Non-Stationary Online Task Offloading in Fog-Enabled Networks

Ting Liu
School of Information Science and Technology
ShanghaiTech University
Email: liuting@shanghaitech.edu.cn

***Abstract*—The recent technological advances related to computing, srotage, cloud, networking and the unstoppable deployment of end-user devices, are all coining the so-called Internet of Things (IoT). To balance the limited-resources IoT devices and intensive-delay, fog computing turns up as an trend. Indeed, fog provides low delay for services demanding real time response, constrained to support low capacity queries. To that end, in this paper, we consider a fog network, in which a fog node only has partial information, we introduce an online algorithm based on combinatorial multi-armed bandit (CMAB) to minimize the latency of the each task by computing offloading.**

1. Introduction

By 2020, the Internet of Things is expected to connect tens of billions of devices over the world, such as smart phones and wearable devices. Consequently, more and more mobile applicaitons such as face recognition, augmented reality and interactive online gaming are emerging and attract great attention. However, this kind of mobile applications typically demand intensive computation and low lantency while the limited battery life and limited computation resource of the IoT devices is difficult to satisfies. To this aim, a promising paradigm called fog computing has been recently proposed [1]. Fog computing inherits the main concepts of Cloud Computing, but move them to the edge of the network. The main goal is to bring computing resources closer to end-user devices, hence enforcing locallity, which imposes low response time and low network load.

By offloading the computation-intensive tasks to the fog node, the quality of computation experience, including the latency and device energy consumption, could be greatly improved. Nevertheless, the efficiency of a fog network system largely depends on the proposed computation offloading policy, which should be carefully designed by taking the characteristics of the comutation tasks and wireless channels into account [2].

Unlike the existing literature which assumes full information knowledge for fog network formation and rely on simple delay models, our goal is to design an online approach to enable an on-the-fly information of the fog network, under uncertainty, while minimizing computational latency, given a realistic delay model.

In this paper, we consider a fog network where the arrival task of each fog node is stochastic and a non-stationary random process, and the fog node offloads tasks to other fog node in the network, given no prior information on channel condition known. We develope a efficient computation peer offloading algorithm based on combinatorial multi-armed bandit (CMAB).

The rest of this paper is organized as follows. In Section 2 , the system model is presented. In Section 3, we proposed the proposed online algorithm. Simulation results are analyzed in Section 4.

2. System Model

Consider a fog-enabled network, where M mobile stations (MSs) and N fog nodes co-exist. To carry out the computational tasks with low-latency, each MS may intelligently offload its computational tasks to the neighboring fog nodes or execute these tasks locally. When one task is generated, the corresponding MS needs to respond immediately, i.e whether to offload the task to fog nodes or process it locally. Obviously, different responses are related to different latencies.

The aim is to find an online strategy to minimize the total latency among all the MSs whenever one task is generated in this network. Note each task is assumed to be generated by each MS randomly and independently, then the probability of two MSs generating tasks simultaneously is 0 if the continuous time is considered. In other words, there is only one task generated by the network each time. Additionally, the decision of offloading the latter task or not would never have impact on the latency of former tasks. This is due to the fact that the tasks are assumed to be cached in a first-input first-output (FIFO) queue, and the latter tasks entering the queue will not influence the latency of the former tasks. Therefore, when we consider to minimize the total latency among all the MSs in an online approach, we just need to find one strategy ensuring that each MS can minimize the corresponding latency. In the rest of this paper, we focus on minimizing the latency of one MS.

Define the set of the concerned nodes as

$$\mathcal{N} := \{\underbrace{1, 2, \cdots, N-1}_{\text{Fog nodes}}, \underbrace{N}_{\text{MS}}\}. \tag{1}$$

Thus one task can be carried out in a particular node belonging to the set $\mathcal{N}$. If the s-th task, the data length of which is denoted as L_s, is carried out at node-n, the total latency should be comprised of three parts, i.e. the transmission delay $L_s \cdot T(n)$, the waiting delay $Q_s(n) \cdot P_s(n)$ in the computation queue, and the processing delay $L_s \cdot P_s(n)$. In particular, $P_s(n)$ is the time needed for processing one bit data, $T(n) = \frac{1}{\alpha \cdot \bar{c}(n)}$ stands for the time needed for transmitting one bit data, where α is the spectrum bandwidth allocated for the concerned MS, and $\bar{c}(n)$ is the spectrum efficiency, both are assumed to be known for simplicity. Additionally, $Q_s(n)$ is the data length of the queue where task-s needs to wait at node-n. The transmission delay is negligible when the task is carried out locally, i.e. $T_s(N) = 0$. The data length and the computational complexity of one task are assumed to be different and vary slowly with time, then the computational complexity for processing one bit task is a random variable, the distribution of which changes slowly with time. Furthermore, the computational ability of each MS or each fog node is different, which makes the time for processing different tasks difficult to be determined. As a result, the processing delay and waiting delay cannot be calculated through the conventional model related to the data length and the CPU frequency as in [3].

3. Online algorithm

In this paper, the waiting delay and the processing delay are regarded as the posterior information, which are obtained via the timestamp feedback provided by the fog nodes or the MS after finishing the corresponding process. Besides, we assume the latency mentioned above dominates the total latency. The latency caused by transmitting computation results and timestamp feedback are ignored since the corresponding data lengths are much smaller than the data lengths of the offloaded tasks.

Due to the uncertainty of this problem, it can be modeled as a non-stationary multi-armed bandit (MAB) problem, where each fog node is regarded as an arm. When the s-th task is generated, we need to determine one node, either a fog node or the MS, to deal with it. This is exactly corresponding to choose one arm to play in the multi-armed bandit problems. The cost of dealing with the k-th task related to node-n is modeled as

$$\bar{X}_k(\gamma, n) = L_k \cdot T(n) + \frac{Q_k(n) + L_k(n)}{Y_k(\gamma, n)} \bar{P}_k(n), \qquad (2)$$

$\gamma \in (0, 1)$ is a discount factor, and

$$\begin{aligned} Y_k(\gamma, n) &= \sum_{s=1}^{k-1} \gamma^{g_k - t_s} \mathbb{1}\{I_s = n\}, \\ \bar{P}_k(n) &= \sum_{s=1}^{k-1} \gamma^{g_k - t_s} P_s(n) \mathbb{1}\{I_s = n\}, \end{aligned} \qquad (3)$$

where g_k stands for the time when the k-th task is generated, and t_s is the time when the feedback of the s-th task is received, I_s represents the index of the node to deal with task-s. $T(n)$ is determined by the spectrum efficiency and the bandwidth, $P_s(n)$ are determined by the feedback. In order to estimate the instantaneous expected reward, the algorithm averages past rewards with a discouted factor γ, which gives more weight to the recent observations.

Algorithm 1 Discounted UCB

1: for task k from 1 to N, choose node $I_k = k$, update $\bar{X}_k(\gamma, n), c_k(\gamma, n)$.
2: $k \leftarrow N$.
3: **while** 1 **do**
4: $k \leftarrow k + 1$.
5: for task k after N, choose node

$$I_k = \arg\max_{1 \le n \le N} -\bar{X}_k(\gamma; n) + 2B\sqrt{\frac{\xi \log y_k(\gamma)}{Y_k(\gamma, n)}}.$$

6: update $\bar{X}_k(\gamma, n), c_k(\gamma, n)$.
end while

The desired algorithm is to find a policy with a small cost. An algorithm that has good theoretical results is upper-confidence bound [4], by applying the UCB policies, we can construct an UCB as $-\bar{X}_k(\gamma, n) + c_k(\gamma, n)$, where the discounted padding function is defined as

$$c_k(\gamma, n) = 2B\sqrt{\frac{\xi \log y_k(\gamma)}{Y_k(\gamma, n)}}, \qquad (4)$$

and

$$y_k(\gamma) = \sum_{n=1}^{N+1} Y_k(\gamma, n). \qquad (5)$$

where B is an upper-bound on the cost and $\xi > 0$ is some appropriate constant.

The proposed algorithm based on the framework of discouted UCB is given in Algorithm 1. Step 1 of Algorithm 1 guarantees that every node n can be played in the first N plays.

4. Numerical Results

This section presents the results achieved by the presented model. In the simulation, we consider a fog network with 10 fog nodes, including a MS. $P_s(n)$ follows a uniform distribution in $(1/3, 1)$ sec/Mbit in each time slot. $\gamma = 0.9, \xi = 3$ during all the simulations.

We consider the performance of the proposed algorithm to other policies: (1) **Local computing**: all the tasks are processed locally; (2) **Random**: the fog nodes are choosed randomly; (3) **Optimal**: given the global knowledge of the fog network and always performs the best node.

The simulation results perform not as well as we thought, I will try to improve it later.

References

[1] F. Bonomi, R. Milito, J. Zhu, S. Addepalli. 2012. Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (MCC '12). ACM, New York, NY, USA, 13-16.

[2] J. Liu, Y. Mao, J. zhang, and K. B. Letaief, "Delay-optimal computation task scheduling for mobile-edge computing under stochastic wireless channel,"IEEE Trans. Wireless Commum., vol. 12, no.9, pp. 4569-4581, Sep. 2013

[3] Y. Yu, J. Zhang, and K. B. Letaief, Joint subcarrier and CPU timeallocation for mobile edge computing, in Proc. IEEE Global Commun.Conf. (GLOBECOM), Washington, DC, Dec. 2016.

[4] Aurlien Garivier and Eric Moulines. On upper-confidence bound policies for non-stationary bandit problems. Preprint, 2008.