



边缘计算使能星地协同网络下的服务部署机制

Service Deployment Mechanism in Edge Computing Enabled Satellite Terrestrial Integrated Network

摘要:在移动边缘计算(MEC)与星地协同网络(STIN)融合的网络架构中,针对卫星网络和边缘计算对时延与资源敏感的特点,以最大化用户服务质量(QoS)为目标,提出基于强化学习的深度Q网络(DQN)算法部署机制。将部署问题描述为一个马尔可夫决策过程(MDP),并把卫星节点的状态和部署行为分别建模为DQN中的状态和动作。通过卫星的计算资源与卫星和用户的通信时延给出奖励值,在神经网络中训练以优化部署行为,进而实现最优部署策略,并对提出的算法做仿真。与其他算法对比的结果表明,在相同的优化目标条件下,DQN算法有较好的性能。

关键词:边缘计算;服务部署;强化学习

Abstract: In the network architecture of mobile edge computing (MEC) and satellite terrestrial integrated network (STIN), the satellite network and edge computing are sensitive to delay and resources. To maximize user's quality of service (QoS), a deployment mechanism based on the reinforcement learning deep Q network (DQN) algorithm is proposed. The deployment problem is described as a Markov Decision Process (MDP). The state and deployment behavior of the satellite nodes are modelled as the state and action in the DQN. The reward value is given by the satellite computing resources and the communication delay between the satellite and the user. Training in the neural network to optimize the deployment behavior achieves the optimal deployment strategy. The proposed algorithm is simulated and compared with other algorithms. The result shows that under the same optimization target conditions, the DQN algorithm has better performance.

Keywords: edge computing; service deployment; reinforcement learning

卢华/LU Hua¹,
段雪飞/DUAN Xuefei¹,
李斌/LI Bin²

(1. 广东省新一代通信与网络创新研究院, 中国 广州 510663;
2. 中兴通讯股份有限公司, 中国 深圳 518057)
(1. Guangdong Communications & Networks
Institute, Guangzhou 510663, China;
2. ZTE Corporation, Shenzhen 518057,
China)

DOI: 10.12142/ZTETJ.202103013
网络出版地址: <https://kns.cnki.net/kcms/detail/34.1228.TN.20210621.1738.002.html>
网络出版日期: 2021-06-22
收稿日期: 2021-05-12

近年来,互联网与通信技术都取得了长足进步。大数据、云计算等新兴技术已经得到广泛运用并成为当前的基础性技术^[1]。受益于5G的大规模使用,物联网(IoT)、虚拟现实(VR)/增强现实(AR)/混合现实(MR)、高分辨率(4K/8K)视频传输得到了进一步推广。然而,以车联网(IoV)、远程医疗、高帧率游戏等为代表的要求响应速度快、时延超低、占用带宽较大的应用,对现有网络体系

架构提出很大的挑战。虽然5G的应用可以缓解部分需求,但是用户与云计算中心通信产生的时延,以及海量数据传输对带宽的占用,与云计算技术本身都是矛盾的。为了解决这些问题,我们需要在数据中心之外,让计算、存储、网络延展到互联网的边缘,甚至到每个家庭的互联网网关上,使服务更加靠近用户。这种技术就是边缘计算^[2-3]。星地协同网络虽然有着很好的发展前景,但也面临着

和上述云计算类似的高数据速率、低通信时延等挑战。移动边缘计算(MEC)技术的引入可以更好地保障用户服务质量(QoS)。

关于边缘计算中服务部署问题的研究有很多。文献[4]将边缘计算系统中的服务部署建模为一个多阶段随机规划问题,设计了一个样本平均近似(SAA)方法以估计多阶段模型中资源函数的期望值,并提出贪心算法来解决基于SAA的并行算法中

每个阶段都需要解决的整数优化问题。针对把服务完全部署到本地的情况,文献[5]将问题建模为非线性整数规划问题,并采用元启发式算法求出近似解。文献[6]将服务部署问题建模为马尔可夫决策过程(MDP),并设计了一种在线算法,同时证明该算法是成本最优的。文献[7]同样将服务部署问题建模为MDP,但采用强化学习中的Dueling-DQN算法(一种改进的DQN算法)进行求解。

不同部署问题的解决方案虽然有很大不同,但基本可以归纳为传统算法和基于学习的方法。传统算法一般将问题描述为规划问题或优化问题,但通常由于问题的复杂性以及多目标约束的存在而变为非确定性多项式(NP)问题,使求解变得困难。而部署问题能够容易被建模为MDP过程,可采用强化学习中的Q-Learning或DQN等算法进行求解。

1 服务部署模型与算法设计

1.1 服务部署模型设计

这里,我们首先对研究问题做一些说明和假设:

- (1)对于每个卫星,除运行轨迹不同外,其他完全相同;
- (2)用户请求的服务相同;
- (3)用户与卫星的距离用时延来描述;
- (4)卫星的可用计算能力与中央处理器(CPU)、内存占用率成反比;
- (5)卫星的CPU和内存消耗是线性的;
- (6)服务在节点上并行计算;
- (7)卫星计算能力存在上限和下限。

为了使用强化学习算法解决服务部署问题,我们需要将其建模为MDP,具体过程如下:

我们需要先明确优化指标和具体的影响因素。本文中,我们选择最小化处理特定数量服务并交付给用户所用总时间为优化目标,如公式(1)所示:

$$\min \sum_{e \in E} \sum_{u \in U_e} (proc_e + delay_{u,e}). \quad (1)$$

公式(1)中, E 表示边缘节点集合, U_e 表示服务部署在节点 e 上的用户集合, $proc_e$ 表示在节点 e 上处理服务需要的时间(根据假设,相同节点上的 $proc$ 相同), $delay_{u,e}$ 表示用户 u 与节点 e 的通信时延。需要说明的是,这里的 $delay$ 不仅代表时延,还代表用户与卫星的物理距离。因此,我们可将时延进行适当的放大,以扩大其在问题中的影响。

MDP是一个四元组 $\langle S, A, P, R \rangle$,分别代表状态、动作、状态转移概率和奖励。本问题中的状态转移概率均为1。下面我们将讨论 S 、 A 与 R 。

边缘节点共有9个,即 $E_{size} = 9$ 。此外,我们还需要确定希望部署的服务数量 n (假设一个用户请求一个服务,即 $U_{size} = n$)。这样我们就可以将状态集定义为 $\dots S = \{s_0, s_1, \dots, s_n\}$ 。 s_i 表示当前已经部署 i 个服务,它仍是一个集合,所包含的状态数可以用简单的排列组合计算得到。其中 s_0 为初始状态, s_n 为终止状态。我们可以将这样的状态集称为简化状态集。相应地,我们可以定义具体状态集,以描述每个服务的具体部署位置(即在哪个节点上)。这个集合共有 $\sum_{i=0}^n 9^i$ 个状态,其中 9^i 是简化状态集中 s_i 扩展到具体状态集的个数。基于前述假设,我们在算法设计中使用简化状态集。可以看出,本问题与传统强化学习问题有所不同:传统问题的状态

转移步数是不确定的,而本问题的状态转移步数是确定的,并且当经过 n 步之后就一定会达到终止状态。

在本问题中,MDP中的动作是把服务部署在某个边缘节点上。我们可以规定服务的部署顺序。对于某个状态集 s_i 而言,要部署的服务就是确定的。此时,动作数量与边缘节点数量一致。本问题的MDP在状态集 s_i 中执行一个动作 a ,随后进入状态集 s_{i+1} 。

奖励是决定算法最终效果的核心。在使用简化状态集时,我们显然不能为状态集 s_i 中的所有状态设置同一个奖励值。单纯地为简化状态集中的每一个状态而定义一个奖励值也是不合理的。因此,在设置奖励值时,我们要按具体状态集来处理。

具体奖励值的设置要参考公式(1)的优化目标。我们可以把每一个状态都当作终止状态。此时 E 代表部署中有服务的节点。我们就可以利用公式(1)来计算当前状态消耗的时间,并计算处理相同服务所花费的基本时间。由于存在时延和节点计算能力下降等因素,实际时间会比基本时间长,因此我们可以通过时间差来确定奖励值。实际时间越长,奖励值就越低。相应的计算公式如公式(2)所示:

$$R = \frac{t_{basic}}{\sum_{e \in E} \sum_{u \in U_e} (proc_e + delay_{u,e})}. \quad (2)$$

1.2 基于服务部署模型的算法设计

当利用强化学习来求解MDP模型时,我们可以采用Q-Learning或DQN算法。在本问题中,即使我们采用简化状态集,随着服务数量的增加,其规模也呈指数级增长,此时不宜采用Q-Learning算法进行求解。因此,本文中我们采用DQN算法。

算法的模型如图1所示。操作环境输入选择的动作,并执行该动作,随后进入下一状态,同时反馈这一步的奖励值和是否到达终止态等信息。这些信息会形成一条记录被存入经验回放区。当经验回放区存储一定数量的记录后,神经网络会从中随机选取一些记录来进行训练,并更新相应的网络参数,选择基于当前网络参数选出的价值最大的动作来让环境执行。新的记录生成后会被继续存入经验回放区。当经验回放区的数据足够多时,新记录将逐渐代替旧记录,以便于那些之前使用价值不大的记录不会再被学习。本文中,我们使用的神经网络有两个隐藏层。神经网络通过反向传播当前Q网络与目标Q网络的差值来优化参数。

奖励值的计算方法可参照公式(2)。假设节点在最佳性能时处理一个服务消耗的时间为 t_0 ,则基本时间 t_{basic} 是所有已部署服务 t_0 的简单求和,如公式(3)所示:

$$t_{\text{basic}} = i \times t_0 \quad (3)$$

公式(3)中, i 表示已部署的服务数。对于实际时间,节点处理服务的时间 $proc$ 与1.1节作出的假设有关。假设在一个节点上部署了3个服务,且CPU空闲率为80%,则每个服务的处理时间均为 $\frac{kt_0}{80\%}$,其中 k 为比例系数。在上述假设的前提下,我们可以直接令 $k=1$ 。因此,公式(2)可以写为公式(4):

$$R = \frac{i \times t_0}{\sum_{e \in E} \sum_{u \in U_e} \left(\frac{t_0}{cpu_free} + delay_{u,e} \right)} \quad (4)$$

神经网络中更新Q值的方式如公式(5)所示:

$$y_j = \begin{cases} R_j & \text{到达终止状态} \\ R_j + \gamma \max_a Q'(\phi(S'_j), A'_j, w) & \text{未到终止状态} \end{cases} \quad (5)$$

公式(5)中, R_j 代表当前奖励值。

γ 为衰减因子($0 \leq \gamma \leq 1$),表示后续奖励值对当前Q值的影响。 Q' 是目标Q网络, $\phi(S'_j)$ 表示下一状态的特征向量, A'_j 表示下一步动作, w 为Q网络中的状态价值函数。

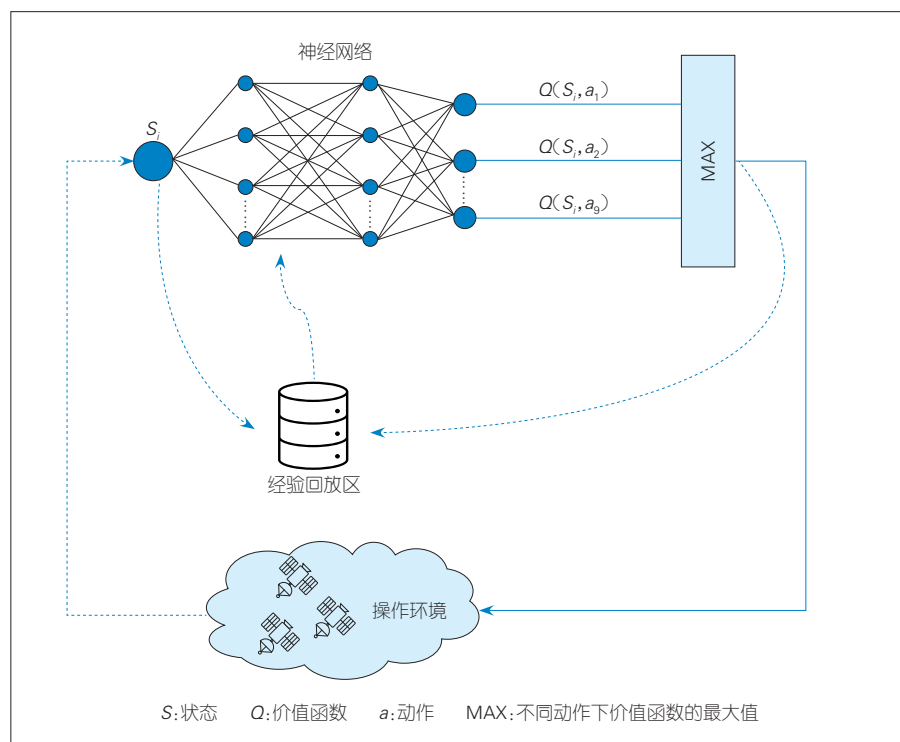
2 实验仿真与结果分析

2.1 实验环境及参数

实验中,我们假定边缘节点数量为9个,用户(服务)数量 n 为20~50个,服务的最短执行时间 t_0 为60 s。为了简化问题,我们假设每个服务都会消耗节点10%的CPU。同时,节点CPU空闲率的下限为10%,即一个节点最多可以同时为9个用户提供服务。如果部署服务多于9个就需要排队等候。显然,在一个节点部署过多服务,不仅会导致每个服务的计算时间变长,还会使需要等待的节点产生更多不必要的等待时延。在上文假设的服务数量下,这显然不是最优策略。强化学习过程中的随机选择动作会导致这些策略被执行和学习,因此,我们要在算法中避免这种情况的发生,即如果采取某个动作后会进入需要排队的状态,就令这一动作无效且下一状态仍为原状态,同时给这次动作一个很低的奖励值,以避免再次作出同样的选择。

用户与卫星的时延是一个难以准确评估的参数。本文1.1节已经指出,时延可代表用户与卫星的物理距离。为了在仿真中模拟现实情况,我们需要对其进行适当放大。经过调试,我们认为,时延分布在1~20 s之间是比较合理的。

此外,本文同时设计了随机部署算法、最短时延贪心算法、均匀部署算法3个参考算法^[8]。我们分析了在不同服务数量条件下4个算法的性能。为了控制无关变量,这3个参考



▲图1 服务部署的深度Q网络算法模型

算法中每一个节点部署服务的数量均不会超过9个,且满足如下条件:

(1)对于随机部署算法,每次部署随机选择节点;

(2)对于最短时延贪心算法,每次部署选择时延最小的节点;

(3)对于均匀部署算法,将服务平均部署到节点中。

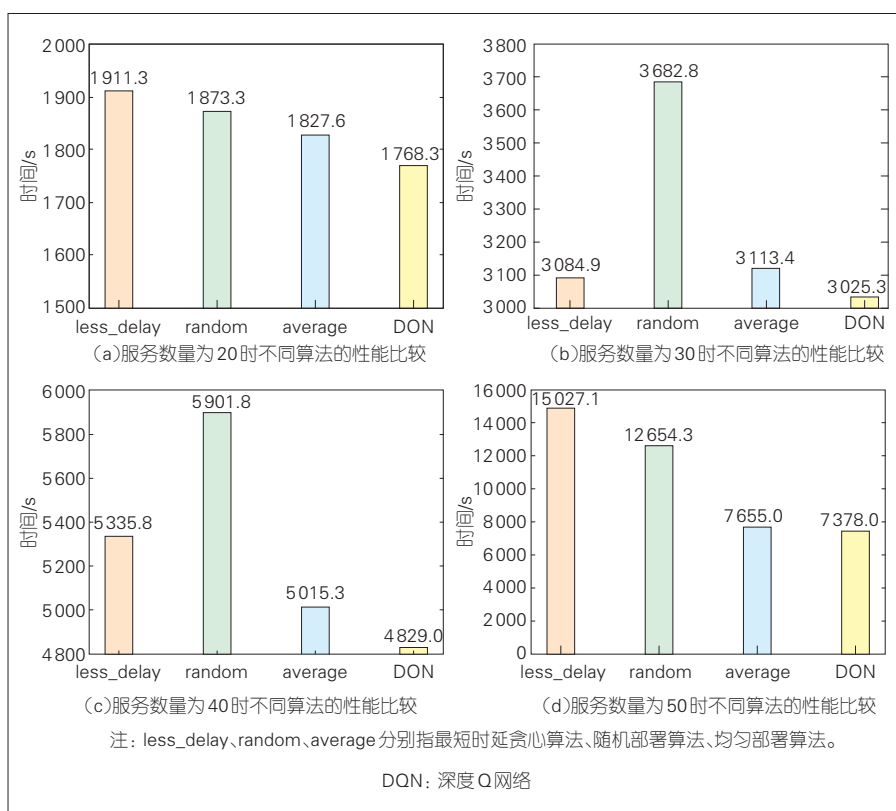
2.2 结果分析

我们选择服务数量 n 分别为20、30、40和50,并进行测试比较。得到的柱状图结果如图2所示。其中,纵坐标表示每种算法处理时延与传输时延之和。为了直观地显示不同情况的算法结果,我们对纵坐标的范围进行适当调整。图3是将柱状图绘制成折线图的结果。

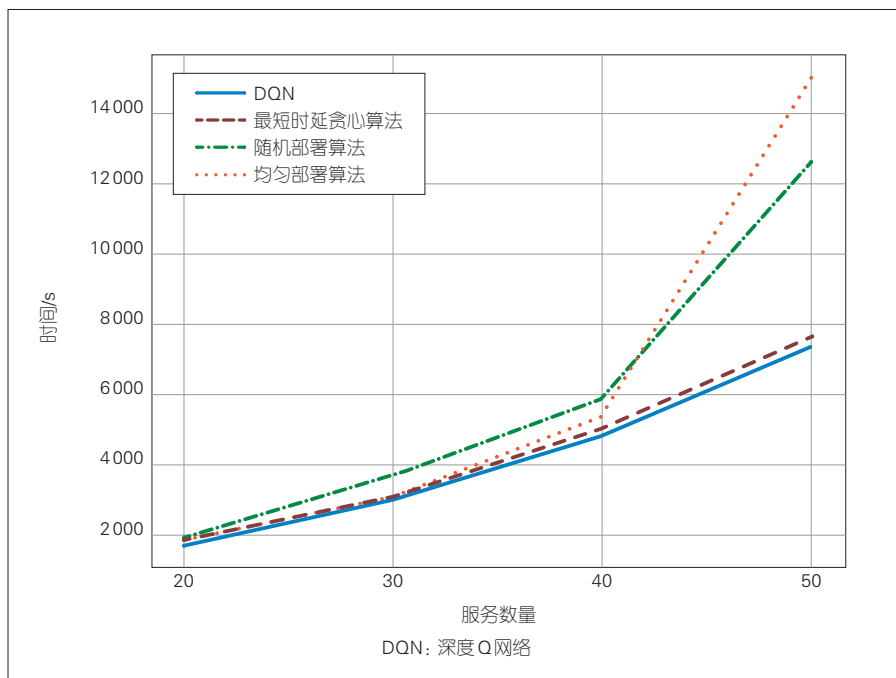
由图2和图3可知,在不同服务数量的情况下,DQN算法的性能均优于另外3种算法。由于对问题作出的一系列假设使最优部署方案接近于均匀部署,因此仿真中的平均部署算法性能与DQN较为接近。在实际问题中,服务对CPU的影响没有那么剧烈,平均部署算法与DQN的真实差距要大于仿真中的差距。此外,在算法设计中,时延对结果的影响小于节点计算能力对结果的影响。因此,基于时延的贪婪算法的性能并不出色,甚至在某些情况下要比随机算法性能更低。

3 结束语

本文中,我们围绕边缘计算使能星地协同网络中的服务部署问题展开研究,将服务部署问题建模为MDP过程,用DQN算法对模型进行求解,并提出详细的算法步骤。我们通过设定基本参数,对算法进行仿真,并将DQN算法与随机部署算法、时延优先贪婪算法、平均部署算法这3个参



▲图2 不同服务数量下的的算法性能比较



▲图3 算法性能比较折线图

考算法进行性能比较,发现DQN算法是解决边缘计算服务部署问题的一种有效算法。

参考文献

- [1] ZHAO J J, XU C Z, MENG T H. Big data-driven residents' travel mode choice: a research overview [J]. ZTE communications, 2019, 17 (3): 9-14. DOI: 10.12142/ZTECOM.201903003

- [2] 丁春涛, 曹建农, 杨磊, 等. 边缘计算综述: 应用、现状及挑战 [J]. 中兴通讯技术, 2019, 25(3): 2-7. DOI: 10.12142/ZTETJ.201903001
- [3] 秦永彬, 韩蒙, 杨清亮. 边缘计算中数据驱动的智能应用: 前景与挑战 [J]. 中兴通讯技术, 2019, 25(3): 68-76. DOI: 10.12142/ZTETJ.201903010
- [4] BADRI H, BAHREINI T, GROSU D, et al. A sample average approximation-based parallel algorithm for application placement in edge computing systems [C]//2018 IEEE International Conference on Cloud Engineering (IC2E). Orlando, USA: IEEE, 2018:198-203
- [5] CHENG Z X, LI P, WANG J B, et al. Just-in-time code offloading for wearable computing [J]. IEEE transactions on emerging topics in computing, 2015, 3(1): 74-83
- [6] WANG S Q, URGANKAR R, ZAFER M, et al. Dynamic service migration in mobile edge computing based on Markov decision process [EB/OL]. [2021-04-06]. <https://arxiv.org/abs/1506.05261>
- [7] ZHAI Y L, BAO T H, ZHU L H, et al. Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling [J]. IEEE wireless communications, 2020, 27(1): 84-91. DOI: 10.1109/MWC.001.1900298
- [8] 严蔚敏, 吴伟民. 数据结构: C语言版 [M]. 北京: 清华大学出版社, 1997

作者简介



卢华, 广东省新一代通信与网络创新研究院网络技术创新中心主任; 研究方向包括 5G 核心网、边缘计算、新型网络架构、软件定义网络、P4 可编程、虚拟化等。



段雪飞, 广东省新一代通信与网络创新研究院网络技术创新中心核心网部门负责人; 研究方向包括 5G 核心网架构、6G 网络架构、空天一体化通信系统等。



李斌, 中兴通讯股份有限公司系统架构师; 主要从事 IP 网络相关技术的研究; 曾获国家科学技术进步奖二等奖。