Distributed Server Resource Optimization in Vehicular Task Offloading Environments using Deep Reinforcement Learning-based Algorithm

Bang Soo Jeong, Lee Mee Jeong*

Ewha Womans Univ.

tnwjd7732@ewhain.net, *lmj@ewha.ac.kr

차량 오프로딩 환경에서 분산 서버 자원 최적화를 위한 심층 강화학습 기반 알고리즘

방수정, 이미정*

Ewha Womans Univ.

Abstract

Recent advances in artificial intelligence (AI) and augmented reality(AR)-based vehicle applications have significantly increased the computational workload for vehicles. Offloading tasks to remote servers can alleviate resource shortages, but using distributed server systems like edge servers introduces challenges due to limitations in computing and storage resources compared to centralized servers. In this paper, we propose an algorithm for efficient load distribution among edge servers in vehicular task offloading environments, aiming to minimize server overload during sudden increases in traffic or intensive tasks. The algorithm forms clusters among edge servers and internally distributes the workload, addressing the server resource optimization problem using workload metrics. This problem is then reformulated as an Markov Decision Process (MDP) and solved through a Deep Reinforcement Learning(DRL)-based algorithm, providing an approximate solution within a short execution time. Our research demonstrates effective resource management in the offloading environment, enhancing the system's capacity for a greater number of users while ensuring satisfaction with vehicle services.

Ⅰ. Introduction

With the rapid progress of AI and AR technologies, active research is underway to improve the safety and convenience of vehicle passengers.[1] These services necessitate extensive data processing and low latency, posing challenges for in-vehicle resources to handle concurrent computationally intensive tasks, despite recent increases in resources.[2]

Offloading is a technology in which tasks from resource-constrained devices are processed by remote servers, and only the results are returned. Utilizing remote servers, especially edge servers located near users, can significantly reduce computation time and network loads, and this approach has gained popularity recently.[3] However, edge servers, which have limited resources compared to centralized ones, are at risk of overloading in areas with increased vehicle traffic or continuous intensive tasks. Additionally, because vehicles are mobile, nearby server resources can fluctuate rapidly. This study aims to effectively utilize distributed server resources, optimize server resource allocation, and propose a DRL-based algorithm.

Ⅱ. System Model and Problem Formulation

We proposes a cluster structure among edge servers and distributing loads within the cluster. Figure 1 illustrates the proposed system architecture. Referring $M = \{0, 1, \dots, m\}$ to the base stations as edge servers, the 0th server represents the cluster head. Assuming that vehicles offload tasks $Q_i = \{D_{i,} C_i, T_i^{\max}\}$ to the nearest base station, three factors represent the size of the task: the amount of resources needed for processing, and the latency requirements, respectively.

The communication between adjacent base stations is wired, enabling the exchange of status information related to the vehicle's data and available resource levels. In Figure 1, 9 servers form a single cluster, with the central edge server acting as the cluster head and the other servers as members. All servers transmit the task data requested by the vehicle to the cluster head server. The head server uses observable states to determine the operational server responsible for handling the task and the resource allocation ratio. Subsequently, after the decision is made, the task data is sent to the operational server, where it is processed and the results are returned.

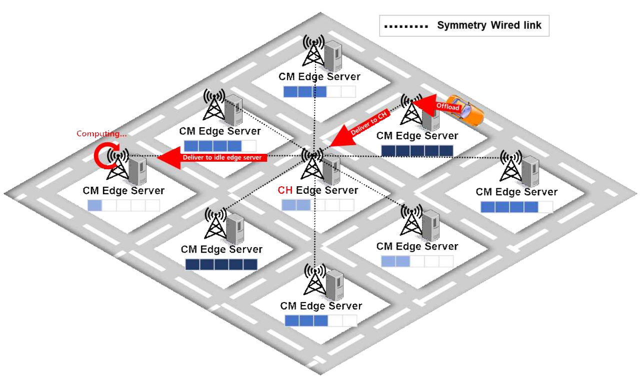

Figure 1. System architecture

A. Communication Model

According to the cluster structure, a single task involves a maximum of two inter-server communications. Assuming unpredictable variations in the wired channel bandwidths, denoted as $b_{m,n}$ between servers m and n at each time interval, the time taken to transmit the task from the server assigned to the vehicle to the cluster head is represented as (1).

$$T_{m,0}^{comm} = D_i / b_{m,0} \tag{1}$$

Subsequently, the time taken by the cluster head to transfer data to the computation server is represented as (2).

$$T_{0,n}^{comm} = D_i / b_{0,n} \tag{2}$$

B. Computation Model

The available resources of edge servers vary based on user density and task arrival rate. Denoting the available resources of server n as F_n, and the proportion of resources allocated by server n to task i as $f_{i,n}$, the computation time is represented as (3).

$$T^{comp} = C_i / F_n f_{i,n} \tag{3}$$

When referring to the server that initially received the task from the vehicle as $a_m \in \{0,1\}, \forall m$, and the computation server as $c_m \in \{0,1\}, \forall m$, according to the communication and computation model, the total processing time T_i within the cluster for the task Q_i can be calculated as shown in (4) where K represents the number of edge servers within the cluster.

$$T_i = \sum_{m=0}^{K} a_m T_{m,0}^{comm} + \sum_{m=0}^{K} c_m T_{0,m}^{comm} + \sum_{m=0}^{K} c_m T^{comp} \tag{4}$$

C. Server Workload Model

Cluster members are required to inform the head of any significant changes in their available resources, which the head will then record and manage. Consequently, the cluster head can calculate the average available resources $\overline{F}$. Accordingly, the standard deviation of the available resources within the cluster σ can be calculated as follows.

$$\overline{F} = \frac{\sum_{m=0}^{K} F_m}{K}, \ \sigma = \sqrt{\frac{\sum_{m=0}^{K} (F_m - \overline{F})^2}{K}} \tag{5}$$

D. Computing Resource Optimization Problem

This study establishes the standard deviation of available resources, as defined earlier, as the objective function for optimization problem P1. The main objective is to efficiently utilize the resources available within the cluster to tackle inter-server load imbalance issues. Additionally, integrating time latency requirements as constraints enables the maintenance of service satisfaction while efficiently managing resources.

Ⅲ. Markov Decision Process (MDP) Formulation

DRL is widely used to optimize solutions in dynamic and unpredictable environments. This paper focuses on addressing the server resource optimization problem by reformulating it as a MDP and applying the Twin Delayed DDPG (TD3) algorithm. The state, action, and reward functions are summarized as follows.

$$\begin{aligned} &P1: \min_{c_m, f_{i,m}} \sigma \\ &s.t. \\ &\quad a_m, c_m \in \{0,1\}, \forall m \\ &\quad \sum_{m=0}^{K} a_m = 1, \ \sum_{m=0}^{K} c_m = 1, \forall m \\ &\quad 0 \le f_{i,m} \le 1, \forall m \\ &\quad T_i \le T_i^{\max} \end{aligned}$$

A. State

The cluster head node acts as the agent in RL, assessing the state to make optimal decisions. The observed state includes: available resources of neighboring servers, wired bandwidth between servers, and task profiles.

B. Action

The agent's decision involves selecting the computation server and determining resource allocation ratios.

C. Reward Function

Failure to meet task latency requirements results in a negative reward of -1. Meeting latency requirements results in a constant reward, and if the cluster's available resource standard deviation decreases as a result of the action, additional rewards are provided. The reward function can be expressed as an equation.

$$R_i = \alpha + \beta \max(0, \sigma_s - \sigma_{s'})$$

σ_s represents the standard deviation in the previous state, and $\sigma_{s'}$ is the standard deviation in the new state transitioned by the action.

Ⅳ. Conclusion

In this paper, we propose a method to address the load imbalance among edge servers in vehicular offloading environments by utilizing a cluster structure to distribute the load regionally and defining the server resource optimization problem. To effectively derive optimal solutions in the rapidly changing vehicular environment, we reformulated the optimization problem as a MDP problem and proposed a learning method based on the TD3 reinforcement learning model. Through this research, we anticipate effectively resolving the load imbalance issues in edge servers for vehicular offloading, ultimately enhancing the system's throughput. This is expected to enhance user satisfaction and have a positive impact on the business's profitability.

REFERENCES

[1] Kuutti, Sampo, et al. "A survey of deep learning applications to autonomous vehicle control." IEEE Transactions on Intelligent Transportation Systems 22.2 (2020): 712-733.

[2] Liu, Jinshi, et al. "RL/DRL meets vehicular task offloading using edge and vehicular cloudlet: A survey." IEEE Internet of Things Journal 9.11 (2022): 8315-8338.

[3] Islam, Akhirul, et al. "A survey on task offloading in multi-access edge computing." Journal of Systems Architecture 118 (2021): 102225.