Joint optimization of edge server and virtual machine placement in edge computing environments

Ayaka Takeda*, Tomotaka Kimura†, and Kouji Hirata‡
* Graduate School of Science and Engineering, Kansai University, Osaka, Japan
E-mail: k087058@kansai-u.ac.jp
† Faculty of Science and Engineering, Doshisha University, Kyoto, Japan
E-mail: tomkimur@mail.doshisha.ac.jp
‡ Faculty of Engineering Science, Kansai University, Osaka, Japan
E-mail: hirata@kansai-u.ac.jp

***Abstract*—In edge computing environments, edge servers are deployed in networks in addition to centralized cloud servers. To design edge computing systems, we should consider multiple problems such as server placement and virtual machine allocation. This paper focuses on the joint optimization problem of edge sever placement and virtual machine placement. The problem determines location of edge servers in a network, and then allocates virtual machines to the edge servers. To do so, this paper proposes the optimization method based on mathematical programming formulations, which take into account the network load and the edge server load. Through numerical experiments, we show the performance of the proposed optimization methods.**

I. Introduction

Recently, the Internet of Things (IoT) technology has rapidly developed, and many IoT services have been provided to users [2]. IoT services use cloud data centers to process data collected by IoT devices [4]. It is expected that a vast amount of data generated by IoT devices will be accumulated on the cloud via the Internet and that data will be analyzed to create new innovations in near future. However, the amount of data drastically increases with the increase in the number of Internet users and IoT devices, which causes large network delay and large processing delay in cloud data centers. In order to resolve this problem, the edge computing technology has been actively studied [6].

In edge computing environments, in addition to cloud data centers, edge servers are deployed at network edges, which are near from users and IoT devices (Fig. 1). The edge servers process data generated by the IoT devices in a distributed manner, and then work together with the cloud data center to further process the data as necessary. Generally, the data processing in the edge servers is performed by virtual machines corresponding to require services. By processing the data in the edge servers, the load on network links can be distributed, which avoids network congestion and reduce network delay (i.e., queueing delay in routers). Furthermore, the load on the cloud data centers can be reduced, which leads to low processing delay. When designing edge computing systems, we should consider multiple problems such as server placement, virtual machine allocation, and traffic routing so as to efficiently utilize the edge computing environments.

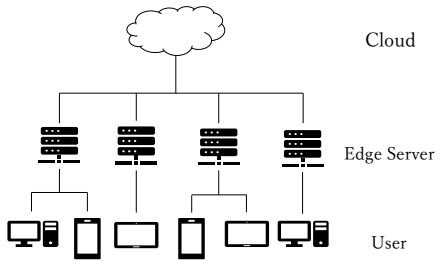

Fig. 1: Edge computing.

In the past, the authors have examined the optimization of edge server placement in [7]. Based on the work, this paper proposes the joint optimization method of edge server placement and virtual machine placement problems. The edge server placement problem determines the location of edge servers in a network. Specifically, it determines which nodes in the network the edge servers are deployed on. Virtual machines are duplicated into the edge servers. The virtual machine placement problem selects edge servers to which virtual machines are allocated. Note that directly solving the joint optimization problem is too difficult from the viewpoint of computational complexity. Therefore, we solve the joint problem in stages. Specifically, we first solve the server placement problem, and then solve the virtual machine placement problem based on the location of the edge servers determined by the server placement problem.

For the edge server placement problem, we adopt two optimization methods providing mathematical programming formulations based on the facility location problem proposed in [7], which takes into consideration the network load or the edge server load. For the virtual machine placement problem, we introduce an optimization method providing a mathematical programming formulation considering the network load. Through numerical experiments, we compare combinations of these optimization methods and show their performance in terms of the network load and the edge server load.

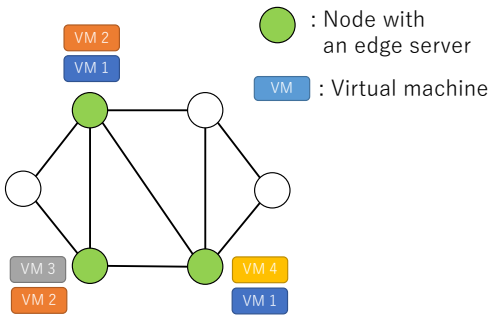

Fig. 2: System model.

II. Edge cloud server placement method and virtual machine placement method

A. System model

Fig. 2 shows the system model assumed in this paper. The symbols used in this paper are listed in Table I. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a given network, where $\mathcal{V}$ denotes a set of nodes and $\mathcal{E}$ denotes a set of links. Let $\mathcal{V}_S \subseteq \mathcal{V}$ and $\mathcal{V}_H \subseteq \mathcal{V}$ denote a set of edge servers and a set of host (i.e., user or IoT device), respectively. Let $\mathcal{N}$ denote a set of distinct virtual machines, each of which provides individual services and can be duplicated into edge servers. Hosts belong to some node. Also, edge servers are placed on some nodes. We here assume that P nodes are selected as edge servers. Each edge server can have multiple virtual machines providing different services. By duplicating virtual machines providing the same service into multiple edge servers, we can balance the load on network links and edge servers. When requiring a service, each host communicate with a corresponding virtual machine in one of edge servers.

In this paper, we propose the joint optimization method of edge server placement and virtual machine placement problems, assuming the above system model. The optimization method aims to minimize the network load and/or the maximum load of edge servers. The edge server placement problem determines the location of edge servers in the network. The virtual machine placement problem selects edge servers to which virtual machines are allocated. Since directly solving the joint optimization problem is too difficult from the viewpoint of computational complexity, we solve the joint problem in two stages. We first solve the server placement problem, and then solve the virtual machine placement problem based on the location of the edge servers determined by the server placement problem. In the following, we introduce the two optimization methods for the server placement problem based on [7], which aim to minimize the network load and the maximum edge server load, respectively. Furthermore, we provide the optimization method for the server placement problem that aims to minimize the network load.

B. Edge server placement problem

TABLE I: List of symbols.

Symbol	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Network consisting of the set $\mathcal{V}$ of nodes and the set $\mathcal{E}$ of links
$\mathcal{V}_C \subseteq \mathcal{V}$	Set of nodes where an edge server can be placed
$\mathcal{V}_H \subseteq \mathcal{V}$	Set of hosts
$\mathcal{V}_S \subseteq \mathcal{V}_C$	Set of edge servers
$\mathcal{N}$	Set of distinct virtual machines
$\mathcal{N}_i \subseteq \mathcal{N}$	Set of distinct virtual machines required by host $i \in \mathcal{V}_H$
$d_{i,j}$	Number of hops from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$ along the shortest path
λ_i	Total amount of traffic required by host $i \in \mathcal{V}_H$
$\lambda_{i,n}$	Amount of traffic transmitted between host $i \in \mathcal{V}_H$ and virtual machine $n \in \mathcal{N}$
P	Number of edge cloud servers to be placed in the network
x_j	Binary variable that is equal to 1 if an edge server placed on node $j \in \mathcal{V}_C$; otherwise, 0
$y_{i,j}$	Binary variable that is equal to 1 if host $i \in \mathcal{V}_H$ communicates with node $j \in \mathcal{V}_C$ where an edge server is placed; otherwise, 0
$z_{i,j,n}$	Binary variable that is equal to 1 if host $i \in \mathcal{V}_H$ communicates with virtual machine $n \in \mathcal{N}_i$ in an edge server $j \in \mathcal{V}_S$; otherwise, 0
$\delta_{n,j}$	Binary variables that is equal to 1 if virtual machine $n \in \mathcal{N}$ is placed on edge server $j \in \mathcal{V}_S$;otherwise, 0
S_n	Size of virtual machine $n \in \mathcal{N}$
C_j	Capacity of edge server $j \in \mathcal{V}_S$
α	Real variable that indicates the maximum edge server load

1) Placement method 1: The first optimization method considers the network load, which is based on the p-median problem [5]. Specifically, it minimizes the sum of the load of network links, which is formulated as the following Integer Linear Programming (ILP) that determines the location of edge servers (i.e., x_j). In the ILP, the decision variables are x_j and $y_{i,j}$.

Minimize

$$\sum_{i \in \mathcal{V}_H} \sum_{j \in \mathcal{V}_C} \lambda_i d_{i,j} y_{i,j} \tag{1}$$

Subject to

$$\forall i \in \mathcal{V}_H; \sum_{j \in \mathcal{V}_C} y_{i,j} = 1 \tag{2}$$

$$\sum_{j \in \mathcal{V}_C} x_j = P \tag{3}$$

$$\forall i \in \mathcal{V}_H, j \in \mathcal{V}_C; \ y_{i,j} \leq x_j \tag{4}$$

(1) is an objective function that minimizes the total amount of traffic requested by each host multiplied by the number of hops from the host to the required edge servers, which means minimizing the network load. (2) represents that each host uses one of the edge servers. (3) represents that the number of edge servers to be placed is equal to P. (4) is the constraint that each host can communicate with a node only when the edge server is placed on it.

2) Placement method 2: The second optimization method aims to minimize the maximum edge server load. It is formulated as the following Mixed Integer Programming (MIP) for the purpose of distributing the load on the edge servers, where

the decision variables are x_j and $y_{i,j}$.

Minimize

$$\alpha \tag{5}$$

Subject to

$$\forall i \in \mathcal{V}_H;\ \sum_{j \in \mathcal{V}_C} y_{i,j} = 1 \tag{6}$$

$$\sum_{j \in \mathcal{V}_C} x_j = P \tag{7}$$

$$\forall i \in \mathcal{V}_H, j \in \mathcal{V}_C;\ y_{i,j} \leq x_j \tag{8}$$

$$\forall j \in \mathcal{V}_C;\ \sum_{i \in \mathcal{V}_H} \lambda_i y_{i,j} \leq \alpha \tag{9}$$

(5) is the objective function indicating that the maximum edge server load given by (9) is minimized. (6)-(8) are the same constraints as (2) to (4) in the placement method 1.

C. Virtual machine placement problem

Based on the location of edge servers (i.e., x_j) determined by the edge server placement problem, we allocate virtual machines to the edge servers so as to minimize the network load. We here assume the situation where each host communicates with corresponding virtual machines. In this paper, we formulate the virtual machine placement problem as the following MIP. It determines edge servers to which respective virtual machines are allocated (i.e., $\delta_{n,j}$) and selects virtual machines which respective hosts use (i.e., $z_{i,j,n}$).

Minimize

$$\sum_{i \in \mathcal{V}_H} \sum_{j \in \mathcal{V}_S} \sum_{n \in \mathcal{N}_i} \lambda_{i,n} d_{i,j} z_{i,j,n} \tag{10}$$

Subject to

$$\forall j \in \mathcal{V}_S;\quad \sum_{n \in \mathcal{N}} \delta_{n,j} S_n \leq C_j, \tag{11}$$

$$\forall i \in \mathcal{V}_H, n \in \mathcal{N}_i;\quad \sum_{j \in \mathcal{V}_S} z_{i,j,n} = 1, \tag{12}$$

$$\forall n \in \mathcal{N};\quad \sum_{j \in \mathcal{V}_S} \delta_{n,j} \geq 1, \tag{13}$$

$$\forall i \in \mathcal{V}_H, j \in \mathcal{V}_S, n \in \mathcal{N}_i;\quad \delta_{n,j} \geq z_{i,j,n}. \tag{14}$$

(10) is the objective function that minimize the network load. (11) represents the capacity constraint of each edge server, which limits the number of virtual machines that can be placed there. (12) means that for each virtual machine, each host communicates with only one edge server. (13) indicates that each virtual machine is duplicated into at least one edge server. (14) is the constraint that each host can communicate only with edge servers having corresponding virtual machines.

TABLE II: Combinations of placement methods.

Label	Edge server placement	Virtual machine placement
Proposal 1	Placement method 1	Proposed MIP
Proposal 2	Placement method 2	Proposed MIP
Random 1	Random placement	Proposed MIP
Random 2-1	Placement method 1	Random placement
Random 2-2	Placement method 2	Random placement
Random 3	Random placement	Random placement

III. Performance evaluation

A. Model

In order to evaluate the performance of the proposed optimization methods, we conduct numerical experiments using networks constructed based on the Barabasi-Albert model [3]. Each node fills the role of a host and an intermediate node. Edge server can be placed on any nodes (i.e., $\mathcal{V} = \mathcal{V}_H = \mathcal{V}_C$). The number P of edge cloud servers placed in the networks is set to 3, 5, and 10. We assume that the route between each host and each edge server is the shortest path in terms of the number of hops. The amount λ_i of traffic required by each host is randomly selected from 0 to 1000 [Mbps]. We assume that the capacity of each link is large enough so that the link can accommodate all traffic flows. The amount $\lambda_{i,n}$ of traffic between each host and each virtual machine is set to the random value such that $\lambda_i = \sum_{n \in \mathcal{N}_i} \lambda_{i,n}$. The size S_n of virtual machine n ($n = 1, 2, \ldots, N$) is equal to $2n$ and the capacity C_j of each edge server is set to 30. In this paper, IBM ILOG CPLEX [1] is used to solve the optimization problems.

B. Results

Figs. 3 and 4 show the network load and the maximum edge server load, respectively, for each placement method listed in Table II, where the number V of nodes is 30 and 60, the number P of placed edge server is 10, and the number N of distinct virtual machines is 8. From this figure, we observe that Proposal 1 can efficiently reduce the network load, but its maximum edge server load is relatively high. This is because it does not take the edge server load into account. On the other hand, Proposal 2 significantly reduces both the network load and the maximum edge server load. This result indicate that the combination of the edge server placement method for the edge load and the virtual machine placement method for the network load works well.

Figs. 5 and 6 show the network load and the maximum edge server load, respectively, as a function of the number N of distinct virtual machines, where the number V of nodes is 30. As we can see from Fig. 5, the network load increases with the number of distinct virtual machines. This is because the number of virtual machines that the edge servers can have is limited. On the other hand, the network load decreases as the number P of edge servers increases. We also observe that Proposal 1 can reduce the network load more efficiently than Proposal 2 for the small values of P. When $P = 10$, the network load of Proposal 1 is almost the same as that of Proposal 2. From Fig. 6, we observe that the maximum edge

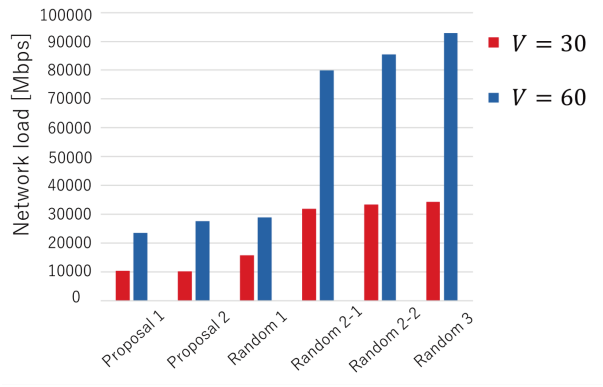

Fig. 3: Network load.

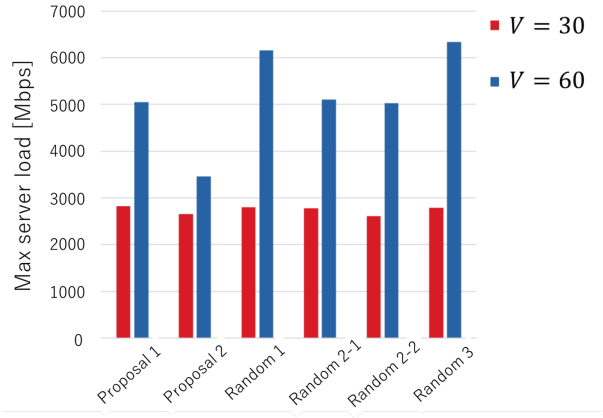

Fig. 4: Maximum server load.

server load of Proposal 2 is basically smaller than that of Proposal 1. These results mean that Proposal 2 has superior performance under situations where many edge servers are placed in the network.

IV. Conclusion

In this paper, we proposed the joint optimization method of edge server placement and virtual machine placement problems. We formulated the optimization problem as mathematical programming, which minimizes the network load and/or the maximum load of edge servers. Through numerical experiments, we showed the performance of the proposed method.

Acknowledgement

This research was partially supported by Grant-in-Aid for Scientific Research (C) of the Japan Society for the Promotion of Science under Grant No. 18K11282.

References

[1] IBM ILOG CPLEX
https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer

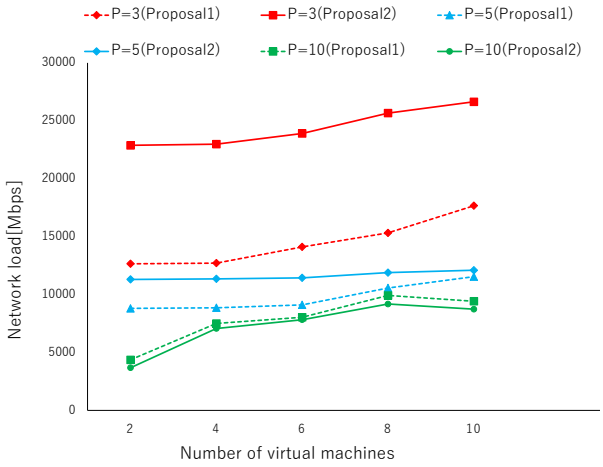

Fig. 5: Network load against the number of distinct virtual machine types.

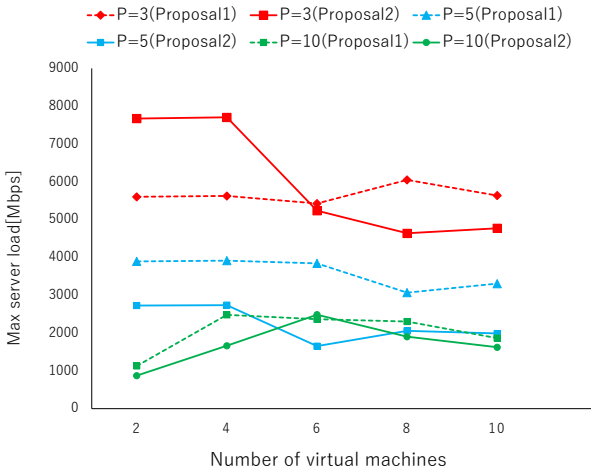

Fig. 6: Maximum edge server load against the number of distinct virtual machines.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey" *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
[3] A. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.
[4] J. Gubbia, R. Buyyab, S. Marusica, and M. Palaniswamia, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 27, pp. 21645-1660, 2013.
[5] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. II: the p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539-560, 1979.
[6] X. Sun and N. Ansari, "Edge IoT: Mobile edge computing for the Internet of Things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22–29, 2016.
[7] A. Takeda, T. Kimura, and K. Hirata, "Evaluation of edge cloud server placement for edge computing environments," in Proc. IEEE International Conference on Consumer Electronics - Taiwan (IEEE ICCE-TW 2019), Yilan, Taiwan, May 2019.