

Poster Abstract: Decentralized Fog Computing Resource Management for Offloading of Periodic Tasks

Slađana Jošilo and György Dán

ACCESS Linnaeus Center, School of Electrical Engineering and Computer Science
KTH, Royal Institute of Technology, Stockholm, Sweden E-mail: {josilo, gyuri}@kth.se

Abstract—Fog computing is recognized as a promising approach for meeting the computational and delay requirements of a variety of emerging applications in the Internet of Things. This work presents a game theoretical treatment of the resource allocation problem in a fog computing system where wireless devices periodically generate computationally intensive tasks, and aim at minimizing their own cost.

I. INTRODUCTION

The emerging paradigm of fog computing brings computing resources close to the network edge. Placing computing resources at the network edge in close proximity of end users and devices may reduce the communication delays and the response times, and is expected to enable a variety of emerging Internet of Things (IoT) applications [1]. Applications that could benefit most are those in need of periodic execution of computationally intensive tasks on energy limited devices, which could be offloaded to fog computing resources, such as surveillance and augmented reality [2], [3].

The proximity of computing resources to devices is a prerequisite for predictable and low response times, but it may not be sufficient. In lack of resource management, when many devices attempt to offload computations simultaneously over a shared communication resource, such as a wireless access point, the communication delays may increase and may become unpredictable. Furthermore, computing resources should be allocated so as to match the assignment of communication resources and computational requirements, to ensure low response times. Thus, the joint management of communication and computing resources is essential for ensuring predictable and low response times for computation offloading.

Joint resource management for fog computing systems, is, however challenging for several reasons. First, fog computing systems are expected to consist of devices that are heterogeneous in terms of their computational capabilities and battery capabilities, and thus devices could have different preferences over the energy consumption and the application response times. Second, the different devices may generate computational tasks that are very different in terms of the amount of data needed to be transmitted and in terms of their complexity. Furthermore, there may be many devices that periodically generate computational tasks with low response time requirements, and thus the tasks have to be scheduled over time and across communication resources. Finally, devices may be autonomous, and hence resource management should respect their individual preferences.

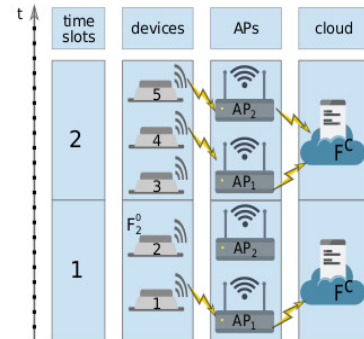


Fig. 1. An example of a fog computing system that consists of $N = 5$ devices, $T = 2$ time slots, and $A = 2$ APs.

In this work we address the problem of managing the communication and computing resources in a fog computing system for computation offloading, and we use game theoretical tools for designing an efficient decentralized algorithm for coordinating the decisions of devices with periodic tasks.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a fog computing system that consists of a set \mathcal{N} of devices that generate computationally intensive tasks periodically every T time units, a set \mathcal{A} of APs and an edge cloud. We consider that each device can choose one time slot from the set $\mathcal{T} = \{1, 2, \dots, T\}$ to perform the computation, and within the chosen time slot it can decide whether or not to offload the computation to the cloud via one of the APs $a \in \mathcal{A}$. We denote by $d_i \in \mathcal{D}_i$ the decision of device i , where $\mathcal{D}_i = \{\mathcal{A} \cup \{0\} \times \mathcal{T}\}$, and 0 corresponds to local computing.

Task model: Device i 's task $\langle D_i, L_i \rangle$ is characterized by two parameters, the mean size D_i of the input data and by the complexity L_i , which is defined as the mean number of CPU cycles required to perform the computation.

Communication model: When it offloads the task, each device i has to transmit the input data of size D_i via one of the APs. We consider that the uplink rate that device i can achieve if it offloads the task via AP a in time slot t is a non-increasing function of the number of devices that offload via the same AP a in the same time slot t . Thus, the time needed for device i to transmit D_i amount of data via AP a in time slot t and the corresponding energy consumption are non-decreasing functions of the number of devices that offload via the same AP a in the same time slot t .

Computation model: In the case of local computing, we consider that the computational capability F_i^0 of device i is the same in each time slot t . Hence, the time needed for device i to perform its task and the corresponding energy

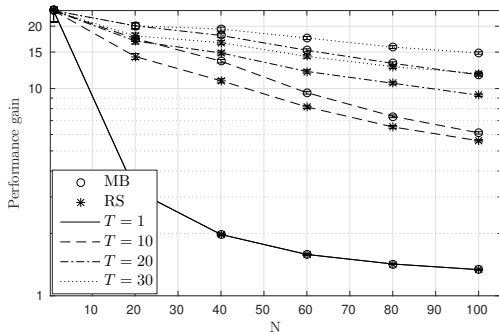


Fig. 2. *Performance gain* vs. number of devices (N).

consumption are independent of the chosen time slot. In the case of computation offloading, we consider that the computational capability F^c of the cloud is the same in each time slot t , but that the computational capability that device i would receive from the cloud in time slot t is a non-increasing function of the total number of devices that offload in the same time slot t . Thus, the time needed for the cloud to perform device i 's task depends on the chosen time slot t , and we consider that it is a non-decreasing function of the total number of devices that offload in the same time slot t .

Cost model: We consider that each device is interested in minimizing the response time and the corresponding energy consumption, and we define device i 's cost C_i as a linear combination of its response time and its energy consumption.

Game theoretical formulation of the problem: Since devices may be autonomous entities that aim at minimizing their own cost, we model the problem as a strategic game $\Gamma = \langle \mathcal{N}, (\mathcal{Q}_i)_i, (C_i)_i \rangle$, in which the players are the devices, and we refer to the game as the *multi-slot computation offloading game* (MSCOG). Our objective is to answer the fundamental question whether there exists a strategy profile in which there is no device that can decrease its cost by changing its strategy, i.e., a pure strategy Nash equilibrium (NE).

Figure 1 shows an example of a fog computing system where five devices can choose one slot out of two time slots to perform the computation and in the chosen time slot devices can offload their tasks via one of two APs.

III. ANALYTICAL RESULTS

Single time slot: We show that the MSCOG for $T=1$ is best response equivalent to a singleton player specific congestion game, and thus it possesses a pure strategy NE. Nonetheless, improvement paths may be cyclic and thus the MSCOG does not allow a generalized ordinal potential function.

Multiple time slots: For $T > 1$ the best-response equivalence does not hold, but as we prove the MSCOG possesses a pure strategy NE for any $T \geq 1$. Our constructive equilibrium existence proof is based on the *MyopicBest* (MB) algorithm, which adds devices one at a time, and lets them play their best replies given the other devices' strategies. We prove that the algorithm has polynomial time complexity, and it can be implemented in a decentralized manner. Consequently, devices do not need to reveal their parameters, but only their most recent decisions.

Structure of NE: We also characterize the structure of NE in general, and we give insight into the structure of the NE computed by the proposed MB algorithm. Based on these

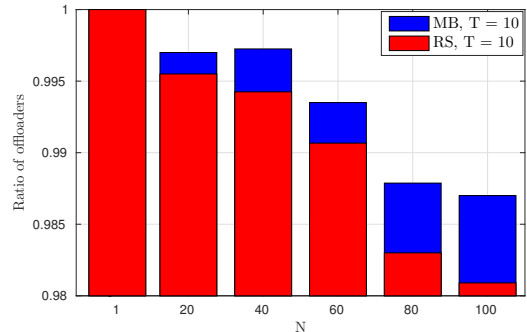


Fig. 3. Ratio of offloaders vs. number of devices (N).

structural results, we have established a bound on the price of anarchy of the strategic game, which serves as the cost approximation ratio for the MB algorithm.

IV. NUMERICAL RESULTS

We use extensive simulations to evaluate the performance of the MB algorithm. We consider that the devices are placed uniformly at random over a square area of $1km \times 1km$, and $A = 5$ APs are placed at random on a *regular grid* with 25 points defined over the area. As a baseline for comparison we use two algorithms. The first (*RandomSlot* (RS)) algorithm lets each device choose a time slot at random, and implements an equilibrium allocation within each time slot. The second algorithm lets all devices perform local execution. We define the *performance gain* of an algorithm as the ratio between the system cost reached when all devices perform local execution and the system cost reached by the algorithm.

Fig. 2 shows the *performance gain* as a function of the number N of devices for four different values of T . The results show that the MB and the RS algorithms can provide significant gains compared to local execution. However, the MB algorithm outperforms the RS algorithm for all values of N and for all values of T except for $T = 1$ when the two algorithms are equivalent. Therefore, coordination between devices is essential for improving the performance of fog computing systems.

Fig. 3 shows the ratio of devices that offload for $T = 10$. The results show that for $N > 1$ the ratio of devices that offload is always higher in the case of the MB algorithm than in the case of the RS algorithm. Hence, coordination between devices allows more devices to offload their tasks, and thus it improves the utilization of fog computing resources.

V. CONCLUSION

Our results show that game theoretical tools can be used for designing efficient algorithms for the joint management of communication and computing resources in fog computing systems, and that coordination between devices can provide significant performance gains and improve resource utilization.

REFERENCES

- [1] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," in *Proc. of IoT-J*, pp. 854–864, 2016.
- [2] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "Pipeneta wireless sensor network for pipeline monitoring," in *Proc. of IPSN*, 2007, pp. 264–273.
- [3] M. Hakkarainen, C. Woodward, and M. Billinghurst, "Augmented assembly using a mobile phone," in *Proc. of ISMAR*, 2008, pp. 167–168.