

一种基于深度强化学习与概率性能感知的边缘计算环境多 workflow 卸载方法



马培银¹ 郑万波² 马 勇³ 刘 航¹ 夏云霓¹ 郭坤银¹ 陈 鹏⁴ 刘诚武⁵

1 重庆大学计算机学院 重庆 400044

2 昆明理工大学理学院 昆明 650500

3 江西师范大学计算机信息工程学院 南昌 330022

4 西华大学计算机与软件工程学院 成都 610039

5 上海交通大学重庆研究院 重庆 401135

(mayuyin@cqu.edu.cn)

摘 要 移动边缘计算是一种新兴的分布式和泛在计算模式,其将计算密集型和时延敏感型任务转移到附近的边缘服务器,有效缓解了移动终端资源不足的问题,显著减小了用户与计算处理节点之间的通信传输开销。然而,如果多个用户同时提出计算密集型任务请求,特别是流程化的 workflow 任务请求,边缘计算环境往往难以有效地进行响应,并会造成任务拥塞。另外,受任务负载、电力供给、通信能力的实时变化等不利因素的影响,边缘服务器本身的性能总是处于波动和变化中,从而为保证任务执行效能和用户感知服务效率带来了挑战。针对上述问题,文中提出了一种基于深度 Q 网络(DQN)与概率性能感知机制的边缘计算环境多 workflow 调度方法。首先对边缘云服务器的历史性能数据进行概率分析,然后利用获得的性能概率分布数据驱动 DQN 模型,不断迭代优化,生成多 workflow 的卸载策略。在实验验证环节,基于边缘服务器位置数据集、性能测试数据和多个科学 workflow 模板,在反映不同系统负载水平的多个场景下进行了模拟实验。实验结果表明,所提方法在多 workflow 执行效率方面明显优于传统方法。

关键词: workflow 调度;边缘计算;概率分布模型;强化学习;深度 Q 网络

中图法分类号 TP301

Multi-workflow Offloading Method Based on Deep Reinforcement Learning and Probabilistic Performance-aware in Edge Computing Environment

MA Yu-yin¹, ZHENG Wan-bo², MA Yong³, LIU Hang¹, XIA Yun-ni¹, GUO Kun-yin¹, CHEN Peng⁴ and LIU Cheng-wu⁵

1 College of Computer Science, Chongqing University, Chongqing 400044, China

2 Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China

3 School of Computer Information Engineering, Jiangxi Normal University, Nanchang 330022, China

4 School of Computer Science and Software Engineering, Xihua University, Chengdu 610039, China

5 Shanghai Jiaotong University Chongqing Research Institute, Chongqing 401135, China

Abstract Mobile edge computing is a new distributed and ubiquitous computing model. By transferring computation-intensive and time-delay sensitive tasks to closer to the edge servers, it effectively alleviates the resource shortage of mobile terminals and the communication transmission overhead between users and computing processing nodes. However, if multiple users request computation-intensive tasks simultaneously, especially process-based workflow task requests, edge computing are often difficult to respond effectively and cause task congestion. In addition, the performance of edge servers is affected by detrimental factors such as task overload, power supply and real-time change of communication capability, and its performance fluctuates and changes, which brings challenges to ensure task execution and user-perceived service efficiency. To solve the above problems, a Deep-Q-Network (DQN) and probabilistic performance aware based multi-workflow scheduling approach in edge computing environment

到稿日期: 2020-09-27 **返修日期:** 2020-12-09 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目: 重庆市研究生科研创新项目(CYS20066, CYB20062); 重庆市科技局技术创新项目(cstc2019jszx-msxm0652, cstc2019jszx-fxyd0385); 四川省科技计划项目(2020JDRC0067, 2020YFG0326); 西华大学人才引进项目(Z202047); 重庆市科技局重点研发计划项目(cstc2018jszx-cyzdX0081); 江西省重点研发计划(20181ACE50029)

This work was supported by the Graduate Scientific Research and Innovation Foundations of Chongqing(CYS20066, CYB20062), Chongqing Technological Innovation Foundations (cstc2019jszx-msxm0652, cstc2019jszx-fxyd0385), Sichuan Scientific Foundations (2020JDRC0067, 2020YFG0326), Research Talent Foundation of Xihua University(Z202047), Chongqing Key R&D Project(cstc2018jszx-cyzdX0081) and Jiangxi Key R&D Project(20181ACE50029).

通信作者: 郑万波(zwanbo2001@163.com)

is proposed. Firstly, the historical performance data of edge cloud servers is analyzed probabilistically, then the DQN model is driven by performance probability distribution data, and iterative optimization is carried out continuously to generate multi-workflow offloading strategy. In the process of experimental verification, simulation experiments are conducted in multiple scenarios reflecting different levels of system load based on edge server Location data set, performance test data and multiple scientific workflow templates. The results show that the proposed method is superior to the traditional method in the execution efficiency of multi-workflow.

Keywords Workflow scheduling, Edge computing, Probability distribution model, Reinforcement learning, Deep Q network

1 引言

随着智能手机的不断发展以及诸多移动端应用的不断出现,用户对移动设备便携式管理以及处理大量及时性信息等方面的需求日益增长。同时,各种新兴的移动应用设备也随着 5G 技术的发展而大量普及,各种交互式游戏、虚拟现实、增强现实等应用程序需要大量计算和高能耗的资源。然而,移动设备有限的电池寿命、计算容量、通信传输能力使得纯粹依赖移动设备本身来承担计算处理任务并不现实,特别是资源饥渴的应用(如视频帧预处理、对象跟踪、实时三维渲染、大规模科学计算、图像图形处理等)。为了解决移动设备中资源限制的问题,将任务卸载到外部的计算节点上,如云服务和边缘服务器^[1],进而使应用需求方以较小的时间代价和传输延迟来实现对计算存储资源的调用和交互,是一种行之有效的方法。

移动边缘计算起源于云计算,指在靠近物体或数据源头的一侧采用集网络、计算、存储、应用核心能力为一体的开放计算构架。边缘计算就近提供最近端服务,其应用程序在边缘侧发起任务请求,产生更快的网络服务响应,从而满足行业在实时业务、应用智能、安全与隐私保护等方面的基本需求。边缘计算处于物理实体和工业连接之间,或处于物理实体的顶端。在 MEC 平台中,在靠近移动用户的网络边缘部署了大量小型专用服务器,用户可以将计算密集型任务卸载到离其很近的边缘云上,在自己的移动终端上平稳地运行一个沉浸式应用程序。单个边缘服务器中的资源往往不足以满足多个用户提出的请求,特别是在为多用户的大规模科学计算任务与流程化工作流任务同时提供服务时。因此,应考虑多个边缘节点和服务器以协同的方式服务多工作流的任务,每个用户可以将自己的工作流任务进行分解并卸载到不同的边缘服务器中执行。

近年来,云环境和边缘计算环境中工作流卸载的问题已经引起了科学界和经济领域学者们的关注。一个工作流(Workflow)通常可以表示为一个含有有限数量任务的定向无环图(Directed-Acyclic-Graph, DAG)。任务的执行须服从工作流的结构约束,即前驱任务必须先于后继任务执行。在边缘计算环境中卸载工作流,需综合考虑对权衡性能、执行代价、可靠性等多方面的约束的优化。众所周知,分布式资源上多约束的多工作流最优卸载问题是一个 NP 难题^[2]。因此,近年来的研究多采取启发式和元启发式算法来生成近优的卸载解^[3-8]。

随着新型机器学习算法的广泛使用,越来越多的工作将强化学习(Reinforcement Learning, RL)和 Q 学习(Q-learning)的思想与算法^[9-11]应用于各种组合优化问题。然而,大多

数现有的相关文献都关注在传统云环境中的工作流调度问题,如何将 Q-learning 的算法和策略应用于边缘计算环境中的多工作流卸载,仍然是一个亟待解决的问题。另外,已有的大部分利用深度学习进行工作流调度的方法,假设云服务的实时性能具有恒定不变或者服从指定分布类型的特性,进而以固定的模型参数输入组合优化模型。我们在后续的实验验证中发现,基于上述假设的策略和算法在实际的边缘计算环境下多工作流调度的性能是实时波动的。为了弥补上述不足,本文提出了一种基于深度 Q 网络(Deep Q Network, DQN)的概率性能感知多工作流卸载方法。区别于大多数现有的方法,该方法假设真实环境下运行的边缘服务器的性能是非稳定且随时间波动的。本文首先对最近一个时间窗口内的边缘服务的历史性能进行概率建模,根据概率分布特性输入组合优化问题,然后利用深度强化学习来进行求解,最后基于真实的公开边缘云服务器位置分布数据集^[12]和性能数据集^[13],进行实验性案例研究。实验结果清楚地表明,本文方法相比多种传统的方法实现了更短的工作流完成时间。

2 相关工作

众所周知,基于分布式资源(如网格、云计算节点、边缘服务器)的多约束、多工作流最优卸载问题是一个 NP 难优化问题,因此采用启发式和元启发式算法来获得近优的卸载方案是行之有效的方法。例如,Kaur 等^[14]提出了一种结合帕累托最优与多目标生物启发过程(Multi-Objective Bio-inspired Procedure, MOBFOA)的云工作流卸载方法。Zhang 等^[15]以优化系统功耗和工作流的可靠性为目标,提出了多目标遗传优化算法(Multi-objective genetic optimization, BOGA)。Casas 等^[16]提出了一种具有高效协调(GA with the Efficient Tune-In, GA-ETI)机制的增广遗传算法,用于优化多工作流在云上的运行时间。Verma 等^[17]采用了一种混合多目标粒子群算法(Multi-Objective Particle Swarm Optimization, MOPSO),来优化云上工作流的运行时间和成本。Zhou 等^[18]引入了一种基于模糊排序的工作流完成时间最小化方法,用于优化运行在异构基础设施云资源上的工作流的成本和运行时间。Habak 等^[19]提出并设计了通过网络边缘移动设备提供云服务的微云系统。他们认为任务卸载是系统设计最重要的部分,并开发了一个优化框架,该框架提供了针对问题的可扩展启发式解。Mao 等^[20]提出了一个多用户移动设备资源管理在线算法,旨在最小化移动设备和移动边缘服务器的能耗,同时满足任务缓冲区稳定性约束。他们还开发了一个延迟改进机制来降低执行延迟。为了确保云资源的有效利用,Tong 等^[21]提出了一种工作负载放置算法,该算法确定目标边缘服务器且为给定任务分配计算能力,其基本思想是

将超出边缘云服务器较低层的能力聚集并服务于边缘云层次结构中较高层的其他服务器。Zhang 等^[22]研究了在满足用户服务质量(Quality of Services, QoS)需求的情况下,边缘计算系统任务卸载问题的成本优化。Zhao 等^[23]考虑了异构云环境中具有延迟限制约束的任务,其目标是合理分配资源,以最大化任务满足延迟需求的概率。

值得注意的是,受任务负载、电力供给、通信能力的实时变化等不利因素的影响,边缘服务器本身的性能总处于波动和变化中。如何描述上述真实环境中的性能波动,并针对性地设计任务调度和卸载策略,也具有相当大的难度。近年来,学者们针对性地提出了基于概率性能感知和概率分析驱动的云/边环境中任务调度和卸载的策略。例如,Huang 等^[24]使用概率质量函数,将服务的 QoS 值表示为离散随机变量来解决服务选择问题,并以最大化组合服务约束的概率值为目标。他们的方法是首先初始化 Web 服务,然后利用概率值为单个任务选择最优服务,最后使用模拟退火算法进行调整,得到最优解。Pan 等^[25]认为科学计算程序由具有波动 QoS 的分散云基础设施支持,目标是在满足完成时间约束的条件下优化工作流成本。他们利用基于时间序列的预测模型和基于关键路径时间估计(简称 CPDE)的 VM 选择策略来解决具有时变性能的工作流调度问题。该方法实时利用云基础设施的性能变化趋势,生成动态工作流调度计划。Li 等^[26]考虑了时变性能,其目标是在满足用户服务水平协议的条件下,最小化部署在基础设施即服务云上的工作流的执行成本。他们使用基于时间序列的方法来捕获动态性能波动,向遗传算法提供 VM 的预测性能,并在运行时生成调度策略。

近年来,深度强化学习的广泛运用为多工作流卸载提供了新的思路^[27-33]。例如,Duan 等^[34]采用顺序协作博弈策略,卸载异构云平台上含有“任务打包模式”的工作流。Cui 等^[35]开发了一种基于强化学习、适用于可变优先级的多工作流实时卸载方法。Iranpour 等^[36]提出了一种面向大规模软件(即服务应用)并基于模糊博弈论模型的分布式负载管理和服务访问控制方法。Wu 等^[37]通过考虑加权适应度值函数,提出了改进的 Q-learning 方法,用于缩短运行时间并实现负载均衡,进而实现云上多工作流卸载的多目标寻优。Dong 等^[38]提出了基于深度强化学习架构的任务调度算法,该算法将具有优先级关系的任务动态地调度到云服务器,以最小化任务执行时间。Peng 等^[39]设计了一种基于强化学习和排队理论的任务调度策略,并利用状态聚合技术加快学习过程,从而优化资源约束条件下的云资源和任务调度的响应时间。Cheng 等^[40]提出了一种基于深度强化学习的新型云系统,该系统具有两阶段的资源配置和任务调度,以降低云服务提供商的能源成本,而这些提供商拥有大规模的数据中心和大量具有依赖关系的用户请求。该系统实现了极高的能耗效率、低拒绝率、较短的运行时间和快速收敛。Wang 等^[41]提出了一种使用深度强化学习求解中心化云服务上多工作流调度的方法,实现了通过迭代训练寻优来确定满足多目标约束的多工作流云上调度的方案。

3 模型与问题描述

3.1 系统模型

如图 1 所示,一个边缘计算环境由多个部署在靠近用户

端的边缘服务器组成。用户可以将计算密集型或延迟敏感的计算处理任务,如虚拟现实、增强现实、高通量科学计算等任务卸载和调度到边缘服务器上运行。本文假设边缘计算环境中存在 n 个用户(用集合 $U = \{u_1, u_2, \dots, u_n\}$ 表示)和 m 个基站(用集合 $B = \{b_1, b_2, \dots, b_m\}$ 表示)。每个用户待执行的任务流程由一个工作流的形式进行组织,用 $W = (T, D)$ 表示。其中, $T = \{t_1, t_2, \dots, t_n\}$ 表示任务集合,每一个任务都有多种类型,且各自的输入数据大小不同; $D = \{d_{i,j} | i, j \in [1, n]\}$ 表示任务之间的依赖关系, $d_{i,j} = 1$ 意味着 t_j 任务只能在 t_i 任务执行完成之后才能执行。用户的移动设备可以通过无线访问点将工作流中的一个或者多个任务卸载到靠近基站的边缘服务器上。本文使用 $S_i = \{s_1, s_2, \dots, s_n\}$ 表示覆盖用户 i 的服务器列表。

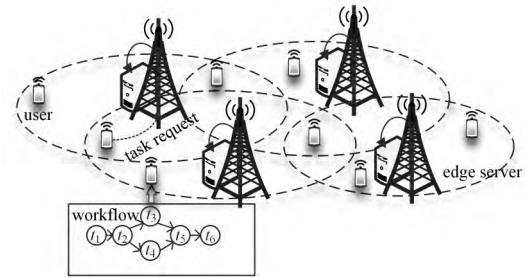


图 1 边缘计算环境

Fig. 1 Edge computing environment

用户 u_i 的任务卸载动作用一个集合 $a_i = \{s_1, s_2, \dots, s_m\}$ 表示,其中 s_j 表示第 j 个服务器所有用户的动作配置文件,用集合 $A = \{a_1, a_2, \dots, a_n\}$ 表示。

根据文献^[42]的分析,一般而言,用户 u_i 可通过边缘计算环境中的无线信道到达服务器 s_j ,其上行数据率通常可用式(1)计算:

$$R_{i,j}(A) = B \cdot \log_2 \left(1 + \frac{p_i g_{i,j}}{\sum_{k \in UL_j} p_k g_{k,j} + \sigma} \right) \quad (1)$$

其中, B 是信道带宽, p_i 是终端数据的传输能力, $g_{i,j}$ 是用户 u_i 到服务器 s_j 的信道增益, σ 是背景噪声功率。由式(1)可以看出,如果过多的用户选择将任务卸载到同一服务器,则上行数据速率降低,进而导致卸载效率降低。

假设用户 u_i 选择将任务 t_j 卸载到服务器 s_k ,根据式(1),卸载数据量大小为 $C_{i,j,k}$ 的输入数据的传输时间一般可用式(2)计算:

$$TT_{i,j,k}(A) = \frac{C_{i,j,k}}{R_{i,j,k}} = \frac{C_{i,j,k}}{B \log_2 \left(1 + \frac{p_i g_{i,j}}{\sum_{k \in UL_j} p_k g_{k,j} + \sigma} \right)} \quad (2)$$

正如前文的分析,受任务负载、电力供给、通信能力的实时变化等不利因素的影响,边缘服务器本身的性能总是处于波动和变化中,这将导致同一个任务或者具有相同计算复杂度的任务,在不同的时间部署在同一个边缘服务或边缘节点上,其执行时间也很可能是不一致的。因此,我们假设边缘服务执行工作流任务的历史完成时间服从非指定类型的一般化概率分布。一般认为,边缘计算环境下的无线信道遵守准静态块衰落规则^[42],这意味着在传输过程中信道的传输速率可视为保持不变。因此,某工作流任意任务的完成时间的概率分布可用式(3)一式(5)计算:

$$T_{i,j,k} = TT_{i,j,k}(A) + TE_{i,j,k} \quad (3)$$

$$PMF_{i,j,k}^{TE}(t) = Prob(TE_{i,j,k}) \quad (4)$$

$$Prob(T_{i,j,k}) = PMF_{i,j,k}^T(t) = PMF_{i,j,k}^{TE}(t - TT_{i,j,k}) \quad (5)$$

其中, TE 为历史运行时间, $PMF(t)$ 表示历史执行时间的概率分布函数。

3.2 问题描述

基于上文的系统模型,本文感兴趣的是,如何在确保距离约束和用户截止时间的前提下,最大化满足完成时间约束的工作流的概率。由此产生的多工作流卸载问题可表示为:

$$\max f = Prob_{avg} = \frac{1}{n} \sum_{i=1}^n Prob(CT_i \leq C_i^s) \quad (6)$$

约束条件如下:

$$dis_{i,j} \leq cov_j, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \quad (7)$$

$$x_{i,j} \leq 1, x_{i,j} = \begin{cases} 1, & \text{if } s_j \text{ is selected for } u_i \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$CT_i \geq 0, C_i^s \geq 0, i \in [1, n] \quad (9)$$

其中, C_i^s 是用户 u_i 的截止完成时间, CT_i 是用户计算任务的实际完成时间; $dis_{i,j}$ 是用户 u_i 到服务器 s_j 的距离, cov_j 是服务器 s_j 的覆盖范围; $x_{i,j}$ 是一个布尔值, 表示同一时刻用户 u_i 只能被卸载到一个服务器 s_j 中执行。

4 基于深度 Q 网络与概率性能感知的多工作流卸载方案

4.1 分解全局约束

为了验证智能体在训练过程中行为的有效性,首先把全局的约束分解成局部约束。给定一个包含 n 个任务的工作流,用 $W = \{t_1, t_2, \dots, t_n\}$ 表示, C^s 是全局完成时间约束,单个任务的局部约束为 $C' = \{C'_1, C'_2, \dots, C'_n\}$ 。本文考虑将全局约束按照单个任务预期完成时间的比例进行划分,具体步骤如下:

(1) 获取覆盖用户 k 的服务器列表,用 $S_k = \{s_1, s_2, \dots, s_n\}$ 表示。

(2) 对于任务 t_i , 其在服务器 s_j 上的完成时间用 PMF 表示。预计完成时间 $e_{i,j}$ 表示为 $\{e_{i,j} \mid \int_0^{e_{i,j}} PMF(X) = 0.5\}$ 。

(3) 任务 t_i 可以被卸载到多个候选服务器 S_k 上执行,任务 t_i 的预期完成时间为 $E_i^s = avg(e_i)$, 其中 $e_i = \{e_{i,1}, e_{i,2}, \dots, e_{i,n}\}$ 。

(4) 对于由任务 $T_g = \{t_1, t_2, \dots, t_n\}$ 组成的任意 p_g , 其预期完成时间为 $E_g^p = \max(E_{t_1}^s, E_{t_2}^s, \dots, E_{t_n}^s)$ 。

(5) 最终,可以将全局约束分解为:

$$C_i^s = C^s \cdot \frac{E_i^p}{\sum_{j=1}^n E_j^p} \quad (10)$$

4.2 DQN 算法

Q-learning 是一种基于时序差分学习的强化学习算法,其基本思想是根据当前环境状态执行 *episode*-贪心策略,使得执行动作得到新的状态奖励函数值,进而利用动态规划思想,通过后继状态来更新 Q 函数值。当强化学习场景中的状态维度很大时,一般的 Q 学习很难完成这样复杂的任务。因此, DQN 算法^[43]被提出,其算法框架如图 2 所示。DQN 算法是一种将 Q-learning 算法和深度神经网络有效结合起来的强化学习算法,同样是基于时序差分学习的思路,并利用深度

神经网络来建立状态动作与奖励,即 Q 函数值间的映射函数关系。一般而言, Q-network 第一层网络的输入节点代表的是状态值向量,中间层节点代表神经网络的隐藏层,输出节点代表对应的 Q 值在训练 DQN 时,通过执行动作与环境交互得到的奖励函数 Q 值,将其作为标签并存储在经验池中;然后从经验池中批量采样并更新网络参数。如此循环,直到训练出一个较好的值函数网络。

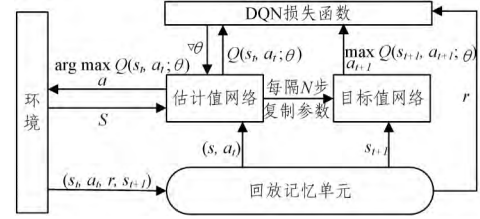


图 2 DQN 算法架构

Fig. 2 DQN algorithm architecture

4.3 基于 DQN 的工作流卸载方法

本文使用 DQN 求解多工作流卸载问题。首先,将更新的值函数定义为:

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[R(a) + \gamma \max_{a' \in A} Q(s', a')] \quad (11)$$

其中, $Q(s, a)$ 是当前的状态动作值函数, $Q(s', a')$ 是下一状态的动作值函数, α 是更新步长, $R(a)$ 是奖励函数, γ 是奖励衰减因子。DQN 损失函数的计算公式如下:

$$L(\theta) = E_{s,a,r,s'} [(Q^*(s, a | \theta) - y)^2] \quad (12)$$

$$y = R(a) + \delta \max_{a' \in A} Q^*(s', a') \quad (13)$$

其中, y 表示目标 Q 网络, Q^* 定期替换其参数。DQN 的执行过程如算法 1 所示。

算法 1 深度 Q 学习算法

初始化经验池存储空间 D, 其容量为 N

用随机权重初始化动作值函数 Q

for episode=1, M do

 初始化序列 $s_1 = \{x_1\}$ 和预处理序列 $\phi = \phi(s_1)$;

 for t=1, T do

 用概率 epsilon 选择一个随机动作 a_t , 否则选择 $a_t = \max_a Q^*(\phi(s_t, a; \theta))$;

 在模拟器中执行动作 a_t , 观察奖励 r_t 和状态 s_{t+1} ;

 将样本 $(\phi_t, a_t, r_t, \phi_{t+1})$ 存入经验池 D 中;

 从 D $(\phi_t, a_t, r_t, \phi_{t+1})$ 中随机均匀抽取样本;

 令

$$y_t = \begin{cases} r_t, & \text{terminal } \phi_{j+1} \\ r_t + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta), & \text{non-terminal } \phi_{j+1} \end{cases}$$

 在 $(y_t - Q(\phi_t, a_t; \theta))^2$ 上执行梯度下降进行更新。

 end for

end for

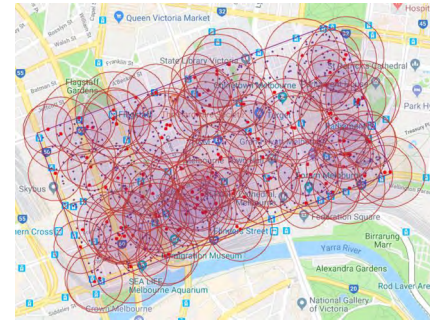
DQN 环境由环境观测、行动空间、政策制定、奖励机制^[44]等组成,需要注意的是,前 3 个组件可以使用标准的 DQN 设置来实现,而奖励设计是基于上述章节给出的优化公式和约束分解配置来开发的。奖励函数设计如下:

$$R_i(a) = Pr(X \leq C_i^s)^3 \quad (14)$$

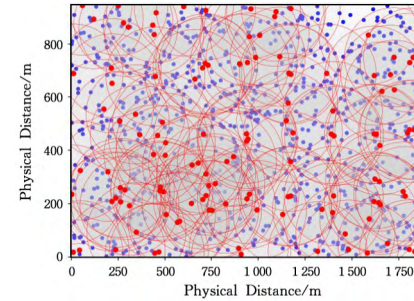
根据上述模型设计,本文算法的时间复杂度分析如下: DQN 中任意层的神经元数等于 N (输入层和输出层的神经元数不能大于 N , 隐含层包含 N 个神经元), 同时前向传播步骤的复杂度为 $O((D+2)N^3)$ 或 $O(DN^3)$, 其中 D 为隐含层数, N^3 是矩阵乘法运算的复杂度。误差的反向传播复杂度为 $O(DN^3)$, 权值的更新复杂度为 $O(DN^2)$, 因此反向传播的复杂度为 $O(DN^3)$ 。在每次迭代中, 前向传播的样本量大小 L 是反向传播的两倍。因此, DQN 算法的时间复杂度为 $O(DN^3 + 2LDN^3)$, 考虑到 D 和 L 是常量, 并且值较小, 忽略常量, 可将其表示为 $O(N^3)$ 。假设 DQN 算法的迭代次数为 K , 则时间复杂度为 $O(KN^3)$ 。假设所有工作流的总任务规模为 $\sum_{i=1}^n \sum_{j=1}^k u_{i,j}$, 用 T 表示, 在预处理阶段根据工作流任务的优先级约束, 动态调整所有任务的调度顺序, 将其存储在任务调度数组 scheduling 中。计算单个任务与边缘服务器之间的距离约束关系, 并将其存储在矩阵 H 中, 其时间复杂度为 $O(Tm)$ 。此时, 算法复杂度为 $O(Tm + KN^3)$ 。执行 DQN 算法时, 从数组 scheduling 中选取一个任务, 相应地在 H 中找到该任务可达的边缘服务器作为状态数组输入。每个任务都需要执行 DQN 算法, 最终得到较优的工作流任务调度策略。因此, 本文算法最终的时间复杂度为 $O(Tm + KTN^3)$, 由于 Tm 是低阶, 可忽略表示为 $O(KTN^3)$ 。

5 案例研究

为了评估本文方法, 通过仿真案例研究从工作流完成时间、损失函数值和算法收敛速度方面进行验证。实验随机生成服务器类型、工作流和任务类型, 并根据文献[12]的数据集来确定边缘服务器和用户的物理位置, 其部署情况如图 3 所示。



(a) 谷歌地图上的数据集



(b) 映射到坐标上的数据集

图 3 墨尔本 BCD 区的边缘服务器和边缘用户

Fig. 3 Edge servers and edge users in Melbourne BCD

实验使用多个著名的工作流模板^[45], 即 CyberShake (地震数据分析科学工作流)、Epigenomics (高通量基因测序数据分析工作流)、Inspiral (引力波测量数据分析工作流)、Montage (天文图像数据分析处理工作流) 和 Sipht (细菌 RNA 数据分析工作流)。它们的工作流结构如图 4 所示。

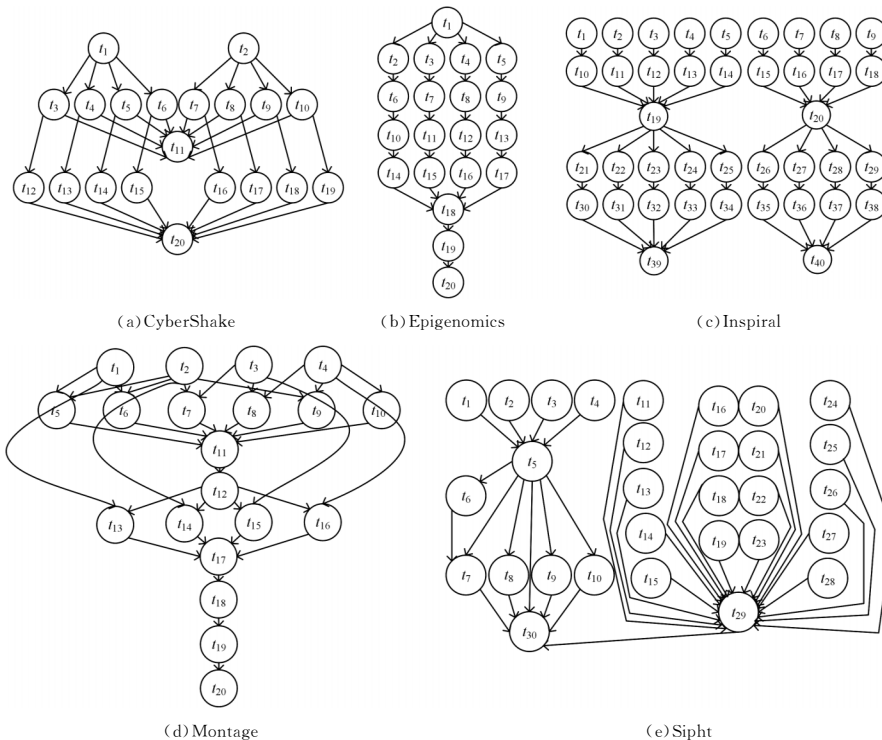


图 4 5 个典型的工作流模板

Fig. 4 Five typical workflow templates

为了模拟边缘服务器执行高计算复杂性任务的场景, 每个工作流涉及的任务对应不同精度 (800 万位、1 600 万位或

3 200 万位)的圆周率科学计算(采用标准的“高斯-勒让德”圆周率计算)程序包实现^[46],如表 1—表 5 所列。

本文实验假设边缘服务器有 3 种类型,即 type1, type2 和 type3,它们有不同的资源配置、性能和收费标准。实验分别在高负载、中等负载、低负载的 3 个不同时间段(测试时间为

2020 年 7 月 15 日早上 10 点、早上 7 点 30 以及凌晨 3 点)测量了在第三方的商业云服务(腾讯云^[47]、阿里云^[48]、华为云^[49])上运行上述圆周率计算所花费的历史时间记录^[13],并以这些数据作为 3 种类型云服务器的历史性能分布数据,如图 5 所示。

表 1 CyberShake 工作流的各个任务对应的圆周率科学计算精度要求

Table 1 Scientific calculation accuracy requirement of pi corresponding to each task of CyberShake workflows

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
800万	1 600万	1 600万	3 200万	1 600万	3 200万	3 200万	1 600万	1 600万	800万	800万	1 600万	1 600万	1 600万	3 200万	1 600万	1 600万	3 200万	3 200万	1 600万

表 2 Epigenomics 工作流的各个任务对应的圆周率科学计算精度要求

Table 2 Scientific calculation accuracy requirement of pi corresponding to each task of Epigenomics workflows

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
1 600万	3 200万	1 600万	3 200万	1 600万	1 600万	3 200万	800万	800万	800万	1 600万	3 200万	3 200万	800万	1 600万	1 600万	1 600万	1 600万	1 600万	1 600万

表 3 Inspiral 工作流的各个任务对应的圆周率科学计算精度要求

Table 3 Scientific calculation accuracy requirement of pi corresponding to each task of Inspiral workflows

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
1 600万	1 600万	1 600万	1 600万	800万	800万	3 200万	800万	800万	1 600万	3 200万	1 600万	800万	1 600万	1 600万	3 200万	800万	3 200万	1 600万	1 600万
t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	t_{30}	t_{31}	t_{32}	t_{33}	t_{34}	t_{35}	t_{36}	t_{37}	t_{38}	t_{39}	t_{40}
800万	1 600万	1 600万	800万	1 600万	800万	1 600万	1 600万	1 600万	1 600万	1 600万	1 600万	3 200万	3 200万	3 200万	1 600万	1 600万	800万	3 200万	3 200万

表 4 Montage 工作流的各个任务对应的圆周率科学计算精度要求

Table 4 Scientific calculation accuracy requirement of pi corresponding to each task of Montage workflows

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	t_{20}
3 200万	1 600万	3 200万	1 600万	1 600万	800万	1 600万	3 200万	1 600万	800万	1 600万	800万	1 600万	1 600万	1 600万	1 600万	3 200万	1 600万	3 200万	1 600万

表 5 Sipt 工作流的各个任务对应的圆周率科学计算精度要求

Table 5 Scientific calculation accuracy requirement of pi corresponding to each task of Sipt workflows

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
1 600万	3 200万	1 600万	3 200万	1 600万	800万	800万	1 600万	1 600万	800万	1 600万	1 600万	1 600万	1 600万	800万
t_{16}	t_{17}	t_{18}	t_{19}	t_{20}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}	t_{26}	t_{27}	t_{28}	t_{29}	
3 200万	1 600万	3 200万	800万	1 600万	800万	1 600万	1 600万	1 600万	1 600万	1 600万	1 600万	1 600万	1 600万	

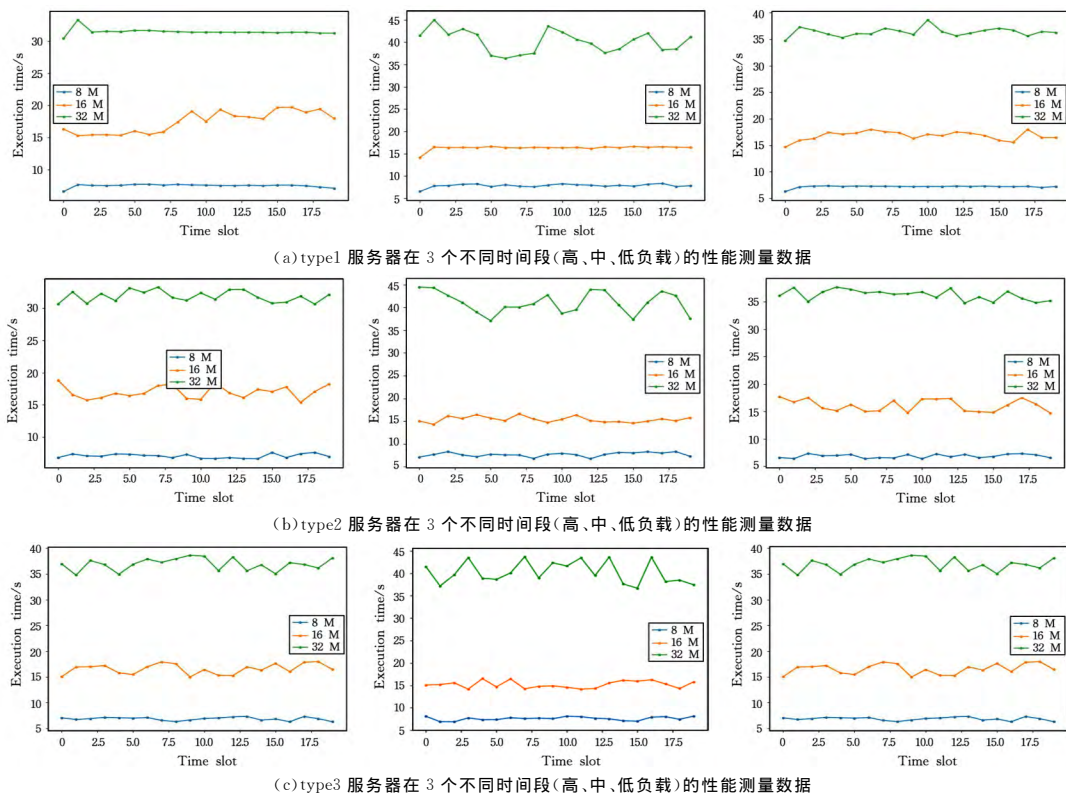


图 5 边缘服务器不同时段的性能记录

Fig. 5 Performance records of edge servers in different time periods

本文将所提方法与其他现有的经典方法,如 NSPSO^[50]和 NSGA^[51],以及基于“备选云服务性能恒定不变”假设的深度强化学习多 workflow 调度算法 DQN-based MARL^[41]进行了比较。

5.1 实验配置

本文使用 Intel Core i7 CPU@2.80 GHz, NVIDIA GeForce GTX 1050 Ti 和 8GB RAM 配置的工作站来测试本文算法和其他方法。表 6 列出了实验的基本参数。

表 6 实验中使用的参数

Table 6 Parameters used in the experiment

参数	值	含义
ϵ	0.3	随机动作的概率
γ	0.9	奖励折扣因子
lr	0.001	梯度下降算法的学习率
Min_e	0.05	ϵ 的最小值
L	512	每一步的样本量
$memory_size$	10000	样本大小
$\epsilon_decrement$	0.00001	ϵ 每次减少的值
$replace_target_iter$	500	网络参数更新周期

5.2 性能评估

基于上述配置和数据集,本文重复调用所提方法以得到 workflow 的最优卸载。本文算法的收敛过程如图 6 所示。

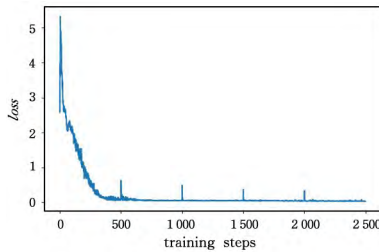


图 6 算法的收敛过程

Fig. 6 Convergence process of proposed algorithm

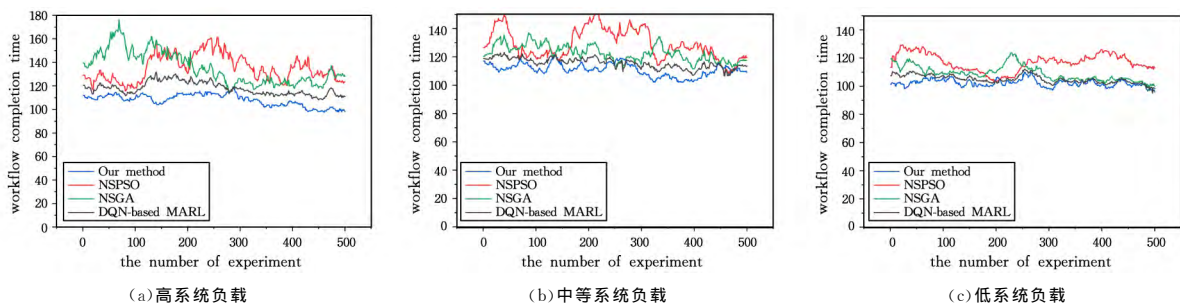


图 8 在 3 个不同时间段工作流的完成时间

Fig. 8 Workflow completion time at 3 different time periods

结束语 本文提出了一种新颖的基于深度强化学习与概率性能感知的边缘计算环境多 workflow 卸载方法。该方法考虑了边缘环境中的边缘服务器具有地理位置分布和性能波动的特性,并利用历史性能数据的概率分析驱动 DQN 模型,最终迭代寻优生成多 workflow 的卸载策略。在实验验证环节,本文基于公开的边缘服务器位置数据集、性能测试数据集和多个科学 workflow 模板开展了模拟实验。实验结果表明,相比多种传统的方法(包括基于“备选云服务性能恒定不变”假设的深度强化学习多 workflow 调度算法),本文方法在多个场景下均取得了更短的多 workflow 完成时间。

可以看出,本文算法经过 500 次迭代后, DQN 已经逐渐收敛到最优解。而从图 7 可以看出,满足全局约束的概率随着迭代次数的增加而增加,当迭代次数为 500 时,概率值趋于稳定。

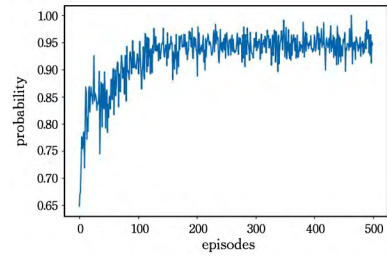


图 7 满足全局约束的概率值

Fig. 7 Probability value satisfying global constraint

实验在不同时间段分别执行本文算法 500 次,实验对比结果如图 8 所示。图 8(a) — 图 8(c) 分别对应高、中、低负载时段。由图 8(a) 可以看出,该算法的工作流完成时间明显短于其他基准算法,这是由于在该时间段内服务器性能波动较大,而本文算法使用概率模型来表示 QoS 而不是把它看作一个常量值,因此在服务器性能波动较大时效果更佳。图 8(c) 显示了低系统负载时服务器性能波动较小的情况,但是本文算法依旧优于其他对比算法,并且从实验结果中可以看出,本文算法的性能表现相比对比算法更加稳定。值得注意的是,即使是相比基于“备选云服务性能恒定不变”假设的深度强化学习多 workflow 调度算法,本文方法也体现出了较为明显的优势。其原因是,我们考虑了备选服务的性能波动,并用概率分析驱动深度强化模型的寻优过程,以充分地体现性能的时变趋势,使调度方案能与备选服务性能变化的趋势相适应。而对比方法则是采用最近一次取得的单点性能值作为 DQN 模型训练过程的输入。

在今后的扩展研究中,我们计划进一步开展以下工作:

- 1) 考虑用户和边缘节点的移动性,研究在非规则移动轨迹下多用户的任务在线卸载的策略;
- 2) 引入移动轨迹预测机制,实现多移动用户的任务卸载决策的模型;
- 3) 考虑在非可信通信条件下的任务失效和传输故障对任务卸载的影响,设计故障容忍和容错的多用户任务卸载的策略和算法等。

参考文献

- [1] CHEN X, LIU Z, CHEN Y, et al. Mobile edge computing based task offloading and resource allocation in 5g ultra-dense net-

- works[J]. IEEE Access, 2019, 7: 184172-184182.
- [2] LYU X, NI W, TIAN H L, et al. Optimal schedule of mobile edge computing for internet of things using partial information [J]. IEEE Journal on Selected Areas in Communications, 2017, 35(11): 2606-2615.
 - [3] ZHANG Y, DU P. Delay-driven computation task scheduling in multi-cell cellular edge computing systems [J]. IEEE Access, 2019, 7: 149156-149167.
 - [4] CAO H, XU X, LIU Q, et al. Uncertainty-aware resource provisioning for workflow scheduling in edge computing environment [C] // 18th IEEE International Conference on Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference on Big Data Science And Engineering, TrustCom/BigDataSE, Rotorua, New Zealand, 2019: 734-739.
 - [5] DENG Y, CHEN, YAO X, et al. Task scheduling for smart city applications based on multi-server mobile edge computing [J]. IEEE Access, 2019, 7: 14410-14421.
 - [6] CHEN J F, CHEN J W, JING P, et al. An improved chaotic bat swarm scheduling learning model on edge computing [J]. IEEE Access, 2019: 58602-58610.
 - [7] MA Y, ZHANG J, WANG S, et al. A Novel Approach to Cost-Efficient Scheduling of Multi-workflows in the Edge Computing Environment with the Proximity Constraint [C] // International Conference on Algorithms and Architectures for Parallel Processing. Springer, Cham, 2019: 655-668.
 - [8] PENG Q, JIANG H, CHEN M, et al. Reliability-aware and Deadline-constrained workflow scheduling in Mobile Edge Computing [C] // 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC). IEEE, 2019: 236-241.
 - [9] BERNAL J, KUSHIBAR K, ASFAW D S, et al. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review [J]. Artificial intelligence in medicine, 2019, 95: 64-81.
 - [10] BOUWMANS T, JAVED S, SULTANA M, et al. Deep neural network concepts for background subtraction: A systematic review and comparative evaluation [J]. Neural Networks, 2019, 117: 8-66.
 - [11] GREKOUSIS G. Artificial neural networks and deep learning in urban geography: A systematic review and meta-analysis [J]. Computers, Environment and Urban Systems, 2019, 74: 244-256.
 - [12] LAI P, HE Q, ABDELRAZEK M, et al. Optimal edge user allocation in edge computing with variable sized vector bin packing [C] // International Conference on Service-Oriented Computing. Springer, Cham, 2018: 230-245.
 - [13] LI W, XIA Y, ZHOU M, et al. Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds [J]. IEEE Access, 2018, 6: 61488-61502.
 - [14] KAUR M, KADAM S. A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling [J]. Applied Soft Computing, 2018, 66: 183-195.
 - [15] ZHANG L, LI K, LI C, et al. Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems [J]. Information Sciences, 2017, 379: 241-256.
 - [16] CASAS I, TAHERI J, RANJAN R, et al. GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments [J]. Journal of computational science, 2018, 26: 318-331.
 - [17] VERMA A, KAUSHAL S. A hybrid multi-objective particle swarm optimization for scientific workflow scheduling [J]. Parallel Computing, 2017, 62: 1-19.
 - [18] ZHOU X, ZHANG G, SUN J, et al. Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT [J]. Future Generation Computer Systems, 2019, 93: 278-289.
 - [19] HABAK K, AMMAR M, HARRAS K A, et al. Femto clouds: Leveraging mobile devices to provide cloud service at the edge [C] // 2015 IEEE 8th international conference on cloud computing. IEEE, 2015: 9-16.
 - [20] MAO Y, ZHANG J, SONG S H, et al. Stochastic Joint Radio and Computational Resource Management for Multi-User Mobile-Edge Computing Systems [J]. IEEE transactions on wireless communications, 2017, 16(9): 5994-6009.
 - [21] TONG L, LI Y, GAO W. A hierarchical edge cloud architecture for mobile computing [C] // IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, 2016: 1-9.
 - [22] ZHANG Y, CHEN X, CHEN Y, et al. Cost efficient scheduling for delay-sensitive tasks in edge computing system [C] // 2018 IEEE International Conference on Services Computing (SCC). IEEE, 2018: 73-80.
 - [23] ZHAO T, ZHOU S, GUO X, et al. Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing [C] // 2017 IEEE international conference on communications (ICC). IEEE, 2017: 1-7.
 - [24] HWANG S Y, HSU C C, LEE C H. Service selection for web services with probabilistic QoS [J]. IEEE transactions on services computing, 2014, 8(3): 467-480.
 - [25] PAN Y, WANG S, WU L, et al. A Novel Approach to Scheduling Workflows Upon Cloud Resources with Fluctuating Performance [J]. Mobile Networks and Applications, 2020: 1-11.
 - [26] LI W, XIA Y, ZHOU M, et al. Fluctuation-aware and predictive workflow scheduling in cost-effective infrastructure-as-a-service clouds [J]. IEEE Access, 2018, 6: 61488-61502.
 - [27] BERTSEKAS D P. Feature-based aggregation and deep reinforcement learning: A survey and some new implementations [J]. IEEE/CAA Journal of Automatica Sinica, 2018, 6(1): 1-31.
 - [28] MAO H, ALIZADEH M, MENACHE I, et al. Resource management with deep reinforcement learning [C] // Proceedings of the 15th ACM Workshop on Hot Topics in Networks. 2016: 50-56.
 - [29] XUE L, SUN C, WUNSCH D, et al. An adaptive strategy via reinforcement learning for the prisoner's dilemma game [J]. IEEE/CAA Journal of Automatica Sinica, 2017, 5(1): 301-310.
 - [30] ZHAN Y, AMMAR H B. Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer [J]. arXiv:1604.03986, 2016.
 - [31] WANG H, HUANG T, LIAO X, et al. Reinforcement learning for constrained energy trading games with incomplete information [J]. IEEE transactions on cybernetics, 2016, 47(10): 3404-3416.

- [32] ZHENG L, YANG J, CAI H, et al. Magent: A many-agent reinforcement learning platform for artificial collective intelligence [J]. arXiv:1712.00600, 2017.
- [33] LOWE R, WU Y I, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[C]// Advances in Neural Information Processing Systems. 2017:6379-6390.
- [34] DUAN R, PRODAN R, LI X. Multi-objective game theoretic scheduling of bag-of-tasks workflows on hybrid clouds[J]. IEEE Transactions on Cloud Computing, 2014, 2(1):29-42.
- [35] CUI D, KE W, PENG Z, et al. Multiple DAGs workflow scheduling algorithm based on reinforcement learning in cloud computing[C]// International Symposium on Computational Intelligence and Intelligent Systems. Springer, Singapore, 2015:305-311.
- [36] IRANPOUR E, SHARIFIAN S. A distributed load balancing and admission control algorithm based on Fuzzy type-2 and Game theory for large-scale SaaS cloud architectures[J]. Future Generation Computer Systems, 2018, 86:81-98.
- [37] WU J H, PENG Z P, CUI D L, et al. A multi-object optimization cloud workflow scheduling algorithm based on reinforcement learning[C]// International Conference on Intelligent Computing. Springer, Cham, 2018:550-559.
- [38] DONG T, XUE F, XIAO C, et al. Task scheduling based on deep reinforcement learning in a cloud manufacturing environment [J]. Concurrency and Computation: Practice and Experience, 2020, 32(11):e5654.
- [39] PENG Z, CUI D, ZUO J, et al. Random task scheduling scheme based on reinforcement learning in cloud computing[J]. Cluster computing, 2015, 18(4):1595-1607.
- [40] CHENG M, LI J, NAZARIAN S. DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers[C]// 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2018:129-134.
- [41] 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.
- [42] GUO S, LIU J, YANG Y, et al. Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing[J]. IEEE Transactions on Mobile Computing, 2018, 18(2):319-333.
- [43] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540):529-533.
- [44] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. arXiv:1312.5602, 2013.
- [45] BHARATHI S, CHERVENAK A, DEELMAN E, et al. Characterization of scientific workflows[C]// 2008 third workshop on workflows in support of large-scale science. IEEE, 2008:1-10.
- [46] ZHANG G M. Gauss-legendre Multi-Repetition Integral formula and its application[J]. Journal of Lanzhou University, 2000(5):30-34.
- [47] Tencent Cloud. Tencent Cloud gradually opens up BAT comprehensively competitive cloud platform[DB/OL]. [2013-09-10]. <http://www.qcloud.com/>.
- [48] Aliyun official website. Aliyun official website[DB/OL]. 2016-9-12. [2016-9-12]. <https://www.aliyun.com/minisite/goods?userCode=om2mzele>
- [49] Qi Chacha-Huawei Cloud. Qi Chacha[DB/OL]. [2020-07-1]. <https://www.qcc.com/product/c893fb05-edc2-4f3b-9f2d-ed4e0c60-bbcd.html>.
- [50] AJEENA BEEGOM A S, RAJASREE M S. Non-dominated sorting based PSO algorithm for workflow task scheduling in cloud computing systems[J]. Journal of Intelligent & Fuzzy Systems, 2019, 37(5):6801-6813.
- [51] MOLLAJAFARI M, SHAHHOSEINI H S. Cost-Optimized GA-Based Heuristic for Scheduling Time-Constrained Workflow Applications in Infrastructure Clouds Using an Innovative Feasibility-Assured Decoding Mechanism [J]. Journal Information Science and Engineering, 2016, 32(6):1541-1560.



MA Yu-yin, born in 1995, postgraduate, is a member of China Computer Federation. Her main research interests include edge computing, service computing and workflows scheduling.



ZHENG Wan-bo, born in 1981, Ph.D., associate researcher. His main research interests include big data intersects with cloud computing, data analysis and mining, mine tunnel emergency management, research on mine tunnel emergency rescue,

disposal information technology and equipment, engineering geophysical exploration, as well as measurement & control technology and instruments.