计算机科学
COMPUTER SCIENCE

基于深度强化学习的数据中心热感知能耗优化方法

李丹阳, 吴良基, 刘慧, 姜静清

引用本文

李丹阳, 吴良基, 刘慧, 姜静清. 基于深度强化学习的数据中心热感知能耗优化方法[J]. 计算机科学, 2024, 51(6A): 230500109-8.

LI Danyang, WU Liangji, LIU Hui, JIANG Jingqing. Deep Reinforcement Learning Based Thermal Awareness Energy Consumption OptimizationMethod for Data Centers [J]. Computer Science, 2024, 51(6A): 230500109-8.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230500109

基于深度强化学习的数据中心热感知能耗优化方法

李丹阳[1]　吴良基[1]　刘　慧[2]　姜静清[3]
1 东北大学软件学院　沈阳 110169
2 东北大学冶金学院　沈阳 112000
3 内蒙古民族大学计算机科学与技术学院　内蒙古 通辽 028000
(2110497@stu.neu.edu.cn)

摘　要　随着数据中心规模的不断扩大,所引起的高能耗、高运营成本和环境污染等问题日益严重,严重影响了数据中心的可持续性。大多数数据中心能耗优化方法为了降低计算能耗,会将任务集中在尽可能少的服务器上,但这样做往往会导致数据中心热点的产生,并且提高了冷却能耗。为了解决这一问题,文中首先对数据中心进行建模,并将数据中心总能耗优化问题建模为一个任务调度问题,并且要求调度过程中不产生数据中心热点。为了解决该问题,文中提出了一种基于深度强化学习的数据中心任务调度方法,并使用奖励塑造对该方法进行优化,在不产生热点的前提下降低数据中心的总能耗。最后,通过仿真环境和真实数据中心负载跟踪数据进行了实验。仿真实验结果表明,所提方法相比其他现有的调度方法能够更好地降低数据中心总能耗,最多降低了25.5%。此外,提出的优化方法还不会产生热点,这进一步证明了其优越性。

关键词:数据中心;能耗优化;热点;任务调度;深度强化学习;奖励塑造
中图分类号　TP181

Deep Reinforcement Learning Based Thermal Awareness Energy Consumption Optimization Method for Data Centers

LI Danyang[1],WU Liangji[1],LIU Hui[2] and JIANG Jingqing[3]
1 Software College,Northeastern University,Shenyang 110169,China
2 School of Metallurgy,Northeastern University,Shenyang 112000,China
3 College of Computer Science and Technology,Inner Mongolia Minzu University,Tongliao,Inner Mongolia 028000,China

Abstract　With the continuous expansion of the scale of data centers,the problems of high energy consumption,high operating costs and environmental pollution are becoming more and more serious,which seriously affect the sustainability of data centers. Most data center energy consumption optimization methods focus tasks on as few servers as possible,so as to reduce computing energy consumption. However,this often leads to the generation of data center hotspots and increases cooling energy consumption. In order to solve this problem,this paper first models the data center,and models the total energy consumption optimization problem of the data center as a task scheduling problem,and requires that no data center hotspots are generated. This paper proposes a task scheduling method based on deep reinforcement learning for data centers,and uses reward shaping to optimize the method to reduce the total energy consumption of data centers without generating hotspots. Finally,experiments are carried out through simulation environment and real data center load trace data. The simulation results show that the proposed method can reduce the total energy consumption of the data center better than other existing scheduling methods,and can reduce the total energy consumption by up to 25.5%. In addition,the proposed optimization method does not generate hot spots yet,which further proves its superiority.

Keywords　Data Center,Energy consumption optimization,Hot spot,Task scheduling,Deep reinforcement learning,Reward shaping

1　引言

随着云计算和大数据产业的快速发展,数据中心已成为支撑现代信息社会运行的重要基础设施[1]。然而,数据中心在运行过程中会消耗大量的电能,该能耗已经占据全球总能耗的2%。更为严重的是,该比例还将继续上升[2]。因此,数据中心的可持续性计算技术备受关注。据亚马逊估计,数据中心的能源开支约占数据中心总成本的42%以上[3]。高能耗问题不仅导致了高成本问题,还导致了碳排放过高[4]、计算性能下降[5]和系统可靠性降低[6]等问题。因此,如何有效降

基金项目:国家自然科学基金(62162050)
This work was supported by the National Natural Science Foundation of China(62162050).
通信作者:吴良基(2171357@stu.neu.edu.cn)

低数据中心的能耗成为了一个亟待解决的问题。

数据中心的能耗主要来自服务器和冷却设施。一方面，数据中心由数千台高性能服务器组成，服务器在运行时需要大量电能来进行计算，即计算能耗。另一方面，为了防止服务器在计算过程中过热，需要冷却设施来移除服务器产生的热量[6]，即冷却能耗。因此，在解决数据中心的能耗优化问题时，应该综合考虑计算能耗和冷却能耗，以实现整体能耗优化，提高数据中心的可持续性。

现有的数据中心能耗优化方法忽视了热量管理问题。大多数的能耗感知优化方法为了降低计算能耗，会将任务集中在尽可能少的服务器上，而其余的空闲服务器会被关闭[7]或者进入低功耗状态[8]。然而，这种能耗优化方法存在一个问题，即过于集中的负载分布会导致数据中心产生热点。热点会对数据中心的可持续性产生负面影响。首先，热点会影响数据中心的总体能耗。为了消除热点，数据中心会提高冷却能耗，从而提高数据中心的总体能耗。其次，热点会影响数据中心的可靠性，导致服务器温度过高而产生故障，进而影响数据中心的可持续性。因此，在优化数据中心总能耗的同时，也应该关注数据中心的热量管理，避免数据中心产生热点。

深度强化学习作为一种强大的机器学习方法，在电子游戏[9]、自动驾驶[10]和机器人[11]等领域都取得了巨大的成功。深度强化学习模型的智能体可以通过与环境的交互来获得奖励信号，以学习最优的行为策略。深度强化学习能够适应动态变化的环境，并能够依据环境给出一个决策结果。在数据中心任务调度问题中，数据中心环境和任务负载是动态变化的，调度方法需要能够适应这种动态性。因此，深度强化学习非常适合用于解决数据中心任务调度问题。

针对以上问题，本文提出了一种基于深度强化学习的任务调度方法，旨在优化数据中心的总能耗，并避免在调度过程中产生热点。由于在真实数据中心中进行训练非常昂贵且耗时，大部分深度强化学习研究[12]都采用仿真环境对模型进行预训练。因此，本文的优化方法也将在仿真环境下进行训练和实验。实验结果表明，本文方法能够有效地降低数据中心的能耗，同时避免了热点的产生。

本文的主要贡献包括：

1)对数据中心的总能耗和热力学环境进行建模，提出了数据中心总能耗优化问题，在不产生热点的前提下优化数据中心总能耗。

2)提出了一种基于深度强化学习的任务调度方法，将任务分配到合适的服务器。在不产生热点的前提下，对数据中心能耗进行优化，本文使用相关领域知识对本文方法进行优化，提高了深度强化学习方法的性能。

3)搭建了仿真实验环境，并且使用两种不同类型的真实数据中心的负载跟踪数据进行了仿真实验。实验结果表明，本文提出的优化方法优于其他方法，并且不会产生热点，证明了本文方法的有效性和实用性。

本文第2章简单介绍了任务调度和深度强化学习的国内外相关工作；第3章介绍了数据中心功耗和温度的相关模型；第4章对数据中心能耗问题进行了形式化描述；第5章给出了基于深度强化学习的任务调度方法；第6章进行了相关仿真实验对比和结果分析；最后总结全文。

2 相关工作

2.1 能耗感知的任务调度

目前，能耗感知的任务调度问题一般被看作是一个非线性优化问题，因此许多学者采用元启发式算法来解决该问题。常见的元启发式算法包括遗传算法[13]、模拟退火算法[14]和引力搜索算法[15]等。文献[13]基于遗传算法，提出了热感知任务调度方法 MITEC-GA，该方法通过将数据中心总能耗作为适应度函数来优化数据中心总能耗。文献[14]提出了基于模拟退火算法的大数据任务调度方法，该方法能有效地降低数据中心总能耗、计算能耗和冷却能耗。除了计算能耗和冷却能耗之外，该文献还额外考虑了虚拟机在迁移过程中的能耗。文献[15]基于二进制引力搜索算法，提出了一种任务调度方法 FC-BGSA，该方法能最小化数据中心的总能耗。此外，该文献还通过一种基于模糊逻辑的自适应控制算法来提高方法的收敛速度。文献[16]提出了两种基于元启发式算法的调度方法 XInt-GA 和 XInt-SQP，用于解决数据中心入口温度峰值问题，以最小化冷却成本。文献[6]提出了一种在线调度方法 ETAS，该方法能够动态整合虚拟机，将过载和负载不足的虚拟机迁移到合适的主机上。

2.2 深度强化学习在任务调度领域的应用

强化学习是一种能够通过与环境的交互来不断优化自身行为的机器学习方法。文献[17]提出了一种基于 DQN 算法的任务调度方法，将具有优先级关系的任务调度到云服务器上，以最小化任务执行时间。文献[18]提出了一种基于深度强化学习的任务调度方法 DeepJS，该方法将任务调度问题视为一个装箱问题，从过往经验中自动获得一个适应度计算方法，从而最小化最大完工时间。文献[19]提出了基于 DQN 算法的任务调度方法 DRL-Cloud。相比其他文献，该文献考虑了数据中心的伸缩性问题和任务之间的依赖关系，减低了运行时间、能源成本。文献[20]提出了 Decima 框架，用于在分布式计算集群上调度任务。Decima 框架考虑了作业间的依赖关系，并且能用于处理连续随机到达的作业。与手工调优的调度启发式相比，Decima 将平均任务完成时间提高了至少 21%。文献[21]提出了一种基于决策树和改进 Q 学习的虚拟机整合算法，该算法使用决策数来实现状态表征。实验结果表明，该虚拟机整合算法在能耗、资源利用率等各方面均有提升。文献[22]提出了一种基于 Q 学习的虚拟机放置策略，并从状态聚合和时间可信度两个方面对算法进行优化。实验结果表明，相比贪心算法、粒子群算法，该放置策略能够显著降低数据中心能耗。

相比任务调度领域的其他深度强化学习方法，本文的深度强化学习算法在热量管理和能耗管理两个方面进行了综合考虑。与其他文献相比，尽管一些文献涉及能耗管理，但它们仅仅关注计算能耗，而未涉及冷却能耗和热量管理。

3 系统模型

典型的数据中心布局如图1所示，它由多台服务器和冷却系统组成。该数据中心共有 N 台服务器为用户提供服务，它们被放置在各个机架上。冷却系统则由多个计算机机房空调(Computer Room Air Condition，CRAC)单元组成，每个

CRAC 单元负责冷却数据中心的一个特定区域。

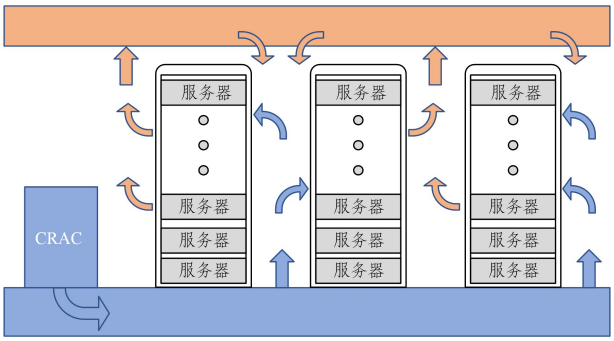


图 1　数据中心布局

Fig. 1　Data center layout

下面将讨论数据中心建模的关键部分：任务模型、服务器功耗模型、冷却模型、温度模型。

3.1　任务模型

在数据中心的运行过程中，每一刻都可能会有零个或多个等待分配的任务到来。这些任务是相互独立的，不会对其他任务产生影响。每个任务都有自己的资源需求，如 CPU、内存、网络等。由于服务器其他组件的功耗和热量相对较低，因此本文仅考虑任务的 CPU 需求。为了表示每个任务的资源需求，本文使用一个元组{cpu，dur}来描述每个任务的资源需求，其中 cpu 代表任务所需的 CPU 资源，dur 代表该任务在所需的 CPU 资源下需要执行的时间。在 t 时刻，数据中心到来的任务集合可以表示为：

$$J(t)=\{j_i \mid 1\leqslant i\leqslant m\} \tag{1}$$

3.2　服务器功耗模型

服务器的功耗主要由其 CPU 利用率决定，并且与 CPU 利用率呈线性关系[23]。为了降低数据中心的功耗，在服务器的 CPU 利用率为 0 时，数据中心会关闭该服务器。第 i 台服务器在 t 时刻的功耗可以表示为：

$$P_i(t)=\begin{cases}P_i^{\text{idle}}+\mu_i(t)\cdot(P_i^{\max}-P_i^{\text{idle}}), & \mu_i(t)>0\\ 0, & \mu_i(t)=0\end{cases} \tag{2}$$

其中，P_i^{idle} 和 $P_i^{\max}$ 是第 i 台服务器在空闲状态下和满载状态下的功率，而 $\mu_i(t)$ 表示第 i 台服务器在 t 时刻的 CPU 利用率。值得注意的是，在异构数据中心，不同服务器的 P_i^{idle} 和 $P_i^{\max}$ 参数可能会不同。

数据中心的计算功耗定义为所有服务器功耗的总和。因此，数据中心在 t 时刻的计算功耗可以表示为：

$$P_{\text{comp}}(t)=\sum_{i=0}^{N}P_i(t) \tag{3}$$

3.3　冷却模型

基于 CRAC 的数据中心冷却过程如图 1 所示。为了有效散热，数据中心通常采用冷热通道的方式进行冷却，其中每个 CRAC 单元负责冷却数据中心的一个特定区域。CRAC 单元将冷空气输送到冷通道，冷通道内的冷空气从机架底部通过通风瓦到达机架顶部。经过服务器后，冷空气变成热空气并流向热通道。热空气通过热通道排出，最终通过天花板收集后送回 CRAC 机组。CRAC 单元的冷却效率可以通过性能系数(Coefficient of Performance，CoP)来衡量，CoP 被定义为计算功耗和冷却功耗之间的比值[6]。因此，在给定的供冷温度和计算功耗的情况下，冷却功耗可以表示为：

$$P_{\text{cool}}(t)=\frac{P_{\text{comp}}(t)}{CoP(T_{\text{sup}})} \tag{4}$$

在不同的数据中心环境下，CoP 值会因物理布局和热力学环境的不同而不同[6]。如果数据中心的 CoP 值越大，CRAC 单元的冷却效率就越高，所消耗的冷却功耗就越低。根据 HP 实验室的拟合结果[24]，CoP 可以表示为关于供冷温度 T_{sup} 的二次函数：

$$CoP(T_{\text{sup}})=0.0068*T_{\text{sup}}^2+0.008*T_{\text{sup}}+0.458 \tag{5}$$

由式(5)可知，当供冷温度增加时，数据中心的冷却效率会提高。因此，提高供冷温度能够有效地提高 CRAC 的冷却效率。

3.4　温度模型

本文将使用热再循环模型对数据中心的热力学环境进行建模，以计算各个服务器的温度。

热再循环模型用于描述数据中心的热再循环现象。数据中心地热再循环现象[16]如图 1 所示，数据中心热通道的空气回流到冷通道，导致服务器入口温度高于供冷温度，且各服务器入口温度不同。热再循环模型使用热分布矩阵[16]来描述这一现象，热分布矩阵 $\boldsymbol{D}$ 是一个 $n*n$ 矩阵，矩阵的元素 $d_{i,k}$ 表示第 k 个服务器对第 i 个服务器入口温度的影响。通过热在循环模型，第 i 台服务器在 t 时刻的入口温度 $T_i^{\text{in}}(t)$ 可以表示为：

$$T_i^{\text{in}}(t)=T_{\text{sup}}+\sum_{k=1}^{n}d_{i,k}*P_k(t) \tag{6}$$

由式(6)可知，即使每台服务器的供冷温度是一致的，但是因为服务器的物理位置和数据中心的热再循环效应，不同服务器的入口温度也可能是不同的。

服务器温度过高会导致其运行不稳定，计算能力下降。本文将热点定义为入口温度超过红线温度的服务器，使用一个二元变量来表示一个服务器是否为热点：

$$H_i(t)=\begin{cases}1, & T_i^{\text{in}}(t)>T_{\text{red}}\\ 0, & T_i^{\text{in}}(t)\leqslant T_{\text{red}}\end{cases} \tag{7}$$

其中，如果 $H_i(t)$ 为 1，则代表第 i 台服务器为热点。反之，如果变量值为 0，则代表该服务器不是热点。

为了保证数据中心的可靠运行，数据中心不应该存在热点。在任何时刻，数据中心都应该保证：

$$\sum_{i=0}^{N}H_i(t)=0 \tag{8}$$

4　问题建模

本文的目标是在不产生热点的前提下，通过任务调度的方式最小化数据中心的总能耗。数据中心调度场景如图 2 所示，为了接收任务、执行任务调度函数和分配任务，数据中心引入了一个集中控制器。在实际的数据中心中，通常有数百台服务器用于处理任务，而仅有一台服务器充当集中控制器的角色。因此，本文不考虑集中控制器所带来的额外开销。

数据中心的总功耗可以表示为计算功耗和冷却功耗之和。因此，数据中心的总功耗可表示为：

$$P_{\text{total}}(t)=P_{\text{comp}}(t)+P_{\text{cool}}(t)=\left(1+\frac{1}{CoP(T_{\text{sup}})}\right)P_{\text{comp}}(t) \tag{9}$$

数据中心的总能耗定义为数据中心总功耗在时间上的累加。因此，数据中心总能耗可以表示为：

$$E_{\text{total}}=\sum_{t}P_{\text{total}}(t) \tag{10}$$

在任意时刻 t，数据中心到达了一个任务集 $J(t)$，调度策略应该依据数据中心当前的状态将这个任务集中所有任务分配到合适的服务器上，从而最小化数据中心的总能耗。本文对问题进行了如下的形式化描述：

$$\begin{aligned}&\text{Given: } J(1), J(2), \cdots\\&\text{minimize: } E_{\text{total}}\\&\text{s.t. } \max(\mu_i(t))\leqslant 1\\&\quad \sum_{i=0}^{N}H_i(t)=0\end{aligned} \tag{11}$$

在解决数据中心任务调度问题时，需要满足 3 个约束，分别是 CPU 容量约束、热点约束和分配约束。首先，CPU 容量约束要求在为任务分配服务器时，必须考虑每台服务器的 CPU 容量。只有当服务器拥有足够的 CPU 资源来处理该任务时，该任务才能被调度到该服务器上。其次，热点约束要求所有服务器的温度都不能超过红线温度，以避免产生热点。最后，分配约束要求任务只能被分配到一台服务器上，并且不允许被重新分配到其他服务器上，以确保任务调度的稳定性和可靠性。

考虑到本问题是一个 NP-hard 问题，并且数据中心的规模一般都较大，精确求解问题中的优化函数对于实时系统来说费时且不可行。因此，本文将提出基于深度强化学习的在线任务调度方法，该方法能够在合理的时间内找到近似最优解。

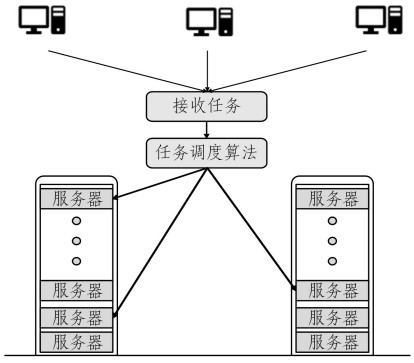


图 2 数据中心调度场景

Fig. 2 Data center scheduling scenario

5 优化方法

本文提出了一种基于深度强化学习的任务调度方法，用于解决数据中心总能耗优化问题。该方法的整体框架如图 3 所示。

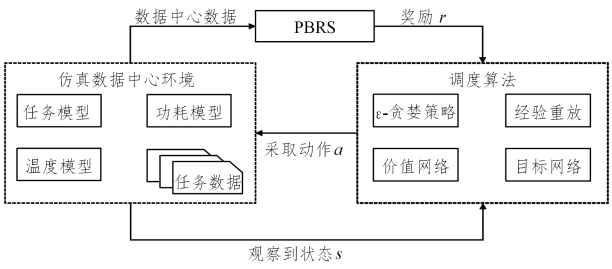


图 3 基于深度强化学习的数据中心任务调度框架

Fig. 3 Task scheduling framework for data center based on deep reinforcement learning

之前的工作大多使用静态规则或传统的启发式算法来解决任务调度问题。与传统的启发式算法相比，深度强化学习模型可以根据当前的状态和奖励信号进行动态的决策，能够适应不同的环境和数据负载，具有更好的适应性和灵活性。当数据中心的环境发生变化时，深度强化学习仍然可以较好地适用。

5.1 动作、状态和奖励

强化学习使用马尔可夫决策过程（Markov Decision Process，MDP）对优化问题进行建模，智能体在离散时间步中不断地从与环境的交互过程中学习并做出决策。本文中，MDP 可以定义为元组 $\{S, A, R, \gamma\}$，其中 S 表示状态空间，A 表示动作空间，R 表示奖励函数，γ 表示折扣因子。

5.1.1 动作

动作空间 A 表示智能体可以执行的所有可能的动作，即将任务分配到某个服务器上。在某一时刻，数据中心到来了一个任务集合等待分配，其中共有 m 个任务。在满足 CPU 容量限制和温度限制的前提下，智能体可以将这些任务分配到任意一台服务器。动作空间的大小最大为 n^m，如此大的动作空间会导致神经网络输出层的神经元过多，使学习变得极其缓慢、困难。

为了解决这个问题，可将当前时间步长划分为 m 个决策点，即将任务集合的分配过程拆分为 m 个阶段。在每个决策点上，智能体需要选择一个服务器将当前任务分配给它，每个决策点上的动作空间的大小减小到 n。这种方法可以显著减小动作空间的大小，从而使得神经网络输出层的神经元数量减少，学习更加高效、快速。决策点 t 处的动作空间为：

$$A_t=\{1, 2, \cdots, n\} \tag{12}$$

其中，当动作的值为 i 时，表示将对应任务分配到第 i 台服务器上。

5.1.2 状态

状态空间 S 表示智能体可以感知到的所有可能的状态，即数据中心当前的状态。状态空间的定义包括 3 个部分：所有服务器的功率信息、所有服务器的入口温度信息和任务信息。状态 s_t 可表示为：

$$s_t=\{\mathbf{P}, \mathbf{T}, j\} \tag{13}$$

其中，矢量 $\mathbf{P}$ 表示数据中心所有服务器的功耗，矢量 $\mathbf{T}$ 表示数据中心所有服务器的入口温度，j 表示数据中心即将要调度的任务。

5.1.3 奖励

在深度强化学习中，为了指导智能体的学习过程，需要设计一个合理的奖励信号。本文旨在设计一种优化方法，通过在给定的数据中心下分配任务，在不产生热点的前提下最小化数据中心的总能耗。为了实现这一目标，奖励函数应该包括两部分，一部分与数据中心的功耗有关，另一部分与数据中心的热点有关。通过这种方式，智能体可以逐步学习到可以在不产生热点的前提下最小化数据中心功耗的动作。奖励函数具体如下：

$$r_t=-P_{\text{total}}(t)-\omega*\sum_{i=0}^{N}H_i(t) \tag{14}$$

其中，热点部分的权重为 ω，用于平衡两个目标。为了避免数据中心产生热点，在奖励函数中应该赋予热点部分较大的权重值，即 ω 的值应当较大。

5.2 深度强化学习算法

强化学习算法可以分为基于值函数的算法和基于策略的算法,这两种类型算法都包含了很多经典算法。为了让深度强化学习算法更快更好地学习,本文使用了基于势能的奖励塑造(Potential Based Reward Shaping,PBRS)和领域知识来指导算法学习。

5.2.1 基于值函数的强化学习算法

基于值函数的强化学习算法旨在构建一个价值函数,该函数用于预测在未来的奖励期望值,以衡量每个状态或状态-动作对的好坏。

深度Q网络(Deep-Q-Network,DQN)算法[9]是一种基于值函数的深度强化学习算法。DQN算法包含两个网络:价值网络和目标网络,这两个网络具有相同的结构。价值网络用于计算当前状态下每个动作的Q值,而目标网络则用于计算下一个状态下每个动作的Q值。在每次训练中,算法通过最小化损失函数来更新价值网络的参数。DQN的损失函数定义为:

$$\mathcal{L}(\theta)=E_{s_t,a_t,r_t,s_{t+1}}[(y-Q(s_t,a_t|\theta))^2] \tag{15}$$

$$y=r_t+\gamma Q(s_{t+1},a'|\theta')$$

其中,θ和θ'分别表示价值网络和目标网络的参数。为了稳定学习,θ'会随着参数θ周期性更新。而y为目标网络输出的目标动作值函数,γ为折扣因子。基于DQN的任务调度算法的伪代码如算法1所示。

算法1 基于DQN的任务调度算法

输入:数据中心初始状态s_0,任务序列C={J(1),J(2),…}

输出:数据中心总能耗E_{total},热点个数h

1. 初始化:随机初始化网络参数,经验重放池R←∅,数据中心总能耗E_{total}←0,热点个数h←0;
2. for t←1 to T do
3. 获取当前时刻的任务集合J(t);
4. for each j∈J(t):
5. 获取当前数据中心的状态s;
6. 使用ε-贪婪策略挑选动作,以概率ε随机选取一个动作a,否则令a=arg $\max_{a'}$(Q(s,a'));
7. 将任务调度到第a台服务器上;
8. 数据中心状态变为s′,智能体获得奖励r;
9. 智能体将(s,a,s′,r)放入经验重放池R中;
10. end for
11. 更新当前数据中心总能耗E_{total}和热点个数h;
12. end for
13. return E_{total}

5.2.2 基于策略的强化学习算法

基于策略的强化学习算法是一种直接对策略进行优化的强化学习算法。它将策略视为状态到动作的映射,策略优化的目标是找到一个最优的映射。

深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)算法[25]是一种基于策略的深度强化学习算法,旨在为智能体找到一种有效的行为策略,以最大化奖励。DDPG算法是由确定性策略梯度(Deterministic Policy Gradient,DPG)算法和DQN算法结合而来的。相比DQN算法,DDPG算法在处理连续动作空间方面具有更强的能力。

DDPG算法基于演员-评论家算法,其中有演员和评论家两类网络。演员实现连续动作值的输出,而评论家则评估动作的执行效果。具体而言,策略函数μ被称为演员,它的输出结果是从动作空间中选择的动作,用于执行动作。价值函数Q被称为评论家,它的输出为TD误差(Temporal Difference Error),用于评价演员给出的行为的优劣。基于DDPG的任务调度算法的伪代码如算法2所示。

算法2 基于DDPG的任务调度算法

输入:数据中心初始状态s_0,任务序列C={J(1),…,J(T)}

输出:数据中心总能耗E_{total},热点个数h

1. 初始化:随机初始化演员μ、批评家Q网络参数,经验重放池R←∅,数据中心总能耗E_{total}←0,热点个数h←0;
2. for t←1 to T do
3. 获取当前时刻任务集合J(t);
4. for each j∈J(t):
5. 获取当前数据中心的状态s;
6. 演员依据状态s直接给出一个动作a;
7. 将任务调度到第a台服务器上;
8. 数据中心状态变为s′,智能体获得奖励r;
9. 智能体将(s,a,s′,r)放入经验重放池R中;
10. 从R中随机批量采样样本训练演员、批评家网络;
11. end for
12. 更新当前数据中心总能耗E_{total}和热点个数h;
13. end for
14. return E_{total}

5.2.3 基于势能的奖励塑造

奖励塑造(Reward Shaping)的基本思想是在原来的奖励函数上加上一个小的中间奖励,通过修改奖励函数来帮助智能体更快地收敛。其中,PBRS已被证明是一种有效的奖励塑造,它基于物理学中的势能概念,使用一个势能函数来看待一个状态的好坏。在PBRS中,修改后的奖励函数可以定义为:

$$r'=r-\Phi(s_t)+\gamma\Phi(s_{t+1}) \tag{16}$$

其中,r和r'分别为修改前和修改后的奖励函数,而Φ函数为势能函数。

在PBRS中,关键问题在于如何设计势能函数。势能函数可以被视为模型设计者如何评估不同状态的好坏程度,势能函数的值越大表示该状态越好。在数据中心中,服务器可能处于空闲状态,此时虽然不执行任何任务,但服务器本身的功率消耗很高。因此,通过关闭空闲服务器可以有效降低数据中心的功耗。基于这个考虑,本文将关闭状态的服务器数量作为势能函数来衡量状态的好坏程度。因此,势能函数的定义为:

$$\Phi(s_t)=\sum_{i=1}^{N}x_i \tag{17}$$

其中,x_i为一个二进制变量,表示当前服务器的状态。如果x_i为1,表示第i台服务器处于关闭状态。

为了明确区分改进前后的算法,本文将使用PBRS改进后的DQN和DDPG算法分别命名为DQN-PBRS和DDPG-PBRS。

6 实验分析

本节通过仿真的方法对所提出的数据中心能耗优化方法进行性能分析。首先介绍实验设置,然后针对实验结果进行分析。

6.1 实验设置

与文献[6]类似，本文的实验仿真环境采用 Python 语言进行编写，并基于第 3 节中的各类模型计算数据中心的能耗与温度。在仿真环境下，每个任务调度算法都将针对给定的任务集合进行调度，最终计算出数据中心的总能耗以及产生的热点个数。深度强化学习算法的训练采用了 PyTorch 深度学习框架，并在 NVIDIA GeForce RTX 3060 显卡上进行了 1 000 个 epoch 的训练。仿真实验在包含两种不同类型的数据中心的仿真环境中进行，其中包括同构数据中心和异构数据中心。实验数据采用了两种不同类型的真实工作负载跟踪数据，用于模拟真实生产环境下的情况。

在本文的仿真实验中，同构数据中心和异构数据中心均采用了 CRAC 冷却技术。同构数据中心中共有 100 台服务器，这些服务器均配置了 Intel Xeon X5675 双核处理器。异构数据中心共有 200 台服务器，其中 100 台服务器配备了英特尔 Xeon X5670 双核处理器，另外 100 台服务器配备了英特尔 Xeon X5675 双核处理器。这两种 CPU 的功耗参数如表 1 所列。

表 1 CPU 参数

Table 1 CPU parameters

CPU 型号	CPU 核数	最小功耗/w	最大功耗/w
Xeon X5670	2	66	247
Xeon X5675	2	58	222

本文使用来自 PlanetLab 系统[26]和 Alibaba 计算集群[27]的真实工作负载跟踪记录作为实验数据。PlanetLab 数据集包括 PlanetLab 系统在几个月内超过 1 000 个虚拟机的历史使用记录，而本文只使用了其中一天的数据。而 Alibaba 数据集则包含了阿里巴巴公司一个生产集群在 12 h 内的集群运行信息，该生产集群包含了大约 1 300 台长时间运行的应用程序和批处理作业的机器。这两个数据集中的任务类型不同。PlanetLab 数据集中的任务具有较低的 CPU 需求和较高的任务到达率，而 Alibaba 数据集中的任务则具有较高的 CPU 需求和较低的任务到达率。总的来说，PlanetLab 数据集反映了互联网数据中心所需处理的任务特征，而 Alibaba 数据集反映了高性能数据中心所需处理的任务特征。

本文采用数据中心总能耗和运行过程中产生的热点个数来评估算法的性能。数据中心总能耗的定义分别如式(9)所示。数据中心热点的定义如式(7)所示。

实验对 DQN 算法和 DDPG 算法在多种实验环境下的性能进行了对比测试。其中，DQN 算法和 DDPG 算法的各个参数如表 2 所列。

表 2 算法参数

Table 2 Algorithm parameters

算法参数	DQN	DDPG
学习率	0.001	0.001
折扣因子	0.9	0.9
学习率	0.001	—
演员学习率	—	0.001
批评家学习率	—	0.001

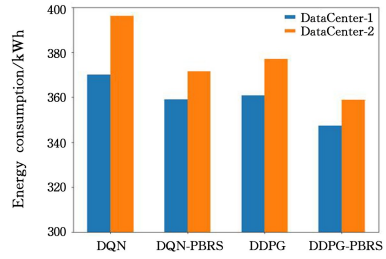
6.2 实验结果与分析

本文进行了两项实验：消融实验和对比实验。两个实验均在两类数据中心以及两个数据集共 4 种实验环境下进行。消融实验旨在验证本文对于深度强化学习算法改进的有效性。而对比实验则旨在比较本文方法与其他相关方法的性能差异。

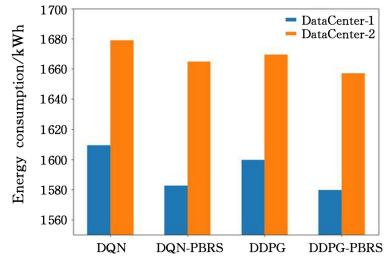
6.2.1 消融实验

为了验证本文提出的改进深度强化学习算法的有效性，本文进行了一项消融实验。在这个实验中，本文比较了基础版的 DQN 和 DDPG 算法，以及本文提出的改进版 DQN-PBRS 和 DDPG-PBRS 算法在仿真环境下的表现。实验结果如图 4 所示，其中图 4(a)给出了 PlanetLab 数据集下的实验结果，图 4(b)给出了 Alibaba 数据集下的实验结果。图 4 中的蓝色柱代表同构数据中心下的实验结果，橘色柱代表异构数据中心下的实验结果。

在消融实验中，4 种算法在任务调度过程中均未产生热点。为了简洁起见，本文只展示 4 种算法在数据中心总能耗方面的对比结果。



(a) PlanetLab dataset



(b) Alibaba dataset

图 4 消融实验

Fig. 4 Ablation experiment

从图 4 中可以看出以下结论。本文提出的改进版 DQN-PBRS 算法和 DDPG-PBRS 算法在数据中心总能耗方面均优于基础版的 DQN 和 DDPG 算法。在 PlanetLab 数据集下，相比基础版的算法，改进版的算法的数据中心总能耗大致降低了 6.5%。在 Alibaba 数据集下，改进版的算法相比基础版的算法，数据中心总能耗降低了 1.8%。从实验结果可以看出，本文对于 DQN 和 DDPG 算法的改进是有效的。

从图 4 中还可以得出以下两个结论。1)在相同的算法和数据集下，同构数据中心的总能耗要低于异构数据中心的总能耗。这是因为在本实验的实验设计中，同构数据中心的服务器能效比优于异构数据中心，同构数据中心在执行相同的任务时消耗更少的能量。2)相比 DQN 系列算法，DDPG 系列算法能够找出更好的任务放置方式。在实验结果中，DDPG 算法优于 DQN 算法，而 DDPG-PBRS 算法也优于 DQN-PBRS 算法。

6.2.2 对比实验

为了验证本文算法在优化数据中心能耗方面和热点方面的效果，本文对其进行了与以下 4 种具有代表性的任务调度算法的对比：随机调度算法(Random)、轮询调度算法(Round Robin、RR)、TASA[28]和 MITEC-GA[13]。这 4 种

对比算法的描述如下：

1）在 Random 算法中，任务被随机分配给一台服务器。

2）在 RR 算法中，任务被轮流分配给服务器，从第一台服务器开始，直到最后一台服务器，然后重新开始循环。

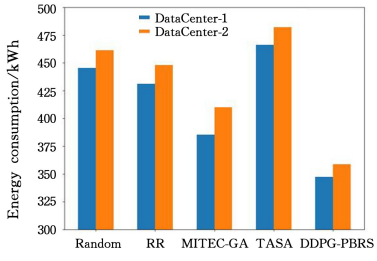
3）在 TASA 算法中，任务集合会根据服务器的温度分布和输入作业的温度来进行调度，最热的任务会被分配到最冷的服务器上。

4）在 MITEC-GA 算法中，调度算法基于遗传算法。为了优化数据中心总能耗，该算法将数据中心总能耗作为适应度函数。

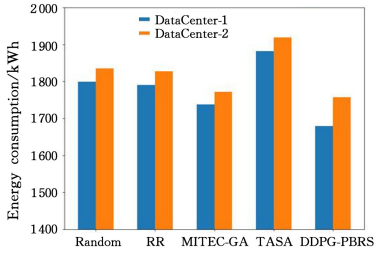
为了验证本文算法的性能，我们在仿真环境下对 DDPG-PBRS 算法与上述 4 种算法进行了对比实验，其结果如图 5 和图 6 所示。其中，图 5 给出了 5 种算法在数据中心总能耗方面的对比结果，而图 6 给出了 5 种算法在热点个数方面的对比结果。

从图 5 中可以看出，本文的 DDPG-PBRS 算法明显优于其他 4 种基线算法。在 PlanetLab 数据集下，相比其他 4 种基线算法，数据中心的总能耗最少降低了 9.8%，最多降低了 25.5%。在 Alibaba 数据集下，相比其他 4 种基线算法，数据中心的总能耗最少降低了 3.4%，最多降低了 10.7%。从实验结果可以看出，本文的 DDPG-PBRS 算法能够更好地优化数据中心能耗。

此外，从图 5 中还可以看出一个现象：DDPG-PBRS 在 Alibaba 数据集的领先幅度较小，领先幅度为 3.4%～10.7%；而在 PlanetLab 数据集下的领先幅度较大，领先幅度为 9.8%～25.5%。本文对此现象做出如下解释：PlanetLab 数据集下的任务到达率高且 CPU 需求量小，因此 PlanetLab 数据集中可行的任务放置方式更多，也就是说问题的搜索空间更大。因为问题的搜索空间更大，好的任务放置策略与差的任务放置策略之间的差距也更大。这也从侧面证明了 DDPG-PBRS 算法的搜索能力比其他算法更强，因此能够比其他算法找到更好的任务调度策略。



(a) PlanetLab dataset

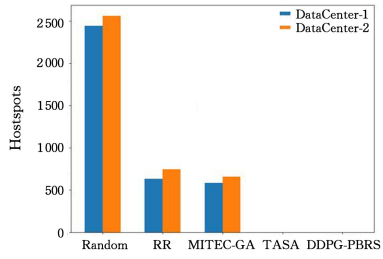


(b) Alibaba dataset

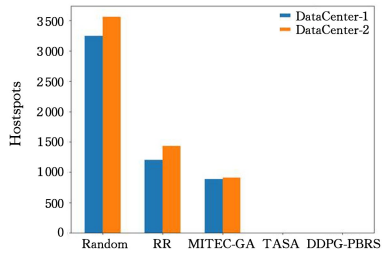
图 5　能耗对比实验

Fig. 5　Energy consumption comparison experiment

从图 6 可以得出以下结论：本文所提出的 DDPG-PBRS 算法在任务调度过程中不会产生热点，并且 TASA 算法也不会产生热点。相比之下，Random 算法会产生大量的热点，而 RR 和 MITEC-GA 算法则会产生少量的热点。这是因为 DDPG-PBRS 算法能够在任务调度中充分考虑数据中心的负载均衡和资源利用率，避免了任务过度集中在某些服务器上，从而避免了热点的产生。通过避免热点的产生，能够使得 CRAC 提高供冷温度，从而节省了大量的冷却能耗。与之相反，Random 算法是一种随机性较高的算法，不能有效地避免任务过度集中，导致了热点的产生。RR 和 MITEC-GA 算法虽然能够一定程度上避免任务过度集中，但是由于算法的局限性，仍然会产生一定数量的热点。



(a) PlanetLab dataset



(b) Alibaba dataset

图 6　热点对比实验

Fig. 6　Hot spot comparison experiment

结束语　随着数据中心数量的快速增长，数据中心的能耗问题愈发严重，严重影响了数据中心的可持续性。本文针对此问题，提出了一种基于深度强化学习的任务调度方法，通过优化任务调度，降低数据中心的能耗而不产生热点。首先，本文对数据中心能耗优化问题进行了形式化定义，并提出了基于深度强化学习的调度方法来解决这个问题。接着，本文使用 PBRS 对深度强化学习算法进行了优化，以进一步提高算法性能。最后，本文设计了一个仿真环境，并将本文方法与其他调度算法进行了对比实验。实验结果表明，本文方法能够有效地降低数据中心的能耗，并且在任务调度过程中不会产生热点，具有优秀的性能表现。

参 考 文 献

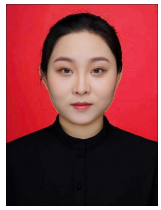
[1] NADJAHI C, LOUAHLIA H, LEMASSON S. A review of thermal management and innovative cooling strategies for data center[J]. Sustainable Computing: Informatics and Systems, 2018, 19: 14-28.

[2] DING J, ZHANG H, LENG D, et al. Experimental investigation and application analysis on an integrated system of free cooling and heat recovery for data centers[J]. International Journal of Refrigeration, 2022, 136: 142-151.

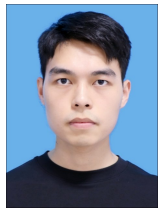
[3] TENG F, YU L, LI T, et al. Energy efficiency of VM consolidation in IaaS clouds[J]. The Journal of Supercomputing, 2017, 73(2): 782-809.

[4] KAPLAN J M, FORREST W, KINDLER N. Revolutionizing

data center energy efficiency[J]. McKinsey & Company, 2008: 1-13.
[5] HE K, LI Z, DENG D, et al. Energy-efficient framework for virtual machine consolidation in cloud data centers[J]. China Communications, 2017, 14(10): 192-201.
[6] ILAGER S, RAMAMOHANARAO K, BUYYA R. ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation[J]. Concurrency and Computation: Practice and Experience, 2019, 31(17): e5221. 1-e5221. 15.
[7] LIN M, WIERMAN A, ANDREW L L H, et al. Dynamic Right-Sizing for Power-Proportional Data Centers[J]. IEEE/ACM Transactions on Networking, 2013, 21(5): 1378-1391.
[8] FELLER E, ROHR C, MARGERY D, et al. Energy Management in IaaS Clouds: A Holistic Approach[C]// 2012 IEEE Fifth International Conference on Cloud Computing. 2012: 204-212.
[9] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with Deep Reinforcement Learning[A]. arXiv, 2013.
[10] KIRAN B R, SOBH I, TALPAERT V, et al. Deep Reinforcement Learning for Autonomous Driving: A Survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.
[11] GU S, HOLLY E, LILLICRAP T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]// 2017 IEEE International Conference on Robotics and Automation(ICRA). 2017: 3389-3396.
[12] 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.
[13] AKBARI A, KHONSARI A, GHOREYSHI S M. Thermal-Aware Virtual Machine Allocation for Heterogeneous Cloud Data Centers[J]. Energies, 2020, 13(11): 2880.
[14] LIU H, LIU B, YANG L T, et al. Thermal-Aware and DVFS-Enabled Big Data Task Scheduling for Data Centers[J]. IEEE Transactions on Big Data, 2018, 4(2): 177-190.
[15] AGHASI A, JAMSHIDI K, BOHLOOLI A. A thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm(FC-BGSA)[J]. Cluster Computing, 2022, 25(2): 1015-1033.
[16] TANG Q, GUPTA S K S, VARSAMOPOULOS G. Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(11): 1458-1472.
[17] 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.
[18] LI F, HU B. Deepjs: Job scheduling based on deep reinforcement learning in cloud data center[C]// Proceedings of the 4th international conference on big data and computing. 2019: 48-53.
[19] 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). Jeju: IEEE, 2018: 129-134.
[20] MAO H, SCHWARZKOPF M, VENKATAKRISHNAN S B, et al. Learning scheduling algorithms for data processing clusters[C]// Proceedings of the ACM Special Interest Group on Data Communication. New York, NY, USA: Association for Computing Machinery, 2019: 270-288.
[21] SHI L, WEN L, LEI S, et al. Virtual machine consolidation algorithm based on decision tree and improved Q-learning by uniform distribution[J]. Computer Science, 2023, 50(6): 36-44.
[22] LU H F, GU C H, LUO F, et al. Virtual machine placement strategy with energy consumption optimization under reinforcement learning [J]. Computer Science, 2019, 46(9): 291-297.
[23] BELOGLAZOV A, ABAWAJY J, BUYYA R. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing[J]. Future Generation Computer Systems, 2012, 28(5): 755-768.
[24] MOORE J D, CHASE J S, RANGANATHAN P, et al. Making scheduling "Cool": Temperature-aware workload placement in data centers[C]// USENIX annual technical conference, general track. 2005: 61-75.
[25] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. arXiv, 2019.
[26] PARK K, PAI V S. CoMon: a mostly-scalable monitoring system for PlanetLab[J]. ACM SIGOPS Operating Systems Review, 2006, 40(1): 65-74.
[27] LU C, YE K, XU G, et al. Imbalance in the cloud: An analysis on Alibaba cluster trace[C]// 2017 IEEE International Conference on Big Data(Big Data). 2017: 2884-2892.
[28] WANG L, VON LASZEWSKI G, DAYAL J, et al. Towards Thermal Aware Workload Scheduling in a Data Center[C]// 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks. 2009: 116-122.



LI Danyang, born in 1997, Ph. D. Her main research interests include green computing and energy saving.



WU Liangji, born in 1999, postgraduate. His main research interests include green computing and reinforcement learning.