Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing

Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing

Xiaoyi Tao, Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE,* Heng Qi, *Member, IEEE,* and Keqiu Li, *Member, IEEE*

***Abstract*—In this paper, we investigate an energy efficiency with performance guaranteed problem in mobile-edge computing. The mobile users desire low energy consumption and performance guaranteed, we propose an energy minimizing optimization problem for mobile-edge cloud computing, that we apply KKT conditions to solve it, and we also present a request offloading scheme for this issue. In particular, the offloading scheme is determined by energy consumption and bandwidth capacity at each time slot. Numerical results demonstrate that our proposed offloading scheme outperforms local computing and entirely offloading method on energy consumption and performance on delay.**

***Index Terms*—Mobile-edge computing; task offloading; energy efficient; optimization.**

I. Introduction

WITH the increasing of mobile devices, more and more mobile applications are striving for computing capacity to provide customized service. These applications demand stringent requirements on real-time communication and intensive computation. However, mobiles are resource constrained devices with limited computation ability and battery capacity. To tackle these problems, mobile-edge computing (MEC) is emerging as a promising technique to provide cloud computing service at the mobile edge network [1]. Compared to mobile cloud computing, edge computing has several advantages such as controllable latency and low energy consumption. Offloading data to a remote cloud brings long latency which would hurt application performance. In the meantime, large data transfer would consume battery largely. Accordingly, rather than applying remote cloud, recent researchers prefer to offload computation tasks to a nearby base station cloud. The computation quality is affected by computation offloading scheme and the condition of the wireless channel. By offloading computation tasks to nearby MEC cloud, the quality of performance including computation latency and energy consumption, can be progressively decreased [2].

Since computation performance and energy consumption competing for resource and these are critical for mobile users [3], the effective computation offloading schemes have attracted huge attention for both mobile cloud and MEC systems. In mobile cloud systems, most works focus on energy minimizing problems. In [4], a dynamic offloading algorithm is proposed in offloading process by determining the data rate of requests. A tradeoff between energy and delay is discussed in [5], and a task scheduling method is proposed with heterogeneous applications. Dynamic radio resource allocation method for computation tasks is described in [6] and [7]. In [8], a user experience based method is proposed to optimize mobile power consumption and computation delay. In MEC systems, most works are also interesting in reducing energy consumption. In [9], a delay-optimal task offloading algorithm is proposed for single user MEC systems. In the meantime, the tradeoff between power and latency are considered for mobile users. Chen [2] study multi-user MEC systems in a distributed manner on game theory. In the device to device communication [10], a joint energy efficient and QoS based method is proposed under strict execution delay for C-RAN based network. Unfortunately, existing works mainly focus on minimizing energy consumption, designing a scheme guaranteed latency from the perspective of energy consumption remains unknown.

In this paper, we address the problem of performance guaranteed computation offloading scheme for mobile-edge computing. Considering the realistic scenarios of base station, we introduce a multi-user MEC system with multiple parallel computation tasks requiring cloud resource. Multiple users lead to resource competition, in this case, a reasonable allocation is desired for radio and CPU cycle. For each computation task, mobile has an expected value of energy consumption. According to this value, we can obtain the worst situation on offloading scheme. We formulate a minimizing energy consumption problem with the constraints on resource capacity and delay. A method is proposed to decide the offloading sequence and decision. In particular, whether to offloading a computation task is determined by mobile energy condition and application requirements. Simulation results show that task offloading scheme is important for tradeoff computation and energy consumption. Besides, latency performance guaranteed under low energy consumption are archived by our proposed algorithm.

II. System Model

In this section, we consider multiple mobiles in one mobile-edge computing system shown in Fig. 1. The MEC servers are regarded as installing computing devices at a wireless access station. The mobile users can access the station resources

X. Tao, H. Qi and K. Li are with the School of Computer Science and Technology, Dalian University of Technology, No 2, Linggong Road, Dalian 116023, China. X. Tao is also a visiting student with Muroran Institution of Technology, Muroran, Hokkaido, Japan. E-mail: taoxiaoyi@mail.dlut.edu.cn, 16061120@mmm.muroran-it.ac.jp; {hengqi, keqiu}@dlut.edu.cn

K. Ota and M. Dong are with Department of Information and Electronic Engineering, Muroran Institution of Technology, Muroran, Hokkaido, Japan. E-mail: {ota, mxdong}@mmm.muroran-it.ac.jp.

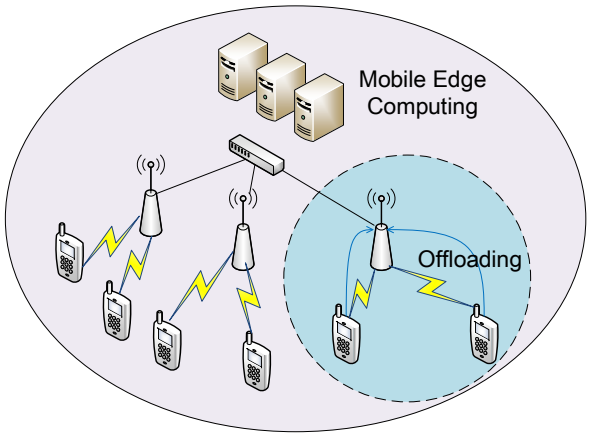


Fig. 1. A multi-user mobile-edge computing system

through a wireless channel, and allocate their computing tasks to the base station which can help mobile users to improve the computation performance. We denote $\mathcal{N} = \{1, 2, \ldots, N\}$ as a set of mobile users which have computational tasks to be completed. The task requirement of mobile n is denoted as a tuple $\{c_n, d_n\}$, where c_n presents required CPU circle and d_n describes the data size of task of mobile n. And l_n denotes the offloading data size of mobile n. In the meantime, we let α_n denote the fraction of offloading task for each user n, that satisfies $l_n = d_n\alpha_n$. For each mobile user, we define a desired power consumption value $\mathcal{E}_n$ and from which we can conduct an energy baseline to compute on MEC server for each mobile. Furthermore, each computation task has a completion time requirement denoted by $\mathcal{T}_n$. We assume that the mobile user is not dynamic in the MEC system. Once a mobile offloads its task to the MEC server, the energy and time consumption of communication and computation are considered.

We first introduce the communication process for MEC system. Mobiles decide whether to offload a task to the MEC server according to its energy consumption and completion performance. The communication consumption should be considered first. Let p_n denote the transmission power for mobile n and g_n is the channel gain of the base station. N_0 denotes the density of noise power of the channel. Let B denote the bandwidth of the channel. The communication rate of mobile n can be defined as

$$r_n = B\log_2(1 + \frac{p_n g_n^2}{N_0 B}). \tag{1}$$

As we known, the transmission rate can also be denoted as $r_n = \frac{l_n}{t_n}$. The transmission power p_n can be calculated by

$$p_n = \frac{1}{g_n^2} h(\frac{l_n}{t_n}) \tag{2}$$

We define a function $h(x) = N_0 B(2^{\frac{x}{B}} - 1)$, which is increasing and convex function while $x > 0$. The energy consumption is determined by task size, power density, and transmission rate. Therefore, if mobile n offloads a task to MEC server, the energy consumption can be denoted as

$$e_{n,off} = \frac{d_n p_n}{r_n} = p_n t_n = \frac{t_n}{g_n^2} h(\frac{l_n}{t_n}). \tag{3}$$

Accordingly, the completion time of offloading a task to edge server contains two parts: communication time and computation time shown in Equation (4). We define the computation ability of edge server as h_n^c. The completion time is denoted as

$$t_{n,off} = \frac{d_n}{r_n} + \frac{c_n}{h_n^c} = \frac{t_n}{g_n^2} h(\frac{l_n}{t_n}) + \frac{c_n}{h_n^c}. \tag{4}$$

We then introduce the local computation model in our system. Mobiles decide to compute task locally, transmission consumption is removed instead of local computation consumption. Here we define f_n as the power consumption per CPU cycle for mobile n. We let h_n denote the computation ability of mobile n. Accordingly, we define

$$e_{n,loc} = f_n c_n \tag{5}$$

as the energy consumption as compute task locally. In the meantime, the completion time is defined as

$$t_{n,loc} = \frac{c_n}{h_n}. \tag{6}$$

The completion time of local computing only connects with the computational ability of the mobile n. Fully local computing or offloading method might ignore the expected energy consumption of users. Consequently, we compute the energy consumption by locally or partial offloading tasks to edge server. Based on the expectation consumption limitation and above computation system model, we formulate an optimization problem for mobile-edge computing system.

III. Optimization Formulation

In this section, we formulate an energy efficiency optimizing problem for MEC systems. We joint consider the energy consumption and task completion time condition for each mobile user. We define α_n as the fraction of offloading task to a cloud server. Therefore, the energy consumption of each mobile n contains locally and partial offloading consumption as shown in Equation (7).

$$e_n = e_{n,off}\alpha_n + e_{n,loc}(1-\alpha_n) = \frac{t_n}{g_n^2} h(\frac{l_n}{t_n})\alpha_n + f_n c_n(1-\alpha_n). \tag{7}$$

Since we consider the computation performance as well, the completion time of one mobile user n can be denoted as

$$t_n = t_{n,loc}(1-\alpha_n) + t_{n,off}\alpha_n. \tag{8}$$

The completion time also includes partial offloading time and local computing time all of which are proportional to original time.

The objective of an optimization problem is a sum of energy consumption for mobiles. It is easy to observe that we calculate the decision for each mobile α_n that how much of its task offloads to the edge server. Our model shows that the decision must satisfy server computation capacity and channel bandwidth. We let $\mathcal{C}$ denote the cloud server CPU computation capacity. In the meantime, the energy consumption and completion time must satisfy the task requirement as well. The constraints of above problem are time and energy consumption. The problem is a convex optimization problem.

Here we consider using Lagrange method to derive a resource allocation scheme. The joint computing and energy efficiency mobile offloading optimizing problem can be formulated as:

$$\min_{\{\alpha_n,t_n\}} \quad \sum_{i=1}^{n}[\frac{d_n p_n}{r_n}\alpha_n + f_n c_n(1-\alpha_n)] \tag{9}$$

$$s.t. \quad \frac{c_n}{h_n}(1-\alpha_n) + (\frac{d_n}{r_n}+\frac{c_n}{h_n^c})\alpha_n - \mathcal{T}_n \leqslant 0 \quad \forall n, \tag{10}$$

$$\frac{d_n p_n}{r_n}\alpha_n + f_n c_n(1-\alpha_n) - \mathcal{E}_n \leqslant 0 \quad \forall n, \tag{11}$$

$$\sum_{n=1}^{n} c_n\alpha_n \leqslant \mathcal{C} \tag{12}$$

$$\sum_{i=1}^{n} r_n \leq B. \tag{13}$$

The optimal problem Equation (9) is a convex optimization problem. Since $h(x)$ is convex, and its multiplier function is also convex such as $\frac{t_n}{g_n^2}h(\frac{l_n}{t_n})$. Furthermore, while $t_n \geqslant 0$, the function $h(x)$ is still convex. Thus, the objective function, the sum of a series of convex equations, remains convex. To solve this convex problem, we define a partial Lagrangian function which is expressed as

$$\begin{aligned}\mathcal{L}(\alpha,t,\lambda,\mu) &= \frac{t_n}{g_n^2}h(\frac{l_n}{t_n})\alpha_n + f_n c_n(1-\alpha_n)\\ &+ \lambda[\frac{c_n}{h_n}(1-\alpha_n) + (\frac{d_n}{r_n}+\frac{c_n}{h_n^c})\alpha_n - \mathcal{T}_n]\\ &+ \mu[\frac{t_n}{g_n^2}h(\frac{l_n}{t_n})\alpha_n + f_n c_n(1-\alpha_n) - \mathcal{E}_n]\end{aligned}$$

where $\lambda \geqslant 0$ and $\mu \geqslant 0$ are the dual Lagrange multiplier which are associated with completion time and energy consumption constraints. Let α_n denote the optimal solution which always exists the feasible condition. Then we apply the KKT condition and transform to following equations:

$$\frac{\partial\mathcal{L}}{\partial\alpha_n^*} = (1+\mu)\frac{t_n}{g_n^2}h(\frac{l_n}{t_n}) - (1+\mu)f_n c_n + \lambda[\frac{c_n}{h_n^c} - \frac{c_n}{h_n}], \tag{14}$$

$$\frac{\partial\mathcal{L}}{\partial t_n^*} = [\frac{\alpha_n^*}{g_n^2} + \mu\alpha_n^*][h(\frac{l_n^*}{t_n^*}) - \frac{l_n^*}{t_n^*}h^{'}(\frac{l_n^*}{t_n^*})] + \lambda\alpha_n^*, \tag{15}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda^*} = \frac{c_n}{h_n}(1-\alpha_n) + (\frac{d_n}{r_n}+\frac{c_n}{h_n^c})\alpha_n - \mathcal{T}_n, \tag{16}$$

$$\frac{\partial\mathcal{L}}{\partial\mu^*} = \frac{t_n}{g_n^2}h(\frac{l_n}{t_n})\alpha_n + f_n c_n(1-\alpha_n) - \mathcal{E}_n. \tag{17}$$

Note that α_n and t_n can be derived from above equation. Accordingly, the optimal scheme for offloading method is computed in the following. We define a dual function $g(\lambda,\mu)$,

$$\begin{aligned}&g(\lambda,\mu) = min\mathcal{L}(\alpha,t,\lambda,\mu)\\ &s.t. \quad 0 \leqslant \alpha_n \leqslant 1.\end{aligned}$$

Consequently, the dual problem can be denoted as

$$\max_{\{\lambda,\mu\}} g(\lambda,\mu) \tag{18}$$

$$s.t. \quad f(\lambda) \geqslant 0, \lambda \geqslant 0, \mu \geqslant 0, \tag{19}$$

where $f(\lambda) \geqslant 0$ and the constraints of (λ,μ) are denoted above. In the following, we analyze the dual function for any feasible solution satisfy the primal problem. Since the primal problem is convex and satisfies the Slater's condition, that is, primal problem (9) and dual problem (19) are strong dualities. Therefore, we can obtain the optimal solution for the primal problem from solving the dual problem (19). In the following, we first solve the dual problem to maximize $g(\lambda,\mu)$ using Lambert function, then we obtain the optimal solution to primal problem (9). Based on KKT conditions, the above fraction functions satisfy dual point and equal to zero. As we know, $h^{'}(x) = \ln 2 N_0 2^{\frac{X}{B}}$ is a derivation of $h(x)$. As shown in Equation (16), assume $z = h(x) - xh'(x)$, we can conduct its inverse function as

$$x = \frac{B}{\ln 2}(W_0(-\frac{z}{N_0 Be} - \frac{1}{e}) + 1), \tag{20}$$

according to Lambert Function. And $r_n = \frac{l_n^*}{t_n^*}$, we can conduct that

$$h(r_n^*) - r_n^* h'(r_n^*) = -\frac{\lambda g_n^2}{1+\mu g_n^2}. \tag{21}$$

For a given $\lambda > 0$ and $\mu > 0$, the optimal solution of this problem can be computed as follows. Based on Equation (20), r_n satisfy that

$$r_n^* = \frac{B}{\ln 2}(W_0(\frac{\lambda g_n^2}{N_0 Be(1+\mu g_n^2)} - \frac{1}{e}) + 1). \tag{22}$$

From Equation (17), we readily conduct the result of α_n

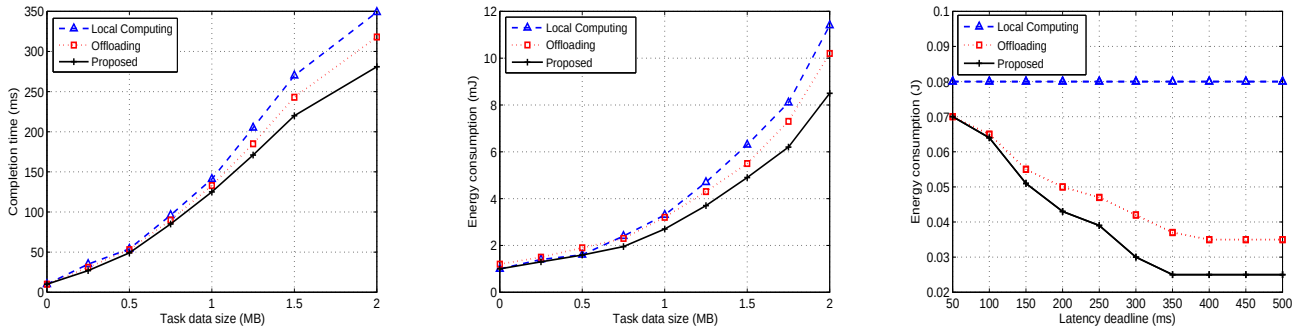
$$\alpha_n^* = \frac{\mathcal{T}_n h_n h_n^c - c_n h_n^c}{h_n^c h_n d_n/r_n + c_n h_n - c_n h_n^c}. \tag{23}$$

Since we know $t_n^* = \alpha_n^* d_n/r_n^*$, we can obtain the optimal solution (α_n^*, t_n^*). These imply that the energy consumption is not tight combined with mobile energy condition. The extreme condition is computation task is all computed locally. Here is an intuitive, if a mobile has sufficient energy to compute tasks locally, offloading task to the cloud must have some improvement on completion time and savings on energy consumption.

IV. Numerical Results

In this section, we discuss the performance of our optimal offloading scheme.The simulation settings are listing as follows. There are 20 mobiles in our simulation. The computing speed of MEC server is set to be h_n^c 10 GHz and the mobile CPU ability is randomly from {0.5, 0.6, ..., 1.0} GHz. The size of the task is basically are the uniformly distributed (0, 2) MB, and the total CPU cycle requirement c_n is 1000 cycles/bits. We set the channel bandwidth $B = 5MHz$, $N_0 = 10^{-12}W$. The expectation energy consumption and latency for each mobile are random from {1, 1.5, 2}W and their desire latencies are random from 100ms to 500ms. The random parameters are independent for various users. We assume that the task offloading is controlled by a centralized controller. We evaluate our proposed partial offloading scheme compared to local computing and full offloading method.

In Fig 2(a), we discuss the completion time as the variance of task size. The completion time of computation task is the



(a) Effect of task size on completion time. (b) Effect of task size on energy. (c) Energy consumption and latency requirment.

Fig. 2. Performance comparison on energy consumption, completion time and influence variance on latency.

main target of evaluating offloading methods. Completion time of offloading task to a cloud contains communication time in wireless network and computation time in edge server. As the increasing of task size, the communication time and computation time are getting larger. Here, we compare to the local computation and offloading communication consumption. From Fig 2(a), we can conclude that partial optimal offloading method reduces completion time of computational tasks. When task size is small, local computing is smaller because of it reduces the latency on the communication process.

Fig 2(b) depicts the energy consumption variance with the task size. It is observed that our proposed scheme outperforms local computing and full offloading method. In the meantime, the offloading method gets a better result especially when the task size becomes larger. Therefore, for large computing task, our method prefers to offload large partial computation tasks to the MEC server to reduce mobile consumption. The full offloading method is expected to have better performance than local computing on energy consumption. Partial offloading computation task to edge server considers the tradeoff between advantages local computing and full offloading methods, hence our scheme reduces energy consumption in total.

We consider the differences of latency requirement have effects on the energy consumption shown in Fig 2(c). Local computing method energy consumption is just affected by the task data size and CPU computing requirement. Therefore, its energy consumption is not influenced by the changes of latency requirement. Obviously, offloading task helps reduce energy consumption. When task latency requirements are low, the offloading and our method has similar performance, because most of the tasks are offloaded to the edge server. As requirement becomes larger, our method adopts an optimal offloading scheme which brings a slight decrease in energy consumption.

V. Conclusion

In this paper, we investigate task offloading scheduling with the guarantee of service performance for mobile-edge computing. To ensure the service quality and mobile energy, we formulate an optimization problem on minimizing energy consumption for mobile users. We design a task offloading algorithm to solve the optimization problem. The edge servers make a decision on each mobile requests based on their priorities and complete performance. Numerical results show that our proposed method can improve consumption and performance on latency.

Acknowledgment

This work is supported by the National Key Research and Development Program of China No. 2016YFB1000205, the State Key Program of National Natural Science of China(Grant No. 61432002), NSFC Grant Nos. 61272417, 61300189, 61370199 and 61672379, JSPS KAKENHI Grant Number JP16K00117, JP15K15976, KDDI Foundation.

References

[1] P. Milan, J. Jerome, Y. Valerie, and A. Sadayuki, "Mobile-edge computing introductory technical white paper," *White Paper*, 2014.

[2] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[3] M. Dong, X. Liu, Z. Qian, A. Liu, and T. Wang, "Qoe-ensured price competition model for emerging mobile networks," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 50–57, 2015.

[4] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[5] J. Kwak, Y. Kim, J. Lee, and S. Chong, "Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2510–2523, 2015.

[6] S. Guo, B. Xiao, Y. Yang, and Y. Yang, "Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, 2016, pp. 1–9.

[7] C. You, K. Huang, and H. Chae, "Energy efficient mobile cloud computing powered by wireless energy transfer," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1757–1771, 2016.

[8] S.-T. Hong and H. Kim, "Qoe-aware computation offloading scheduling to capture energy-latency tradeoff in mobile clouds," in *Sensing, Communication, and Networking (SECON), 2016 13th Annual IEEE International Conference on*, 2016, pp. 1–9.

[9] J. Liu, Y. Mao, J. Zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing systems," in *Information Theory (ISIT), 2016 IEEE International Symposium on*, 2016, pp. 1451–1455.

[10] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. T. Yang, "Energy-efficient resource allocation for d2d communications underlaying cloud-ran-based lte-a networks," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 428–438, 2016.