# Dynamic Mobile Cloudlet Clustering for Fog Computing

Yuanjie Li, Nguyen Tung Anh, Azhar Saeed Nooh, Kuwon Ra,
Minho Jo
***Department of Computer Convergence Software, Korea University, Sejong Metro, S. Korea***
*wongul929@korea.ac.kr, {tunganhbgbk, asknooh}@gmail.com, kwla103@naver.com, minhojo@korea.ac.kr*

## Abstract

*Although mobile devices have obtained more and more capabilities such as CPU power, memory, and battery, they still cannot satisfy the requirement of high resource consuming applications. Fog Computing is one of the solutions for offloading the task of a mobile. However the capability of fog server is still limited due to the high deployment cost. In this paper, we propose a dynamic mobile cloudlet cluster policy (DMCCP) to use cloudlets as a supplement for the fog server for offloading. The main idea is that by monitoring each mobile device resource amount, the DMCCP system clusters the optimal cloudlet to meet the requests of different tasks from the local mobile device. In simulation results, we demonstrated that our proposed policy can cluster the approximated optimal cloudlets for different resource demands of a task in a real-time processing time compared to the optimal method.*

**Keywords:** Cloudlet, Fog computing, Mobile cloud computing, Dynamic cloudlet cluster, Offloading

## 1. Introduction

Nowadays, mobile devices have become a necessary part of human life all over the world. However, the computing power of each mobile device is still not enough to satisfy the high requirement of applications. Cloud computing is one of the solutions. Cloud computing plays a remote server role for sharing the massive amount of on-demand services and provides flexible information technology (IT) resources for different applications demand. Lots of enterprises take advantage of cloud computing to allow high-performance services on mobile applications for the users. Despite cloud computing benefits, it faces the challenges of solving delayed response caused by the far-away distance between cloud and users.

To solve the weak points of cloud computing, many researchers have proposed their techniques and schemes relate by using fog computing paradigm. Fog computing, known as the mobile edge computing (MEC) defined by European Telecommunications Standards Institute (ETSI), brings the remote services more closely to the users. This new fog server can run as close as possible to users. The fog server deploys at the edge of the network

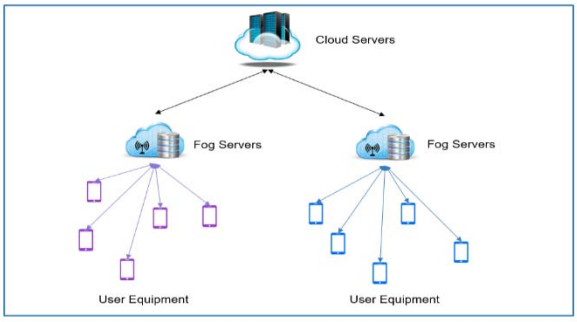


**Figure 1. Simplified fog computing architecture.**

such as access points (AP) [1]. As shown in **Fig. 1,** the fog computing architecture consists of cloud servers, fog servers, and end-user equipment. In this architecture, each user equipment connects with one of fog servers, and the fog server can communicate with each other through a cloud server. Fog server at the edge of the network provides low latency and location awareness services which increases the quality-of-service (QoS) for real-time demand response applications.

Since fog servers should be closer to the users, they are required to be deployed at local sites as many as possible. In order to reduce the deployment costs, fog servers should be light-weight, but unfortunately it is limited with IT resource [2]. Because the limitation of the capacity of fog server, the fog server will not be able to handle them in a short time if receives many requests simultaneously from end-user equipment. Recent research has proposed the use of clusters of mobile devices called cloudlet as a supplement for the cloud for offloading [3].

Cloudlets are accessible by the user equipment via a wireless connection. As defined in [4], cloudlets are a cluster of computers or mobile devices that provide the computing power for use by nearby mobile devices. Cloudlets have so far been proposed to assist mobile users for data storage and task processing.

In this paper, we propose a dynamic mobile cloudlet cluster policy (DMCCP) for fog server offloading. We assume that the fog server will dynamically cluster the cloudlet corresponding to different resource demands of local mobile device task. As shown in **Fig. 2,** all available mobile devices, which are potential IT resource for cloudlet should be pre-connected to a fog server via wireless network. In order to efficiently use the potential IT resource pool, we use the mobile resource monitor

(MRM) in fog server to observe available resources amount of each mobile device [5]. All the information of the mobile device such as remaining central processing unit (CPU), memory, and battery level will be sent to fog server when fog server requests for the monitoring data through MRM. If a single mobile device requests a task to the fog server, the fog server will cluster a cloudlet by DMCCP based on resource demands of this specific task.

In experimental simulation, we evaluated the performance of the DMCCP. The results show that our DMCCP can cluster the approximated optimal cloudlets for different resource demands of a task in a short processing time compare to the optimal method.

The remaining part of this paper is organized as follows.

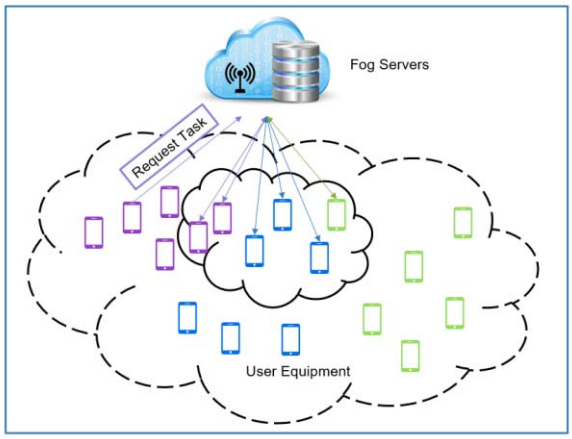


**Figure 2. Mobile cloudlet architecture for fog server.**

In Section 2, we explain our proposed policy, mathematical formulations, and heuristic method for DMCCP. We show the experimental result in Section 3. In the last, we conclude our proposed work.

## 2. Proposed cloudlet clustering policy and algorithm

Before clustering a cloudlet, we use MRM to monitor the status of every available mobile device. **Table 1** shows the criteria of resource status which addresses current mobile device resource availability. Status of CPU (%), memory (%) and battery (%) represents the current remaining CPU power, memory and battery life of a mobile device, respectively. Status of network capacity (%) represents the current network availability of the mobile device. Application means how much an application task of the local mobile device can be carried out by other nearby available mobile device. We assume that M1 represents a local mobile device which requests a task from its application (APP), M2 represents nearby mobile device having the APP and M3, another nearby mobile device which does not the APP. In this case, M2's application status becomes 100, and M3's application status becomes 0. We consider M2 to provide more computing power than M3 in a cloudlet.

**Table 1. Mobile device resource status.**

| Device Resource (%) | Weight |
|---|---|
| CPU | 0.2 |
| Memory | 0.2 |
| Battery | 0.3 |
| Network Capacity | 0.2 |
| Application | 0.1 |

The MRM in fog server will periodically request for above resource status data from all mobile devices and update the received data in the fog server.

In our work, the resource amount calculation formula is proposed for evaluating each mobile device resource amount. Resource amount $\boldsymbol{\sigma(r_i)}$ is computed by the following equation:

$$\sigma(r_i) = W^R \times S$$

$$S = \begin{bmatrix} Remaining\ CPU(\%) \\ Remaining\ memory(\%) \\ Remaining\ battery(\%) \\ Network\ Capacity(\%) \\ Application \begin{cases} 100\ if\ application\ is\ setup \\ 0\ if\ application\ not\ available \end{cases} \end{bmatrix}$$

Symbol $\boldsymbol{\sigma(r_i)}$ denotes the total resource amount of mobile device $\boldsymbol{r_i}$ which can be provided for cloudlet. $\boldsymbol{W^R}$ is weight of the different mobile resources, and $\boldsymbol{S}$, set of resources status.

In our work, the weight of mobile resources is determined by an observation-based heuristic approach. Based on the observation of usage of each resource in real mobile devices, the remaining battery status, compared to other resources status, gives the most significant contribution to the mobile device resource amount. For example, Lt us assume mobile device M1 has high remaining CPU power and memory with low battery level. M2 has medium remaining CPU power and memory with high battery level. When fog server starts to cluster a cloudlet by DMCCP, due to low battery level fog server will not choose M1 as a member of cloudlet even if M1 can provide more computing power than M2. Thus in **Table 1,** we scored 30% of total weight for remaining battery level and in the same way, 20% of total weight, for remaining CPU, memory and current network capacity, it occupies respectively. For the application status, if the current mobile device has installed an APP, we set the score to be 100. If not, we set the point to be '0'.

For the optimal method to cluster the cloudlet, we need to consider every possible cloudlet among available mobile devices. However, based on our simulation results shown in Section 3, as the number of available mobile devices increases, the optimal method shows exponential growth of clustering time. Thus the optimal method cannot be used in real world. Hence, we propose a heuristic method called DMCCP to achieve our goal:

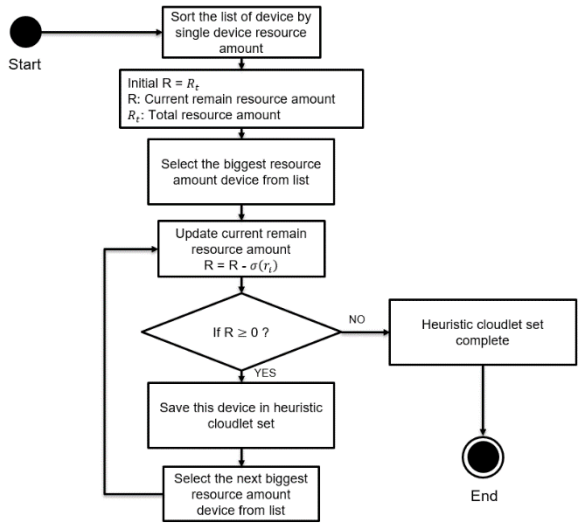


**Fig. 3. Heuristic method flowchart for DMCCP.**

**Fig. 3.** Illustrates the proposed heuristic method flowchart for DMCCP. The detailed algorithm steps are as follows:

**Step 1.** When the fog server receives a task from a local device, DMCCP will calculate the total resource amount ($R_t$) required by the task. Meanwhile, DMCCP sorts all current available mobile devices from big resource amount $\boldsymbol{\sigma(r_0)}$ to small resource amount $\boldsymbol{\sigma(r_i)}$, and stores these available mobile devices in sequence in the list.

**Step 2.** Initialize the current remaining resource amount $R$ equal to total resource amount required by the task $R_t$.

**Step 3.** Select one available mobile device with the biggest resource amount from the list, subtract the biggest one from the $R$ and update new $R$.

**Step 4.** If the current $R$ is still greater than or equal to zero, then save this available mobile device in heuristic cloudlet set, which means this mobile device is selected as a member of cloudlet.

**Step 5.** Select the next available mobile device with the second biggest resource amount from the list, subtract the second biggest one from $R$ and update new $R$.

**Step 6.** Repeat the loop from Step 4 to Step 5 until the current $R$ violates the condition in Step 4 and jump to Step 7.

**Step 7.** The fog server starts to cluster the cloudlet based on the current heuristic cloudlet set.

## 3. Experimental Result

We evaluate the performance of the optimal method and our proposed heuristic method DMCCP under different numbers of available UEs and tasks.

For the experiment, we clustered cloudlets randomly in fog server for performance comparison with the optimal and DMCCP method.

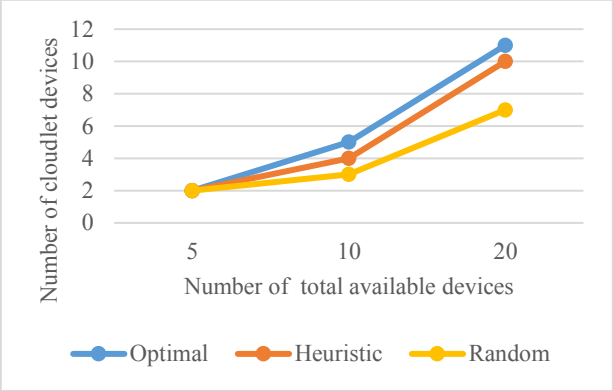


**Figure 4. The number of cloudlet devices depending on a different number of total available mobile devices for fog server.**

As shown in **Fig. 4**, we compare the ability of each method to find the potential cloudlets under the limited number of available mobile devices. The results show that the optimal method can find the largest number of potential cloudlets, and our proposed heuristic DMCCP also can find the number of potential cloudlets, approximated to the optimal number. The random method shows lousy performance compared to other methods.

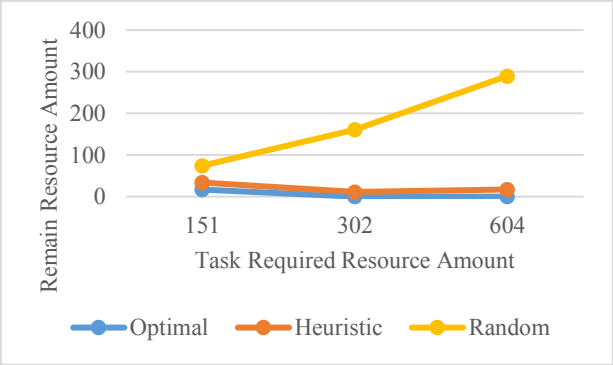


**Fig. 5. The remaining resource amount depending on different total resource amount required by a task.**

**Fig. 5** represents how much resource amount is left in a local device after cloudlet clustering. Less remaining resource amount indicates the higher performance of offloading by cloudlet.

According to the results, the performance of our proposed heuristic DMCCP approximately reaches the optimal method. Although the optimal method can cluster the optimal cloudlet for a task, it shows terrible clustering time in the simulation due to NP-hard time complexity. On the contrary, the clustering time of the proposed heuristic DMCCP is less than one millisecond. Therefore, this experimental result confirms our proposed heuristic DMCCP is the rational choice for clustering cloudlet.

## 4. Conclusion

In this paper, we proposed an efficient heuristic DMCCP for fog server offloading. Our simulation results

show that the proposed real-time DMCCP can be practically used to overcome limited resources of fog computing while its performance is almost close to the optimal method. For future research, we will try to balance the resource amount of cloudlets under required execution time.

## References


[1] Stojmenovic, Ivan, and Sheng-Wen. "The fog computing paradigm: Scenarios and security issues." *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on. IEEE*, 2014.

[2] Luan, Tom H., et al. "fog computing: Focusing on mobile users at the edge." *arXiv preprint arXiv:1502.01815*, 2015

[3] S. Clinch, J. Hawkes, A. Friday, N. Davies, and M. Satyanarayanan, "How close is close enough? Understanding the role of cloudlets in supporting display appropriation by mobile users," *in Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on. IEEE*, pp. 122–127, 2012.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. "The case for VM-based cloudlets in mobile computing." *IEEE Pervasive Computing*, 8(4):14 –23, oct.-dec. 2009.

[5] Kwon K, Park H, Jung S, et al. "Dynamic Scheduling Method for Cooperative Resource Sharing in Mobile Cloud Computing Environments." *KSII Transactions on Internet and Information Systems,* 10.2: 484-503, 2016.