

Article

# Hierarchical Mobile Edge Computing Architecture Based on Context Awareness

Juyong Lee <sup>1</sup>  and Jihoon Lee <sup>2,\*</sup>

<sup>1</sup> Department of Electronic and Information System Engineering, Sangmyung University, Cheonan 31066, Korea; juyonglee0208@gmail.com

<sup>2</sup> Department of Smart Information and Telecommunication Engineering, Sangmyung University, Cheonan 31066, Korea

\* Correspondence: vincent@smu.ac.kr; Tel.: +82-41-550-5441

Received: 27 June 2018; Accepted: 12 July 2018; Published: 17 July 2018



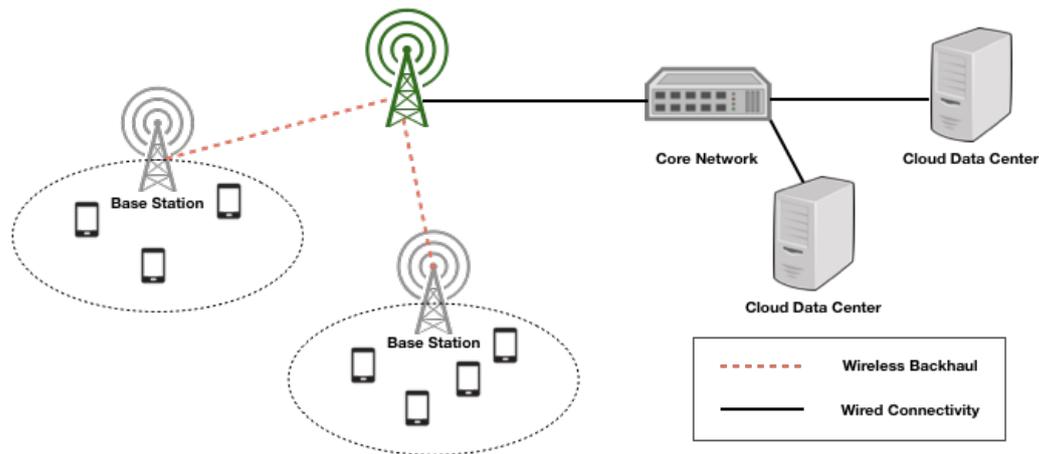
**Abstract:** Due to the recent developments in mobile network technology and the supply of mobile devices, services that require high computing power and fast access speed, such as machine learning and multimedia streaming, are attracting attention. Mobile Edge Computing (MEC) has emerged. MEC allows servers to be located close to users to efficiently handle these services and provides users with ultra-low latency content delivery and powerful computing services. However, there has been a lack of research into the architecture required to efficiently use the computing power and resources of MEC. So, this paper proposes hierarchical MEC architecture in which MEC servers (MECS) are arranged in a hierarchical scheme to provide users with rapid content delivery, high computing performance, and efficient use of server resources.

**Keywords:** mobile edge computing; multi-access edge computing; mobile edge computing architecture; hierarchical mobile cloud computing architecture

## 1. Introduction

With the explosion of mobile and Internet of Things (IoT) devices and the evolution of mobile networks, a variety of services that require low latency and high speed are emerging. To provide services that meet these requirements, cloud computing services, such as Dropbox, Google Cloud Platform, and Amazon, have emerged. Also, with the development of many studies and technologies related to cloud computing, cloud computing could provide a well established distribution model and application platform. Thanks to the computing paradigm, there is growing interest in new applications and services. For example, there are real-time online games, powerful computing operations like machine learning and ultra-high definition (UHD) streaming that requires very low latency and high access speeds. In the meantime, due to the need for high computing power by such services, the cloud computing platform has been considered as a solution to handle such services. Figure 1 shows the existing cloud computing environment. Even though existing cloud computing platforms have performed well, computation-intensive services with their powerful computing capabilities have trouble, and they cannot efficiently use cloud computing resources because there is no consideration of situations requiring ultra-low latency and high computing performance.

Furthermore, the enormous amount of data exchanged between user devices and remote cloud servers causes a data tsunami, which leads to saturation and brings down the back-haul networks. In order to solve such problems, new network architectures, named Fog Computing and Mobile Edge Computing (MEC), which place a small cloud with various computing functions close to the user devices, are under active research [1–5].



**Figure 1.** The existing cloud computing environment.

The initial concept of Fog Computing was proposed by Cisco, which is a service infrastructure that accesses the computation and storage resources of network equipment. This service provides data computation, storage, and application services, like the cloud, but it does not feel like a central server. That is, Fog Computing seamlessly extends cloud computing to the edge for the secure control and management of domain specific hardware, software, storage, and network functions within the domain and enables secure rich data processing applications across the domain. Similarly, the concept of MEC [6] was proposed by the European Telecommunications Standard Institute (ETSI) and is defined as a new platform that provides IT and cloud computing capabilities within a radio access network (RAN) in close proximity to mobile users. In other words, MEC, which is considered a promising and emerging paradigm for the provision of powerful computing capabilities in proximity to mobile devices in 5G networks, enables fast and popular content delivery of delay-sensitive applications at the back-haul capacity of limited mobile networks. In this paper, we focus on a proposed hierarchical architecture based on the MEC environment. Although many studies on the MEC environment have been conducted, existing works have little regard for the efficient arrangement and structure between the central cloud server and the MEC server (MECS) [7–10]. In addition, there is a problem in that MECS and cloud computing resources cannot be efficiently used due to insufficient consideration of operations specific to the content and computing operation types in the MEC environment; thus, the user service is delayed.

So, this paper proposes an architecture that enables users to provide low-latency services and efficient use of server resources by hierarchically arranging MECS specialized for content type and computing operation type. The basic idea of the proposed architecture is to efficiently utilize resources of MECS and to provide users with services specific to their content type and computing type based on operations through a hierarchical MECS configuration with service profile information.

The rest of the paper is organized as followed. Section 2 describes the existing cloud computing architecture. Section 3 presents the proposed hierarchical MEC architecture and then, we present the evaluation results. Finally in Section 5, we conclude this paper.

## 2. Problems of the Existing Cloud Computing Architecture

### 2.1. Propagation Distance and Multi-User Access Problem

In order to deliver real-time multimedia content and the results of computing operations to users quickly, the content–propagation distance is critical to the transmission latency. However, in the existing mobile cloud computing environment, mobile consumer devices need to deliver data of interest to the data center, which is located a long distance from the location of the data. Also, this central mobile cloud computing brings about long latency, because various types of data are transmitted

across multiple networks, including wireless access networks, back-haul networks, and the Internet, which require their own traffic control, routing, and network management tasks. There have been a lot of attempts to solve these problems, for example, by using a cloudlet that integrates several cloud servers and MECS located near the user [11–13]. However, these alternatives also require a large number of computing resources to be shared when lots of users send requests to a server are located at the edge and the cloudlet, resulting in increased processing latency for the computing requests and high delivery latency for the content requests due to system resource sharing [14,15]. Thus, low service latency and fast computing operations are not guaranteed if the content requester is far away from the content source, or when there is a large number of users connect to the MECS or the cloud data center.

### 2.2. Cloud Computing Resource Management Problem

Multimedia services require not only many computationally intensive tasks (i.e., encoding, decoding, and transcoding, etc.) but a large storage capacity. Moreover, computing-related services that require robust memory and computing performance (i.e., machine learning, image analysis, etc.) lead to high energy consumption. Because of their compact size, mobile and IoT devices have limited energy capacity; therefore, computing power and high energy consumption are key factors. There is a possibility for unnecessary energy consumption, resulting from wasted computing resources, since the provision of services suited to the characteristics of the cloud and MEC server and the efficient server arrangement structure are not considered [16,17]. So, excessive energy consumption and inefficient processing overhead happen when a large number of user requests is received, which leads to resource waste, because existing schemes adopt a fixed resource allocation policy for user requests.

### 2.3. Task Migration Problem

Existing cloud computing services operate in a way that users delegate their own processing to a data center that is rich in computing resources specific to a specific task. This method is efficient in terms of energy efficiency, because the data center handles tasks that cannot be processed due to the limited resources of the user's device. However, when a large number of user requests are concentrated in the data center, the computing resources are not properly allocated, and therefore, long service latency is caused. This has become a critical factor for multimedia services.

### 2.4. Improved Edge Cloud Approaches

To solve the aforementioned problems of the central cloud system, the hierarchical cloud architecture was presented. Kiani, et al. [18], proposed a hierarchical mobile edge computing (HI-MEC) architecture. It includes three hierarchical levels of field with shallow and deep cloudlets in Long Term Evolution-Advanced (LTE-advanced) mobile network environments. The HI-MEC scheme allocates cloudlet resources and communication resources to mobile users in order to maximize the profit of a service provider. It is assumed that the quantity of computing resources for mobile users is given. In contrast, our proposed scheme considers various kinds of content and therefore, provides dynamic task migration. Anagnostopoulos, et al. [19] proposed a sensor centric context-awareness system. The sensor centric system includes contextual sensing information and an abstraction model. However, it does not consider the edge computing system. Liang et al. [20] presented a hierarchical edge cloud architecture to efficiently serve the peak loads from users. It includes a workload placement algorithm to execute a mobile program on the edge. However, it lacks consideration of the workload placement according to content type, and so it cannot provide dynamic migration of workload. Meanwhile, our proposed scheme targets the MEC environment and is designed to perform dynamic task migration based on the context information of user applications, which guarantees fast service responses by the hierarchical MEC server architecture.

### 3. Proposed Context-Aware Hierarchical Mobile Edge Computing Architecture

The point of the proposed context-aware hierarchical mobile edge computing (CoHEC) architecture is to hierarchically deploy MECS at the edge of existing mobile cloud infrastructures in order to provide users with low-latency content delivery as well as efficient use of computing resources and services tailored to various applications. We assumed that the mobile cloud data center was the root of the proposed scheme. First, the proposed CoHEC architecture underwent a profile-based performance scoring process for MECS pre-configured by a network operator or an internet service provider (ISP) to efficiently use edge cloud computing resources and provide services tailored to the various applications of users. Then, when the user requests computing task migration and content delivery to the MECS, each task is executed at the MECS with the computing capability suitable for the service. If the task requires a more powerful computing function, the current MECS migrates the task to other MECS in the upper hierarchy. The detailed operations are as follows.

**Phase 1. Edge Performance Scoring:** To utilize MECS efficiently, an “Edge Performance Scoring” process is performed that measures the system’s own computing capabilities and selects hierarchy layers. For this, each MECS sends a Hierarchy Edge Profile Request (HEPR) message to the central mobile cloud data center to request a profile to perform the Edge Performance Scoring process.

Figure 2 shows the format and examples of HEPR messages, and Table 1 shows their fields and meanings. The Edge Identifier field of the HEPR message is an identifier that uses the SHA-256 hash value used for the mobile central cloud data center to identify the MECS requesting the profile. The HEPR field indicates the profile request message, and a timestamp records the starting time of the transmission of the HEPR message in order to recognize the communication status between the mobile central mobile cloud data center and the MECS. On receiving the HEPR message, the central mobile cloud data center sends an HEPR Acknowledgment (HEPRA) message including the profile to the MECS. Figure 3 shows the format and examples of HEPRA messages. The profile contained in the HEPRA message indicates the definition of the task and its own unique identification information that determines whether the service provided by the service provider and the cloud service manager can be appropriately performed.

**Hierarchy Edge Profile Request Message**

Identifier	HEPR	Timestamp
------------	------	-----------

**Example>**

043d247347 · · · · e07f3481400d	HEPR	95251
---------------------------------	------	-------

**Figure 2.** Hierarchy Edge Profile Request (HEPR) message format and example.

**Table 1.** Hierarchy Edge Profile Request (HERP) message fields and description.

Field	Detailed Description
Identifier	Edge identification hash value using the SHA-256 algorithm
HEPR	Packet type indication field for profile request
Timestamp	Field for checking network status

The MECS that has distributed the profile performs the Edge Performance Scoring process based on the contents defined in the profile and sends the Hierarchical Edge Registration Message, including the result, to the central mobile cloud data center corresponding to the root in the hierarchical structure. That is, each MECS transmits a profile request to a central mobile cloud data center and performs a profile-based operation. That is, each MECS carries out a profile-based operation in which each MECS transmits a request to a central mobile cloud data center and then the central mobile cloud

data center transmits the related response to an efficient operation of the MECS and the MECS selects a specific task.

**HEPR Acknowledgement Message**

Identifier	HEPRA	Timestamp	Profile	Signature
<b>Example&gt;</b>				
043d247347 · · · · e07f3481400d	HEPRA	95251	Profile	84275f4d · · · · 2dce81b

Figure 3. HEPR Acknowledgement message format and example.

**Phase 2. Configuration of the Hierarchy Layer:** In response to the HEPR message, the MECS transmit their results to the central mobile cloud data center. Then, the MECS transmits the Hierarchical Edge Registration (HER) message to the central mobile cloud server, including the execution result and its own identifier as a response to the HEPRA message.

Figure 4 shows the format and example of the HER message. The operation timestamp field is used to record the time at which each operation defined in the profile is performed. The time when the MECS sends the HER message to the central mobile cloud data center is recorded in the request timestamp field. Then, it is used to check the network status between the central mobile cloud data center and the MECS. Also, to ensure secure registration between the MECS and the central mobile cloud data center, the signature generated by the identifier, and the hash value from the central mobile cloud data center is used. The central mobile cloud data center checks the operation timestamp field of the HER message transmitted by the MECS and determines the hierarchy level of the MECS according to its own policy.

**Hierarchy Edge Registration Message**

Identifier	HER	Operation Timestamp	Request Timestamp	Signature
<b>Example&gt;</b>				
043d247347 · · · · e07f3481400d	HER	12398491	13392322	75f4d824 · · · · 21b8dce

Figure 4. Hierarchy Edge Registration Request message format and example.

After determining the hierarchy layer of the MECS, the central mobile cloud data center transmits a HER Acknowledgment (HERA) message, including a task type and hierarchy layer information suitable for the MECS. After receiving the HERA message, the MECS configures its task type and layer information. In other words, each MECS can provide layer-specific services through the Edge Performance Scoring process and hierarchical level that is configured based on its own computing ability. Figure 5 shows the format and example of the HERA message.

**HER Acknowledgement Message**

Identifier	HERA	Task Type	Level	Signature
<b>Example&gt;</b>				
043d247347 · · · · e07f3481400d	HERA	Content Migration	2	75f4d824 · · · · 21b8dce

Figure 5. HER Acknowledgement message format and example.

**Phase 3. Context Awareness Migration:** Since the proposed CoHEC architecture constructs a hierarchical architecture of MECS segmentalized through the Edge Performance Scoring process, it is possible to perform the operation of content transmission and specific services efficiently. The proposed CoHEC architecture consists of task migration and content migration.

3.1. Content Migration Operation

MECS with either a large storage capacity or poor computation ability are selected for the content migration operation. The proposed content migration operation provides content with a low latency to the end users through the caching function and hierarchical MECS configuration. First, when receiving the content transmission request from the user, the MECS performs a content lookup process to determine whether the related content exists in its own repository. If the same content is cached in the repository, the MECS deliver the matched content, not from the central mobile cloud data center. On the other hand, if the matched content is not found, a request message is sent to the neighboring MECS.

Figure 6 shows the content type migration request message format and example. Since the hierarchy information of the MECS is recorded in the level field of the request message, the MECS can recognize the layer information of the MECS that transmitted the request message. First, the MECS receiving the content request message checks the level field of the request message to see if the sender of the request message is of the same hierarchical level. In cases where the request message is from the same layer, the corresponding content is transmitted if the requested content exists in its own repository. If the requested content does not exist, the request message is discarded. If the MECS that transmitted the request message does not receive the content, this means that there is no content corresponding to the same layer. Therefore, the upper layer information is recorded in the level field of the request message, and the newly recorded request message is transmitted to the MECS. After receiving the content request message, the MECS checks the level field of the request message and immediately discards the content request message if the message is sent from a different hierarchy level. That is, the proposed architecture performs a content request to the MECS existing in the same hierarchy and performs a content migration request to an upper layer when there is no content in the same hierarchy.

**Content Type Hierarchy Edge Migration Request Message**

Identifier	Task Type	Level	TASK	Signature
<b>Example&gt;</b>				
043d247347 · · · · e07f3481400d	Content Migration	1	mdpi_guide.pdf	75f4d824 · · · · 21b8dce

Figure 6. Content type migration request message format and example.

3.2. Computing Task Migration Operation

MECS with high performance computing capability perform the computing migration type operations. First, they determine whether to execute the task in their own device environment or not according to the cost policy set by the user (i.e., the battery priority mode and the computing operation priority mode). A mobile user who decides to perform a computing migration by its own policy sends a request message, including its computing task, to a neighboring MECS. Figure 7 shows the computing type migration request message format and example. Upon receiving the request message, the MECS checks the type of request message field and determines if it can be handled when the type is computing migration. Even if the MECS has enough computing capability, it can send a migration request message to the same hierarchy level, especially when it has a large computing workload and cannot provide the requested services. When a MECS in the same level of hierarchy that has received the request sends a response message to the request, it delegates the computing load sent by the user

to process the response and returns the result. If delegation cannot be handled by the adjacent MECS due to a large computing workload, the request is sent to the upper layer and processed. On the other hand, if it does not have enough computing capability, it delegates the computing workload to the upper layer. In other words, the computing load that can be handled in the same hierarchy level is migrated to neighboring MECS in the same layer; otherwise, it is delegated to the MECS in the upper layer. Figure 8 shows the message flow of the proposed CoHEC architecture.

**Computing Type Hierarchy Edge Migration Request Message**

Identifier	Task Type	Level	TASK	Signature
------------	-----------	-------	------	-----------

**Example>**

043d247347 · · · · e07f3481400d	Computing Migration	3	AR_TRAKING	75f4d824 · · · · 21b8dce
---------------------------------	---------------------	---	------------	--------------------------

Figure 7. Computing type migration request message format and example.

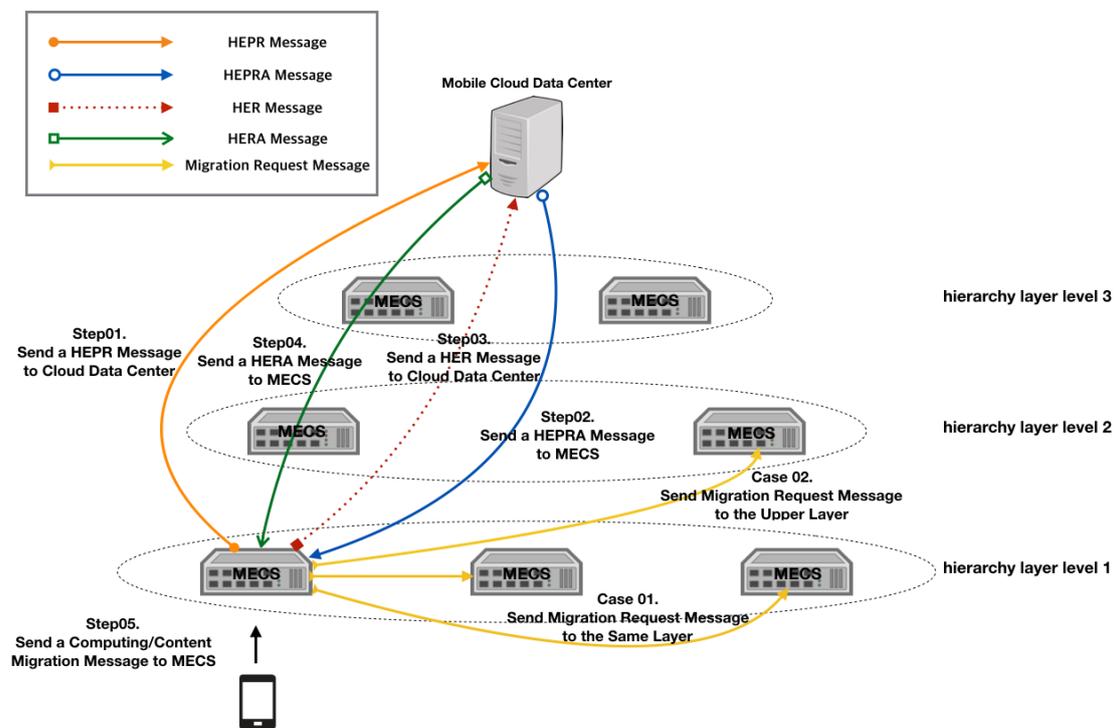


Figure 8. Message flow of the proposed context-aware hierarchical mobile edge computing (CoHEC) architecture.

**4. Performance Evaluation**

To assess the improvement of the proposed CoHEC architecture, we created a series of simulations where the number of mobile users that transmitted a request message to the server was varied. To do this, we simulated the cloud and MEC environments using the CloudSim and EdgeCloudeSim simulation tools, respectively [21–23]. First, we constructed the hierarchical MEC network topology with three edge level layers, where each MECS in hierarchy level 1 was attached to one wireless router. The link bandwidth was assumed to be 100 Mbps. We set the number of users to be from 100 to 400. We assumed that each MECS could handle migration requests of up to 200 mobile users. Also, to express the complexity of a computing operation, we set the number of cycles of the Central Processing Unit (CPU). For example, if a computing task with 1 gigacycle is offloaded to the server, its

execution time would be 1/2.5 s. We compared the efficiency of various existing network architectures with the efficient use of server resources and the provision of low latency content.

Figure 9 shows the average latency of the response when a user migrates a low complexity computing operation to a server based on the number of users. The average latency is the response time excluding the computation time after computing a migration request. As the number of users who delegate the computing operation to the server increases, existing mobile cloud data centers have significantly higher latency than the flat MEC architecture and the proposed CoHEC architecture. As the existing mobile cloud data centers are located at relatively long distance from the user, they have a high average latency time. Unlike the existing mobile cloud architecture, it is possible for a flat MEC to provide a low latency service to a user because it performs computing on an MECS close to the user. However, the average latency time of a flat MEC increases when the user's requests for computing migration operations increase. On the other hand, since the proposed architecture operates by migrating computation operations to upper hierarchy level MECS, it can provide low latency services even when the number of user requests increases. In addition, if the number of hierarchy layers increases, the MECS with high computing performance will be placed in the upper layer, so that the service can be provided to the user with lower latency. That is, if the hierarchical MEC architecture can be migrated to the upper layer MECS when its computing capability is exceeded, it will be possible to provide a service with a lower latency than the flat MEC architecture even if the number of users increases.

Figure 10 shows the content download completion time according to the number of users requesting content. For this purpose, a high volume of multimedia video was chosen to evaluate the possibility of providing fast content as the user content migration requests increase. Existing mobile cloud architecture can provide multimedia content to the user if the user capacity of the mobile cloud data center is not exceeded. However, if the content request increases and the mobile cloud data center cannot process all of the users' requests, the mobile cloud data center requires a long downloading time, because it performs the content providing operations according to the order of the requests. The flat MEC architecture enables faster content download than mobile cloud architecture because the adjacent MECS of the user provides a caching function. On the other hand, the proposed CoHEC architecture provides a content caching function by hierarchically structuring MECS, so that it is possible to perform a download quickly if the neighboring MECS has the content requested by the user. In addition, if the number of hierarchy layers increases, the number of MECS capable of delegating content migration operations and performing caching functions can be increased, so that the proposed CoHEC architecture can download content quickly, even if the number of users increases. However, the proposed CoHEC architecture works well by delegating the computing load to the upper layer. However, when the number of users that can be processed by MEC increases, the performance is diluted due to the transmission of the migration request message of the neighboring MECS and the delay time of the neighboring MECS search. Nevertheless, since the proposed CoHEC architecture can delegate the computing load to the upper layer, it can operate more efficiently than the existing MEC architecture.

Figure 11 shows the energy consumption according to computing operation type. We conducted an experiment on the energy consumption required for content transmission, video transcoding, and image processing. To do this, we performed video transcoding with 240p and 1080p resolution and experimented with face recognition application to measure the energy consumption of complex computing tasks [24]. Since the existing mobile cloud architecture and the flat MEC architecture simply provide computing resources without consideration of service type, they rely on their computing resources, regardless of the complexity of computation, and consume a lot of energy, making efficient energy use impossible. On the other hand, each MECS performs the Edge Performance Scoring process based on the profile, so it is possible to provide suitable services to users. That is, each MECS can perform the service that is suitable to them through the Performance Scoring process, which results in efficient use of their resources and a quick response time to the users' requests.

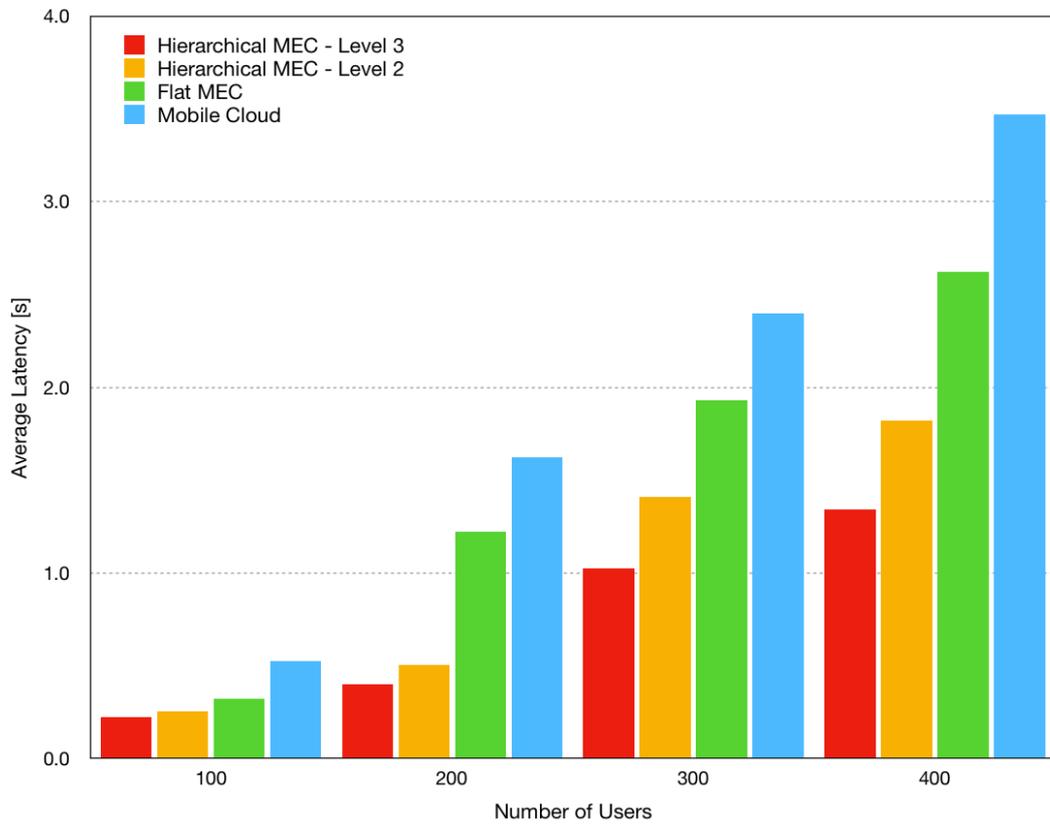


Figure 9. Average latency time.

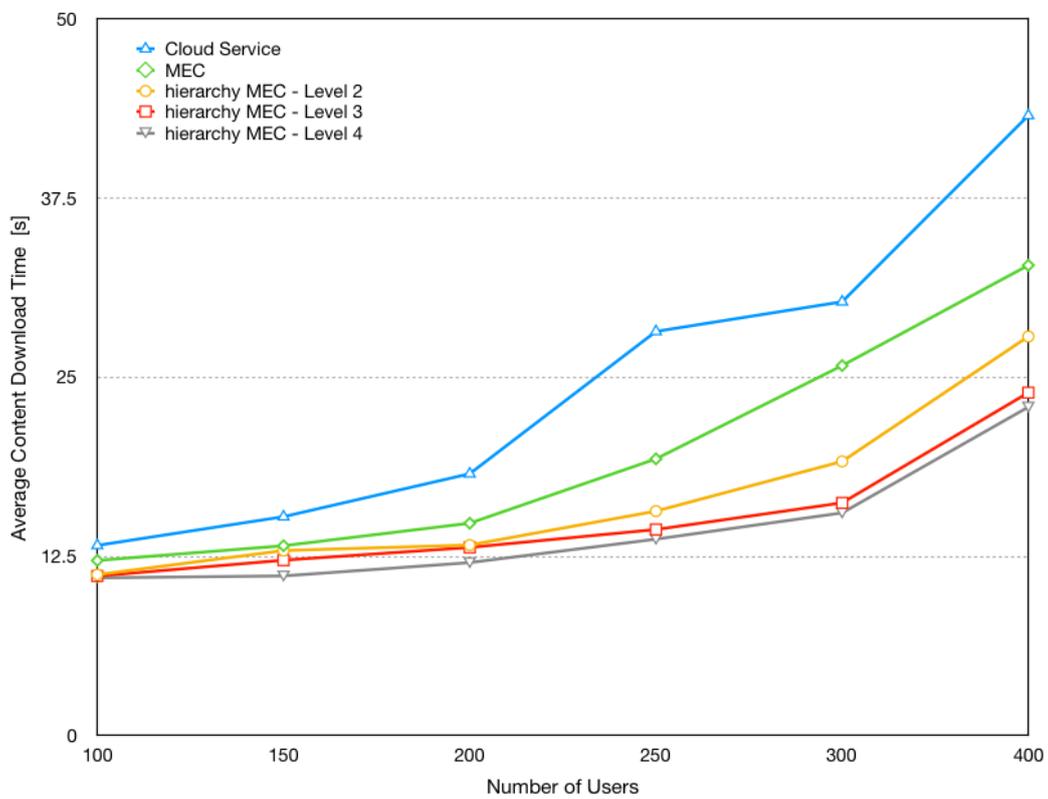
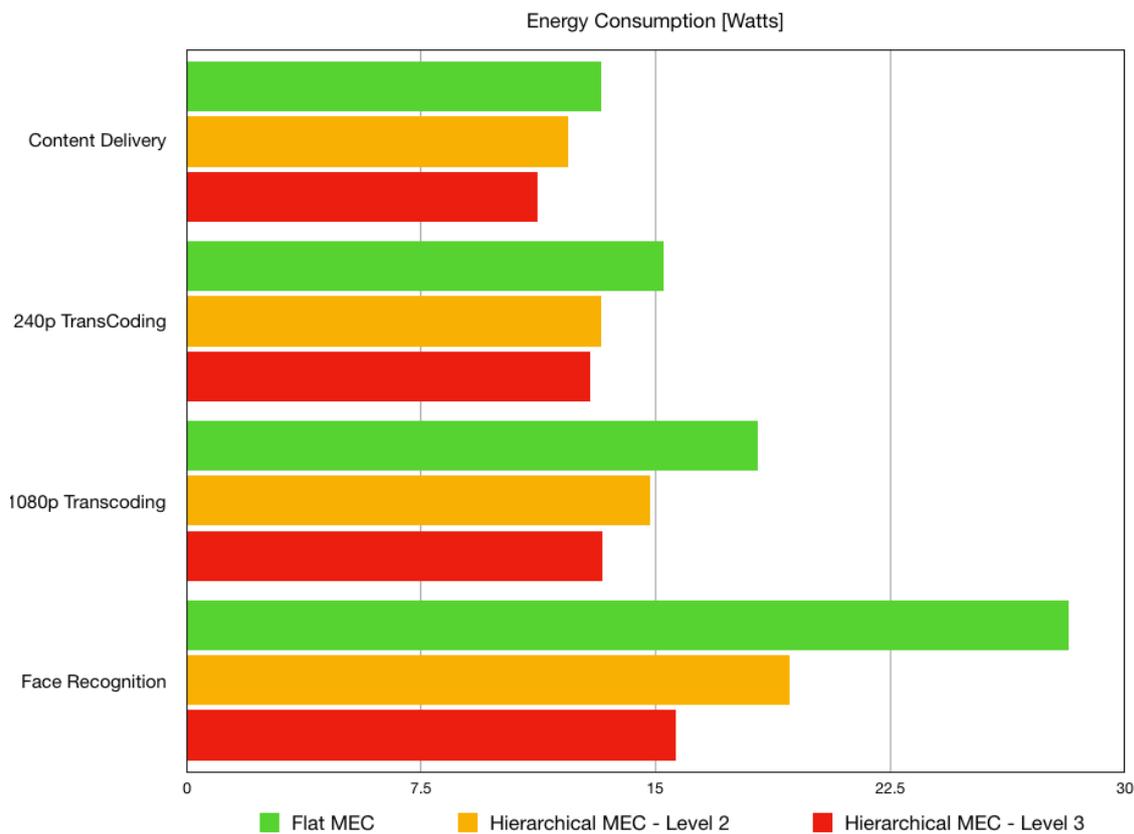


Figure 10. Average content download time.



**Figure 11.** The energy consumption of different types of computing operations.

Finally, Theorem 1 shows the processing method of the computing task requested by the user according to each computing architecture. The proposed hierarchical architecture performs the task of sending a task/content migration message to select a MEC server in the upper hierarchy. When there are many users connected to the MEC servers, the proposed hierarchical architecture is more efficient than the existing MEC, cloud computing and Fog Computing architectures because it delegates the task to the upper layer. However, if all the resources of MEC servers located in the upper hierarchy are exhausted, the network overhead increases and delays occur because of the task delegation message transfer behavior.

**Theorem 1.**

$$Cloud = Task_{User\ Request} / Computing\ Power_{Cloud}$$

$$MEC = Task_{User\ Request} / Computing\ Power_{MEC} + Task_{Migration} / Computing\ Power_{Cloud}$$

$$CoHEC = MEC_{task} + Task_{Migration} / Computing\ Power_{NeighboringMEC}$$

**5. Conclusions**

This paper makes the following points. First, it shows that the existing cloud data center has powerful computing capability and it is possible to efficiently delegate user computing operations. However, the problem is that the existing mobile cloud data centers are located relatively far away from users, and therefore, they cannot provide a fast response to users. We solved this problem through providing a low-latency service by utilizing the MECS which places the server at the edge located near the user. Nevertheless, there was no consideration of an efficient MEC architecture, and it has been shown that a large number of users cannot provide a low-latency service when a request is transmitted. Therefore, in this paper, we proposed a hierarchical MEC architecture that provides a low-latency

service even if many users send requests through the hierarchical arrangement of the MECS. That is, each MECS can perform a service that is suitable to them through the Performance Scoring process, so that it is possible to use of their resources efficiently and response to user requests quickly. This also shows that server energy can be saved by using server resources effectively.

**Author Contributions:** J.L. (Juyong Lee) and J.L. (Jihoon Lee) conceived and designed the experiments; J.L. (Juyong Lee) performed the experiments; J.L. (Juyong Lee) and J.L. (Jihoon Lee) analyzed the data; J.L. (Juyong Lee) contributed reagents/materials/analysis tools; J.L. (Juyong Lee) and J.L. (Jihoon Lee) wrote the paper.

**Funding:** This research was supported by a 2017 Research Grant from Sangmyung University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CoHEC	Context-Aware Hierarchical Mobile Edge Computing
IoT	Internet of Things
UHD	Ultra-High Definition
MEC	Mobile Edge Computing
MECS	Mobile Edge Computing Server
ETSI	The European Telecommunications Standard Institute
RAN	Radio Access Network
HI-MEC	Hierarchical Mobile Edge Computing
ISP	Internet Service Provider
LTE-advanced	Long Term Evolution-Advanced
HEPR	Hierarchy Edge Profile Request
HEPRA	Hierarchy Edge Profile Request Acknowledgement
HER	Hierarchical Edge Registration
HERA	Hierarchical Edge Registration Acknowledgement
CPU	Central Processing Unit

## References

- Chiang, M.; Zhang, T. Fog and IoT: An overview of research opportunities. *IEEE Int. Things J.* **2016**, *3*, 854–864. [[CrossRef](#)]
- Stolfo, S.J.; Salem, M.B.; Keromytis, A.D. Fog computing: Mitigating insider data theft attacks in the cloud. In Proceedings of the 2012 IEEE Symposium on Security and Privacy Workshops, San Francisco, CA, USA, 24–25 May 2012; pp. 125–128.
- Luis, M.; Luis, R. Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing. *ACM SIGCOMM Comput. Commun.* **2014**, *44*, 27–32.
- Mao, Y.; Zhang, J.; Letaief, K. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3590–3605. [[CrossRef](#)]
- Nunna, S.; Kousaridas, A.; Ibrahim, M.; Dillinger, M.; Thuemmler, C.; Feussner, H.; Schneider, A. Enabling real-time context-aware collaboration through 5G and mobile edge computing. In Proceedings of the IEEE Information Technology-New Generations (ITNG), Las Vegas, NV, USA, 13–15 April 2015; pp. 601–605.
- Patel, M.; Naughton, B.; Chan, C.; Sprecher, N.; Abeta, S.; Neal, A. Mobile-edge computing introductory technical white paper. In *Mobile-Edge Computing (MEC) Industry Initiative*; ETSI: Sophia Antipolis, France, 2015.
- Mach, P.; Becvar, Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [[CrossRef](#)]
- Li, H.; Shou, G.; Hu, Y.; Guo, Z. Mobile edge computing: Progress and challenges. In Proceedings of the IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, UK, 29 March–1 April 2016; Volume 10, pp. 142–149.
- Chen, T.; Matinmikko, M.; Chen, X.; Zhou, X.; Ahokangas, P. Software defined mobile networks: Concept, survey, and research directions. *IEEE Commun. Mag.* **2015**, *53*, 126–133. [[CrossRef](#)]

10. Yixue, H.; Min, C.; Long, H.; Hossain, M.; Ahmed, G. Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access* **2018**, *6*, 11365–11373.
11. Chen, M.; Hao, Y.; Lai, C.; Wu, D. On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Commun. Mag.* **2008**, *53*, 18–24. [[CrossRef](#)]
12. Jararweh, Y.; Tawalbeh, L.; Ababneh, F.; Dosari, F. Resource efficient mobile computing using cloudlet infrastructure. In Proceedings of the IEEE Ninth International Conference Mobile Ad-hoc and Sensor Networks (MSN), Dalian, China, 11–13 December 2013; pp. 373–377.
13. Sun, X.; Ansari, N. Green cloudlet network: A distributed green mobile cloud network. *IEEE Netw.* **2017**, *31*, 64–70. [[CrossRef](#)]
14. Tsiropoulou, E.E.; Paruchuri, S.T.; Baras, J.S. Interest, energy and physical-aware coalition formation and resource allocation in smart IoT applications. In Proceedings of the IEEE Information Sciences and Systems (CISS) 51st Annual Conference, Baltimore, MD, USA, 22–24 March 2017; pp. 1–6.
15. Tsiropoulou, E.E.; Mitsis, G.; Papavassiliou, S. Interest-aware energy collection & resource management in machine to machine communications. *Ad Hoc Netw.* **2018**, *68*, 48–57.
16. Wang, C.; Liang, C.; Yu, F.R.; Chen, Q.; Tang, L. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 4924–4938. [[CrossRef](#)]
17. Wang, C.; Liang, C.; Yu, F.R.; Chen, Q.; Tang, L. Joint computation offloading, resource allocation and content caching in cellular networks with mobile edge computing. In Proceedings of the IEEE Communications (ICC) International Conference, Paris, France, 21–25 May 2017; pp. 1–6.
18. Kiani, A.; Ansari, N. Toward Hierarchical Mobile Edge Computing: An Auction-Based Profit Maximization Approach. *IEEE Int. Things J.* **2017**, *4*, 2082–2091. [[CrossRef](#)]
19. Anagnostopoulos, C.B.; Tsounis, A.; Hadjiefthymiades, S. Context awareness in mobile computing environments. *Wirel. Pers. Commun.* **2007**, *42*, 445–464. [[CrossRef](#)]
20. Rodrigo, N.; Calheiros, R.; Anton, B.; Cesar, A.; Rajkumar, B. A hierarchical edge cloud architecture for mobile computing. In Proceedings of the 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9.
21. Rodrigo, N.; Calheiros, R.; Anton, B.; Cesar, A.; Rajkumar, B. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Softw. Pract. Exp.* **2008**, *41*, 23–50.
22. Sonmez, C.; Ozgovde, A.; Ersoy, C. EdgeCloudSim: An environment for performance evaluation of Edge Computing systems. In Proceedings of the Second International Conference on Fog and Mobile Edge Computing (FMEC), Valencia, Spain, 8–11 May 2017; pp. 39–44.
23. Sonmez, C.; Ozgovde, A.; Ersoy, C. Performance evaluation of single-tier and two-tier cloudlet assisted applications. In Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops), Paris, France, 21–25 May 2017; Volume 10, pp. 302–307.
24. Chen, X. Decentralized Computation Offloading Game for Mobile Cloud Computing. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 974–983.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).