Poster: Edge-cloud Enhancement – Latency-aware Virtual Cluster Placement for Supporting Cloud Applications in Mobile Edge Networks

Xuan Liu, Bo Cheng, Meng Wang, Junliang Chen
Beijing University of Posts and Telecommunications
{liuxuan0527,chengbo,mengwang,chjl}@bupt.edu.cn

ABSTRACT

Mobile edge networks benefit cloud applications in particularly providing a shorter response latency for mobile terminals. However, the cumulative increase of mobile terminals and emerging cloud applications, poses new challenges for edge networks. To combat this issue, the promising idea of mobile micro-clouds (MMCs) is proposed to enhance edges and the cloud. In this paper, we investigate the problem of virtual cluster (VC) placement in MMCs, to minimize the average response latency with various requests among multiple cloud applications. Then a hybrid swarm intelligence approach is proposed to optimize VC placement scheme, for a trade-off between the average response latency and the overall VC placement cost. The preliminary evaluation results show the effectiveness and efficiency of our approach.

KEYWORDS

Mobile edge computing, virtual cluster placement, cloud computing, average response time

ACM Reference Format:
Xuan Liu, Bo Cheng, Meng Wang, Junliang Chen. 2019. Poster: Edge-cloud Enhancement – Latency-aware Virtual Cluster Placement for Supporting Cloud Applications in Mobile Edge Networks. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3300061.3343388

1 INTRODUCTION

The physical limitations, such as memory capacity, processing ability, battery life, make mobile terminals incapable of executing complex computational tasks from many emerging cloud applications[1]. A remedy for this obstacle is allowing the backend cloud to perform complex tasks for mobile terminals. However, this mechanism may result in long request latency and heavy traffic load in the cloud. To alleviate the heavy traffic and provide cloud services in close proximity to the mobile terminals, mobile edge computing has been introduced to integrate mobile computing, network control and storage to edge networks [1, 2]. The newly emerging idea of mobile micro-clouds (MMCs) is introduced to make mobile terminals fast and reliably access cloud services [3]. As illustrated in Figure 1, MMCs are connected to the backend cloud by backhaul network at the network edge. Each MMC supports the backend cloud with a small-size cloud and provides cloud services to mobile terminals.

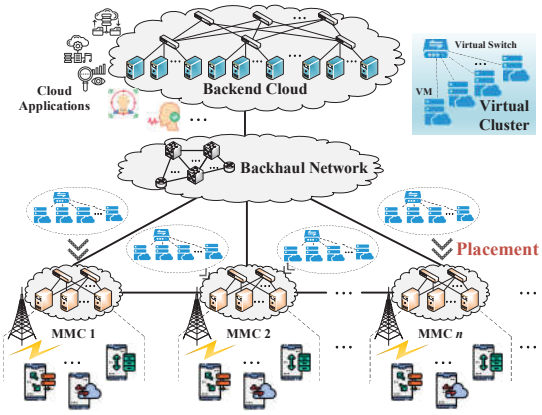

Figure 1: Diagram of VC placement in MMCs.

As the study in [4] indicates, deploying virtual machines (VMs) supporting cloud applications in edges can not only improve the network resource utilization, but also greatly enhance the QoE by meeting various critical requirements of response delay. The study in [5] places VM replicas in mobile edge network servers, with the consideration that a VM replica on a server is able to support a cloud application. To allow each cloud application to specify resource demands and construct an inner-connecting virtual network in MMCs, we employs virtual cluster (VC) placement for supporting

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico

ACM ISBN 978-1-4503-6169-9/19/10.
https://doi.org/10.1145/3300061.3343388

cloud applications in MMCs. As shown in Figure 1, a VC describes a star-shaped topology connecting virtual machines to a virtual switch. A high-level comprehensive survey on VC placement problem has been discussed in the study [6]. Despite of the promising benefits brought by VC placement in MMCs, the cost of VC placement increases as more VCs are placed in MMCs [3, 4].

In this paper, we build an average response latency model and an overall VC placement cost model. Then we formulate the latency-aware VC placement problem as a joint optimization problem. A swarm intelligence approach is proposed to resolve this optimization problem. Our work also raises some interesting open problems, which are discussed in the last section.

2 PROBLEM FORMULATION

Consider that the edge network is modeled by a connected graph $G = (V, E)$, where $V = \{v_1, v_2, \dots v_i, \dots\}$ is the set of MMCs, and $E = \{e_1, e_2, \dots e_k, \dots\}$ is the set of physical links among MMCs. The set of cloud applications is denoted as $A = \{a_1, a_2, \dots a_j, \dots\}$, and there are n_i mobile terminals within the service region of MMC v_i. We denote the $|V| \times |A|$ matrix X as the VC placement scheme, where $|V|$ and $|A|$ are the number of MMCs and applications, respectively. The element x_{ij} of X indicates that a VC supporting application a_j is placed in MMC v_i when $x_{ij} = 1$; otherwise, $x_{ij} = 0$.

For all requests of application a_j in those received requests, if there is no VC supporting application a_j in MMC v_i, these a_j related requests will be distributed to other MMCs or the backend cloud [5]. We use $\theta^j_{i,o}$ to represent the proportion of requests of application a_j from MMC v_i to MMC v_o, with $\theta^j_{i,o} \in [0, 1]$. Thus, the proportion to the backend cloud is $1 - \sum_o \theta^j_{i,o}$.

Edge Network Latency. Assume that L^j_k denotes the network latency for the request of application a_j on the link e_k, and $I^k_{i,o}$ denotes whether the transmission path from MMC v_i to MMC v_o includes the link e_k. We use $\mathcal{T}^j_{i,o}$ to represent the overall edge network latency for requests of application a_j from MMC v_i to MMC v_o, as formulated in the equation (1), where ϕ^j_i is the ratio of application a_j requests to the number of mobile terminals n_i within the service region of MMC v_i; and $|E|$ is the size of the set E.

$$\mathcal{T}^j_{i,o} = \theta^j_{i,o} \cdot (\phi^j_i \cdot n_i) \cdot \left(\sum_k^{|E|} L^j_k \cdot I^k_{i,o} \right) \cdot x_{oj} \tag{1}$$

Processing Latency in MMC. Assume that the request generation of application a_j follows a *Poisson Distribution* with the average rate λ_j. The equation (2) formulates the average request arrival rate of application m_j in MMC v_i, denoted as R^j_i.

$$R^j_i = \sum_o^{|V|} \theta^j_{o,i} \cdot (\phi^j_o \cdot n_o) \cdot \lambda_j \cdot x_{oj} \tag{2}$$

Let μ^j_i represent the average service rate of application a_j in MMC v_i. We assume that μ^j_i is larger than R^j_i, to guarantee the stability of the whole system. Actually, we can adjust the tunable proportion parameter $\theta^j_{i,o}$ to guarantee this constraint. Therefore, the average processing latency of application a_j in MMC v_i, denoted as $\mathcal{P}^j_i$, is shown in the equation (3), where $\mu^j_i > R^j_i$.

$$\mathcal{P}^j_i = \frac{R^j_i}{\mu^j_i - R^j_i} = \frac{\sum_o^{|V|} \theta^j_{o,i} \cdot (\phi^j_o \cdot n_o) \cdot \lambda_j \cdot x_{oj}}{\mu^j_i - \sum_o^{|V|} \theta^j_{o,i} \cdot (\phi^j_o \cdot n_o) \cdot \lambda_j \cdot x_{oj}} \tag{3}$$

Network Latency to Backend Cloud. As mentioned above, when MMC v_i can not provide enough computational resources to mobile terminals for application requests, that is, when computational nodes suffer heavy overload in MMC v_i, some of those requests will be dispatched to other MMCs or the backend cloud. Assume that b_j denotes the average network latency of backend cloud for requests of application a_j. Therefore, the overall network latency to backend cloud for serving requests of application a_j from edges, denoted as $\mathcal{B}_j$, is formulated as shown in the equation (4).

$$\mathcal{B}_j = b_j \cdot \sum_i^{|V|} \phi^j_i \cdot n_i \cdot \left(1 - \sum_o^{|V|} (\theta^j_{i,o} \cdot x_{oj})\right) \tag{4}$$

Average Response Latency Minimization. The latency for offloading application requests from mobile terminals contains following portions: 1) wireless transmission latency of uplink application requests from mobile terminals to the corresponding MMC; 2) network latency of distributing application requests in MMCs; 3) network latency of processing requests in MMC; 4) network latency to backend cloud; 5) wireless transmission latency of downlink response results towards mobile terminals. Note that different VC placement schemes will not affect the wireless transmission latency of uplink requests and downlink response. To formulate the average response latency, we previously calculate the overall application requests from all mobile terminals, denoted as R^* and presented in the equation (5).

$$R^* = \sum_j^{|A|} \sum_i^{|V|} (\phi^j_i \cdot n_i) \tag{5}$$

Consequently, minimization of the average response latency is formulated in the equation (6) with constraints (7)-(10), where $\mathcal{F}(X)$ denotes the average response latency function. We use D_j and C_i to denote the resource demand of application a_j and the resource capacity of the MMC v_i, respectively. Note that D_j and C_i are summary notations, which may contain a group of resource objectives. We use "$\preceq$" to denote "no more than" relation between D_j and C_i.

$$\min : \mathcal{F}(X) = \frac{1}{R^*} \cdot \sum_{j}^{|A|} \left(\mathcal{B}_j + \left(\sum_{i}^{|V|} \left(\mathcal{P}_i^j + \sum_{o}^{|V|} \mathcal{T}_{i,o}^j \right) \right) \right) \quad (6)$$

$$\text{s.t. } x_{ij}, x_{oj} \in \{0,1\}, \ \forall v_i, v_o \in V, \ \forall a_j \in A \quad (7)$$

$$\text{s.t. } \sum_{o}^{|V|} \theta_{i,o}^j \leq 1, \ \forall v_i \in V, \ \forall a_j \in A \quad (8)$$

$$\text{s.t. } \theta_{i,o}^j - x_{oj} \leq 0, \ \forall v_i, v_o \in V, \ \forall a_j \in A \quad (9)$$

$$\text{s.t. } \sum_{j}^{|A|} x_{ij} \cdot D_j \leq C_i, \ \forall v_i \in V, \ \forall a_j \in A \quad (10)$$

Overall Cost of VC Placement in MMCs. As the work in [5] indicates, the overall cost of VC placement can be considered as the synchronization between VCs and resource demand of VCs. The overall VC placement cost function, denoted as $C(X)$, can be formulated as a quadratic function in the equation (11), where $\sigma_j \cdot \sum_i^{|V|} x_{ij}$ is the weighted number of VCs supporting application a_j. We use the weight σ_j to facilitate calculating the overall cost of VC placement.

$$C(X) = \omega_1 \cdot \left(\sum_{j}^{|A|} (\sigma_j \cdot \sum_{i}^{|V|} x_{ij}) \right) + \omega_2 \cdot \left(\sum_{j}^{|A|} (\sigma_j \cdot \sum_{i}^{|V|} x_{ij}) \right)^2 \quad (11)$$

Joint Optimization Function. The average response latency $\mathcal{F}(X)$ and the overall cost of VC placement in MMCs $C(X)$ are two conflicting optimization objectives. If each MMC has corresponding VCs for supporting each applications, all application requests from mobile terminals can be served in the nearest MMC. In this case, the average response latency will be the minimum $\mathcal{F}_{min}$, but the overall cost of VC placement will be the maximum C_{max}. Conversely, if no VC is placed on any MMC, all application requests need to be delivered to the backend cloud. In this case, the average response latency will be the maximum $\mathcal{F}_{max}$, but the overall cost of VC placement will be the minimum C_{min}. We formulate a joint optimization function as follows, where $0 < \alpha < 1$.

$$\min : \alpha \cdot \frac{\mathcal{F}(X) - \mathcal{F}_{min}}{\mathcal{F}_{max} - \mathcal{F}_{min}} + (1-\alpha) \cdot \frac{C(X) - C_{min}}{C_{max} - C_{min}} \quad (12)$$

$$\text{s.t. Constraints (7)-(10)} \quad (13)$$

3 PROPOSED APPROACH

With analyses in [5, 6], the VC placement problem is NP-hard and commonly needs heuristic algorithms to solve. According to the *No-Free-Lunch Theorem* in optimization problems [7], no heuristic algorithm is categorically suitable for the VC placement optimization problem and better than any other algorithms. One of the main reasons is that most of the heuristic algorithms are based on stochastic operations. We propose a swarm intelligence approach (SIA) that assembles several effective and efficient heuristic algorithms, to obtain as optimal as possible VC placement schemes. We conduct our preliminary evaluation with settings following the works in [3, 5]. As Figure 2 shows, SIA performs seven eligible algorithms to find the optimal solution. In brief, these seven algorithms include a greedy algorithm (RFF), a graph-based approximation algorithm (MKC), and some population-based heuristics. The box plots show that no algorithm is absolutely better than another one.

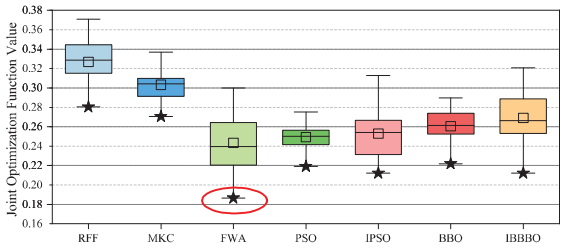

Figure 2: Box plots of JOF values with 30 runs

4 ONGOING WORK

We plan to further study the VC placement on edges, and raise pending problems as follows. How to dynamically place VCs in MMCs for adaptively serving cloud application requests? If those collected algorithms in SIA exchange and share solution information, can SIA obtain a better solution? If so, what mechanism with do those algorithms exchange and share solutions? We attempt to answer the questions in the ongoing work.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No. 2017YFB14006 03) , Natural Science Foundation of China (No.61772479).

REFERENCES

[1] T. G. Rodrigues, K. Suto, H. Nishiyama and N. Kato. Hybrid Method for Minimizing Service Delay in Edge Cloud Computing Through VM Migration and Transmission Power Control. *IEEE Transactions on Computers*, 66, 5 (2017), 810-819.

[2] Y. Mao, C. You, J. Zhang and K. B. Letaief. Mobile Edge Computing: Survey and Research Outlook. *CoRR*, abs/1701.01090 (2017).

[3] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer and K. K. Leung. Dynamic Service Placement for Mobile Micro-Clouds with Predicted Future Costs. *IEEE Transactions on Parallel and Distributed Systems*, 28, 4 (2017), 1002-1016.

[4] Y. Zhou, F. R. Yu, J. Chen and Y. Kuo. Resource Allocation for Information-Centric Virtualized Heterogeneous Networks With In-Network Caching and Mobile Edge Computing. *IEEE Transactions on Vehicular Technology*, 66, 12 (2017), 11339-11351.

[5] L. Zhao and J. Liu. Optimal Placement of Virtual Machines for Supporting Multiple Applications in Mobile Edge Networks. *IEEE Transactions on Vehicular Technology*, 67,7 (2018), 6533-6545.

[6] M. Rost, C. Fuerst, and S. Schmid. Beyond the Stars: Revisiting Virtual Cluster Embeddings. *ACM SIGCOMM COMP COM*, 45, 3 (2015), 12-18.

[7] A. J. Lockett and R. Miikkulainen. A Probabilistic Reformulation of No Free Lunch: Continuous Lunches Are Not Free. *Evolutionary Computation*, 25, 3 (2017), 503-528.