Optimal Control-based Computing Task Scheduling in Software-Defined Vehicular Edge Networks

Ershuai Peng, Zhiyuan Li
Department of Cybersecurity Jiangsu University Zhenjiang, Jiangsu 212013, China

三等奖

Abstract

MEC(Mobile edge computing) is one of the key technologies to promote the development of 5G. Mobile edge computing server is the small base-station equipped with computation and storage capacity. It allows nearby-cars to access in a wireless way. Be different with cloud computing, Mobile edge computing can meet the strict requirements in network delay of the cars but the limitation computation capacity cannot meet all requirements of the cars. To solve this issue, This paper introduces OCTS(Optimal Control-based Computing Task Scheduling) algorithm. It can develop efficient resource allocation strategies and make full used of these resource. Software-Defined network is introduced to avoid obtaining local optimum and we use adjustable parameters to make the algorithm performance well in different circumstance. Experiments show that this method can effectively improve the utilization of resources and meet the requirements of computation delay and network delay for vehicle's tasks.

Optimal Control-based Computing Task Scheduling algorithm

1. Framework of System

In the user layer, when the vehicles in the area covered by a RSU(Road Side Unit), they can send tasks to the RSU with wireless connections. In the MEC layer. Each MEC server is connected with a RSU. The RSU trans pond these tasks to the MEC servers and then the MEC server calculate these tasks and send them back to the vehicles via the RSU. The MEC servers in different area are connected by wired communication lines in the same LAN (Local Area Network). The MEC servers can send or receive tasks to or from other servers by these lines. In the control layer, the SDN controller connect with these MEC servers by wired communication lines. The SDN controller can not only obtain the information of these servers, such as the CPU utilization, memory utilization, etc but also control the task scheduling among these servers.

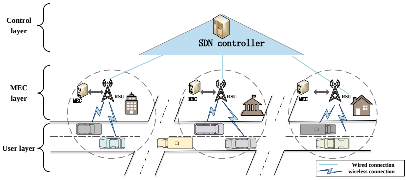

Figure 1. Framework of System

2. System Model

Task scheduling can not only decrease the High load servers' computation delay but also increase Computation resource utilization. However, task scheduling would increase the network delay of the LAN. To calculate the network delay, we assume that the tasks arrive at server i from others servers follow a Poisson distribution. We also assume that the network delay is t when there is no network congestion.

Due to the limited computation capacity of MEC servers and disequilibrium of the tasks' arrival in the aspect of time and space, part of the MEC servers need to process a large number of tasks and at the same time part of servers are processing a small number of tasks. It will lead to excessive computation delay and a waste of computation resources. In this circumstance, we have to scheduling tasks among MEC servers to alleviate above situation.

According to Modern Control Theory, MEC servers' CPU utilization can be denote as state variables. The tasks that from both vehicles and servers cane be denoted as the system's input, which have influence on the state variables.

Computation delay is the result of servers processing tasks. To simplify the computation, we assume that the tasks arrive at server i at time t follow a Poisson distribution and each task would be processed with mt i CPU clock periods.

According to Modern Control Theory, MEC servers' CPU utilization can be denote as state variables. The tasks that from both vehicles and servers cane be denoted as the system's input, which have influence on the state variables.

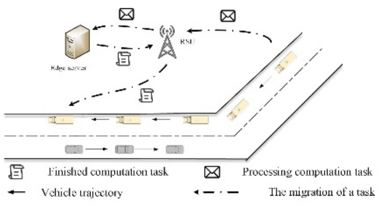

Figure 2. Diagram of MEC server at work

3. Algorithm

Algorithm 1: OCTS

Input: task arrival numbers, CPU usage of all SBSs, Control parameter ξ, $\mathbf{J}(\mathbf{u}(t))$

Output: optimal control $\mathbf{u}^*(t)$, optimal trajectory $\mathbf{x}^*(t)$

1 **for** $t = 0$;$t \le T$ **do**
2 Calculate the expected CPU usage of SBSs $C_e\%$;
3 Calculate the fitted curve $\lambda(t)$ and $\rho^{\lambda}(t)$ based on task arrival numbers;
4 Establish the state vector $\dot{\mathbf{x}}$ based on 3 and 4;
5 Solving the formula (10);
6 get $\mathbf{u}^*(t)$ and $\mathbf{u}^*(t)$;
7 Update the CPU usage of SBSs;
8 **end**
9 return $\mathbf{u}_1^*(t), \cdots, \mathbf{u}_i^*(t), \cdots, \mathbf{u}_T^*(t)$;

Simulation

1. Experimental Description

In this part, systematic simulations are carried out to evaluate the performance of optimal control-based computing task scheduling in Software-Defined Vehicular Edge Networks. We install SUMO and Mini-Net as platform on a computer equipped with i7 CPU and 16GB memory. We assume that in a given region, there are 5 MEC servers connected in a same LAN (Local Area Network).

2. Experimental Result

As shown in figure 3, in the initial state, the CPU utilization of the five servers is about 20%, 20%, 30%, 30% and 80% respectively. The total experiment duration was 100s. The experimental results showed that the CPU utilization of each server did not decrease significantly without optimization. We uses OCTS algorithm in initial states. As is shown in figure 5, The CPU utilization on servers 1 and 4 has increased from about 20% to about 33%, servers 2 and 3 have increased from about 30% to about 35%, and server 5 has decreased from about 80% to about 45%.

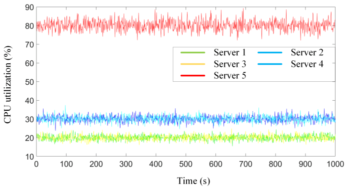

Figure 3. CPU Utilization on The First Initial State

Conclusion

In this paper, we investigate OCTS algorithm to solve unbalanced-load in MEC servers in Software-Defined Vehicular Edge Networks. This paper consider that task scheduling will cause additional network delay and that in different circumstances users may have the different requirements. This paper introduce a parameter x to adjust network delay caused by task scheduling and meet the different requirements of the users. Simulation results show that the optimal control-based computing task scheduling method can effectively reduce the imbalance-load caused by the arrival of user tasks and it effectively reduce the load of high-load servers and improve the resource utilization of idle servers.

References

[1] D. Linthicum, "Responsive data architecture for the internet of things," Computer, vol. 49, no. 10, pp. 72–75, 2016.
[2] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," Business Horizons, vol. 58, no. 4, pp. 431–440, 2015.
[3] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Communications Surveys Tutorials, vol. 19, no. 4, pp. 2322–2358, 2017.
[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," Internet of Things Journal, IEEE, vol. 3, no. 5, pp.637–646, 2016.
[5] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—a key technology towards 5g," ETSI white paper, vol. 11, no. 11, pp. 1–16, 2015
等等

Contacts

Linkman：Ershuai Peng
Mobile Phone：17851007351
Email: 2221808035@stmail.ujs.edu.cn