A Perspective on the Parallel Computation Offloading in the Fog Computing Systems

Hoa Tran-Dang, Dong-Seong Kim

***Abstract*—The fog computing paradigm is introduced to improve the performance of IoT-enabled systems in terms of service delay owing to the capability of task execution nearby the data sources (i.e., IoT devices). However, the fog-based computing methods do not always reduce the delay as compared to the cloud-based ones since the heterogeneity and resource limitation of fog devices can incur the excessive delay in the congested queues. This short paper introduces a perspective to solve this problem by using the parallel computation concept enabled by the task division.**

***Index Terms*—Fog-enabled IoT Systems, Fog Computing, Task Offloading, Parallel Computation, Task Division.**

I. Introduction

The Internet of Things (IoT) paradigm has been widely adopted in practical applications such as smart cities, smart grids since it enables providing smart services and informed decision makings for monitoring, control, and management purposes [1]. Currently, the mutual benefits gained from the combination of fog and cloud enable the resulting IoT-fog-cloud systems to provide uninterrupted IoT services with various QoS requirements for the end users along the things-to-cloud continuum. However, employing the fog computing raises another concern regarding decisions whether the tasks should be processed in the fog or in the cloud. There are many factors impacting on the offloading decision policies such as offloading criteria, application scenarios [2]. Basically, in the most of existing offloading techniques the tasks are probably offloaded to the best surrogate nodes, which have the most ample resources (e.g., large storage capacity, high speed processing) [3]. However, these solutions face significant challenges regarding the workload distribution among the complicated heterogeneous fog devices. The challenge is further amplified by increasing the rates of service requests, which probably make the task queues of resource-rich fog nodes longer. As a result, the requirements of latency-sensitive applications can be violated because of excessive waiting time of long queue. Furthermore, the reliance on the remote cloud servers to fulfill the tasks may not help in improving the situation due to high communication delay or networking related disturbance.

Many existing offloading algorithms consider to use parallel computation concept to resolve the aforementioned issues. These solutions derive partial offloading policies based on the division of input data division [4], [5]. Accordingly, a task can be divided into two portions, which are then processed by local device and neighbor fog nodes in the parallel manner to reduce the task delay. However, the delay can be reduced more as the task can be divided into multiple parts to exploit the parallel computation benefit as well as improve the resource utilization of resource-poor devices. This perspective is the main motivation to derive a framework of offloading in this paper.

Hoa Tran-Dang and Dong-Seong Kim are with department of IT Convergence Engineering, Kumoh National Institute of Technology, Korea, e-mail: {hoa.tran-dang, dskim}@kumoh.ac.kr. Corresponding Author: Dong-Seong Kim

Corresponding Author: Dong-Seong Kim, e-mail: dskim@kumoh.ac.kr.

II. System Model and Problem Formulation

A. System Model

This paper considers a simple architecture of fog computing system as illustrated in Fig. 1, which includes a fog controller and N neighbor fog devices.

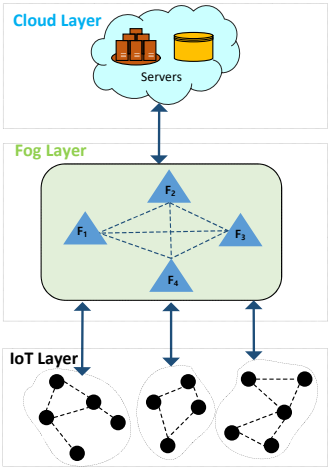

Fig. 1: A typical three-tier architecture of IoT-Fog-Cloud system for providing specific kinds of IoT services.

From the above stated perspective, each primary host is based on the available resources and workload state of its neighborhood to select the efficient service host. Each fog maintains it own neighbor resource table containing the updated information about the available resources. These tables are updated and shared periodically among the neighboring nodes to support the primary host to make offloading decisions. Table I shows an example of neighbor resource table stored by the fog node F_1, which records the resource states of neighbors with respect to residual memory (M_r), clock frequency, round-trip time (RTT), and waiting time in queue (W).

Node ID	Fog specification & Resource Status			
	M_r (MB)	Frequency (GHz)	RTT (ms)	W (ms)
F_2	200	10	2.5	350.2
F_3	100	5	3.1	500
F_4	400	2.5	4.8	239.1

TABLE I: Resource table of neighbors of fog node F_1

B. Problem Formulation

Fig. 2 illustrates the concern problem in IoT-Fog-Cloud systems, in which the primary host F_2 cannot process a service request A due to lack of resources. Meanwhile, offloading the

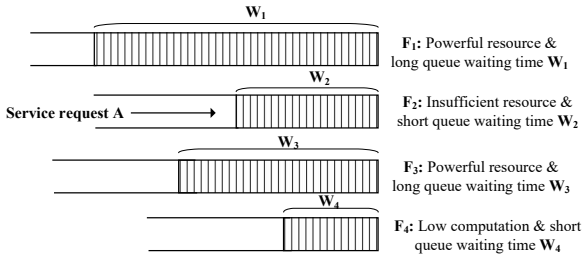

Fig. 2: The heterogeneity and unbalanced workload of fog environment expose issues in offloading tasks.

task to the fog neighbors F_1 and F_3 may lead to extensive delay since there are high workloads in queues of these fog nodes. In addition, F_4 can offload the task but resulting long delay due to low computational capability. Such issue urges a need to design an algorithm for efficiently allocating which fogs process which tasks in order to achieve the minimized average delay.

III. Delay Reduction through Parallel Computation

A task T_k is represented by (A_k, B_k), where A_k is size of task and B_k is required central processing unit (CPU) cycles. T_k can be divided up into at most $N+1$ substasks $\{ST_{k,1}, ..., ST_{k,N+1}\}$, which can be processed in parallel by the local device and its N neighbors. Fig. 3 illustrates the parallel computation in the fog enabled by partial offloading.

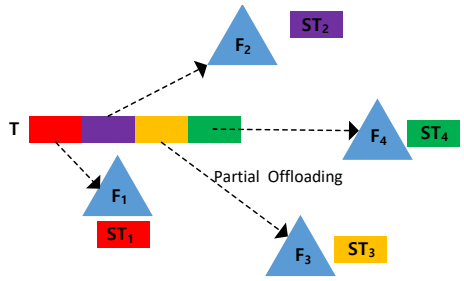

Fig. 3: The parallel computation is enabled by partial offloading.

$\alpha_k = \{\alpha_k^0, \alpha_k^1, ..., \alpha_k^N\}$ is a vector specifying the division of task data. Specially, α_k^i is a data fraction divided from a task k. Basically, $0 \leq \alpha_k^i \leq 1$ and $\sum_{j=0}^{N} \alpha_k^i = 1$ as a task k can be divided up to at most $N+1$ portions. Correspondingly, A_k^i is size of data subset α_k^i, $A_k^i = \alpha_k^i A_k$.

$\beta_{k,j}^x = 1$ if $ST_{k,j}$ is processed by fog x, $(x = 1, .., N+1)$. Since $ST_{k,j}$ can be only processed by a single device, thus $\sum_{x=1}^{N+1} \beta_{k,j}^x = 1$, $\forall k, j$.

The delay for executing T_k is denoted as D_k, which can be derived by $D_k = \max\{D_{k,1}, ..., D_{k,N+1}\}$, where $D_{k,j}$ is delay for executing subtask $ST_{k,j}$. Since a subtask $ST_{k,j}$ is processed only by the local device x or a neighbor node y, a generic model for calculating $D_{k,j}$ is follow:

$$D_{k,j}^x = \frac{B_k}{R^{xy}} + \beta_{k,j}^x \frac{A_{k,j}^x}{\mu^x} + \beta_{k,j}^y \frac{A_{k,j}^x}{\mu^y} \quad (1)$$

In this equation, R^{xy} is transmission rate from a node x to a node y. As the task (substask) is processed locally, the transmission delay is vanished. Thus, we set $R^{xx} = \infty$. μ^x is the computation capacity of node x (in cycles/s). Note that $\beta_{k,j}^x + \beta_{k,j}^y = 1$. Consequently, to minimize D_k, we need to find α_k and $\beta_{k,j}$ such that $D_k = \min\max\{D_{k,1}, ..., D_{k,N+1}\}$.

IV. Conclusions and Future Works

Offloading the tasks to the best neighbor fogs may contribute to the excessive delay due to the long queuing tasks of powerful nodes. In addition, the workload distribution is unbalanced among the heterogeneous fog environment, which results in the underutilized resources. This short paper introduced the perspective of parallel computation, which can reduce the task execution delay by partial offloading substasks to different fog nodes. This novel policy opens potential issues regarding the implementation, and evaluation in the future works.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2020-2020-0-01612) and supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003), and Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (NRF-2020R1I1A1A01073019).

References

[1] H. Tran-Dang and D. Kim, "An information framework for internet of things services in physical internet," *IEEE Access*, vol. 6, pp. 43 967–43 977, 2018.

[2] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities," *Future Generation Computer Systems*, vol. 87, pp. 278–289, Oct. 2018.

[3] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 998–1010, April 2018.

[4] Z. Liu, Y. Yang, K. Wang, Z. Shao, and J. Zhang, "Post: Parallel offloading of splittable tasks in heterogeneous fog networks," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3170–3183, 2020.

[5] H. Tran-Dang and D.-S. Kim, "FRATO: Fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," vol. 32, no. 10, pp. 2491–2508.