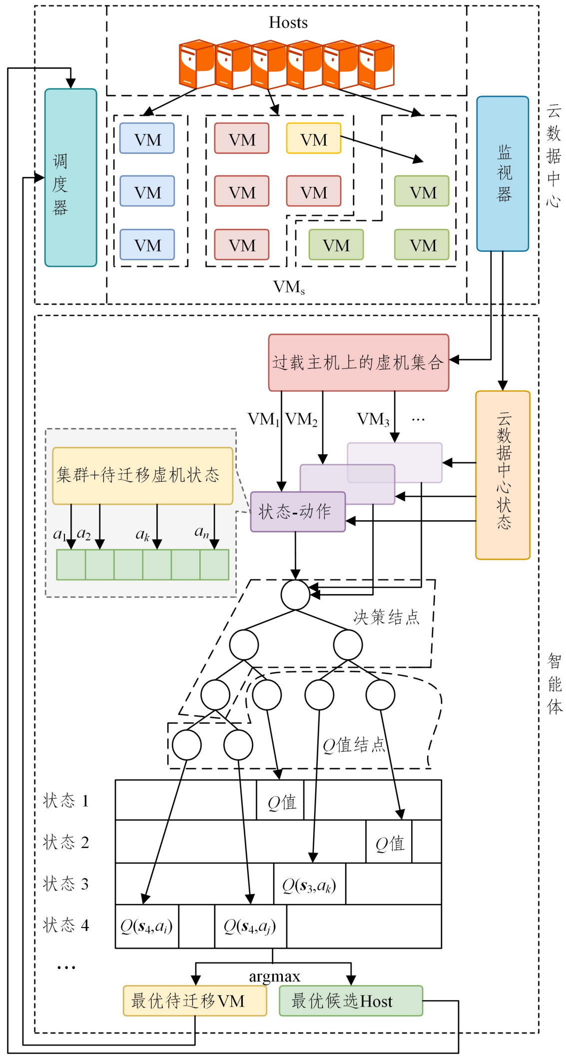


图 1 云数据中心虚拟机整合框架

Fig. 1 Framework of VM consolidation in cloud data center

图 1 上侧是经典的云数据中心环境,右侧所示的监视器模块负责记录运行时所有虚拟机及所属主机的相关元数据及动态负载。当出现过载主机时,下侧所示的智能体会首先获取此时过载主机上的虚拟机集合和云数据中心对应的状态信息,并将该信息组合成该虚拟机的状态-动作向量,然后依次进行状态-动作价值(即 Q 值)计算,其次再选取最大 Q 值对应的虚拟机和主机作为最佳待迁移虚拟机和最佳候选主机,最后通过调度器完成虚拟机迁移操作并更新状态记录。

4.2 虚拟机整合问题的 MDP 建模

4.2.1 动作表示

在实际场景中,单次整合操作通常仅针对少部分虚拟机。假设云数据中心虚拟机数量为 M,主机数量为 N,则某过载主机上虚拟机到所有主机的映射数量高达 M^N 个,因此在设计动作表征时应尽可能低维化动作表征以减少其稀疏性。为此,将虚拟机迁移的动作定义为 $a_t=i,\forall i\in[1,N]$,表示 t 时刻该虚拟机迁移到编号为 i 的主机上。

4.2.2 状态表示

由于真实云数据中心场景下的虚拟机资源占用量存在波动性且异构主机间的虚拟机满载数量存在较大差异,

因此，直接将云数据中心状态按不同资源维度进行张量化建模容易出现大量稀疏状态，从而导致模型出现欠拟合现象。为此，可将 t 时刻的环境状态定义为如下表格型一维向量：

$$S_t=(\boldsymbol{v}_t,\boldsymbol{c}_t)^{\mathrm{T}}$$

其中，$\boldsymbol{v}_t$ 表示 t 时刻待迁移虚拟机的资源描述量，$\boldsymbol{c}_t$ 表示 t 时刻云数据中心集群资源描述量，具体可定义为：

$$\boldsymbol{c}_t=(host_t^1,host_t^2,\cdots,host_t^N)^{\mathrm{T}}$$

其中，$host_t^i$ 表示 t 时刻云数据中心第 i 台主机的资源描述量，它与 $\boldsymbol{v}_t$ 均由每一维资源的当前占用量和满载量分别拼接而成（如在仅考虑 CPU 和 RAM 的场景中，$host_1$ 由编号为 1 的主机对应的当前 CPU 占用量、CPU 满载量、当前 RAM 占用量和 RAM 满载量组成）。此外，本文的待迁移虚拟机由上一时刻集群状态和动作对应的最大 Q 值选取得到，具体定义如下：

$$\boldsymbol{v}_{t+1}\leftarrow\underset{v}{\operatorname{argmax}}\,Q((\boldsymbol{v},\boldsymbol{c}_t),a_t) \tag{6}$$

4.2.3　奖励函数设计

本文的主要优化目标是能耗、云数据中心资源利用率和服务质量。因此，具体奖励函数设计如下。

（1）能耗：该奖励由云数据中心在执行虚拟机迁移前和迁移后的能耗差分表示。若差分为正数，则说明迁移操作有助于降低能耗；反之，则说明该操作在增加能耗。假设 t 时刻云数据中心的能耗为 E_t，则该时刻能耗即时奖励为：

$$r_t^{\mathrm{power}}=E_t-E_{t+1} \tag{7}$$

（2）云数据中心资源利用率：该奖励由执行虚拟机迁移前和迁移后 CPU 和 RAM 的联合资源占用率差分表示。若差分为正数，则表示当前迁移策略有助于提升云数据中心资源利用率；反之，则表示该操作在降低利用率。假设 t 时刻云数据中心的资源利用率为 U_t，则该时刻云数据中心资源利用率奖励为：

$$r_t^{\mathrm{util}}=U_t-U_{t+1} \tag{8}$$

（3）服务质量：主要通过虚拟机迁移后是否会导致任一维度资源过载来体现。若过载，则返回相应 $over\_rate$（过载率）的相反数作为惩罚：

$$r_t^{\mathrm{QoS}}=-over\_rate \tag{9}$$

最后，将以上不同目标对应的奖励加权求和得到单步综合奖励：

$$r_t=\begin{cases}r_t^{\mathrm{QoS}}, & \text{if overload}\\ \alpha\cdot r_t^{\mathrm{power}}+\beta\cdot r_t^{\mathrm{util}}, & \text{otherwise}\end{cases} \tag{10}$$

其中，超参数 $\alpha,\beta\in R^+$ 表示缩放因子，具体设置及依据详见 5.2 节。

4.3　离线训练算法设计

4.3.1　按均匀分布进行动作选择

强化学习中经典 Q-learning 算法的 Q 值更新公式如下：

$$\Delta Q(\boldsymbol{S},A)\leftarrow\alpha[R+\gamma\underset{a}{\operatorname{argmax}}\,Q(\boldsymbol{S}',a)-Q(\boldsymbol{S},A)] \tag{11}$$

式(11)中的 argmax 算子将下一时刻状态中 Q 值最大的动作作为最优动作，该方法假设 Q 表或 Q 网络有较高准确率，但该假设在智能体训练初期往往并不成立，导致出现最大化偏差问题。为了缓解该问题，双学习机制被提出，该方法使用两个独立 Q 表或 Q 网络分别进行训练，在选择动作时，将其中一个 Q 表或 Q 网络中值最大的动作代入到第二个 Q 表中进行评估，具体公式如下：

$$\Delta Q_1(\boldsymbol{S},A)\leftarrow\alpha[R+\gamma Q_2(\boldsymbol{S}',\underset{a}{\operatorname{argmax}}\,Q_1(\boldsymbol{S}',a))-Q_1(\boldsymbol{S},A)] \tag{12}$$

以上双 Q 学习更新公式由于训练数据的不同，其两个 Q 表或 Q 网络的动作价值分布也不相同。同时，如图 2 所示，若将双学习再扩展到三 Q 学习甚至多 Q 学习，随着 Q 表或 Q 网络的个数趋于无穷，其本质即为按动作分布来选取下一时刻状态的期望价值对应的离散动作，但动作分布随着动作价值的变化而变化，难以实时地准确估计，而采用均匀分布可以更高频地选取到期望价值对应的动作，以此逼近状态的期望价值。因此，区别于文献[31]，本文采用均匀分布来选取评估动作以逼近真实的期望价值。具体到虚拟机整合场景的 Q 值更新式如下：

$$\Delta Q(\boldsymbol{S},A)\leftarrow\alpha[R+\gamma Q(\boldsymbol{S}',a\sim U(i_{\min},i_{\max}))-Q(\boldsymbol{S},A)] \tag{13}$$

其中，$i_{\min}$ 表示云数据中心中编号最小的主机，$i_{\max}$ 表示云数据中心中编号最大的主机，$a\sim U(i_{\min},i_{\max})$ 表示从动作空间中按均匀分布选取下一时刻状态对应的动作。

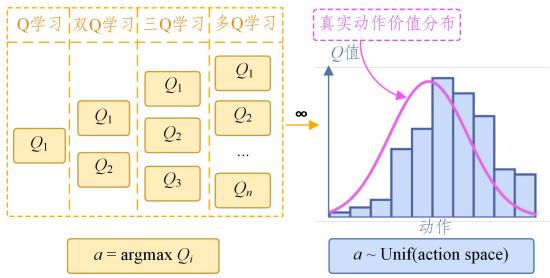


图 2　从 Q 学习扩展到多 Q 学习

Fig. 2　From Q-learning to multiple Q-learning

4.3.2　采用监督学习训练模拟器

本文首先将真实场景中的云数据中心动态负载数据（详情见 5.1 节中的 cluster-trace-v2018 数据集）加载到云仿真平台 CloudSim 中，然后计算每次调度后的过载情况和综合奖赏，将此时的云数据中心和虚拟机状态进行记录并生成数据集。同时，为了更好地提高模拟器的准确性和真实性，本文从该数据集中分离出表 1 中的模拟器训练数据集。其中，过载二分类数据集用来训练判断当前状态是否过载的二分类模型[32]，而奖励回归数据集用来训练根据当前状态和执行动作输出精准综合奖励的回归模型。

表 1　用于训练强化学习模拟器的数据集

Table 1　Datasets for training reinforcement learning simulator

Datasets	size	Positive-negative Ratio
Rewards Regression	134 223	—
Overload Classification	436 000	4∶6

基于以上数据集，本文使用多种常用的监督学习方法（使用原始默认超参数）对模拟器进行建模并得到对比结果，如表 2 所列，其中过载二分类任务采用准确率（Accuracy，ACC）进行评估，奖励回归任务采用均方误差（Mean Square Error，MSE）进行评估。

表 2 不同监督模型在分类和回归任务上的比较

Table 2 Comparison of different supervised models on classification and regression tasks

Models	$ACC\uparrow$/%	$MSE\downarrow$
ResNet	70.29	5.6530
MLP	68.28	5.7110
SVM	75.57	9.9136
Naive Bayes	98.55	—
KNN	99.99	0.2994
ID3	99.99	—
CART	99.99	0.1723
RF	96.99	0.0539
LightGBM	99.99	0.7861
XGBoost	99.99	0.3696
CatBoost	**99.99**	**0.0563**

由表 2 可知，相比其他常用的统计机器学习和深度学习模型，决策树模型在表格型数据上的分类和回归任务中整体表现卓越[33-34]，特别是其中的 CatBoost 模型[35]取得了最佳效果，这表明决策树在表格型数据中的特征提取能力整体较优[36]。因此，本文在具体实验中也采用 CatBoost 模型进行分类树和回归树训练并将其作为强化学习模拟器。在实际智能体训练过程中，该模拟器会基于输入的状态-动作二元组先通过分类树判断执行下一步动作后是否会过载，若出现过载现象，则依据过载率施行相应惩罚；反之，则对非过载状态-动作进行奖励预测。

4.3.3 基于决策树的特征提取

基于表 2 中结果得到的推论，本文将深度强化学习模型中的深度神经网络替换为 CatBoost 梯度提升决策树，以增强模型对表格型数据的特征提取能力。具体如图 1 下侧部分所示。其中，输入为状态-动作二元组，而输出为对应 Q 值。在训练的过程中，随着 $Q(s,a)$ 被更新，梯度提升决策树会不断接收新的样本进行训练，并通过建立新的决策树来拟合当前模型的梯度以减少残差，从而实现 Q 值的更新。

4.3.4 DTQL-UD 算法伪代码

本文通过改进 Q-learning 算法的更新公式并引入决策树模型来替代深度神经网络进行云数据中心状态的表征，以更好地提高表格型数据的特征提取能力，具体训练伪代码如算法 1 所示。

算法 1 DTQL-UD 算法

输入：待迁移虚拟机状态序列 $\mathbf{v}_1$，云数据中心负载序列 $\mathbf{c}_1=\{h_1,h_2,\cdots,h_N\}$

输出：奖励回归树 Q

1. 采用强化学习模拟器中的回归树初始化回归树 Q，其中 $\mathbf{s}\in\mathcal{S}^+$，$a\in\boldsymbol{A}(s)$；
2. for episode←1 to N do
3. 经验回放缓存 $\mathcal{D}\leftarrow\emptyset$，环境状态 $\mathbf{s}_1\leftarrow\{\mathbf{v}_1,\mathbf{c}_1\}$；
4. for t=1 to T do
5. 以概率 ε 随机选取一个动作 a_t，否则令 $a_t\leftarrow\arg\max_a Q^*(s_t,a)$；
6. 执行动作 a_t 并观察所得奖励 r_t 和更新后的云数据中心状态 $\mathbf{c}_{t+1}$；
7. 更新下一步的待迁移虚拟机 $\mathbf{v}_{t+1}\leftarrow\arg\max_v Q^*(\{\mathbf{v},\mathbf{c}_t\},a)$，下一步环境状态 $s_{t+1}=\{\mathbf{v}_{t+1},\mathbf{c}_{t+1}\}$，和当前步 Q 值；

$$y_t=\begin{cases}r_t, & \text{for terminal } \mathbf{s}_{t+1}\\ r_t+\gamma Q(\mathbf{s}_{t+1},a\sim U(\boldsymbol{A}(\mathbf{v}_{t+1}))), & \text{otherwise}\end{cases}$$

8. 存储 $(y_t,(\mathbf{s}_t,a_t))$ 至经验回放缓存 $\mathcal{D}$ 中；
9. end for
10. 使用经验回放缓存 $\mathcal{D}$ 增量化训练回归树 Q；
11. end for
12. return 奖励回归树 Q

在算法开始阶段，初始化 $Q(\boldsymbol{s},a)$ 为预训练回归树模型，如可采用强化学习模拟器中的回归树。在算法训练阶段：1）在每幕开始时初始化一个经验回放缓存 $\mathcal{D}$ 为空集；2）初始化环境状态 $\boldsymbol{s}_1$ 为集群负载状态 $\boldsymbol{c}_1$ 和待迁移虚拟机 $\boldsymbol{v}_1$ 按序列拼接成的一维向量；3）以概率 ε 随机选择下一步动作，反之选择当前状态对应的 Q 值最大的动作；4）执行动作并观察所获的即时奖励 r_t 和下一步云数据中心负载状态 $\boldsymbol{c}_{t+1}$，而下一步的最佳待迁移虚拟机 $\boldsymbol{v}_{t+1}$ 则通过最大 Q 值选出并结合下一步云数据中心负载构成下一步新环境状态 $\boldsymbol{s}_{t+1}$；5）在动作空间上按均匀分布选择离散动作并得到当前 Q 值的评估量 y_t，然后将 y_t 与当前状态-动作二元组一并存储到经验回放缓存 $\mathcal{D}$ 中；6）使用数据集 $\mathcal{D}$ 对回归树 Q 进行增量化训练。

5 仿真实验与分析

5.1 实验环境

本文采用两种不同功耗模型的异构主机和 4 种不同类型的虚拟机进行仿真实验，其中虚拟机的配置信息参考 Amazon EC2[37]的实例类型。主机和虚拟机的详细配置信息如表 3 和表 4 所列（本实验中的物理资源为 CPU 和 RAM），不同主机的完整功耗模型如表 5 所列。本实验中的仿真云数据中心环境包含 100 台主机和 200 台虚拟机，实验开始前将不同类型配置的虚拟机按均匀分配处理，使得不同类型的实例数量相等。为验证算法的有效性，本文采用 CloudSim 4.0[3]来评估各算法的性能，仿真开始时 CloudSim 加载指定负载文件并将其分配给预先指定的虚拟机，之后每间隔 5min 进行一次负载数据采样，并模拟云数据中心运行 24h。

表 3 主机配置

Table 3 Host configuration

Physical host	Hardware configuration
ProLiant DL560 Gen9	Intel Xeon E5-4669 v3, 2.1GHz, 2 core, 4GB
ProLiant DL325 Gen10	AMD EPYC 7551P, 2.0GHz, 2 core, 4GB

表 4 虚拟机配置

Table 4 VM configuration

Size	Large	Medium	Small	Micro
CPU/Mips	2500	2000	1500	500
Memory/GB	3.0	1.5	1.0	0.5

表 5 主机功耗模型

Table 5 Host power model

CPU utilization/%	Power consumption/(kW·h)	
	EPYC 7551P	Xeon E5-4669
0	61.7	86.3
20	105.0	218.0
40	127.0	289.0
60	145.0	340.0
80	161.0	433.0
100	181.0	557.0

本文采用的 cluster-trace-v2018 数据集为阿里巴巴开源

云数据中心跟踪数据[38]，该数据集记录了 2018 年长达 8 天不同在线应用容器和离线计算任务在 4 000 台服务器上的 CPU 和 RAM 动态负载。本实验从数据集中随机抽取同一时间周期内不同虚拟机上对应的负载文件进行任务负载模拟。

5.2　对比方法

本文在虚拟机选择和虚拟机重放置两个子问题上采用传统启发式和基于深度神经网络的强化学习算法进行对比实验，对于上游任务中的主机过载检测问题则采用排列组合方式进行交叉实验，具体对比方法如下：

(1)主机过载检测：通过指定的静态阈值来判断当前主机是否过载，当某主机的任一维度资源利用率达到或超过该阈值时被认定为过载。实验中的不同静态阈值依次设定为 Thr(0.6)，Thr(0.8)和 Thr(1)。

(2)虚拟机选择：当主机过载时，需要根据算法选择该主机上合适规格的虚拟机以便后续迁移。本实验选取主流的随机选择(Random Select，RS)、最低利用率(Minimum Utilization，MU)和最小迁移时间(Minimum Migration Time，MMT)3 种方法与其他子任务的方法进行交叉组合实验。其中 MU 方法选择当前资源利用率最低的虚拟机，而 MMT 方法选择迁移时间最少的虚拟机。由于强化学习策略直接将虚拟机选择和放置两个问题合并处理，因此强化学习方法无须组合现有虚拟机选择方法。

(3)虚拟机重放置：本实验选取了主流开源云平台 OpenStack 中常用的两种虚拟机重放置策略，包括首次适应(FF)和降序首次适应(FFD)。

(4)D3QN[39]：基于竞争架构和双学习的深度 Q 网络模型(Dueling Double DQN，D3QN)，在原深度 Q 网络架构基础上通过将隐藏层解耦为优势函数和价值函数双通道独立输出，从而进一步缓解强化学习中 Q 值的过估计问题。考虑到实际训练中出现的梯度弥散问题，本实验选取 ResNet 作为 D3QN 中的神经网络。

此外，由于 DTQL-UD 算法仅在 Q 值更新公式中进行改进，因此 DTQL-UD 与 DTQL 共享相同的超参数。同时，本文基于表 1 中的奖励回归数据集得到能耗和资源利用率奖励值的箱线图，如图 3 所示。其中图 3 左侧两个箱线图表示原数据集。考虑到能耗与资源利用率奖励之间的平衡，本文设置能耗缩放因子 $\alpha=0.9$，资源利用率缩放因子 $\beta=4.9$，并得到图 3 右侧经过加权后的箱线图。具体各算法的超参数设置如表 6 所列。

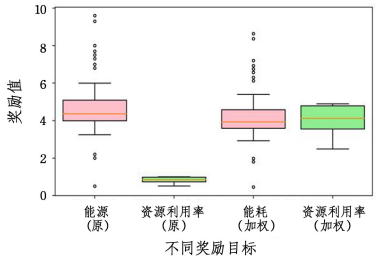


图 3　回归数据集中能耗和资源利用率的奖励值箱线图

Fig. 3　Reward value boxplot for energy consumption and resource utilization in regression datasets

表 6　各算法超参数设置

Table 6　Algorithm hyperparameter setting

Hyperparameter	D3QN-UD	D3QN
Energy reward scaling factor α	0.90	0.90
Utilization reward scaling factor β	4.90	4.90
Exploration rate ε	0.01	0.01
Learning rate of agent α_1	0.99	0.99
Discount factor γ	0.95	0.95
Learning rate of decision tree α_2	—	0.10
Decision tree depth d	—	8.00

5.3　评估指标

考虑到 3.4 节目标函数以及虚拟机整合策略性能评估的有效性，本小节使用到的评价指标具体如下：

(1)能耗：见 3.1 节式(2)。

(2)云数据中心资源利用率：见 3.2 节式(3)。

(3)服务质量：见 3.3 节式(4)。

(4)虚拟机迁移次数：由于处于迁移状态的虚拟机会导致自身约 90%的性能下降以及其所属主机约 10%的性能下降[3]，因此，虚拟机整合策略应尽可能减少虚拟机迁移的次数。

此外，为了更全面地进行对比并突出本文算法的优化效果，本实验还补充了剩余活跃主机数量和两种强化学习方法(基于深度学习与基于决策树)之间的累计平均奖励及其训练耗时对比。

5.4　实验结果

5.4.1　强化学习策略的累计奖励和训练耗时

图 4 给出了基于深度学习与基于决策树的强化学习及其改进策略共 4 种方法在 5 次重复实验下的累计平均奖励对比。其中，D3QN-UD 和 DTQL-UD 分别表示在原方法基础上将 Q 值更新公式中目标 Q 值的动作选取方法替换为按均匀分布选取动作。而 DTQL 与 DTQL-UD 均采用相同 CatBoost 预训练模型进行训练，因此在训练初期可达到约 3 500 的累计奖励值，并于第 11 000 步时开始稳定收敛。

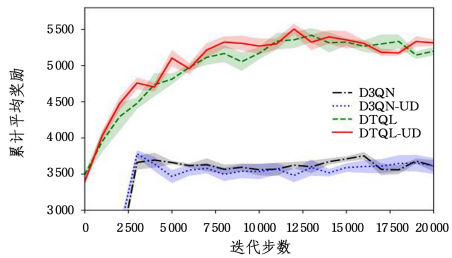


图 4　两种强化学习及其改进策略平均累计奖励对比

Fig. 4　Comparison of average cumulative rewards for two RL models and their improvement strategies

图 4 中的结果表明，在虚拟机整合场景下，基于决策树的强化学习方法在收敛时比基于深度学习的强化学习方法多获取了约 46%的平均累计奖励，这意味着前者更容易学习到较优的虚拟机整合策略。这是由于决策树在表格型数据上有更优的特征提取能力[35]，以此可以更好地识别出不同的负载状态，从而有助于 Q 值的准确评估。同时，从图中可见，基于神经网络的 D3QN 模型过早地收敛到一个累计奖励较低的水平，我们分析这是神经网络在本任务中的特征提取能力不足，

从而难以区分相似状态所致。此外，采用均匀分布(UD)更新Q值的方法可获得与原方法几乎相同的累计平均奖励，这表明UD的寻优能力不差于原方法，从而验证了UD方法的有效性。

图5给出了4种强化学习策略在不同主机规模下的每幕训练耗时。本实验结果在操作系统版本为MacOS 11.1的MacBook Pro主机上进行5次重复实验得到，主机配置为4核Intel(R)Core(TM)i5 CPU @ 2.40 GHz，8 GB 2133 MHz LPDDR3型号内存。以上策略均使用Python语言编程，且D3QN和D3QN-UD均基于PyTorch 1.6.0框架实现。

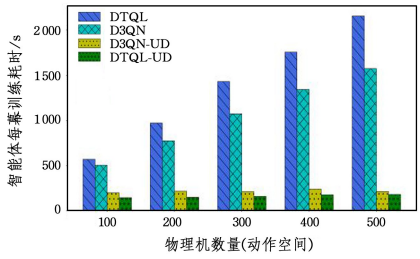


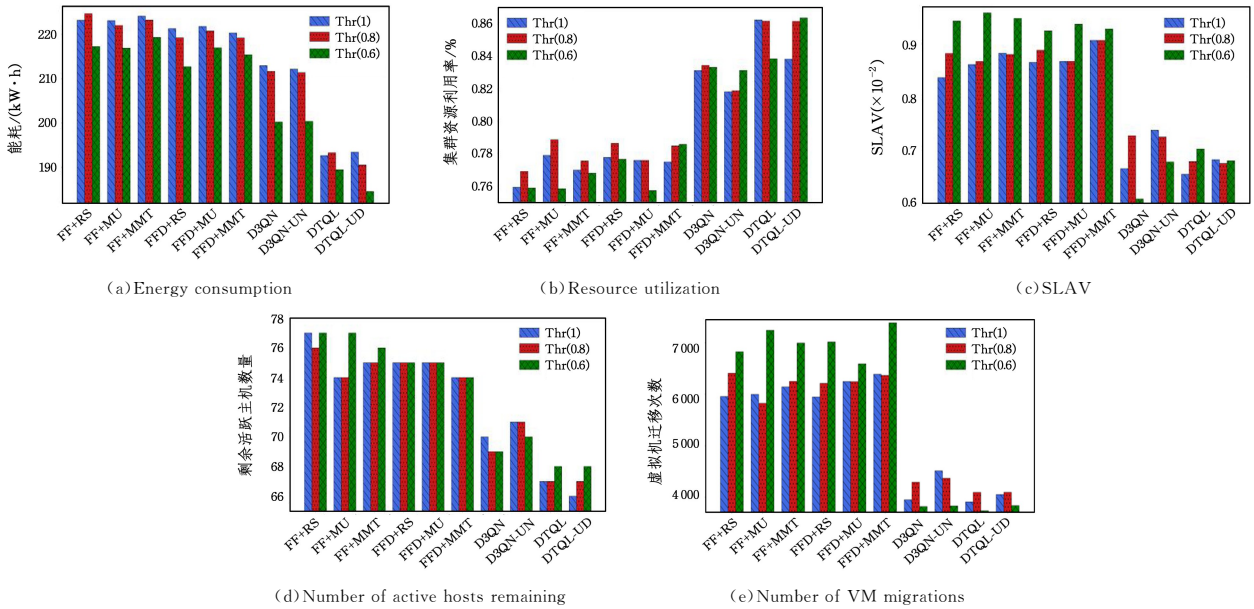
图5 两种强化学习模型及其改进算法的训练耗时对比

Fig. 5 Comparison of training time-consuming for two RL models and their improvement strategies

图5中的结果表明，智能体按均匀分布选取动作时，其训练耗时少于原方法，且当云数据中心主机数量增加到500台时，基于决策树的强化学习方法中，DTQL-UD的平均每幕训练耗时仅为DTQL的约8%；而在基于深度学习的强化学习方法中，D3QN-UD的平均每幕训练耗时仅为D3QN的约13%，即按均匀分布选取动作的耗时随动作空间规模的增长几乎未发生变化，而原有DTQL方法和D3QN方法的耗时与离散动作规模具有正比例线性关系。其关键原因在于按均匀分布选取动作时既不需要因为argmax算子而遍历每个动作价值，也不需要通过增加额外Q网络进行训练。综上所述，对于大规模离散动作任务，采用均匀分布选取动作可大幅降低强化学习方法的训练耗时。

5.4.2 虚拟机迁移性能

图6给出了传统启发式和基于深度学习与基于决策树的强化学习及其改进方法等10种算法在能耗、资源利用率、用户服务质量、剩余活跃主机数量和VM迁移次数指标上的对比。其中，传统启发式调度策略由不同虚拟机选择策略和虚拟机重放置策略排列组合而成，如FF+RS组合策略表示当出现过载或欠载主机时，首先在该主机上通过随机选择策略选择待迁移VM，然后通过首次适应策略找到目标主机并进行迁移。



(a) Energy consumption (b) Resource utilization (c) SLAV

(d) Number of active hosts remaining (e) Number of VM migrations

图6 不同虚拟机整合算法的性能对比

Fig. 6 Performance comparison of different virtual machine consolidation algorithms

图6中的结果表明，与传统启发式调度策略(FF+RS等)相比，基于深度学习的强化学习算法(D3QN和D3QN-UD)在各性能指标上均有所提升：平均能耗降低4%～5%，资源利用率提高6%～10%，剩余活跃主机数量减少7%～8%，用户服务质量提升18%～23%，VM迁移次数减少28%～45%。基于决策树的强化学习算法(DTQL和DTQL-UD)在各性能指标上表现更优：平均能耗降低13%～15%，资源利用率提高11%～13%，剩余活跃主机数量减少10%～11%，用户服务质量提升19%～22%，VM迁移次数减少35%～43%。综上所述，基于强化学习的4种整合算法的整体性能优于传统启发式调度策略，分析其中的原因主要有以下两方面：一是强化学习智能体通过累计奖励来寻找较优的下一步动作，在一定程度上避免了智能体陷入局部最优解；二是4种算法均将虚拟机选择和虚拟机重放置问题整合到一起求解，使得模型拥有更全局的视野，从而可以更灵活地求解多目标优化问题。

此外，结合图5可知，D3QN-UD和DTQL-UD方法在大规模离散动作空间任务中保持较低训练耗时的同时，在所有评估指标上依然与原方法有着几乎同等的性能表现。这是由于按均匀分布选取下一时刻目标动作的方式在缓解模型训练

初期产生的最大化偏差的基础上可以更高频地选取到期望价值对应的动作，以此逼近状态的期望价值，这样既不需要训练额外的Q网络，也无须通过argmax算子来遍历动作空间中的所有动作，因此可大幅缩短训练耗时。

结束语　本文研究了云资源调度中的虚拟机整合问题，并针对现有策略在能耗、资源利用率和用户服务质量等多指标上的性能瓶颈，提出了一种基于决策树和由均匀分布改进Q学习的虚拟机整合算法(DTQL-UD)。该算法通过均匀分布改进Q学习更新公式中的动作选择方式以缩短训练耗时，并采用决策树代替神经网络进行状态表征以提高表格型数据的特征提取能力，同时使用监督学习模型训练模拟器以增强模拟器仿真度。实验结果表明，DTQL-UD与现有启发式方法相比，可同时提升多个性能指标，并可随云数据中心规模的扩充，大幅缩短现有强化学习方法的训练耗时。

由于数据量的限制，本文中的强化学习模拟器和真实环境间还存在一定差异。此外，本文仅针对云数据中心中的CPU和RAM两个维度的资源进行建模。在未来的工作中，可以加入网络带宽进行建模以充盈资源维度并在实际生产环境(如OpenStack)中进行测试，从而提升真实场景下的云数据中心性能。

参 考 文 献

[1] HAMEED A, KHOSHKBARFOROUSHHA A, RANJAN R, et al. A Survey and Taxonomy on Energy Efficient Resource Allocation Techniques for Cloud Computing Systems[J]. Computing, 2016, 98(7): 751-774.

[2] QURESHI A, WEBER R, BALAKRISHNAN H, et al. Cutting the Electric Bill for Internet-scale Systems[J]. ACM SIGCOMM Computer Communication Review, 2009, 39(4): 123-134.

[3] CALHEIROS R N, RANJAN R, BELOGLA-ZOV A, et al. CloudSim: A Toolkit for Modeling and Dimulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms[J]. Software: Practice and Experience, 2011, 41(1): 23-50.

[4] HU Z G, XIAO H, LI K Q. Virtual Machine Consolidation Algorithm Based on Multi-objective Optimization in Cloud Computing[J]. Journal of Hunan University(Natural Sciences), 2020, 47(2): 116-124.

[5] HIEU N T, DI FRANCESCO M, YLÄJÄ-ÄSKI A. Virtual Machine Consolidation with Multiple Usage Prediction for Energy-efficient Cloud Data Centers[J]. IEEE Transactions on Services Computing, 2020, 13(1): 186-199.

[6] YU X, LI Z Y, SUN S, et al. Adaptive Virtual Machine Consolidation Based on Deep Reinforcement Learning[J]. Journal of Computer Research and Development, 2021, 58(12): 2783-2797.

[7] PRABHA B, RAMESH K, RENJITH P N. A Review on Dynamic Virtual Machine Consolidation Approaches for Energy-Efficient Cloud Data Centers[M]//Data Intelligence and Cognitive Informatics. Springer, Singapore, 2021: 761-780.

[8] WANG K, QU H, ZHAO J H. Multi-objective Optimization Method Based on Reinforcement Learning in Multi-domain SFC Development[J]. Computer Science, 2021, 48(12): 324-330.

[9] XIE S Q, CHEN Z T, XU C, et al. Environment Upgrade Reinforcement Learning for Non-differentiable Multi-stage Pipelines[J]. Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition), 2020, 32(5): 857-858.

[10] SUTTON R S, BARTO A G. Reinforcement Learning: an Introduction[M]. Massachusetts: MIT Press, 2018.

[11] CHENG Z K, YAN X L, CHENG W S, et al. Research on Coke Quality Prediction Model Based on Gradient Boosting Decision Tree[J]. Journal of Chongqing Technology and Business University(Natural Science Edition), 2021, 38(5): 55-60.

[12] VAN HASSELT H, GUEZ A, SILVER D. Deep Reinforcement Learning with Double Q-learning[C]// Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, Arizona, USA: AAAI Press, 2016: 2094-2100.

[13] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous Control with Deep Reinforcement Learning[J]. arXiv: 1509.02971, 2015.

[14] FUJIMOTO S, VAN HOOF H, MEGER D. Addressing Function Approximation Error in Actor-critic Methods [C]// International Conference on Machine Learning(ICML). PMLR, 2018: 1587-1596.

[15] ABDULLAH M, LU K, WIEDER P. A Heuristic-Based Approach for Dynamic Vms Consolidation in Cloud Data Centers[J]. Arabian Journal for Science and Engineering, 2017, 42(8): 3535-3549

[16] BELOGLAZOV A, BUYYA R. Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers[J]. Concurrency and Computation: Practice and Experience, 2012, 24(13): 1397-1420.

[17] FARAHNAKIAN F, ASHRAF A, PAHIKKALA T. Using Ant Colony System to Consolidate VMs for Green Cloud Computing[J]. IEEE Transactions on Services Computing, 2015, 8(2): 187-198.

[18] SINGH N, DHIR V. Hypercube Based Genetic Algorithm for Efficient Vm Migration for Energy Reduction in Cloud Computing [J]. Statistics, Optimization & Information Computing, 2019, 7(2): 468-485.

[19] ZHANG Y, WANG Y, WANG H. Energy-Efficient Task Scheduling for DVFS-enabled Heterogeneous Computing Systems Using a Linear Programming Approach[C]// 2016 IEEE 35th International Performance Computing and Communications Conference(IPCCC). IEEE, 2016: 1-8.

[20] ANASTASOPOULOS M, TZANAKAKI A, SIMEONIDOU D. Stochastic Energy Efficient Cloud Service Provisioning Deploying Renewable Energy Sources[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(12): 3927-3940.

[21] RASOULI N, RAZAVI R, FARAGARDI H R. EPBLA: Energy-efficient Consolidation of Virtual Machines Using Learning Automata in Cloud Data Centers[J]. Cluster Computing, 2020, 23(4): 3013-3027.

[22] MA Z J. Research on Energy-aware Virtual Machine Consolidation Technology in Cloud Computing Environment[D]. Guangzhou: South China University of Technology, 2021.

[23] CHEN T. Research on Dynamic Integration Strategy of Virtual Machine Based on MOPOS Algorithm[D]. Xi'an: Xidian University, 2021.

[24] HAGHSHENAS K, PAHLEVAN A, ZAPATER M, et al. Magnetic: Multi-agent Machine Learning-based Approach for Energy Efficient Dynamic Consolidation in Data Centers[J]. IEEE Transactions on Services Computing, 2022, 15(1): 30-44.

[25] DING W, LUO F, GU C, et al. Performance-to-power Ratio Aware Resource Consolidation Framework based on Reinforcement Learning in Cloud Data Centers[J]. IEEE Access, 2020, 8: 15472-15483.

[26] THEIN T, MYO M M, PARVIN S, et al. Reinforcement Learning based Methodology for Energy-efficient Resource Allocation in Cloud DataCenters[J]. Journal of King Saud University-Computer and Information Sciences, 2020, 32(10): 1127-1139.

[27] HUANG N X, YIN X, YUE Y L, et al. An Improved Deep Reinforcement Learning Algorithm Based on Meta-learning[J]. Journal of Yangzhou University (Natural Science Edition), 2021, 24(3): 19-23.

[28] FAN J Y, LIU Q. Off-policy MaximumEntropy Deep Reinforcement Learning Algorithm Based on Randomly Weighted Triple Q-Learning[J]. Computer Science, 2022, 49(6): 335-341.

[29] MASOUMZADEH S S, HLAVACS H. Int-egrating VM Selection Criteria in Distributed Dynamic VM Consolidation Using Fuzzy Q-Learning[C]// Proceedings of the 9th International Conference on Network and Service Management(CNSM). IEEE, 2013: 332-338.

[30] KUSIC D, KEPHART J O, HANSON J E, et al. [J]. Cluster Computing, 2009, 12(1): 1-15.

[31] BELLEMARE M G, DABNEY W, MUNOS R. A Distributional Perspective on Reinforcement Learning[C]// International Conference on Machine Learning(ICML). PMLR, 2017: 449-458.

[32] OU D X, ZHANG X Y, ZHAO Y, et al. Urban Rain Transit Train Accident Delay Time Prediction Based on GBDT Cascade Classification Method[J]. Urban Mass Transit, 2022, 25(10): 65-70.

[33] YIN C Y, SHAO C F, HUANG Z G, et al. Investigating Influences of Multi-scale Built Environment on Car Ownership Behavior Based on Gradient Boosting Decision Trees[J]. Journal of Jilin University (Engineering and Technology Edition), 2022, 52(3): 572-577.

[34] LIU J, ZHAO J, FENG Y M, et al. Power Load Forecasting in Power Internet of Things Based on Gradient Boosting Decision Tree[J]. Smart Power, 2022, 50(8): 46-53.

[35] PROKHORENKOVA L, GUSEV G, VOROBEV A, et al. CatBoost: Unbiased Boosting with Categorical Features[C]// Advances in Neural Information Processing Systems(NIPS). 2018: 1-11.

[36] GORISHNIY Y, RUBACHEV I, KHRU-LKOV V, et al. Revisiting Deep Learning Models for Tabular Data[J]. arXiv: 2106.11959, 2021.

[37] HABIBA, KHAN M I. Reinforcement Learning based Autonomic Virtual Machine Management in Clouds[C]// 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV). IEEE, 2016: 1083-1088.

[38] CHENG Y, CHAI Z, ANWAR A. Characteri-zing Co-located Datacenter Workloads: An Alibaba Case Study[C]// Proceedings of the 9th Asia-Pacific Workshop on Systems. 2018: 1-3.

[39] WANG Z, SCHAUL T, HESSEL M, et al. Dueling Network Architectures for Deep Reinforcement Learning[C]// International Conference on Machine Learning(ICML). PMLR, 2016: 1995-2003.

SHI Liang, born in 1996, postgraduate. His main research interests include cloud resource scheduling and reinforcement learning.

LI Jianhui, born in 1973, Ph.D, professor, Ph.D supervisor. His main research interests include cloud computing, distributed systems, and artificial intelligence for IT operations.

(责任编辑:何杨)