

基于 DDPG 的边缘计算任务卸载决策研究

张轩 胡晓辉

兰州交通大学 电子与信息工程学院

DOI:10.12238/acair.v2i3.8637

[摘要] 随着通信技术的不断发展,海量的移动应用数据给原有的服务设备带来了巨大的挑战。边缘计算(Mobile Edge Computing, MEC)作为新的计算范式能够有效应对这一挑战。本文针对MEC任务计算卸载问题,建立了一种多用户多服务器的MEC环境,考虑了一种任务可分的情况,可将任务部分卸载到MEC服务器上,同时考虑了任务的时延和能耗,将最小化时延和能耗的加权和作为优化目标,提出了一种基于DDPG动态卸载算法。一系列仿真结果表明,该算法与基线相比,本研究提出的算法能够有效的降低时延和能耗。

[关键词] 边缘计算; 任务卸载; DDPG; 部分卸载

中图分类号: G623.58 **文献标识码:** A

Research on Decision-making of Offloading Edge Computing Tasks Based on DDPG

Xuan Zhang Xiaohui Hu

College of Electronic and Information Engineering, Lanzhou Jiaotong University

[Abstract] With the continuous development of communication technology, the massive amount of mobile application data poses great challenges to existing service devices. Edge computing, as a new computing paradigm, can effectively deal with this challenge. This paper focuses on the problem of task offloading in edge computing, and establishes a multi-user multi-server MEC environment, considering the case of task divisibility and taking into account both task delay and energy consumption. The weighted sum of minimizing delay and energy consumption is proposed as the optimization objective, and a task offloading algorithm based on DDPG is proposed. A series of simulation results show that compared with The baseline algorithm, the algorithm proposed in this study can effectively reduce both delay and energy consumption.

[Key words] Edge computing; task offloading; DDPG; partial offloading

1 绪论

随着移动互联网时代的来临,移动数据流量迅猛增长,传统的云计算模式面临着严峻的延迟和网络拥塞问题。边缘计算的兴起,通过计算卸载技术,优化任务执行的时延和能耗。

2023年,刘耿旗等^[1]人提出了一种基于蛙跳算法的任务卸载模型。考虑了多边缘多设备的环境,以时延和能耗的加权和作为优化目标,优化卸载决策。2024年,周天清等^[2]人考虑了多蜂窝无线回程网络的MEC环境。以最小化终端的平均时延为目标,考虑了基站和缓存模式的选择,提出了一种基于粒子群算法的算法。Zhu等^[3]人考虑一种联合计算和缓存框架,旨在降低用户的延迟和能耗。提出了一种基于深度确定性策略梯度的强化学习算法,实现了用户的任务卸载和任务缓存决策研究。

尽管已有研究提出多种卸载策略,但在实际操作中,任务的计算卸载决策仍面临复杂且动态变化的网络环境,资源利用率低下等难题。为此,本文提出了一种新的任务卸载决策方法,它基于深度确

定性策略梯度(Deep Deterministic Policy Gradient, DDPG)^[4]算法,以解决任务卸载问题。

2 基于DDPG的任务卸载决策研究

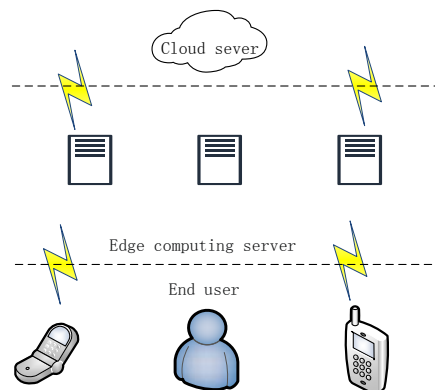


图1 MEC系统模型

针对多用户环境下的任务计算卸载问题, 本文构建了一个多用户多服务器的MEC环境模型。在该框架下, 任务可被拆分配给MEC服务器。其中包括了N个终端用户, M个MEC服务器和一个云计算服务器。计算任务为可拆任务, 允许部分卸载至边缘服务器上执行。系统模型如图1所示。

2.1时延模型

在多用户多服务器的MEC环境中, 时延对终端用户体验感极其重要。由于处理结果通常较小, 因此本文考虑返回时延为0。本文构建了包含任务上传、计算处理两个部分的时延模型。

处理时延主要由终端用户的处理时延 T_u^d 、MEC服务器处理时延 T_m^d 和云服务器的处理时延 T_c^d 构成。终端用户的处理时延

$$T_u^d = \frac{(1 - a_u) \cdot Q \cdot c}{p_u} \quad (1)$$

其中 Q 、 a_u 、 c 、 p_u 分别表示任务的大小, 终端用户卸载到边缘服务器的卸载率, 单位任务所需要的计算频次和终端用户的计算频率。

MEC服务器处理时延 T_m^d 为:

$$T_m^d = \frac{(1 - a_m) a_u \cdot Q \cdot c}{p_m} \quad (2)$$

其中 a_m 和 p_m 分别表示边缘服务器卸载到云服务中心的卸载率和MEC服务器的计算频率。

云服务器处理时延 $T_c^d a(t)$ 为:

$$T_c^d = \frac{a_u a_m \cdot Q \cdot c}{p_c} \quad (3)$$

其中 p_c 表示云服务器的计算频率。

上传时延由任务数据大小、上传速率共同决定, 受无线信道状态及用户与服务设备距离影响显著。本研究使用正交频分多址技术因此数据的从终端设备传输到边缘设备的传输速度 v_u 和边缘设备传输到云计算平台的传输速度可以表示为:

$$\begin{aligned} v_u &= B_u \cdot \log_2(1 + \rho_u \cdot \lambda_u \cdot \sigma^{-2}) \\ v_m &= B_m \cdot \log_2(1 + \rho_m \cdot \lambda_m \cdot \sigma^{-2}) \end{aligned} \quad (4)$$

其中 B_u 、 ρ_u 和 λ_u 分别表示终端用户的上行带宽、上行功率和信道增益。 B_m 、 ρ_m 和 λ_m 分别表示边缘服务器的上行带

宽、上行功率和信道增益, σ 表示高斯噪声。

在本文的MEC环境中, 上传时延主要由两个部分构成, 终端用户到MEC服务器的上传时延 T_u^{tra} 和边缘服务器上传到云服务器的上传时延 T_m^{tra} 。终端用户到MEC服务器的上传时延 T_u^{tra} 可以表示为:

$$T_u^{tra} = \frac{a_u \cdot Q}{v_u} \quad (5)$$

边缘服务器上传到云服务器的上传时延 T_m^{tra} 可以表示为

$$T_m^{tra} = \frac{a_u a_m Q}{v_m} \quad (6)$$

本文考虑终端用户和边缘设备以及云服务器, 能够同时对自己分配到的任务进行处理。因此系统的时延如公式(1)所示。

$$T = \max\{T_u^d, T_u^{tra} + T_m^d, T_u^{tra} + T_m^{tra} + T_c^d\} \quad (7)$$

2.2能耗模型

MEC环境中各物理设备的能量消耗包括无线数据传输过程中的能耗 E^{tra} 、处理任务时的能耗 E^p 构成。

无线数据传输过程中的能耗 E^{tra} 由终端用户上传到边缘设备的能耗 E_u^{tra} 和边缘设备上传到云服务器的能耗 E_m^{tra} 构成。可以表示为:

$$E^{tra} = E_u^{tra} + E_m^{tra} \quad (8)$$

其中 $E_u^{tra} = \rho_u T_u^{tra}$, $E_m^{tra} = \rho_m T_m^{tra}$ 。

处理任务时的能耗 E^d 由终端用户的处理能耗、边缘服务器的处理能耗和云服务器的处理能耗构成。可以表示为:

$$E^d = E_u^d + E_m^d + E_c^d \quad (9)$$

其中 $E_u^d = p_u^d \cdot T_u^d$, $E_m^d = p_m^d \cdot T_m^d$ 和 $E_c^d = p_c^d \cdot T_c^d$ 。

3 基于DDPG算法任务卸载决策算法

3.1优化目标

本文将能耗和时延的加权和作为本文的优化目标, 如公式10所示。

$$\begin{aligned} \min & \alpha T + \beta E^p \\ \text{s.t.} & 0 \leq a_u \leq 1 \\ & 0 \leq a_m \leq 1 \end{aligned} \quad (10)$$

根据第二小节所提出的系统模型, 本文使用了一种基于DDPG的任务卸载决策算法, 解决本文提出的优化问题。

3.2 基于DDPG的任务卸载决策算法

首先本文要将问题转换为马尔可夫决策过程。马尔可夫决策过程由状态空间 $s(t)$ 、动作空间 $a(t)$ 和奖励 $r(t)$ 构成。

状态空间: 状态空间由每个时间片内终端用户产生的任务量 $Q_{i,j}(t)$ 构成。因此状态空间可以表示为:

$$s(t) = \{Q_{i,j}(t)\} \quad (11)$$

算法1: DDPG算法流程

算法1: DDPG 算法流程

```
1: 初始化 Actor 神经网络参数 $\theta^\mu$  和 Critic 神经网络参数 $\theta^Q$ ;
2: 初始化经验回放缓冲区: 初始化一个大小为 N 的经验回放缓冲区, 用于存储过去的经验元组 $(s(t), a(t), r(t), s(t+1))$ ;
3: for episode = 1, M do:
4:   初始化环境状态 $s(0)$ ;
5:   for t = 1, T do:
6:     根据当前策略 Actor 选择动作 $a_t = \mu(s_t | \theta^\mu) + \mathcal{N}$ ;
7:     执行动作 $a(t)$ , 观察奖励 $r(t)$  和新状态 $s(t+1)$ ;
8:     将经验元组 $(s(t), a(t), r(t), s(t+1))$  存储到经验回放缓冲区中;
9:     从经验回放缓冲区中随机采样一个批次的经验元组 $(s(t), a(t), r(t), s(t+1))$ ;
10:    计算目标 Q 值;
11:     $y_i = r(i+1) + \gamma Q'(s(i+1))$ ;
12:     $\mu'(s(i+1) | \theta^{\mu'}) | \theta^{Q'}$ ;
13: 通过最小化损失函数 $loss = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i | \theta^Q))^2$  更新 Critic 网络参数 $\theta^Q$ ;
14: 更新 Actor 网络参数 $\theta^\mu$  通过沿着策略梯度的方向上升 $\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a | \theta^Q) | s = s_i$ 
 $a = \mu(s_i) \nabla_{\theta^\mu} \mu(s | \theta^\mu) |_{s_i}$ ;
15: 更新目标网络参数;
16:  $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$ ;
17:  $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$ ;
18: end for;
19: end for;
```

动作空间: 动作空间由每个时间片内终端用户将任务传输到边缘服务器的卸载率 $a_u(t)$ 和云服务器的卸载率 $a_m(t)$ 和卸载到哪个MEC服务器 $a^s(t)$ 。因此动作空间可以表示为:

$$a(t) = \{a_u(t), a_m(t), a^s(t)\} \quad (12)$$

奖励: 在每个时间片中本文的研究目标是 minimize 延迟 $T(t)$ 和能耗 $E^p(t)$ 的加权和, 因此奖励可以表示为:

$$r(t) = -(\alpha T(t) + \beta E^p(t)) \quad (13)$$

3.3 算法流程

本文设计了一种基于DDPG的动态卸载决策算法。该算法旨在解决多智能体系统中任务卸载问题。算法1为DDPG的算法流程。

3.4 仿真实验与分析

实验模拟环境所需的参数如表1, 网络中参数设置如表2。

表1 实验模拟环境所需的参数

参数名称	参数设置
用户数	20
MEC服务器数量	4
终端用户的上传带宽	20MHz
MEC服务器的上传带宽	40 MHz
终端用户的上传功率	16W
MEC服务器的上传功率	16W
用户产生单个任务最大数据量	1GB
噪声指数	9dB
终端用户的计算频率	4GHz
MEC服务器的计算频率	8GHz
云计算平台的计算频率	16GHz
终端用户处理功率	20W
MEC服务器处理功率	40W
云服务器的处理功率	60W

在构建的多用户多服务器MEC环境中, 仿真实验中采用平均时延、平均能耗和奖励指标, 对比分析了基于改进DDPG算法的性能。

在模型收敛后, DDPG、PPO、A2C、DQN在处理500M任务的时延如图2所示。仿真结果表明, DDPG相较于其他基线算法在时延方面有良好的性能。

在模型收敛后, DDPG、PPO、A2C、DQN在处理500M任务的能耗如图3所示。仿真结果表明, DDPG相较于其他基线算法在能耗方面有良好的性能。

表2 网络参数

参数名称	参数设置
经验池大小	1000000
批次大小T	256
激活函数	Tanh
优化器	Adam
Actor 网络学习率	0.0001
Critic 网络学习率	0.0001
折扣因子	0.99
更新步长	10
隐藏层大小	256
训练轮数M	500

楚看到DDPG算法的奖励明显高于基线算法。

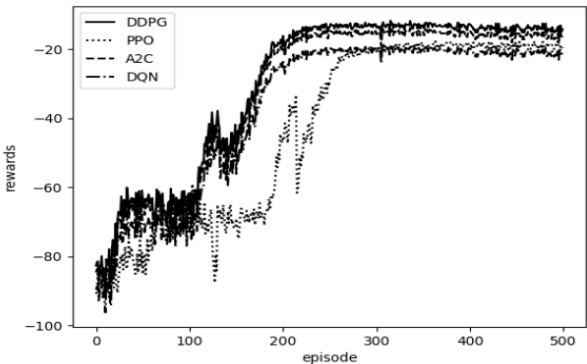


图4 DDPG、PPO、A2C和DQN时延对比图

4 总结

在本研究中,构建了多用户多服务器的MEC环境,对任务的时延和能耗问题进行了详细分析,且将时延与能耗的加权和作为本文研究的核心优化目标,实现了在MEC场景下任务卸载决策。在我们的模型中,考虑了任务可分的实际情况,并通过DDPG算法来解决任务卸载策略优化的问题。

在仿真实验与分析部分,经过模拟实验,我们的研究结果具备说服力。与基线算法相比,改进后的DDPG算法在降低任务执行的时延和能耗方面展现出了显著优势。。

[参考文献]

[1]刘耿旗,张旭秀,马洪源,等.多边缘节点场景下的任务卸载算法[J].信息与控制,2023,52(05):679-688.

[2]周天清,许铭.基于PSO的联合任务卸载与缓存算法研究[J].无线电工程,2024,54(03):517-525.

[3]Zhu X,Jia Z,Pang X,etal.Joint Optimization of Task Caching and Computation Offloading for Multiuser Multitasking in Mobile Edge Computing[J].Electronics,2024,13(2):389.

[4]Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971,2015.

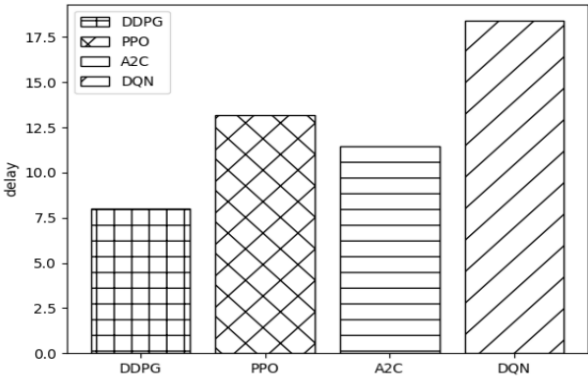


图2 DDPG、PPO、A2C和DQN时延对比图

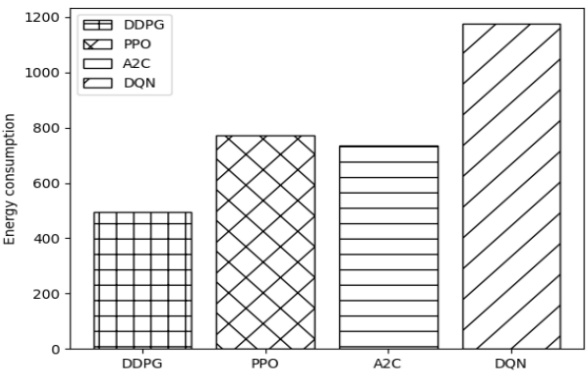


图3 DDPG、PPO、A2C和DQN时延对比图

图4为本文算法和基线算法的奖励对比图。从图中,可以清