Efficient Multi-user Computation Scheduling Strategy Based on Clustering for Mobile-Edge Computing

Qing-Yan Lin, Guang-Shun Li, Jun-Hua Wu(✉), Ying Zhang, and JiaHe Yan

School of Information Science and Engineering, Qufu Normal University, Rizhao 276800, China
shdwjh@163.com

Abstract. The Mobile Edge Computing (MEC) is a new paradigm that can meet the growing computing needs of mobile applications. Terminal devices can transfer tasks to MEC servers nearby to improve the quality of computing. In this paper, we investigate the multi-user computation offloading problem for mobile-edge computing. We study two different computation models, local computing and edge computing. First, we drive the expressions for time delay and energy consumption for local and edge computing. Then, we propose a server partitioning algorithm based on clustering. We propose a task scheduling and offloading algorithm in a multi-users MEC system. We formulate the tasks offloading decision problem as a multi-user game, which always has a Nash equilibrium. Our proposed algorithms are finally verified by numerical results, which show that the scheduling strategy based on clustering can significantly reduce the energy consumption and overhead.

Keywords: Mobile edge computing · Offloading decision · Node clustering · Optimal strategy · Nash equilibrium

1 Introduction

The growing popularity of mobile devices has revolutionized mobile applications [1]. As function of cloud computing increasingly move to the edge of the network, a new trend of computing has emerged. It is estimated that many edge computing devices will be deployed on the edge of the network [2, 3]. The network edge has enough capacity to support mission-critical and latency-critical tasks on mobile devices [4]. This mode is called Mobile Edge Computing (MEC). This MEC paradigm can provide low latency, high bandwidth and computing agility in the computation offloading process [5]. It can support applications and services with reduced latency and improved QoS [6, 7]. This article is organized as follows. In Sect. 2, we introduce the related work. In Sect. 3, we describe the system model of MEC. In Sect. 4, we proposed a clustering algorithm

X. Zhang et al. (Eds.): CloudComp 2019/SmartGift 2019, LNICST 322, pp. 266–274, 2020.
https://doi.org/10.1007/978-3-030-48513-9_22

for mobile device. Section 5 introduces the task scheduling and offloading algorithm based on game theory. Section 6 shows the experiment evaluation. Finally, we will draw conclusions in Sect. 7.

2 Related Work

Wang and Yuan et al. [8] described the placement problem of energy-aware edge server as a multi-objective optimization problem, and designed a placement algorithm based on particle swarm optimization for energy-aware edge servers to find the optimal solution. Zeng et al. [9] expressed the task scheduling problem as a mixed integer nonlinear programming problem and solved its high computational complexity, and proposed a computationally effective solution. Song and Gao et al. [10] used cloud atomization technology to transform physical nodes into virtual machine nodes. Adila Mebrek et al. [11] used energy and quality of service (QoS) as two important indicators of fog performance. Chen et al. [12] describe multi-user multi-task unloading as NP-hard problem, and use the separable semi-deterministic relaxation problem to obtain the lower bound of the system overhead, and obtain the optimal performance of the system under multi-parameter conditions. Qi et al. [13] consider the user's job size, service invocation time, and service quality level, a set of experiments are designed, deployed, and tested to validate the feasibility of our proposal in terms of cost optimization. Chen and Liang et al. [14] consider a general multi-user mobile cloud computing system, the mobile users share the communication resource while offloading tasks to the cloud. Zhu et al. [15] formulated a minimum energy consumption problem in deadline-aware MEC system.

3 The Description of MEC

Mobile Edge Computing (MEC) can be defined as an implementation of edge computing that introduces compute and storage capacity to the edge of the radio access network, reducing latency by dropping the cloud and service platform to the edge of the network [16]. In the mobile edge computing system, the edge server is connected to the core network through a wired link, and the edge server can directly be a mobile device [17]. Most of the computing tasks can be processed at the edge node without entering the cloud core network, so the computing and communication load of the cloud core network can be significantly reduced. Redundant computing resources at the edge of the network can also be fully utilized, and mobile devices and IoT devices can achieve lower computational task completion delays [18] (Fig. 1).

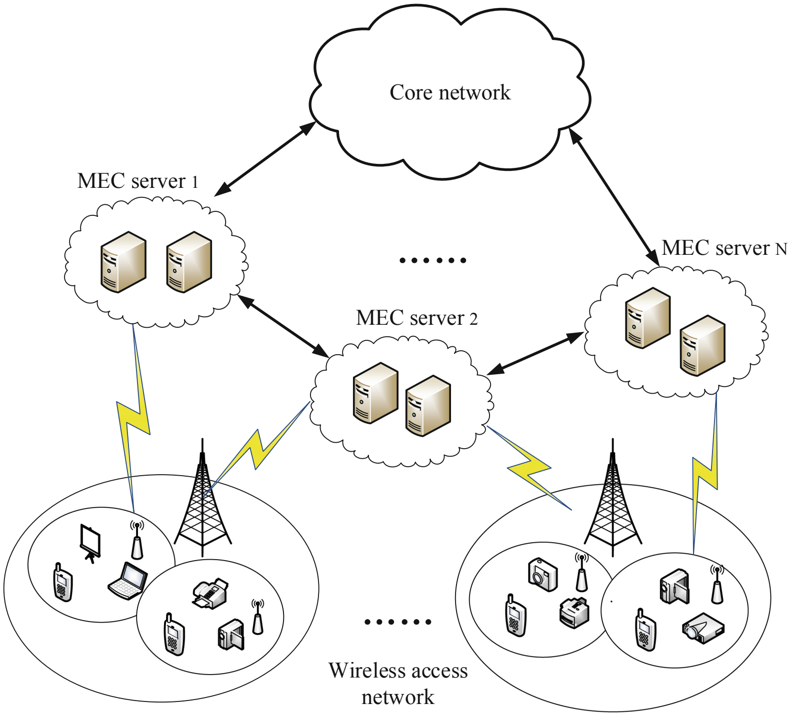

Fig. 1. MEC deployment architecture diagram

4 Graph-Based Energy-Clustering Algorithm

Now, we introduce the system model of mobile edge computing system. We assume that there are some mobile devices, in which each device has a task to complete. Each device can connect directly to its neighbors.

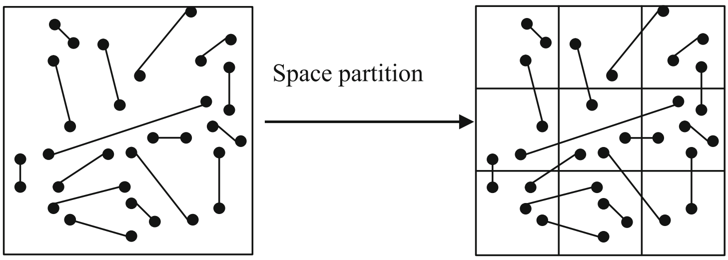

Fig. 2. The graph of nodes in the area

In this paper, the network topology diagram composed of the physical nodes of the edge computing is abstracted into an undirected weighted connected graph From a network system standpoint we consider a mobile device deployment as presented in

Fig. 2. The first graph represents the region where the node is located, the second graph represents the discretized distribution of nodes after space partition. Now, we will use the node set as the starting point for node partitioning.

The Fig. 3 represents the interactions between the nodes the area.

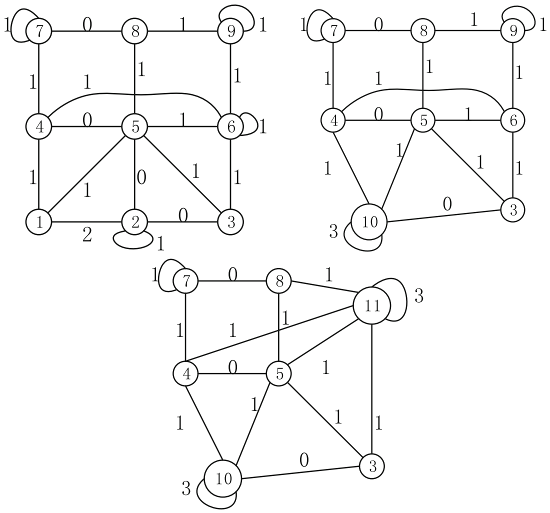

Fig. 3. The interaction graph $G_{\text{int}} = (C, E_{\text{int}})$

- (1) First, we look for the edge with the highest weight and get the amount of resources for each node, and the highest weight between node i and node j is lower than the cluster maximum weight $M_{\max}$, the sum of resource of node i and node j is also lower than the cluster maximum resource $res_{\max}$.
- (2) Then, we cluster node i and node j, updating the graphs G_a and G_{int}. The weight of the link between the new node and its neighbor is determined by the sum of the link weights between the former node i and its neighbors and the former node j and its neighbors.

5 System Model and Problem Describe

5.1 System Model

Local Computing Model

The communication delay mainly consists of transmission delay T_m^n of tasks transmitted through wireless network and execution time T_i^m of computing tasks at nodes. When task is executed at local layer, the execution time T_i^m is expressed as

$$T_i^m = \frac{K_i}{F_i} \tag{1}$$

The energy consumption of task i at the local device is

$$E_i^m = c_n T_i^m \tag{2}$$

c_n is the CPU power of local mobile device, So the transmission delay can expressed as

$$T_m^n = \frac{K_i}{w\log(1+\frac{p_m g_{mn}}{\omega_0})} \tag{3}$$

When the task want to be transmitted to other device for execution, the device chooses a channel, and the energy consumption during transmission is as follows $E_{trans}^{mn} = r_m^n \times K_i$, the total energy consumption is

$$E_i^{mn} = E_{trans} + E_i^n \tag{4}$$

Among them, E_{trans} is the energy consumed by the transmission process, and E_i^n is the energy consumed by other device to execute the task. Therefore, the total overhead is

$$\begin{aligned} &\min \sum_{i=1}^{n}\left[\sum_{j=1}^{n}\alpha_l\left[E_i^m x_{ij} + E_i^{mn}(1-x_{ij})\right] + \lambda_l\left[T_i^m x_{ij} + T_i^n(1-x_{ij})\right]\right] \\ &s.t \quad i \in (1,n),\ j \in (1,n) \\ &\qquad T_i^m + T_i^n \le L_i \end{aligned} \tag{5}$$

α_l, λ_l are two weighting factors which indicate the weights of time consumption and consumption in the decision-making process respectively.

MEC Computing Model

$$\varphi_{ij} = \begin{cases} 0 \text{ Task excute at local mobile device} \\ 1 \text{ Task excute at MEC device} \end{cases} \tag{6}$$

When $\varphi_{ij} = 1$, tasks needs to be uninstalled to the MEC device for execution, energy consumption E_n^{edge} can be divided into transmission energy consumption $E_{n,trans}^c$ and calculation energy consumption $E_{n,com}^c$.

$$E_n^{edge} = E_{n,trans}^{edge} + E_{n,com}^{edge} \tag{7}$$

$$E_{n,trans}^{edge} = P_i \cdot T_{i,trans}^e, \forall i \in N \tag{8}$$

P is the unit energy consumption when the mobile device accesses the channel, $E_{n,com}^c$ is the energy consumed by task execution at the MEC layer.

$$E_{n,com}^{edge} = T_{i,com}^e \cdot \varepsilon_e \tag{9}$$

The communication delay mainly consists of transmission delay $T_{i,trans}^e$ and execution time $T_{i,com}^e$ of computing tasks.

$$T_{i,trans}^e = \frac{K_i}{r_n} \tag{10}$$

$$T_{i,com}^{e} = \frac{K_i}{F_c} \tag{11}$$

Therefore, the total overhead of MEC is

$$\min \sum_{i=1}^{n} \alpha_c \left[E_{n,trans}^{edge} + E_{n,com}^{edge} \right] + \lambda_c \left[T_{n,trans}^{e} + T_{n,com}^{e} \right] \tag{12}$$

6 Evaluation and Analysis

In this section, we use the MATLAB software simulation method to evaluate the performance of the proposed unloading strategy. In this simulation, the parameters of the task are set as, The calculated value of the task is, the time delay constraint; The computing power of mobile devices is. The computing power of MEC devices is. The parameters of the Wi-Fi wireless channel are set as (Fig. 4):

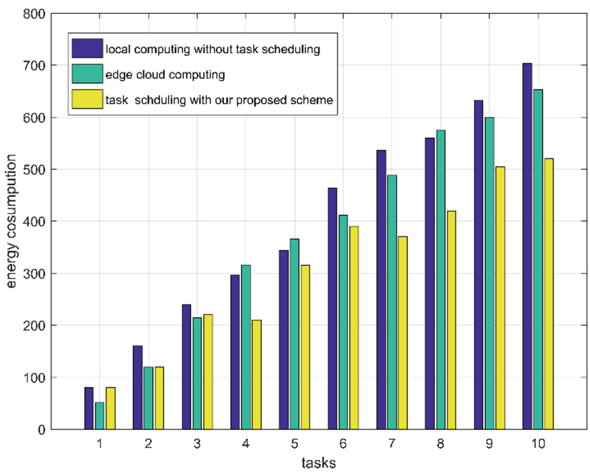

Fig. 4. The consumption of task owner

The experimental results show that when the amount of tasks increases, the local computing task scheduling of our proposed scheme has less energy consumption than local computing without task scheduling and all tasks to edge cloud computing. We can see that when $I \leq 3$ h, the energy consumption between our proposed solution and all tasks of the edge layer calculation is not much different. When $I > 3$, due to the increased workload, all transmission to the edge layer will consume a large amount of transmission energy, while the local device is idle. At this time, the game theory-based scheduling scheme is better.

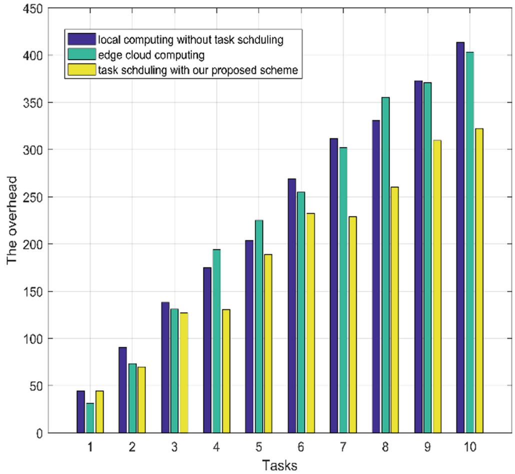

Fig. 5. The overhead for mobile device

From Fig. 5, we can see that the local computing without task scheduling has the largest overhead. When the number of tasks is small, there is a little difference of the total overhead between our schemes and the edge cloud computing. When the number of tasks is low, the total cost of the two solutions is basically the same. As the number of tasks increases, unloading all tasks to the edge cloud creates a large delay. This leads to high latency, and the energy consumption of uploading tasks to the edge layer is also increasing at the same time. Our schemes has higher performance.

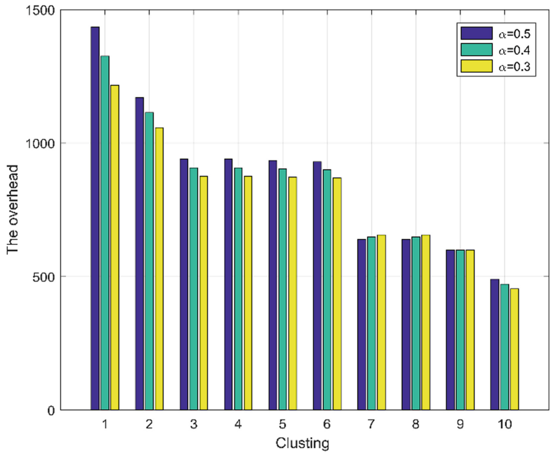

Fig. 6. The overhead of cluster

We select different values for the weight parameters α. The Fig. 6 shows that when α is gradually larger, the proportion of energy consumption in total overhead increases, and in our method, it is more suitable for calculating tasks with low delay requirements.

7 Conclusion

This paper analyzes the task scheduling problem based on self-organized edge computing, and proposes a graph-based server region clustering algorithm. The game scheduling based task scheduling mechanism mainly considers energy consumption and aims to minimize mobile devices. In this paper, the communication model and the computational model are proposed firstly. The graph-based server clustering algorithm is proposed, and the algorithm is applied to the edge computing. Then the task unloading problem is analyzed, which effectively reduces the edge computing task scheduling energy consumption.

References

1. Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. **34**, 3590–3605 (2016)
2. Wang, S., et al.: A survey on mobile edge networks: convergence of computing, caching and communications. IEEE Access **5**, 6757–6779 (2017)
3. Jin, L., Li, S., Yu, J.G., He, J.B.: Robot manipulator control using neural networks: a survey. Neurocomputing **285**, 23–34 (2018)
4. Shahzadi, S., Iqbal, M., Dagiuklas, T., Qayyum, Z.U.: Multi-access edge computing: open issues, challenges and future perspective. J. Cloud Comput. **6**(1), 30 (2017)
5. Zhang, K., et al.: Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. IEEE Access **4**, 5896–5907 (2016)
6. Corcoran, P., Datta, S.K.: Mobile-edge computing and the internet of things for consumers: extending cloud computing and services to the edge of the network. IEEE Consum. Electron. Mag. **5**(4), 73–74 (2016)
7. Qi, L.Y., et al.: A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Futur. Gener. Comp. Syst. **88**, 636–643 (2018)
8. Li, Y., Wang, S.: An energy-aware edge server placement algorithm in mobile edge computing. In: 2018 IEEE International Conference on Edge Computing (EDGE), San Francisco, pp. 66–73 (2018)
9. Zeng, D., Gu, L., Guo, S., Cheng, Z., Yu, S.: Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans. Comput. **65**(12), 3702–3712 (2016)
10. Song, N., Gong, C., Xingshuo, A.N., Zhan, Q.: Fog computing dynamic load balancing mechanism based on graph repartitioning. China Commun. **13**(3), 156–164 (2016)
11. Mebrek, A., Merghem-Boulahia, L., Esseghir, M.: Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing. In: IEEE International Symposium on Network Computing and Applications, pp. 1–4 (2017)
12. Chen, X., Jiao, L., Li, W., Fu, X.: Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans. Networking **24**(5), 2795–2808 (2016)

13. Qi, L.Y., Yu, J.G., Zhou, Z.L.: An invocation cost optimization method for web services in cloud environment. Sci. Program. **2017**, 9 (2017)
14. Chen, M.H., Liang, B., Min, D.: Joint offloading decision and resource allocation for multi-user multi-task mobile cloud. In: IEEE International Conference on Communications, Kuala Lumpur, pp. 1–6 (2016)
15. Zhu, T., Shi, T., Li, J., Cai, Z., Zhou, X.: Task scheduling in deadline-aware mobile edge computing systems. IEEE Internet Things J. **6**(3), 4854–4866 (2019)
16. Deng, R., Lu, R., Lai, C., Luan, T.H., Liang, H.: Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet Things J. **3**(6), 1171–1181 (2016)
17. Pham, Q.V., Leanh, T., Tran, N.H., Hong, C.S.: Decentralized computation offloading and resource allocation in heterogeneous networks with mobile edge computing (2018)
18. Zheng, J., Cai, Y., Yuan, W., Shen, X.S.: Stochastic computation offloading game for mobile cloud computing. In: IEEE/CIC International Conference on Communications in China, Chengdu, pp. 1–6 (2016)