[文章编号] 1007－7405(2021)06－0569－08 DOI:10.19715/j.jmuzr.2021.06.12

一种基于强化学习的CDN流量调度系统设计方法

林春莺

（集美大学理学院，福建 厦门 361021）

[摘要] 针对传统的CDN流量调度系统大多采用启发式方法或规划方法，存在维护成本高，实时性不足等缺点，提出一种基于深度强化学习的CDN流量调度系统设计框架。该框架基于马尔科夫链设计了故障告警网络来触发调度，建立了基于stacking模型的质量评估奖励函数，并在此基础上对流量调度进行定义和建模，构建了基于DQN的深度强化学习模型。最后，通过仿真实验验证了该调度框架的有效性。

[关键词] CDN；流量调度；强化学习；神经网络

[中图分类号] TP 393.2

A Design Method of CDN Traffic Scheduling System Based on Reinforcement Learning

LIN Chunying

(School of Science, Jimei University, Xiamen 361021, China)

Abstract: The traditional CDN traffic scheduling system mostly adopts heuristic method or planning method, which has the disadvantages of high maintenance cost and poor real-time performance, resulting in weak quality of service control. This paper studies a design framework of CDN traffic scheduling system based on deep reinforcement learning. The framework defines and models traffic scheduling, firstly it triggers scheduling by constructing fault alarm network based on Markov chain, and constructs deep reinforcement learning model based on DQN which constructs quality evaluation reward function based on stacking model. Simulation results verify the effectiveness of the scheduling framework.

Keywords: content delivery network; traffic scheduling; reinforcement learning; neural network

0 引言

内容分发网络（content delivery network，CDN）是一种在服务提供方和消费方之间，通过架设节点或者服务器，让用户就近获取所需的内容，从而达到缓解网络拥塞，提高用户访问网站响应速度的目的。基于这种服务结构，如何有效管理不同服务的资源调度分配，保证高质量的服务是一个需要深入研究的问题。特别是当资源处于超负荷运转或网络波动时，如何通过调度来及时选择合适的替代资源，是保证服务质量的关键。传统业界的调度方式一般采用基于规则的方式[1]，即，先收集大量服务器资源的指标（比如负载情况、网络情况、物理位置、服务冗余度等），再统计评价指标，最后以此来制定规则。但指标量越大则导致调度规则越复杂，使修改和维护成本更高，灵活性更差，同时调度策略往往只考虑服务质量，而没有考虑现实的成本问题。

[收稿日期] 2021－04－23

[作者简介] 林春莺（1978—），女，讲师，从事边缘计算，数据挖掘，网络安全方向研究。

近年来强化学习（reinforcement learning，RL）[2]因其强大的探索能力和自主学习能力，在游戏[3]、机器人控制[4]、交通控制[5]等领域都有广泛的应用。在资源调度策略领域，文献［6］利用 Q 值强化学习，将虚拟机资源调度描述成马尔科夫决策过程，并设计了动作的奖励函数，实现虚拟机的资源调度策略；文献［7］将深度强化学习的思路应用到微电网在线优化调度过程中；文献［8］应用强化学习算法求解置换流水车间调度问题；文献［9］提出 DeepRM 模型，将资源调度系统的状态信息建模成图像形式，将获取的图像信息输入到卷积神经网络中，通过卷积神经网络对图像信息进行快速特征提取，用强化学习的方法对神经网络中的参数进行迭代更新，形成最终的策略；文献［10］使用分析集群状态和机器状态的两层深度强化学习模型来共同完成对集群资源和能耗的管理工作。这些应用都是将现实问题转化为强化学习问题，通过训练智能体（agent）以及定义符合领域知识的环境反馈，进行动态灵活的学习。

为了解决 CDN 资源调度中存在的复杂规则问题，同时综合考虑服务质量和成本，本文拟结合强化学习的优势，对 CDN 资源调度问题进行重新定义，提出一种基于深度强化学习的 CDN 资源调度系统设计方法，以避免人为指定调度规则带来的不准确、维护困难以及成本波动等问题。

1 系统设计

1.1 总体框架设计

本文提出的基于深度强化学习的流量调度系统的设计内容，主要包括调度的智能告警触发模块、综合质量和成本的调度评估模块、强化学习模块。其工作原理主要为：1）采集节点和服务器相关指标以及故障数据，构建基于马尔可夫模型的智能报警网络，产生流量调度的增量带宽需求；2）对节点和服务器的带宽情况、服务能力、用户覆盖情况、响应时间等进行综合考虑，建立质量评估模型（强化学习环境的奖励函数的组成部分）；3）对节点和服务器的计费类型、计费系数等进行综合考虑，建立成本评估模型（强化学习环境的奖励函数的组成部分）；4）针对增量带宽需求，结合质量和成本评估奖励函数，在线上受限环境（model-free）或者虚拟环境（model-base）中进行迭代学习，建立深度强化学习模型；5）在全网环境部署模型后，根据调度产生的数据不断进行在线学习，完善调度系统。

1.2 故障报警网络设计

建立故障报警网络是为了能及时发现出现服务质量的节点或服务器，触发流量调度。本研究通过采集近期故障报警历史数据，建立基于马尔科夫链的异常检测算法，进行智能告警。主要有：1）采集故障报警历史数据。历史数据指因监控服务器而产生的从底层机器指标（CPU、请求数、响应时间）到高层机器指标（客户投诉）报警的时间序列。2）对故障报警历史数据进行离散化，设立低、中、高三个告警级别，从而建立不同告警级别前后相连的马尔科夫链。3）基于报警马尔可夫链，建立故障报警网络。当发生低级别机器指标报警时，根据网络内各个报警级别间的转化关系及概率，预测高级别报警产生的概率，若概率较高，则会以较大概率触发调度。

1.3 基于 Stacking 模型的质量评估模型

在基于强化学习的调度系统中，当调度系统生成一个动作时，需要环境给出奖励来指导整个学习过程。本研究为强化学习提供服务质量的奖励函数。当一个带宽增量需求提出时，虽然机器的承载带宽可承载，但是由于业务特性不同，对机器的 CPU、内存等要求是不同的，若不加以区分就会造成机器资源耗竭，影响服务质量。因此，本研究提出基于 Stacking 模型的服务质量评估模块，用来预测加量后机器资源的 CPU 使用率，并以此作为机器负载的判断标准。过程步骤如下：

1）采集训练数据：采集线上机器资源的业务属性（带宽量、http 请求数）和机器属性（内存大小、磁盘类型、CPU 核数、CPU 主频大小）作为输入特征，采集机器资源的多核 CPU 实时使用率均值为预测值，同时统计不同业务特征每 1 Mibit/s 带宽对应增加的 http 请求。

2）数据集划分：将历史 n 天数据作为训练集，第 $n+1$ 天数据作为测试集，并采用滑动窗口方式在时间轴上将数据集不断划分。

3）基模型训练：先使用 Xgboost、RandomForest、Lightgbm 算法作为基模型，采用 K 折交叉验证方式对每种模型进行训练，然后每种基模型训练后会得到 K 个子模型以及对训练集样本的 CPU 使用率预测结果，再保存每种基模型的各个子模型。

4）模型融合：用 K-NearestNeighbor（KNN）进行模型融合，将步骤 3）中每种基模型下的子模型对应训练集的 CPU 使用率预测结果作为 KNN 模型的输入特征，将训练集真实 CPU 使用率作为 KNN 模型的输出，设置训练模型近邻个数为 5（见图 1 左边部分）。

5）预测阶段：假设要给机器资源加量 500 Mibit/s 带宽量，实时采集当前机器的业务属性（带宽量、http 请求数）和机器属性（内存大小、磁盘类型、CPU 核数、CPU 主频大小），在此基础上将增量的 500 Mibit/s 按照统计系统换算成 http 请求数加到当前值中，带宽量也加 500 Mibit/s，训练好的模型对增量后的特征进行预测，得到 CPU 使用率，如图 1 所示。

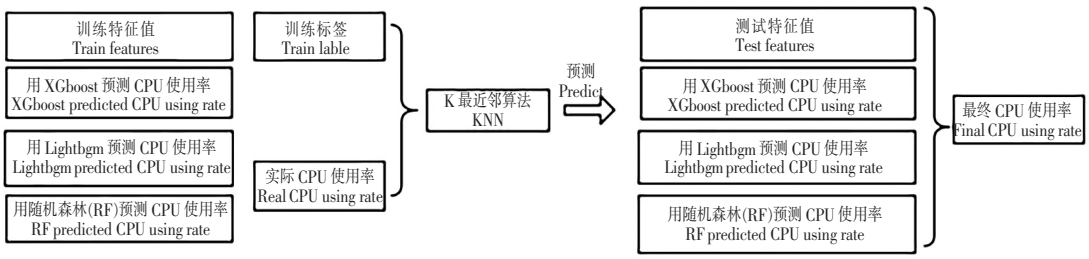

图 1　Stacking 模型训练图

Fig.1　The pinpline of stacking model

1.4　深度强化学习调度模型设计

本文所要解决的 CDN 资源调度问题为服务的增量需求问题。当客户请求访问增加，导致当前服务的机器资源负载增加，出现服务响应时间变长，服务质量降低，此时需要为当前的客户请求增加额外的机器资源。其调度结构如图 2 所示。其中：增量需求包含带宽增量（如需要增加 300 Mibit/s）和地理属性（如福建）两个重要属性；服务器资源是有额定带宽上限的机器资源，可用资源池为机器剩余的可用带宽，不同的机器资源有不同的价格系数，代表了机器的成本因素；调度策略为强化学习的智能体（agent）结构，调度策略的目标是既要满足带宽增量的要求，又要兼顾服务质量和成本的均衡。

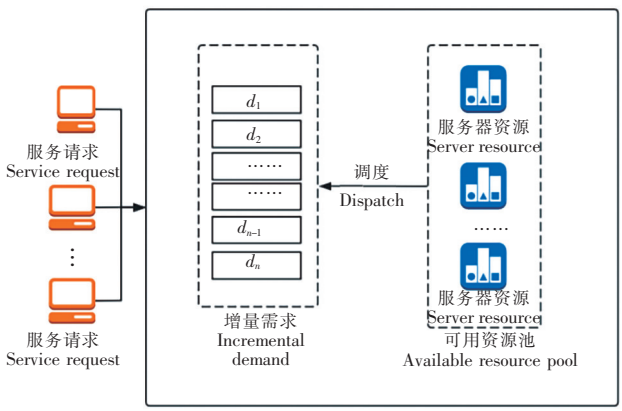

图 2　CDN 资源调度结构图

Fig.2　Structure of CDN resource scheduling

1.4.1　问题建模

与传统的机器学习相比，强化学习更注重与环境的交互，通过不断与环境交互获得奖励学习，以此得到最佳的动作（如图 3 所示）。

本文将 CDN 增量需求的调度问题转化为基于有限的马尔科夫决策过程。这种过程可以用

图 3　强化学习架构图

Fig.3　Framework of reiforcement learning

一个五元组表示 $(S,A,P_a(s_t,s_{t+1}),R_a(s_t,s_{t+1}),\gamma)$。其中：$S$ 为状态集合；A 是动作集合；$P_a(s_t,s_{t+1})$ 为在时刻 t 的状态 s_t 经过执行动作 a 可以在 $t+1$ 时刻转换到状态 s_{t+1} 的转换概率；$R_a(s_t,s_{t+1})$ 为执行动作 a 所产生的奖励值；γ 为折扣因子，值处于［0，1］区间，用来表示奖励值对累积奖励值的影响权重。对于本文要解决的 CDN 增量需求的调度问题，动作 A 为将可用资源池中服务器资源分配给某个增量需求；状态 S 为分配后增量需求满足情况和资源的剩余情况；而奖励 R 为分配后 CDN 服务产生的服务质量和成本。本文设计的强化学习的目标就是学习最佳的策略（h），能使整个任务过程的累积奖励值最大。

本文采用 Q 学习方法中的动作评价函数 $Q(s,a)$ 来描述在状态 s 时 agent 选择动作 a 后所得到的最大累积奖励。在 agent 训练过程中，总是选择 Q 最大值的动作：

$$Q(s_t,a_t) = r(s_t,a_t) + \gamma_{\max} Q(s_{t+1},a_{t+1})。$$

1.4.2 策略设计

本文采用基于卷积网络的 DQN(deep Q-network)[11] 算法来训练 agent 智能体。DQN 算法使用参数为 θ 的深度卷积神经网络作为动作值函数的网络模型，用模型 $Q(s,a,\theta)$ 来模拟最佳 $Q(s,a)$。在训练过程中，该网络模型生成每个动作的 Q 值，采用递增的 ε-greedy 策略来选择动作，生成一系列的状态、动作和奖励值。DQN 训练模型如图 4 所示。

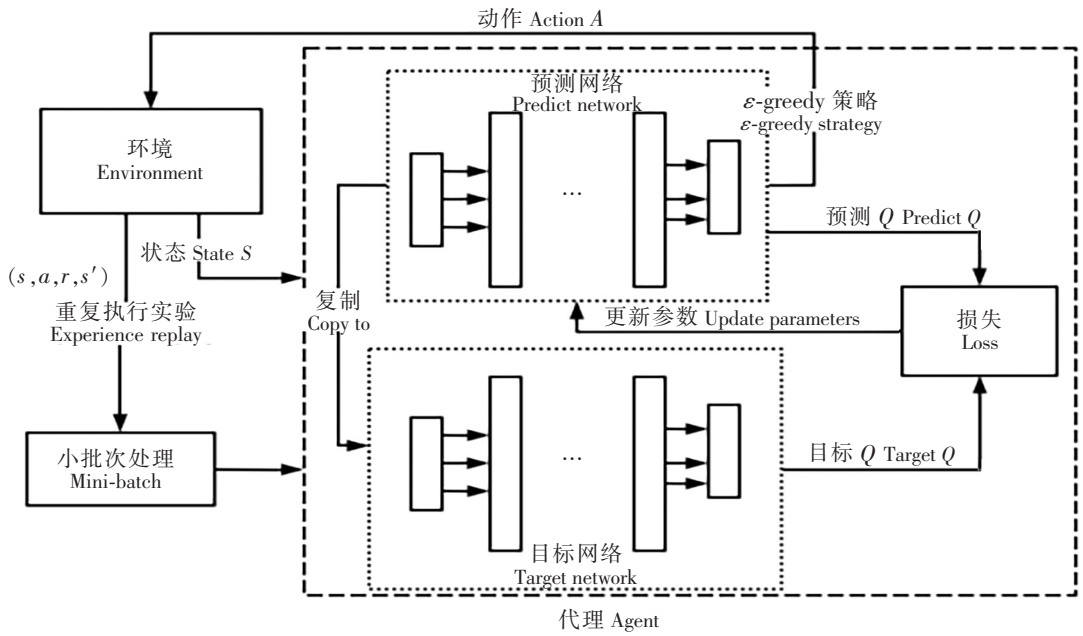

图 4 DQN 训练图

Fig.4 Train pipline of DQN

DQN 采用均方误差作为深度网络的损失函数，公式为 $L_i(\theta_i) = E[(r+\gamma \max_{a'} Q(s',a';\theta_i) - Q(s,a;\theta_i))^2]$。其中：$\gamma$ 为折扣因子；θ_i 为第 i 次迭代的网络参数；s',a' 为下一个状态和动作。

DQN 采用 mini-batch 方式的随机下降法来实现对目标损失函数的优化。每产生一个动作 a 和环境交互后，神经网络都会进行一次迭代学习，同时更新参数，直到收敛。

1.4.3 状态空间

策略中状态空间包含可用的机器资源池和增量带宽需求。由问题定义可知，资源池和增量需求都是以数值属性带宽量（如 500 Mibit/s）表示，这样动作设计（分配资源）时只能整个分配或者不分配，使资源分配方式受限。所以在状态空间设计时，需将资源和需求量拆分成更小的单元（如 100 Mibit/s）。如图 5 所示，左边为 5 个带宽需求，有颜色的部分代表最小单元带宽，第一个

需求为增强 500 Mibit/s；右边为当前资源池中每台机器可用带宽资源，同样有颜色的表示最小带宽单元，第一个带宽资源为 700 Mibit/s。将有颜色的部分用 1 表示，空白的地方用 0 表示，就可以将资源状态转换为图像矩阵的形式，作为 DQN 深度卷积网络的输入，从而提取重要特征。由于 DQN 神经网络需要保持输入的固定大小，状态空间设计时，设定固定 M 个需求，而对于剩余的需求则可以放在等待队列。

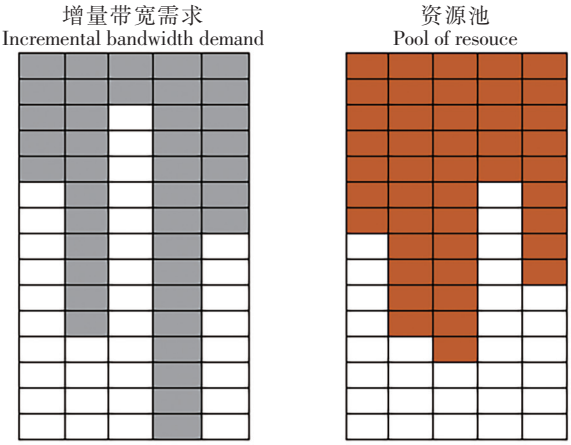

图 5　状态空间示意图

Fig.5　The diagram of state space

1.4.4　动作空间

策略中的动作空间为将资源分配给带宽需求的动作集合。设策略中包括 M 个带宽需求，N 个机器资源，则总的状态空间数量为 $N \times M + 1$ 个，$\{0,1,2,3,\cdots,N \times M - 1,\varphi\}$，$\varphi$ 表示空动作，不做分配。在每次迭代时，调度器将资源分配给合适的带宽需求，并更新系统中的资源状态，直到所有的带宽需求都分配完或者所有的资源都分配完才中止迭代。

1.4.5　奖励函数

本策略中针对 CDN 资源调度问题，兼顾质量和成本的目标，设计了一个基于模型的评估函数。当一个机器资源分配给一个带宽需求时，除了要满足需求的带宽，还要使机器资源在地理位置上尽可能相同，这样才能保证就近访问，提高响应时间。另外，还需要考虑加量后机器资源是否会出现高机器负载。虽然机器资源带宽是允许承接设定的带宽量，但由于业务特性不同，使用 CPU 资源不同，即使同样的带宽量产生的负载也会不同，如网页和流媒体的 CPU 资源使用率就完全不同。服务成本要求尽可能选择价格系数小的机器资源。

因此，策略中的奖励函数可以表示成：$r = (m \times g \times l)/p$。其中：$m$ 表示满足带宽的比例，可以在分配后计算；g 表示地理位置的匹配程度，可以根据带宽需求的地理属性和机器资源的地理属性匹配情况，按照不同层级设定不同的比例（如同城为 1，同省为 0.8，同大区为 0.5，跨大区为 −0.5，跨国为 −1）；l 表示加量后负载情况；p 为价格系数，是机器资源的固有属性。

对于如何判断加量后会不会产生机器负载问题，本策略先采用 Stacking 质量评估模型进行预测，得到增量带宽后的 CPU 使用率；再根据 CPU 使用率的不同等级设置不同的分数，如 >90% 为 −1，[80%,90%] 为 0.2，[50%,80%] 为 0.6，<50% 为 1。

1.5　深度强化学习调度模型训练

1.5.1　训练参数

在 DQN 网络训练中，为使 agent 在训练前期对最优策略的探索力度增加，采用递增的 ε − greedy 策略来选择动作。设 ε 的初始值为 0.5，最大值为 0.9，增幅为 0.001；折扣因子为 0.95；经验池规模为 30 000；迭代次数为 1000；采用 Mini-batch 训练方法，设 batch-size 为 32；采用随机梯度下降方法更新 Q 网络参数，优化器为 Adam，学习率为 0.001。每 C 个训练回合后将当前 Q 网络的参数值复制给目标 Q′网络，并更新一次目标网络参数。

1.5.2　受限环境调度系统

CDN 是一个复杂的内容分发网络，在训练本文提出的基于强化学习的 CDN 流量调度系统时，需要在不影响现有网络基础上构造一个可行的环境。受限环境（model-free）调度系统是指系统的学习过程、调度过程都是在真实的环境下进行，同样，接收到的结果反馈也来自于真实的环境。model-free 调度系统结构图见图 6。

本文构建 CDN 受限环境的目的在于从 CDN 全网环境中规划出一个小范围的环境，使得深度强化

学习模型在训练和验证期间的一系列试错行为产生的负面影响，被控制在一个较小的范围，避免给全网的服务质量和成本带来波动。该受限环境可以根据物理位置、运营商或者不同等级的用户等进行划分。

本文提出的设计方法分成 CDN 受限环境下的训练阶段和 CDN 全网环境下的应用阶段。

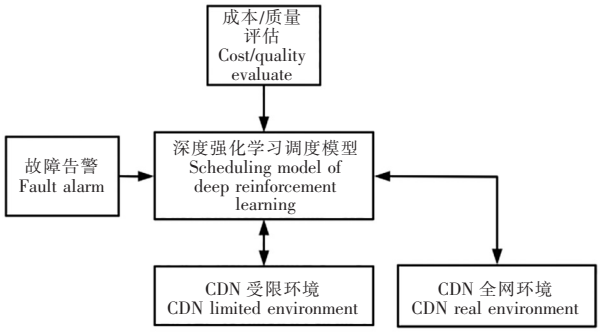

图 6 Model-free 调度系统结构图

Fig.6 The structure diagram of model-free scheduling system

在训练阶段，若出现触发调度的情况，则设置 CDN 受限环境下资源配置情况为模型输入状态，模型会基于当前网络中的参数，输出替代资源的挑选概率或者分数，并根据一定策略选择替代资源，然后评估系统对替代资源的服务情况进行评估并反馈给模型，模型接收到反馈信号后，根据反向传播算法，调整网络参数，使后续调度能够朝着全局最优的方向进行。

当训练达到一定迭代次数或者结束条件时，会生成性能较好的调度模型。该调度模型会应用到 CDN 全网环境上，在后续的运行过程中仍然持续进行在线学习。若发生触发调度的情况时，模型会根据学习的结果选择可能对未来产生正面影响的替代资源进行服务，同时评估系统会对替代资源的选择进行评估，调度模型根据反馈信号进行参数调整。

1.5.3 虚拟环境调度系统

虚拟环境（model-based）调度系统是指系统的学习过程是在虚拟环境中完成，而调度过程是在真实环境中进行。虚拟环境调度系统结构图如图 7所示。

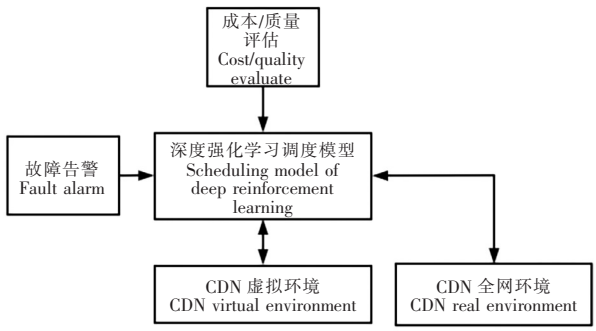

图 7 Model-based 调度系统结构图

Fig.7 The structure diagram of model-based scheduling system

CDN 虚拟环境是指根据线上 CDN 全网环境虚拟出的一个环境。该环境支持调度系统与其进行交互，且支持根据调度结果，模拟线上 CDN 全网环境对其的响应。其评估系统的工作方式和原理与受限环境调度系统一样，但由于是虚拟环境，调度触发条件中的故障报警功能可以用简单的元胞自动机进行模拟。

基于 CDN 虚拟全网环境来训练模型有两点优势：

1）线上真实环境需要等待调度触发条件出现，才能进行调度，评估调度结果，累积训练数据。在这样的情况下让模型的性能收敛，可能需要一个比较长的时间周期。如果是在虚拟环境中，可以通过元胞自动机进行触发条件模拟，相较于真实环境下能有千倍甚至万倍的效率对模型进行训练，大大缩短训练时间周期。

2）CDN 受限环境是为了减小模型在训练期间的试错行为对线上服务质量和成本带来的负面影响，而规划出的 CDN 全网环境的一个子集。该子集环境虽然能够将负面影响限制在一个较小的范围内，但是却不能完全消除影响。另外，由于是一个 CDN 的子集环境，在该环境上产生的用于模型训练的数据是一个局部数据集，可能与 CDN 全网环境产生的全局数据集不是来自同一个分布，可能会导致深度强化学习模型出现过拟合现象，而不能很好地泛化到全网环境。而 CDN 虚拟环境则解决了以上两个问题，一方面，它在虚拟环境中的任何操作都不会对真实环境造成影响；另一方面，虚拟环境是模拟真实的全局环境，避免了数据层面的局限性。

2　仿真实验结果分析

在阿里云申请了全球 10 个机房节点资源，20 台服务器设备，对本文算法进行实验仿真。

首先，训练 Stacking 质量评估模型。实验定义 ABS（实际 CPU – 预测 CPU） <5 表示模型预测准确，模型预测精度 = 模型预测准确数量/样本总数。预测结果显示，CPU 的精度整体可以达到 95%。从在线运行结果（见图 8）可知，21:03 实际带宽 2.57 Gibit/s，预期加量 700 Mibit/s，模型判断不能加量；21:12 实际带宽 3.15Gibit/s，机器的健康值为 0.7，加量导致机负载超负荷。由实验可知，质量评估模型能有效评估加量后的机器质量情况，能够为强化学习提供有效的奖励反馈。

图 8　质量评估模型线上验证图

Fig.8　The online result of quality evalution model

图 9 显示了强化学习训练过程不同迭代的奖励值。由图 9 可知，随着迭代的增加，智能体 agent 完成调度任务所获得的奖励值不断增长直至收敛。

在线上部署强化学习过程中，对比开启强化学习调度前后的效果（见图 10），发现本算法在不同产品线上不仅能满足合适的带宽要求，而且机器的服务质量也有明显的提升。

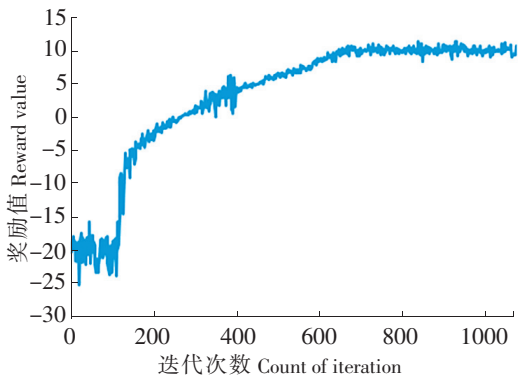

图 9　强化学习迭代奖励值收敛图

Fig.9　Reward value of reinforzement learing with iteration

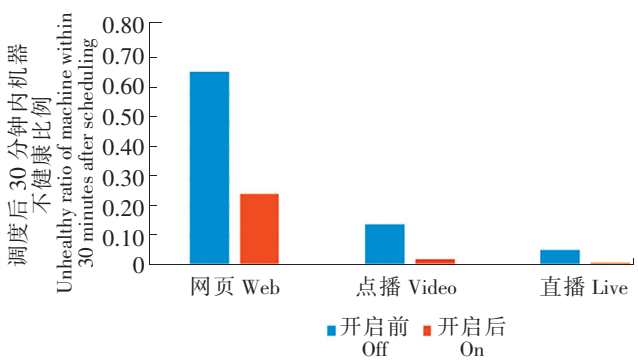

图 10　强化学习调度前后服务质量对比

Fig.10 Before and after quality comprison of deep reinforcement learning

在带宽调度上，线上实验结果表明强化学习模型可以有效地控制节点跑高。如图 11 所示，节点跑高后 1 min 内开始降量。

最后对比了基于规则调度系统和深度强化学习调度系统在不同场景中的调度耗时。由表 1 可见，深度强化学习调度系统提升效果明显。

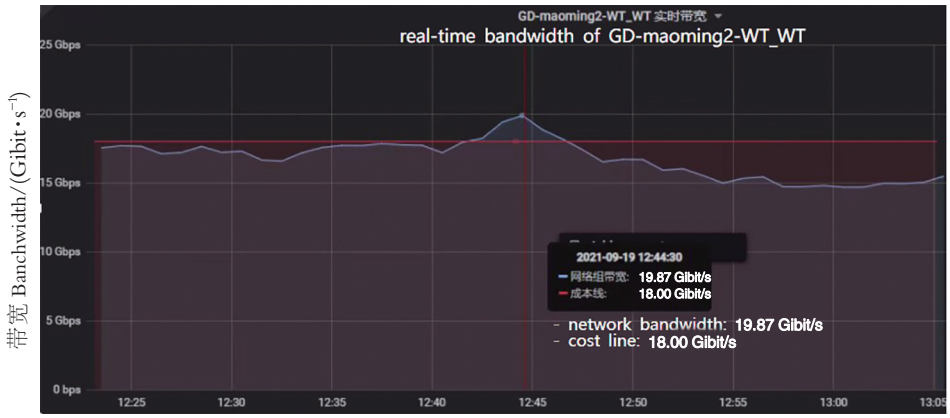

图 11 节点跑离调度图

Fig.11 The schedling result of node when exceeding the linit

表 1 调度耗时对比图

Fig. 1 Time consuming comparison of scheduling

场景 Scene	传统调度 Traditional scheduling	深度强化学习调度 Scheduling of deep reinforcement learning	提升效果 Enhancing effect
节点故障调度 Scheduling of node fault	3 min	30 s	83%
节点跑高调度 Scheduling of node when exceeding the limit	3 min	30 s	83%
健康值调度 Scheduling of health value	3 min	10 s	94%

[参 考 文 献]

[1] 雷爱民，祖兆研，王家文. 基于机器打分的 CDN 调度策略的研究 [J]. 信息系统工程，2018(3)：1.

[2] MOUSAVI S S，SCHUKAT M，HOWLEY E. Deep reinforcement learning：an overview [C] //Proceeding of SAI Intelligent Systems Conference. Cham：Springe，2018.

[3] MNIH V，KAVUKCUOGLU K，SILVER D，et al. Playing atari with deep reinforcement learning [J]. Computer Science，2013.

[4] KOBER J，BAGNELL J A，PETERS J. Reinforcement learning in robotics：a survey [J]. The International Journal of Robotics Research，2013，32(11)：1238-1274.

[5] 舒凌洲，吴佳，王晨. 基于深度强化学习的城市交通信号控制算法 [J]. 计算机应用，2019，39(5)：5.

[6] 李天宇. 基于强化学习的云计算资源调度策略研究 [J]. 上海电力大学学报，2019，35(4)：399-403.

[7] 季颖，王建辉. 基于深度强化学习的微电网在线优化调度 [J/OL]. 控制与决策 [2021-09-01]. https:/doi-org/10.13195/j.kzyjc.2021.0835.

[8] 张东阳，叶春明. 应用强化学习算法求解置换流水车间调度问题 [J]. 计算机系统应用，2019，28(12)：195-199.

[9] 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. Atlanta：IEEE，2016：50-56. DOI:10.1145/3005745.3005750.

[10] LIU N，LI Z，XU J，et al. A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning [C] //37th International Conference on Distributed Computing Systems (ICDCS). Atlanta：IEEE，2017：372-382. DOI:10.1109/ICDCS.2017.123.

(责任编辑 朱雪莲 英文审校 黄振坤)