



# A Cloud Service Enhanced Method Supporting Context-Aware Applications

Zifan Liu<sup>1</sup>(✉), Qing Cai<sup>2</sup>, Song Wang<sup>3</sup>, Xiaolong Xu<sup>2</sup>, Wanchun Dou<sup>1</sup>,  
and Shui Yu<sup>4</sup>

<sup>1</sup> State Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing, China

nju.liuzf@gmail.com, douwc@nju.edu.cn

<sup>2</sup> School of Computer and Software,  
Nanjing University of Information Science and Technology, Nanjing, China

1392134662@qq.com, njuxlxu@gmail.com

<sup>3</sup> State Grid Anhui Electric Power Company, Hefei, China

wsong09@163.com

<sup>4</sup> School of Information Technology, Deakin University, Melbourne, Australia  
shui.yu@deakin.edu.au

**Abstract.** Mobile cloud computing is emerging as a powerful platform for running demanding applications migrated from mobile devices to a remote cloud. For some real-time or urgent deadline-constrained applications, the migration process generates intolerable transmission latency. Cloudlets co-located with Access Points (APs) are considered as an efficient way to reduce such transmission latency. However, it is still a challenge to manage the cloudlets that have been deployed for fixed context-aware applications to achieve cost savings. In view of this challenge, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgment. Then a relevant cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental and analytical results demonstrate the validity of our proposed method.

**Keywords:** Cloud service · Context-aware applications  
Mobile cloud computing · Cloudlet · Cost savings

## 1 Introduction

With increasing resource requirements of mobile applications in computation, communication and storage, mobile cloud computing is emerging as an effective way to realize on-demand resource provisioning for mobile applications [1, 2]. Mobile cloud computing provides abundant storage resources and high computing capability for migrated demanding applications, but at the same time, such application migration generates long transmission latency [1]. And the latency

is intolerable for users in some mobile applications, such as interactive gaming, speech recognition, video playback [3, 4], to name a few.

As an efficient and effective cloud deployment paradigm, cloudlets are deployed around mobile devices to reduce the transmission latency for mobile applications [5, 6]. Generally, cloudlets are resource-rich and self-managed which can be organized by harnessing the personal idle servers of other individual users or directly provisioned by the network operators (NOs) [1]. Mobile devices can offload their mobile applications to nearby cloudlets rather than remote clouds for processing. As a result, the physical proximity between mobile users and cloudlets is beneficial to transmission time reduction for workload offloading.

The cloudlets are in active mode when they are placed to enhance the local cloud services supporting context-aware applications. These active cloudlets consume overwhelming power which leads to high running cost. When there are no mobile devices within cloudlet coverage, the cloudlets in active mode are unnecessary to run. Such cloudlets should be shut down to achieve cost savings for energy consumption.

Here, an example of cloudlet management is presented to illustrate the problem investigated in this paper. There are a number of mobile devices with fixed context-aware applications, and 5 cloudlets, i.e., *A, B, C, D*, and *E*, with APs distributed in 3 rooms (i.e., Room1, Room2, and Room3) of a conference center, as shown in Fig. 1. In Fig. 1, cloudlets *A, B, C* are placed in Room1, cloudlets *D* and cloud *E* are placed in Room2 and Room3, respectively. In Fig. 1(a), there are no mobile devices within the coverage of cloudlets *C* and *E*, but they consume a certain amount of energy. In such circumstance, cloudlets *C* and *E* should be shut down to achieve cost savings as illustrated in Fig. 1(b).

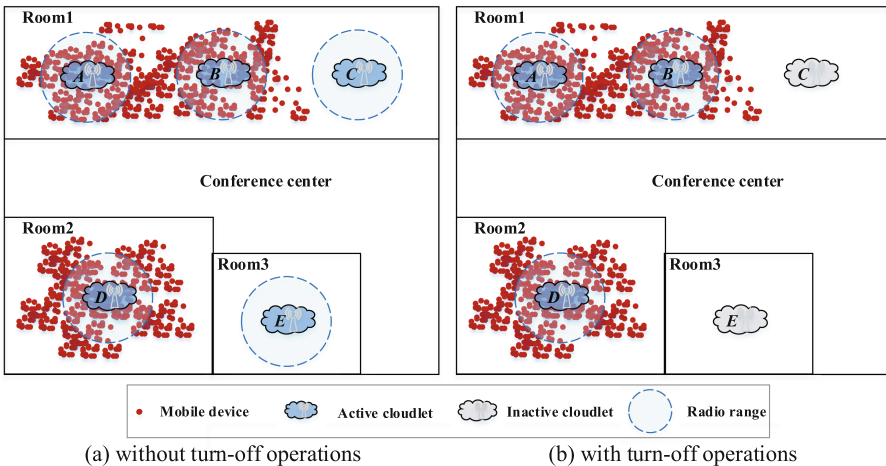


Fig. 1. An example of cloudlet management

However, current researches mainly focus on cloudlet placement to reduce energy consumption and network delay [7, 8]. Based on these observations, it is still a challenge to manage the cloudlets that have already been deployed for fixed context-aware applications to achieve cost savings.

In view of this challenge, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgments. Then a cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental results demonstrate that the proposed method is both effective and efficient.

The rest of the paper is organized as follows. A cloudlet management principle is presented in Sect. 2. Section 3 elaborates a cloud service enhanced method supporting context-aware applications. Experimental evaluations are conducted in Sect. 4 to demonstrate the validity of our method. Some related work is described in Sect. 5. Section 6 concludes the paper and gives an outlook on possible future work.

## 2 Cloudlet Management Principle

In this section, formalized concepts are given to facilitate our further discussion for cloud service enhancement supporting context-aware applications. To simplify the discussion, key terms used in our cloudlet management principle are summarized in Table 1.

**Table 1.** Key terms and descriptions

Terms	Description
$S$	The set of the points in device activity area
$DA$	The set of device activity area, $DA = \{da_1, da_2, \dots, da_N\}$
$da_n$	The $n$ -th device activity area
$MD$	The set of mobile devices, $MD = \{md_1, md_2, \dots, md_M\}$
$md_n$	The set of mobile devices in $da_n$ , $md_n = \{md_{n,1}, md_{n,2}, \dots, md_{n,Z}\}$
$mp_m$	The position $md_m$ , $mp_m = (mpx_m, mpy_m)$
$cl_{n,i}$	The $i$ -th cloudlet in $da_n$
$cp_{n,i}$	The central position of $cl_{n,i}$ , $cp_{n,i} = (cpx_{n,i}, cpy_{n,i})$
$dc_{n,i}$	The device collection of $cl_{n,i}$
$r_{n,i}$	The coverage radius for $cl_{n,i}$
$IC_{n,i}$	The indoor cloudlet coverage collection of $cl_{n,i}$
$\rho$	The density threshold for cloudlet placement judgment

To enhance cloud service supporting fixed context-aware applications, a 3-tier mobile cloud infrastructure with cloudlet is introduced, as shown in Fig. 2.

In this infrastructure, the cloudlets are co-located with APs, the remote cloud provides data storage and computing service, and the mobile devices are clients that can access the mobile cloud service through wireless networks. Compared to the traditional client-server communication model [1] without cloudlet, it is more powerful to reduce access latency by leveraging such 3-tier infrastructure. Via the cloudlets placed nearby, mobile devices can get direct cloud computing resources through AP to access cloudlets in the network.

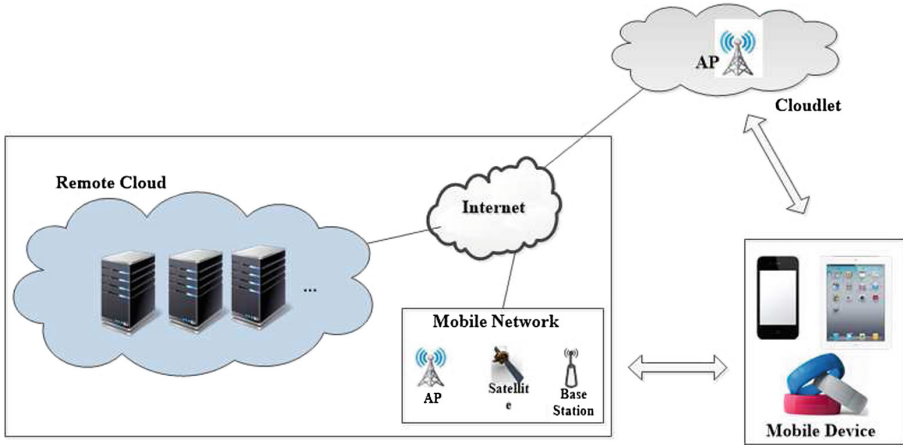


Fig. 2. Mobile cloud infrastructure with cloudlet

Generally, there are various shapes of cloudlet management areas in practice. As other shaped areas can be assembled by multiple rectangles of different sizes, the cloudlet management area is defined by a rectangle in this paper.

**Definition 1 (Cloudlet Management Area).** *Cloudlet management area is defined by the  $x$ - $y$  plane with definite ranges of  $x$ -axis and  $y$ -axis, denoted as  $S = \{(x, y) | 0 \leq x \leq X, 0 \leq y \leq Y\}$ .*

Within the cloudlet management area, there are a number of device activity areas, denoted as  $DA = \{da_1, da_2, da_N\}$  ( $DA \subseteq S$ ), where  $N$  is the number of device activity areas in  $S$ . Each device activity area has the  $x$ - $y$  plane to show the range.

**Definition 2 (Divisional Device Activity Area).** *For the  $n$ -th ( $1 \leq n \leq N$ ) device activity area  $da_n$ , it is defined by the  $x$ - $y$  plane with definite ranges of  $x$ -axis and  $y$ -axis, it is denoted as  $da_n = \{(dax_n, day_n) | 0 \leq dax_n \leq X_n, 0 \leq day_n \leq Y_n\}$ .*

Within the cloudlet management area, there are a number of mobile devices distributed randomly, denoted as  $MD = md_1, md_2, \dots, md_M$ , where  $M$  is the number of devices in  $S$ . Each mobile device has a position to show their locations.

**Definition 3 (Mobile Device Position).** For the  $m$ -th ( $1 \leq m \leq M$ ) mobile device  $md_m$ , the device is located at some positions that is a 2-tuple in  $S$ , denoted as  $mp_m = (mpx_m, mpy_m)$ , where  $mpx_m$  and  $mpy_m$  are the  $x$ -axis value and the  $y$ -axis value of  $mp_m$ , respectively.

Each cloudlet is co-located with an AP. Let  $cl_n$  be a collection of cloudlets located within  $da_n$  and  $z_n$  ( $z_n \geq 1$ ) be the total number of cloudlets in  $cl_n$ . Then the total number  $K$  of cloudlets in  $S$  can be calculated by

$$K = \sum_{n=1}^N z_n \tag{1}$$

Each deployed cloudlet has a central position, which is beneficial to confirm the covered mobile devices.

**Definition 4 (Cloudlet Central Position).** For the  $i$ -th ( $1 \leq i \leq z_n$ ) cloudlet  $cl_{n,i}$  in the  $n$ -th device activity area  $da_n$ , the relevant central position is location where AP is placed, which is a 2-tuple in  $da_n$ , denoted as  $cp_{n,i} = (cpx_{n,i}, cpy_{n,i})$ , where  $cpx_{n,i}$  is the  $x$ -axis value and  $cpy_{n,i}$  is the  $y$ -axis value of  $cl_{n,i}$ .

The cloudlet can enhance the cloud service for users within cloudlet coverage area. In order to identify whether the cloudlets need to turn off, it is important to detect the device collection of cloudlet central positions.

**Definition 5 (Cloudlet Coverage Collection).** For the  $i$ -th cloudlet  $cl_{n,i}$  in  $da_n$ , the corresponding device collection is defined by  $dc_{n,i} = \{md_m | dis(mp_m, cp_{n,i}) \leq r_{n,i}, 1 \leq m \leq M\}$ , where  $dis(mp_m, cp_{n,i})$  is calculated by

$$dis(mp_m, cp_{n,i}) = \sqrt{(mpx_m - cpx_{n,i})^2 + (mpy_m - cpy_{n,i})^2} \tag{2}$$

Some mobile devices may be covered simultaneously by multiple cloudlets in different areas. In the circumstances, the real device collection of a cloudlet central position should subtract the number of devices outdoor. Suppose there are a set of mobile devices in  $n$ -th device activity area  $da_n$ , denoted as  $MD_n = \{md_{n,1}, md_{n,2}, \dots, md_{n,Z}\}$ , where  $Z$  is the number of devices in  $da_n$ . Then an indoor cloudlet coverage collection is defined as follows.

**Definition 6. Indoor Cloudlet Coverage Collection** For the  $i$ -th cloudlet  $cl_{n,i}$  in  $da_n$ , the corresponding indoor cloudlet coverage collection, denoted as  $IC_{n,i}$ , is defined by

$$IC_{n,i} = dc_{n,i} \cap MD_n \tag{3}$$

Figure 3 gives an example of indoor cloudlet coverage collection of cloudlet, the radius of cloudlet is  $r$  placed in the center  $O$ . The devices  $a_1, a_2, a_3, a_4$  and  $a_5$  are covered by cloudlet, but only devices  $a_3$  and  $a_4$  are in Room1. So the indoor cloudlet coverage collection of cloudlet is  $\{a_3, a_4\}$ .

In this paper, the coverage density plays a key role in our process of cloudlet. To achieve the goal of cost savings for mobile users within  $S$ , a cloudlet management principle is presented as a measurement for cloudlet status management.

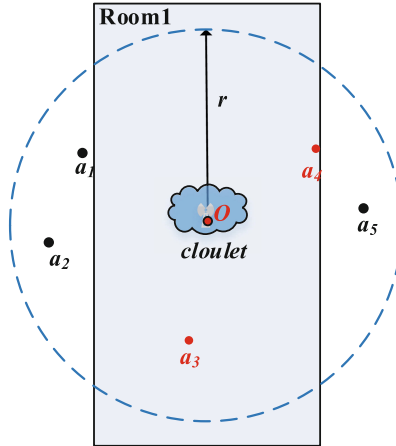


Fig. 3. An example of indoor cloudlet coverage collection of cloudlet

**Definition 7.** *Cloudlet Management Principle* If the  $i$ -th cloudlet  $cl_{n,i}$  with central position  $G(x, y)$  runs in active mode to serve mobile users within  $da_n$ , the number of indoor cloudlet coverage collection  $IC_{n,i}$  of  $cl_{n,i}$  should satisfy the condition that  $|IC_{n,i}| \geq \rho$ , where  $\rho$  is a density threshold for cloudlet status management judgment.

### 3 Cloud Service Enhanced Method Supporting Context-Aware Applications

In this section, a cloud service enhancement method supporting context-aware applications is presented for the mobile cloudlet placement. This method consists of the three steps specified in detail by Fig. 4.

- Step1:** Mobile device positions identification. The positions of mobile devices are identified by the mobile location technology for further cloudlet management in the following steps.
- Step2:** Device collection generation. The device collection of each cloudlet is derived from the mobile device positions gained by Step1. Depending on device area constraints, devices outside will be removed from corresponding collections.
- Step3:** Cloudlet management. In this step, the device collections will be estimated based on the cloudlet threshold. A greedy algorithm is utilized for selecting the least necessary cloudlets. All the selected cloudlets are turned on while others are shut down.

Fig. 4. Specifications of cloud service enhanced method supporting context-aware applications

### 3.1 Step1: Mobile Device Positions Identification

In order to present our cloud service enhanced method, it is essential to locate the mobile devices for analysis. Currently, Mobile Location Based Service (MLBS) is a pretty proven technology, which can accurately locate the mobile devices via a mobile terminal or a personal geographical through wireless network [9, 10]. According to the basic principles of mobile location, MLBS can be broadly divided into two categories: network-based location technology and terminal-based location technology [11]. Besides, mobile devices also can be located via combining these two location technologies.

Here, considering the Internet surfers in a cloud service enhanced network, network-based location technology is more appropriate to locate the mobile devices. Even if the locations of mobile devices change, it can be accurately located in no time. Taking advantage of such technology, a set of device positions are identified for further cloudlet management.

### 3.2 Step2: Device Collection Generation

The cloudlets are dispersedly deployed all over the device area and they will affect the largest number of mobile devices if they are all active. Yet not all the cloudlets can be turned on according to the cost saving consideration. Besides, in real world, the mobile devices usually scatter in an uneven distribution, thus it is appropriate that we select several cloudlets to cover considerable mobile devices. On this purpose, the collection of mobile devices that each cloudlets covers is identified.

We consider the device positions  $MP = \{mp_1, mp_2, \dots, mp_N\}$  gained by Step1 and the coverage radius  $r$  of every cloudlet. For each cloudlet  $cl_{n,i}$ , all the mobile devices, the distances of which and  $cl_{n,i}$  are less than  $r$ , are added to corresponding device collection  $dc_i$ . Additionally, as  $cl_{n,i}$  is in the device area  $da_n$ , it cannot affect the devices outside of this area, thus all the mobile devices outside are deleted from  $dc_{n,i}$ .

Algorithm 1 specifies the generation process of device collections. For each collection, the devices outside the corresponding device area are deleted.

### 3.3 Step3: Cloudlet Management

Take advantage of Step2, the cloudlets which should be active are determined in this step. The device collections gained by Step2 differ in size and it is a waste to turn on a cloudlet which covers only few mobile devices. Therefore, a cloudlet management principle  $\rho$ , defined in Definition 5, is presented to filter the device collections. Concretely,  $\rho$  represents the least number of mobile devices that an active cloudlet should cover. All cloudlets that dissatisfy such threshold are shut down to reduce energy cost.

Furthermore, the coverage area of these cloudlets may overlap partly. Consider the case that numerous mobile devices gather in the overlapped area of two cloudlets and few in separate areas. In view of the cloudlet threshold, both two

---

**Algorithm 1.** Device Collection Generation( $MD, CL$ )

---

**Require:** A set of mobile device  $MD$  and a set of cloudlet  $CL$ .**Ensure:** A set of device collections.

```

1: for  $i = 1 \rightarrow N$  do
2:   for  $j = 1 \rightarrow cl_n.size$  do
3:     if  $dis(mp_i, cl_{i,j}) < r_{i,j}$  then
4:       add  $md_i$  to  $dc_{i,j}$ 
5:     end if
6:   end for
7: end for
8: for  $i = 1 \rightarrow N$  do
9:   for  $j = 1 \rightarrow cl_n.size$  do
10:     $ic_{i,j} \leftarrow dc_{i,j}$ 
11:    for  $mp_k$  in  $dc_{i,j}$  do
12:      if  $mp_k$  is not in  $da_i$  then
13:        Delete  $mp_k$  from  $ic_{i,j}$ 
14:      end if
15:    end for
16:  end for
17: end for

```

---

cloudlets should be turned on, which generates a waste compared with turning on only one cloudlet of them. In order to overcome such problem, a greedy algorithm is employed in our method. First, the largest device collection, denoted as  $dc_i$ , is chosen and  $cl_i$  is added in result collection, denoted as  $clo$ . Second, for all the mobile devices in  $dc_i$ , delete them from every device collection. Repeat the two steps until sizes of all device collections are smaller than  $\rho$ . Then the cloudlets in result collection are turned on, while remaining cloudlets are shut down.

Algorithm 2 specifies management process of all cloudlets in device management area. The set of device collections is generated by Step1. All the cloudlets in  $clo$  are turned on while others are shut down.

## 4 Experimental Evaluation

In this section, HANA in-memory database is employed to evaluate the performance of our proposed cloud service enhanced method supporting context-aware applications.

### 4.1 Experiment Settings

In our experiments, two nodes are engaged to create a HANA cloud, including a mater and a slave, and the configuration of the hardware and the software is specified in Table 2.

To simplify the experimental evaluation on our method, a case study is presented. The shape of device management area is a square with sides of 140m.



**Algorithm 2.** Cloudlet management( $IC$ )**Require:** A set of device collections  $IC$ .**Ensure:** The cloudlet management policy.

```

1:  $clo \leftarrow \emptyset$ 
2: for  $i = 1 \rightarrow N$  do
3:   while true do
4:      $ic$  is the largest device collection in  $da_i$ 
5:     if  $|ic| > \rho$  then
6:       add  $ic$  to  $clo$ 
7:       for each  $ic_{i,k}$  in  $da_i$  do
8:          $ic_{i,k} \leftarrow ic_{i,k} - ic$ 
9:       end for
10:    else
11:      break
12:    end if
13:  end while
14: end for
15: for  $ic_{i,j}$  in  $IC$  do
16:   if  $ic_{i,j}$  is in  $clo$  then
17:    turn on  $cl_{i,j}$ 
18:   end if
19: end for

```

**Table 2.** The experiment context

	Client	HANA Cloud
Hardware	Lenovo Thinkpad T430 machine with Intel i5-3210M 2.50 GHz processor, 4 GB RAM and 250 GB Hard Disk	Master/slave: HP Z800 Workstation Intel(R) Multi-Core X5690 Xeon(R), 3.47 GHz/12M Cache, 6cores, 2 CPUs, 128 GB ( $8 \times 8$ GB + $4 \times 16$ GB) DDR3 1066 MHz ECC Reg RAM; 1 disk on the master and 2 disks on the slave: 2TB,7.2K RPM SATA Hard Drive
Software	Windows 7 Professional 64bit OS and HANA Studio	SUSE Enterprise Linux Server 11 SP3 and SAP HANA Platform SP07

There are five separate device activity areas, denoted as area  $a, b, c, d, e$ , in the device management area. In each device activity area, one or more cloudlets have been deployed and most space of those areas is covered by at least one cloudlet. Over the whole device management area, a set of mobile device positions is randomly generated. In practice, the cloudlet management principle  $\rho$  is set to 15 and the coverage radius  $r$  is set to 25 m. The detailed experimental parameters are specified in Table 3.

**Table 3.** The experiment context

Parameter item	Domain
The maximum $x$ -axis value $X$	140 m
The maximum $y$ -axis value $Y$	140 m
Cloudlet management principle $\rho$	15
The radio range $r$ of all mobile cloudlets	25 m
The number of mobile devices	200
The maximum $x$ -axis value $X$	140 m

## 4.2 Performance Evaluation

In this section, performance evaluations are presented to discuss the coverage number of mobile devices by using our method compared with this value without cloudlet management. For the device activity areas without cloudlet management, these cloudlets cannot be shut down. As numerous mobile devices are distributed in the device management area, the cloudlet covered mobile devices under the above two situations, i.e., with cloudlet management by our method and without cloudlet management are presented.

In Fig. 5, the distribution of mobile devices is shown without cloudlet management. Most devices gather in area  $a$ ,  $b$ , and  $d$ . Concretely, the right bottom part of area  $a$ , the upper part of area  $b$  and the center part of area  $d$  are the dense region of mobile devices. In our experiments, the walls of each activity area are radiation resistant and the cloudlet cannot affect the mobile devices that outside of corresponding area. The experimental results are illustrated by Fig. 6.

In Fig. 6, the cloudlet coverage number in each activity area is compared with total device number in such area. From this comparison, we can find that nearly all mobile devices are covered by cloudlets. Under this situation, all cloudlets are active that generates a certain amount of running costs.

Then our method is conducted on the same conditions. The management results of our method after cloudlet management are presented in Fig. 7. Compared with Fig. 5, only 4 of the 10 cloudlets are active. Furthermore, the cloudlet coverage number, as shown by Fig. 8, is 175 out of total 200, which is quite close to the number, 184 out of 200, before leveraging our method.

From the experimental evaluation, we can find after cloudlet management by our method, the cloudlet coverage value is similar to this value without cloudlet management. But our method can greatly reduce the energy consumption and the running costs of cloudlets, since only 4 of total 10 cloudlets running after cloudlet management.

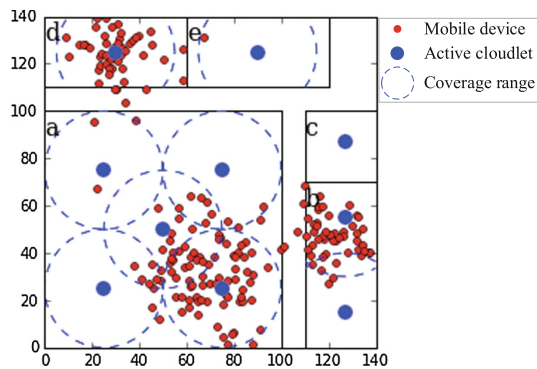


Fig. 5. Mobile device distribution and cloudlets status before using our method

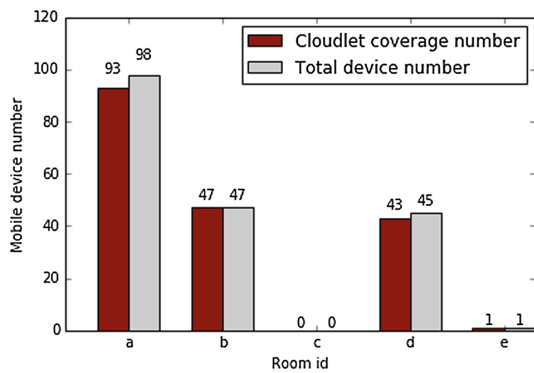


Fig. 6. Cloudlet coverage numbers in each device activity area before using our method

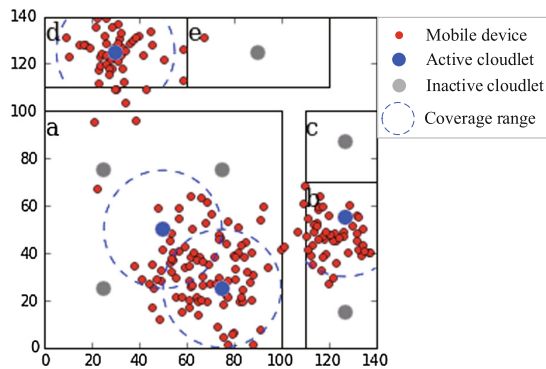


Fig. 7. Mobile device distribution and cloudlets status after using our method

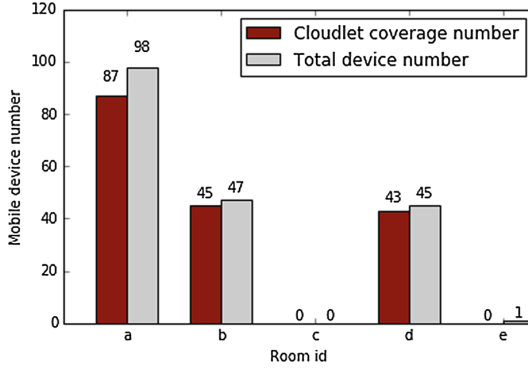


Fig. 8. Cloudlet coverage numbers in each device activity area after using our method

## 5 Related Work and Comparison Analysis

Currently, many researches of mobile cloud computing have been proposed to improve the computing capacity of mobile devices [1, 12]. However, the clouds are geographically far away from mobile users, in the result of which a huge latency generates during the workload offloading between mobile devices and remote clouds. Thus cloudlet is applied close to users to provide compute capability and data storage, which can greatly decrease the response time. They have been fully investigated in [13–17], to name a few.

In [13], the author proposed a network architecture, which is cost-effective via combining localized and distributed mini-clouds and fast-deployable wireless mesh networks. Xia *et al.* [14] devised an efficient online algorithm to solve an online location-aware offloading problem in a two-tiered mobile cloud computing environment, which consists of a local cloudlet and remote clouds. In [16], a framework named PEFC (Performance Enhancement Framework of Cloudlet) was proposed to enhance the finite resource cloudlet performance by increasing cloudlet resources. Artail *et al.* [17] formulated a more ubiquitous solution to decrease high network latency of remote cloud services, which relies on a network of cloudlets distributed within a geographic area. Despite the increasing momentum of cloudlet research, the cost of cloudlets has largely been overlooked.

Current researches mainly focus on cloudlet placement to reduce energy consumption and network delay. Wu and Ying [18] designed a novel virtual currency tailored for the cloudlet-based multi-lateral resource exchange framework, which made the resource exchange efficiently. Ravi and Peddoju [19] proposed a model to tackle the mobility and energy efficiency of mobile cloud computing. But they ignore the energy consumption of the redundant cloudlets. The best solution is to effectively manage the cloudlets usage. Thus we formulate a cloudlet management principle to save energy consumption of the cloudlets.

## 6 Conclusion and Future Work

In this paper, a cloud service enhanced method supporting context-aware applications is proposed in this paper. Specifically, a cloudlet management principle is designed to provide a reference for cloudlet status judgments. Then a cloud service enhanced method is proposed to decide which active cloudlets should be shut down. Finally, the experimental and analytical results demonstrate that the proposed method is both effective and efficient.

For future work, we plan to apply our cloud service enhanced method to real-world cloudlet platform. Besides, we will analyze energy consumption of the cloudlet management in different environment. And we intend to find a better energy-efficient method to manage cloudlets status.

**Acknowledgement.** This research is partially supported by the National Science Foundation of China under Grant No. 61672276, No. 61702277, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

## References

1. Li, Y., Gao, W.: Code offload with least context migration in the mobile cloud. In: IEEE Conference on Computer Communications (INFOCOM), pp. 1876–1884 (2015)
2. Xu, Z., Liang, W., Xu, W., Jia, M., Guo, S.: Efficient algorithms for capacitated cloudlet placements. *IEEE Trans. Parallel Distrib. Syst.* **27**(10), 2866–2880 (2016)
3. Cohen, J.: Embedded speech recognition applications in mobile phones: status, trends, and challenges. In: IEEE International Conference on Acoustics, Speech and Signal (ICASSP), pp. 5352–5355 (2008)
4. Mohiuddin, K., Mohammad, A.R., Raja, A.S., Begum, S. F.: Mobile-CLOUD-mobile: is shifting of load intelligently possible when barriers encounter? In: IEEE International Conference on Information Science and Digital Content Technology (ICIDT), vol. 2, pp. 326–332 (2012)
5. Rimal, B.P., Van, D.P., Maier, M.: Cloudlet enhanced fiber-wireless access networks for mobile-edge computing. *IEEE Trans. Wirel. Commun.* **16**(6), 3601–3618 (2017)
6. Dinh, H.T., Lee, C., Niyato, D., Wang, P.: A survey of mobile cloud computing: architecture, applications, and approaches. *Wirel. Commun. Mob. Comput.* **13**(18), 1587–1611 (2011)
7. Xu, Z., Liang, W., Xu, W., Jia, M., Guo, S.: Capacitated cloudlet placements in wireless metropolitan area networks. In: IEEE Conference on Local Computer Networks (LCN), pp. 570–578 (2015)
8. Jia, M., Cao, J., Liang, W.: Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Trans. Cloud Comput.* **5**, 2168–2161 (2015)
9. Wang, H., Wang, Z., Shen, G., Li, F., Han, S., Zhao, F.: WheelLoc: enabling continuous location service on mobile phone for outdoor scenarios. In: IEEE Conference on Computer Communications (INFOCOM), pp. 2733–2741 (2013)

10. Constandache, I., Gaonkar, S., Sayler, M., Choudhury, R.R., Cox, L.: EnLoc: energy-efficient localization for mobile phones. In: IEEE Conference on Computer Communications (INFOCOM), pp. 2716–2720 (2009)
11. Lee, H.J., Wicke, M., Kusy, B., Guibas, L.: Localization of mobile users using trajectory matching. In: Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, pp. 123–128 (2008)
12. Chen, X.: Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. Parallel Distrib. Syst.* **26**(4), 974–983 (2015)
13. Khan, K.A., Wang, Q., Grecos, C., Luo, C., Wang, X.: MeshCloud: integrated cloudlet and wireless mesh network for real-time applications. In: IEEE Conference on Electronics, Circuits, and Systems (ICECS), pp. 317–320 (2013)
14. Xia, Q., Liang, W., Xu, Z., Zhou, B.: Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. In: IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 109–116 (2014)
15. Bahtovski, A., Gusev, M.: Multilingual cloudlet-based dictionary. In: IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 380–385 (2014)
16. Whaiduzzaman, M., Gani, A., Naveed, A.: PEFC: performance enhancement framework for cloudlet in mobile cloud computing. In: IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), pp. 224–229 (2014)
17. Artail, A., Frenn, K., Safa, H., Artail, H.: A framework of mobile cloudlet centers based on the use of mobile devices as cloudlets. In: IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 777–784 (2015)
18. Wu, Y., Ying, L.: A cloudlet-based multi-lateral resource exchange framework for mobile users. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), pp. 927–935 (2015)
19. Ravi, A., Peddoju, S.K.: Mobility managed energy efficient Android mobile devices using cloudlet. In: IEEE Conference on Students' Technology Symposium (Tech-Sym), pp. 402–407 (2014)