Design and Evaluation of In-situ Resource Provisioning Method for Regional IoT Services

Yugo Nakamura[1,2], Teruhiro Mizumoto[1], Hirohiko Suwa[1], Yutaka Arakawa[1], Hirozumi Yamaguchi[3], and Keiichi Yasumoto[1]

[1]Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
[2]Research Fellow of the Japan Society for the Promotion of Science
[3]Osaka University, Suita, Osaka 565-0871, Japan
Email: {nakamura.yugo.ns0, yasumoto}@is.naist.jp

***Abstract*—In an era where billions of IoT devices are deployed, edge/fog computing paradigms are attracting attention for their ability to reduce processing delays and mitigate waste of communication resources. However, since the computing system assumed by edge/fog paradigms have heterogeneity (in terms of the computing power of devices, network performance between devices, device density, etc.), provisioning computational resources according to computational demand becomes a challenging constrained optimization problem. In this paper, we propose in-situ resource provisioning method consisting of in-situ resource area selection with adaptive scale out and in-situ task scheduling based on tabu search algorithm. We conducted a simulation study in a target regional area where 2,000 IoT devices and 10 IoT services are deployed to evaluate the effectiveness of the proposed algorithm. The simulation results show that our proposed algorithm can obtain higher user QoS compared to conventional resource provisioning algorithms.**

***Keywords*—Internet of Things, Edge computing, Resource provisioning.**

I. Introduction

Recently, new and promising computing paradigms including Edge and Fog Computing have recently been attracting attention [1]. These paradigms have adopted an approach that assigns delay-sensitive tasks to computation resources close to service users. Inspired by these paradigms, we have proposed a new IoT data processing framework called the Information Flow of Things (IFoT) [2] for processing, analyzing, and curating IoT data in a real-time and scalable manner based on distributed processing among in-situ IoT devices. The IFoT framework aims to flexibly utilize computational resources of edge IoT devices existing near the data source, aiming to realize and maintain delay-sensitive regional IoT services efficiently and at low cost. In delay-sensitive regional IoT services, raw sensor data must be quickly (within delay constraints) converted to valuable information and IoT content such as regional map services with various timely information (e.g., crowdedness at shops/sightseeing spots, buses/vehicles location) by executing a set of tasks programmed in a task graph according to user's request. Therefore, how to effectively assign computational resources (i.e., IoT devices) to each IoT services considering QoS is a key challenge. In addition, such distributed systems composed of various IoT devices have heterogeneity with respect to the computing power of devices, network performance between devices, devices density, etc. Consequently, we propose an in-situ resource provisioning method which aims to efficiently provision computational resources of edge IoT devices to satisfy the computational demand required for each service.

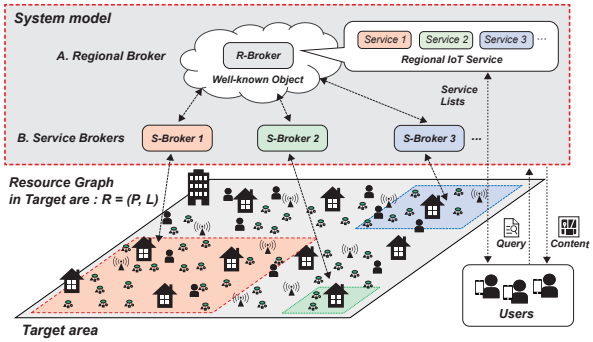

Fig. 1. Target area and system model of proposed method

II. In-situ resource provisioning

We show the target area and system model of the proposed method in Fig. 1. In the target area, various regional IoT services that effectively utilize raw sensor data generated by IoT devices are provided. Users are allowed to get a content from a service s by sending a query q. The area of each service s is called the service area. Each service s is required to select suitable computation resources and execute the task graph G_q generated by a query q.

As heuristics, we employ the following two basic policies: *(1) In-situ task allocation*: query's task graph G_q is preferentially allocated and processed by IoT devices in the service area $area(q.s)$ and its extended resource area according to computational demand of q, and *(2) FCFS task allocation*: for the set of queries Q, all tasks of $\cup_{q \in Q} G_q$ are assigned to the available IoT devices in a first-come-first-serve (FCFS) manner.

The system model consists of two types of brokers: *regional broker (or R-broker)* shown in Fig.1 (A) and *service brokers (or S-broker)* shown in Fig.1 (B). The R-broker is a well-known server that all devices know beforehand[1]. The R-broker is responsible for management of all processors (computational resources) existing in the target area and regional IoT services deployed in that area, while the S-broker manages the resource provisioning (task assignment) of a specific service in charge. Each S-broker assigned for each service s carries out admission control of each query for s and scheduling execution of its task graph.

[1]A cloud server or a fog server is typically selected as an R-broker, but a powerful IoT device/gateway can also be selected.

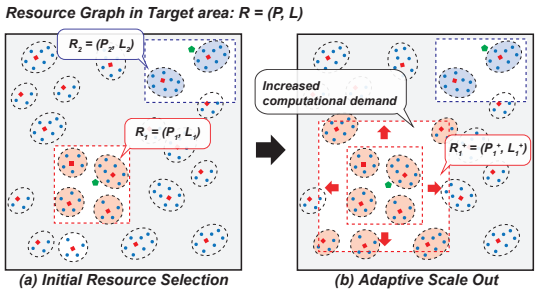

Fig. 2. In-situ initial resource provisioning and Adaptive scale-out

Specifically, the proposed method consists of the following three steps: (Step 1) service activation with initial resource selection as shown in Fig.2 (a), (Step 2) admission control of each query and adaptive scale-out of resource selection area (If the service's computational demand has increased) as shown in Fig.2 (b), and (Step 3) task scheduling by using tabu search algorithm for the admitted query's task graph.

III. Evaluation

We conduct the evaluation of the proposed method in a simulated environment. In the simulation, 2,000 IoT devices are randomly arranged in a target area as shown in Fig 3. Number of R-broker, high power IoT devices (master node), low power IoT devices (slave nodes), IoT services are 1, 600, 1400, and 10, respectively. The parameters related to processing power and network speed are determined based on the actual measurement. Queries are issued to each service s based on the Poisson arrival process of arrival intensity $\lambda = 1/(min)$, and input data size of each task graph G_q is randomly selected in the range of 1 to 20 (MB). Two conventional methods: random task scheduling and greedy task scheduling were set as comparison targets against the two proposed methods (1: tabu search scheduling, 2: tabu search scheduling & adaptive scale out).

Fig. 4 shows cumulative distribution function(CDF) curves of the delay time for processing queries. The average delay time of each method is: 17.83s for random, 14.95s for greedy, 9.79s for proposed method 1, 10.07s for proposed method 2. This result shows that the proposed methods 1 and 2 have shorter delay times than random and greedy methods.

Fig. 5 shows the number of accepted queries for each service. The x-axis is a service ID arranged in ascending order

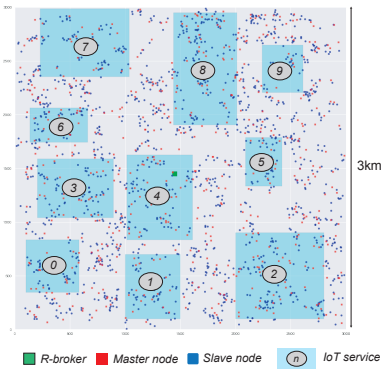

Fig. 3. Simulation environment

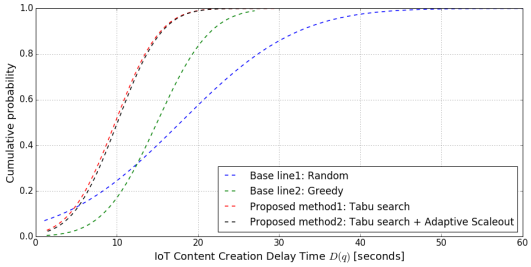

Fig. 4. CDF curve of IoT content creation delay

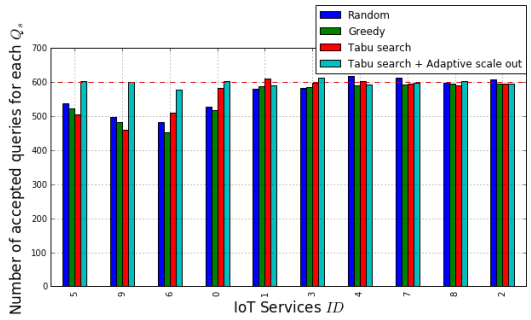

Fig. 5. Number of accepted queries per each service

of service area size. As shown in Fig.5, random, greedy and proposed method 1 show that the number of accepted queries decreases when the area size is small (service ID = 5, 9, 6, 0). On the other hand, in the proposed method 2, almost the same number of queries are accepted regardless of area size. This result shows that many queries can be accepted even for services with small area sizes by using adaptive scale-out.

We summarize the evaluation results as follows. We confirmed that the tabu search scheduling shortens the average delay time for processing queries (Fig. 4) and adaptive scale-out increases the number of accepted queries (Fig. 5). Reducing the delay time means improving the quality of service. An increase in the number of accepted queries means an improvement of service availability. From this, the proposed method contributes to the improvement of QoS.

IV. Conclusion

In this paper, we proposed the in-situ resource provisioning method composed of in-situ resource area selection with adaptive scale-out and in-situ task scheduling based on tabu search technique. The evaluation results from the simulation showed that proposed method shorten delay of query processing and improve Quality of Service (QoS) of users in delay-sensitive IoT services.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Numbers 17J10021, 16H01721 and 26220001.

References

[1] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 5, pp. 37–42, 2015.

[2] K. Yasumoto, H. Yamaguchi, and H. Shigeno, "Survey of real-time processing technologies of iot data streams," *Journal of Information Processing*, vol. 24, no. 2, pp. 195–202, 2016.