Shen *et al. EURASIP Journal on Wireless Communications and Networking* (2019) 2019:141
https://doi.org/10.1186/s13638-019-1455-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

ECPM: an energy-efficient cloudlet placement method in mobile cloud environment

Chao Shen[1,2], Shengjun Xue[1,3*] and Shucun Fu[1,2]

Abstract

The development of mobile cloud computing has greatly improved the computing and storage performance of mobile devices. And mobile cloud computing is undoubtedly the necessary way to solve the performance of the process for mobile applications with high performance requirements. However, migrating the mobile applications to the cloud brings about a migration delay, which is intolerable for high real-time demanding applications. This can be technically achieved by expanding mobile cloudlets, co-located with access points (AP). Then, how to deploy the cloudlet to reduce the energy consumption has become a part of major challenges in the current study. In this paper, we propose an energy-efficient cloudlet placement method, named ECPM, to effectively reduce the number of cloudlets, so as to achieve the energy savings. Specifically, the clustering of mobile devices at each time is obtained, and then the cloudlet will move to the clustering to achieve the best use of energy. Finally, the experimental results demonstrate that the proposed method is energy saving.

Keywords: Mobile cloud computing, Cloudlet placement, Energy-efficient

1 Introduction

With the popularity of mobile devices, people's demands for mobile applications are also increasingly widespread. There are many mobile applications with high computational power and high response latency. In this situation, the computing power of the mobile device itself is unable to meet the processing requirements of the applications; the mobile cloud is consequently applied to such situations, to solve the current problems faced by mobile devices. Mobile cloud computing can move applications from mobile devices through remote migration to the cloud for processing, which can greatly improve the computing power [1–3]. However, the current cloud and the mobile side of the distance are relatively far, and consequently, the migration process may bring a relatively high migration delay. This delay is intolerable for some applications that require stringent response time, such as natural language processing, face recognition, interactive games, etc. [4–6].

In order to overcome the huge delay caused by remote application migration, a small cloud called "cloudlet" is deployed to a near place to provide enhanced cloud services for users [7–9]. The cloudlets can be divided into two types: first, users connect the small servers provided by network operators to provide small servers, through the access to access points (AP), and this is self-organization [2, 10, 11]; second, the cloudlets that provide the formation of idle resources through a number of P2P networks under the mobile devices [12–14]. With the cloudlet, mobile devices can migrate their applications to the nearest cloudlet. In this way, users can greatly reduce the huge delay caused by remote migration, to achieve the response time required by a particular application.

The cloudlet resource is scarce and costly; therefore, it cannot be deployed wherever it is needed. In order to strengthen the cloud service experience for as many users as possible, how to effectively use the limited cloudlet resource has become a hot topic of the current researches [15]. Specifically, in a particular area and given a certain number of cloudlets, how to deploy the cloudlet to cover

*Correspondence: ksjxue@163.com
[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[3]School of Computer Science and Technology, Silicon Lake College, Suzhou, China
Full list of author information is available at the end of the article

the largest number of mobile devices in this area, that is, to achieve the most efficient utilization rate of cloudlet.

At present, a lot of researches have been done on the deployment of the cloudlet, mainly focusing on how to let the cloudlets cover as many network nodes as possible [1, 16, 17]. But the current deployment for the mobility of large flow is almost undesirable, it is significantly only for the people that in fixed work locations daily [18–20]. Therefore, in this article, a concept, cloudlets, is introduced to solve this problem in the huge changes of flow occasions. The goal of mobile cloud is to set the fewest cloudlets, to minimize cloudlet energy consumption, to maximize the utilization efficiency, and to effectively enhance the user's cloud services experience.

Now, let us look at a realistic application scenario. Inside a large event center, some active person groups are randomly dispersed. Most of the active people of the crowd will connect the wireless network through the AP to access the equipment networking. In order to strengthen the activities of the crowd experience, each AP will be equipped with cloudlets to enhance cloud services [21–23]. But for the crowd within the active area, their activities are irregular, and even the frequency of activities is high. For example, in Fig. 1, a large crowd aggregation will move from location 1 to location 2 at a certain moment. However, the coverage range of AP is limited; then, in order to make the crowd get access to cloud services, the sponsor need to equip a lot of cloudlets. However, in the frequent activities of people, some of the cloudlet signal can cover very few people at certain moments, which results in a lot of energy waste. In order to save energy, the best way is to fix a small number of cloudlets, and the rest of cloudlets follow the flow of people, always to provide cloud services to most people [24–26]. As shown in Fig. 1, when the population in position 1 moves to position 2, it is absolutely feasible to remove the cloudlet of position 1 to position 2, which does not increase the energy consumption and meanwhile provides the majority of the population reliable and enhanced cloud services. At present, the moving cloudlets can be moved by moving the robot that equipped with AP [27–29].

How to effectively set the movement of the cloudlets to reduce energy waste is still a difficult problem. In this paper, we propose a dynamic method of cloudlet placement, which can adjust the location of the cloudlet in real time according to the location change of the mobile device to provide more effective cloud service. First of all, through the K-means algorithm, we can obtain the k central aggregation point positions for all the equipment, and these k positions are the locations where cloudlets need to be placed at the moment, then matching the current location of the cloudlet and the target location, and finally moving the mobile cloud to the destination through the cloudlet movement strategy proposed in this paper. Through the simulation experiment, it is proved that the proposed method is effective, which can save the number of cloudlet and obtain the most efficient utilization rate.

Our paper is organized as follows. Section 2 puts forward the strategy of moving cloudlets. Section 3 designs an energy-efficient cloudlet placement method. In Section 4, through the experimental comparative analysis, we proved our proposed method can make better use of the cloudlet, from the perspective of achieving the effect of green energy. The last two chapters describe some related works and prospects for future work.

2 Cloudlet placement and movement analysis for mobile devices

This part mainly describes the dynamic cloudlet placement method and defines some parameters used in the construction of the method. The specific definition is shown in Table 1.

The cloudlet is playing an important role in the cloud architecture, which is shown in Fig. 2. The mobile devices are connected to the cloudlet through the wireless network or AP. Mobile cloud and remote cloud provide data

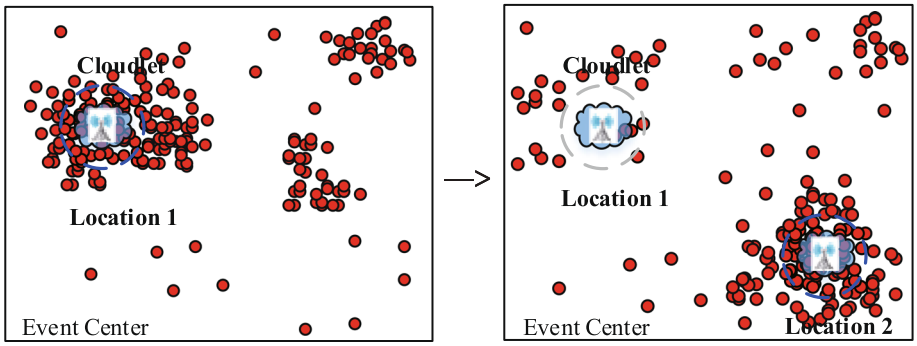

Fig. 1 The movement of people in a realistic application scenario

Table 1 Key parameters and descriptions

Item	Description
AREA	The collection of the entire activity area AREA $= \{(x,y)\,\vert 0 \le x \le W, 0 \le y \le H\}$
DEVICE	A collection of mobile devices DEVICE $= \{d_1, d_2, \ldots, d_N\}$
Device_n	Represents the nth device in the mobile device collection
$\text{Deviceposition}_n\,(t)$	The location of the mobile device at time t.
CLOUDLET	The collection of cloudlets CLOUDLET $= \{\text{cloudlet}_1, \text{cloudlet}_2, \ldots, \text{cloudlet}_M\}$
AP	The collection of AP $P = \{p_1, p_2, \ldots, p_M\}$
Cloudlet_m	The mth mobile cloud
$\text{Cloudposition}_m\,(t)$	The center location of mobile cloud at time t.
Radium_m	The coverage radius of mobile cloud cloudlet_m
$\text{Devicecollection}_m\,(t)$	The device collection at time t
Thresh	The threshold to determine whether to place mobile cloud
TotalN (t)	The total number of covered devices by all cloudlets
$\text{Devicecollection}_{m,P}\left(t'\right)$	At time t, the device collection of cloudposition_m located at the position of $P\,(x,y)$

storage and computing services. Compared to the traditional client-server model [30–32], the cloudlet architecture with mobile cloud can effectively reduce the access delay. Because the mobile device from the cloudlets are relatively close, getting connected to the cloudlet through the AP so as to obtain the cloud storage and computing performance is very fast.

Of course, the above cloudlet architecture can enhance the mobile cloud services well for users. In order to enable mobile users to get access to cloud service better, this paper will be set in a rectangular area, because almost all shapes can be made through the combination of the rectangle. The definition of the rectangular area is as follows.

Definition 1 (Device moving range) *The entire device moving range is defined in an $x-y$-axis plane and defines the region range* $\text{AREA} = \left\{(x,y)\,\vert 0 \le x \le W, 0 \le y \le H\right\}$.

In the device moving range, a large number of mobile devices are randomly distributed, where the mobile device is defined as:

$\text{DEVICE} = \{\text{device}_1, \text{device}_2, \ldots, \text{device}_N\}$, where N is the number of mobile devices in area AREA. There is a coordinate to represent each mobile device.

Definition 2 (Device location at time t) *For the nth* $(1 \le n \le N)$ *mobile devices* device_n, *the coordinate location at time t is* $\text{deviceposition}_n\,(t) = (\text{devicepositio}x_n\,(t), \text{devicepositio}y_n\,(t))$ *(where* $\text{devicepositio}x_n\,(t)$ *represents the abscissa and* $\text{devicepositio}y_n\,(t)$ *represents the vertical position of the mobile device).*

The mobile cloud is usually loaded on the AP to enhance the user's cloud service experience, assuming that there are M cloudlets in area AREA, then the cloudlet can be defined as $\text{CLOUDLET} = \{\text{cloud}_1, \text{cloud}_2, \ldots, \text{cloud}_M\}$. Each cloud corresponds to an AP, so there are M APs, which are defined as $P = \left\{p_1, p_2, ..., p_M\right\}$. In order to determine the location of the cloudlet, we also need to define the center of the cloudlet, as shown in Definition 3.

Definition 3 (Central position of cloudlet_m at time t) *For the mth* $(1 \le m \le M)$ *cloudlet* cloudlet_m, *each moment has a central position (i.e., where the AP is located), which is defined as:* $\text{cloudposition}_m\,(t) = (\text{cloudpositio}x_m\,(t), \text{cloudpositio}y_m\,(t))$. *Where* $\text{cloudpositio}x_m\,(t)$ *represents the abscissa and* $\text{cloudpositio}y_m\,(t)$ *represents the ordinate position in the coordinate system.*

As long as within the signal coverage of the cloudlet, users will be able to enjoy the enhanced cloud services. So, the cloudlet will be placed in the most densely populated areas of the user; meanwhile, we need to monitor real-time cloudlet coverage within the device set. We define the radius of the cloudlet as r_m; then, the collection of devices covered by cloudlets can be easily obtained.

If the device set in the coverage range of the cloudlet_m is $\text{devicecollection}_m\,(t) = \{\text{device}_n\,(t)\,\vert\, \text{dis} \le r_m, 1 \le n \le N\}$, the distance $\text{dis}\{\text{deviceposition}_\text{n}(\text{t}), \text{cloudposition}_m(t)\}$ can be measured with Euclidean distance [33, 34], which is calcaulated by:

$$\text{dis}\left(\text{deviceposition}_n\,(t), \text{cloudposition}_m\,(t)\right) = \sqrt{\begin{array}{l}(\text{devicepositio}x_n\,(t) - \text{cloudpositio}x_m\,(t))^2 \\ +(\text{devicepositio}y_n\,(t) - \text{cloudpositio}y_m\,(t))^2\end{array}} \tag{1}$$

Definition 4 (Cloudlet placement strategy) *At time t, if the cloudlet* cloudlet_m *satisfies the coverage condition at position* $P\,(x,y)$: $\text{devicecollection}_m\,(t) \ge \text{thresh}$ (where thresh is the threshold to determine whether the cloudlet is placed), then the cloudlet can be placed at position $P\,(x,y)$, otherwise removed.

In the entire equipment activity area AREA, it is necessary to maximize the number of the covered devices, where the number of covered devices is calculated as:

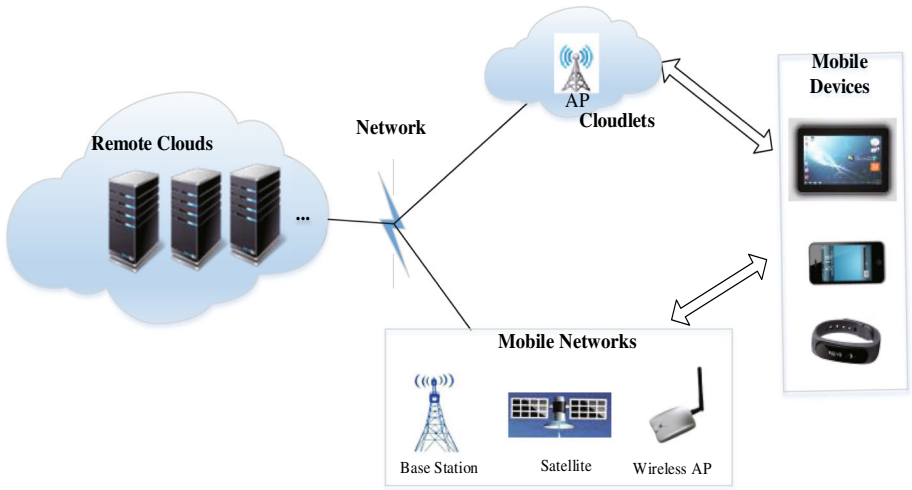

Fig. 2 The architecture of cloudlet communication for mobile devices

$$\text{TotalN}(t) = |\bigcup_{m=1}^{M} dc_m(t)|. \tag{2}$$

Assuming that at time t, the cloudlet cloudlet_m is placed at position $P(x, y)$; after the time period (t, t'), some mobile devices are moved in the area AREA, and in order to keep the highest utility efficiency of the cloud and save energy, it is necessary to define a cloudlet movement strategy.

Definition 5 (Movement strategy of mobile cloudlets) *To the time t', if the mobile device set on the location $P(x, y)$ and the mobile device set on the location $P'(x', y')$ satisfy the placement strategy:* $\text{devicecollection}_{m,P'}(t') \geq \text{devicecollection}_{m,P}(t')$, *and there is no other closer cloudlet near the position* $\text{devicecollection}_{m,P'}(t') \geq \text{devicecollection}_{m,P}(t')$, *then the* cloudlet_m *will move from position P to P'.*

3 Energy-efficient cloudlet placement method

This section presents a dynamic placement method of mobile cloud, which consists three steps: device center location recognition, cloudlet location determination, and dynamic placement of cloudlet, as shown in Fig. 3.

3.1 Center location recognition

The cloudlets should be moved to the locations with intensive mobile devices, to provide high-quality services. Therefore, it is very necessary to achieve the central locations of the mobile devices.

We consider the mobile device distribution in the active area, where the mobile device can be represented by a set: $\text{DP} = \{\text{deviceposition}_1, \text{deviceposition}_2, ..., \text{deviceposition}_N\}$. Because the cost of the mobile cloud is expensive, we should set the clouds as few as possible and make the cloudlets cover as much as possible to ensure the highest efficiency clouds and save the cloudlet energy consumption. The most appropriate

Step 1: Center Location Recognition. In order to make the use of the highest efficiency of the cloudlet, we must know the location of cloudlet by gathering the positions of mobile devices. These locations can be determined by the K-means clustering algorithm. After determining the location, it can be abstracted into a path map to be processed.

Step 2: Energy-aware Location Confirmation. For the above obtained central locations, we need to filter out the locations that do not meet the conditions, according to the cloud clouding strategy, to achieve energy savings. Then we select the nearest cloud for each location and do the location matching, to obtain the center of the location where the mobile cloud should reach.

Step 3: Dynamic Cloudlet Movement. Generate the moving trajectory of the cloudlets according to the regional structure. We apply the DAG to assist the movement of the cloudlets, to avoid obstacles to generate a mobile path. In addition, in order to save energy, we keep the original position of the cloudlet unchanged which does not have destination position.

Fig. 3 Main steps of the proposed energy-efficient cloudlet placement method

way to do this is to put the cloudlets in the equipment of the most intensive areas, that is, to find a number of cluster center of mobile devices, and K-means clustering algorithm is employed to obtain the clustering centers.

In addition, considering the various obstacles that may be encountered when moving the cloudlet, the path is abstracted into a graph, and the points and edges of the graph are expressed as $V = \{v_1, v_2, ..., v_U\}$ and $E = \{e_1, e_2, ..., e_U\}$, respectively. This graph is just an abstraction of the entire device moving range, and the following is a definition of the path map.

Definition 6 (Device moving range path map) *The path map is used to represent the center of the cluster and to help the cloudlet to plan the trajectory better, and is described as:* $G = (E, V, W)$ *(where V is the position where the cloud can move, E represents the movement path of the cloudlet, and W represents the weights on all positions)* [35, 36].

According to the abovementioned path map, the measured center position of the device can be adjusted to the nearest node V.

Algorithm 1 specifically describes the identification process of these central locations, which are the location where the cloudlet should actually be placed.

Algorithm 1 Center position recognition (DP, V)

Require: The set of mobile device location DP; The set of cloudlet available locations V.

Ensure: The set of central positions for cloudlet placement CP.

Initialize the number of centers, i.e., k, as N
Confirm the cluster centers by using K-means algorithm
$CP \leftarrow \emptyset$
for i=1 to N **do**
  $MD = dis\,(cp_i, v_1)$
  $PP = v_1$
  for j=1 to U **do**
    Calculate $dis\left(cp_i, v_j\right)$ by Eq.(1)
    if $dis\left(cp_i, v_j\right) < MD$ **then**
      $MD = dis\left(cp_i, v_j\right)$
      $PP = v_j$
    end if
  end for
  $cp_i = PP$
  Remove v_i from V
end for
return CP

3.2 Energy-aware location confirmation

In order to facilitate the energy-efficient cloudlet placement, several mobile device aggregation centers were obtained in Section 3.1. Dense mobile devices act around the center of the cluster, so that the cloudlets can cover the largest crowd, making the use of efficiency be maximized. Of course, maybe only a small number of mobile devices are around the center of the cluster obtained in step 1, if the clouds are placed inside such centers, the use of efficiency will be low, resulting in a lot of waste of energy consumption; therefore, a placement rule of the cloud need to be developed to save energy.

After filtering out the unnecessary equipment centers, we need to find the target location for each cloudlet. In this paper, the initial position of each cloudlet will be given, and consequently the distance between the cloudlet and the central location can be determined, so that the nearest equipment gathering center cloudlet be found for the merit location matching.

We try to make full use of the running cloudlets and sleep the cloudlets that serve only several a little number of mobile devices. Algorithm 2 filters the center locations of the cloudlets that do not meet the criteria (which is definitely measured by the density threshold thresh) and calculates the distance between each central location and the cloudlet.

Each obtained central position need a cloudlet to move on; thus, there are some cloudlet-location pairs. The pair matching process could help to improve the efficiency of dynamic cloudlet placement. Algorithm 3 matches the mobile device center location and the initial location of the cloudlets.

Algorithm 2 Energy-aware location filtering (DP, CP)

Require: The set of mobile device location DP; The set of device clustering central locations CP.

Ensure: The updated set of cloudlet central locations.

for i=1 to $|CP|$ **do**
  $dp_i \leftarrow \emptyset$
  for j=1 to M **do**
    Get the location of dp_j
    Calculate the distance $dis\left(cp_i, dp_j\right)$ by Eq. (1)
    if $dis\left(cp_i, dp_j\right)$ is less than the threshold r **then**
      Add dp_i to dc_i
    end if
  end for
  if $|dc_i| < thresh$ **then**
    Remove cp_i from CP
    $|CP| = |CP| - 1$
  end if
end for
return CP

Algorithm 3 Cloudlet-location pair matching (CP, L)

Require: The set of cloudlet central positions CP; The set of cloudlet L.
Ensure: The pairs of cloudlets and the relevant central locations.

for i=1 to $|CP|$ **do**
  Copy L to ml_i
  for j=1 to M **do**
    Calculate the distance between cp_i and ml_j by Eq.(1)
  end for
  Sort the ml_i in the increasing order of the obtained distances
end for
$num = |CP|$
while $num > 0$ **do**
  for j=1 to M **do**
    if cp_i has not a pair of cloudlet **then**
      Assign cp_i as the pair of cloudlet $ml_{num,1}$
    end if
    Remove $ml_{m,1}$ from $ml_2, ml_3, .., ml_M$
    Update $ml_2, ml_3, .., ml_M$
  end for
  $num = num - 1$
end while

3.3 Dynamic cloudlet movement

This section designs a cloud service enhancement method for the cloudlet to provide support for applications in the mobile cloudlet environment. In step 2, the weight path map is provided, and the movement trajectory of each cloud is determined according to the weight path map.

The initial position and destination of all the cloudlets are set in the graph $G = (E, V, W)$, so that the problem of moving path of the cloudlet can be transformed into the shortest path search problem. We use Dijkstra algorithm to find the shortest path, and the results will be converted to the move trajectory of the mobile cloudlet.

In the real world, the cloudlet cannot directly be moved to another place directly, and there are many restrictions for the moving of cloudlets that the cloudlets can only choose several directions. To give the help of cloudlet movement, we try to monitor the cloudlets under the control of reasonable movement traces.

Algorithm 4 describes the process of dynamic cloudlet movement, to determine the movement path of the cloudlets, which provides the most efficient cloud enhancement service through moving cloudlet to the nearest clustering center to achieve the best energy savings.

Algorithm 4 Dynamic cloudlet movement

Require: The path graph G without weights.
Ensure: The moving traces of all the cloudlets.

Get the central locations CP by using **Algorithm 1**
Update the central locations by using **Algorithm 2**
Get the pairs of central locations and cloudlets by using **Algorithm 3**
for i=1 to $|DP|$ **do**
  for j=1 to M **do**
    for k=1 to $|CP|$ **do**
      if l_j is the pair of cp_k **then**
        Get the shortest trace between cp_i and l_j by the Dijkstra Algorithm
        Record the moving path from l_j to cp_i
      end if
    end for
  end for
end for
Update the moving trace for all the cloudlets

4 Experiment evaluations

In this chapter, we mainly compare the experimental method and the experimental results of DBSCAN algorithm to verify the effectiveness of the method.

4.1 Experiment settings

In order to simplify the experiment, simulation is used in the experiments. The shape of the active area is set to a square with the side length being 280 m. In this area, a large number of mobile devices are randomly distributed. In order to be close to the real world, the distribution of mobile devices is set to satisfy the law of Gaussian distribution. The specific parameter settings are shown in Table 2.

4.2 Results and discussion

In this section, the experimental results are evaluated, mainly based on the comparison of the movement, equipment coverage, and energy saving.

Figure 4 compares the mobile device coverage between the method in this paper and DBSCAN algorithm based on different numbers of cloudlets. Based on the above

Table 2 Parameter settings

Parameter	Values
X: active area length X	280 m
Y: active area length Y	280 m
Threshold in the cloudlets placement strategy	40
The radius of cloudlet movement	60 m
The number of AP	{1, 2, 3, 4, 5, 6}
The number of mobile devices	{300, 400, 500, 600, 700, 800}
Time intervals between experiments	40 min

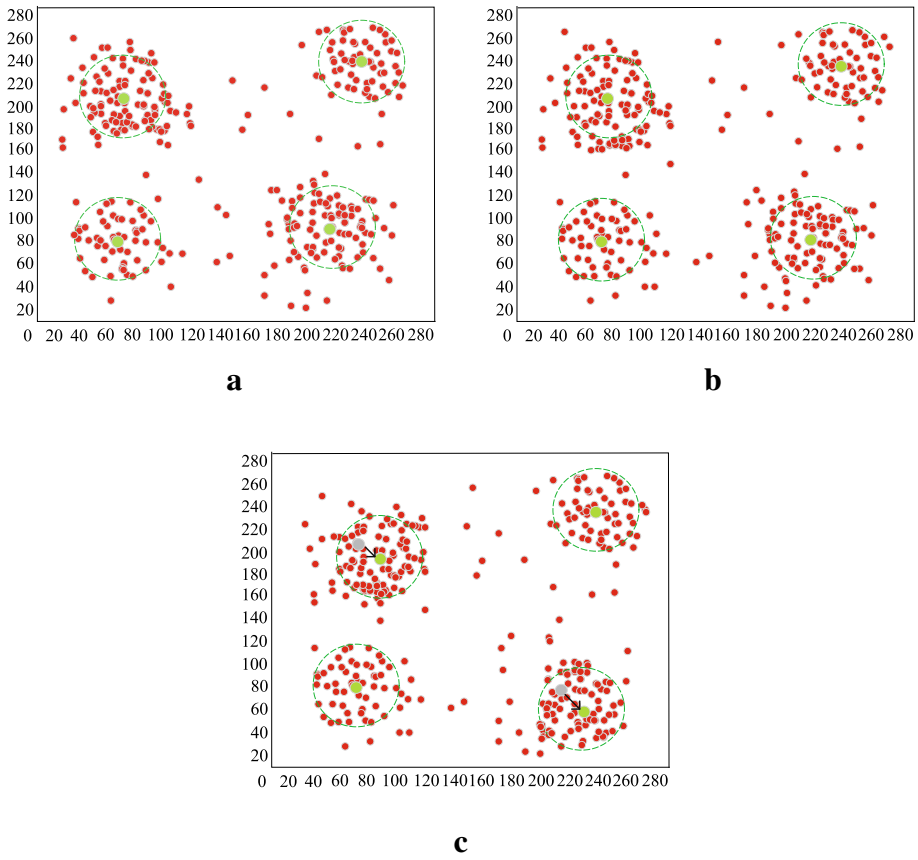

Fig. 4 Distribution of mobile devices at three different moments of the cloudlet movement. **a** States at time t_0. **b** States at time t_1. **c** States at time t_2

assumptions, each simulation experiment was performed 40 different data records with different mobile device locations, and the average coverage value was obtained as the final experimental result. As shown in Fig. 4, we can find that when the time is at t_2, the position of the mobile device changes compared to the time t_0. According to the center location recognition proposed in this paper, two clustering centers can be found to change, so as to better cover the mobile devices range.

As can be seen in Fig. 5, the method we proposed can cover more mobile devices than the DBSCAN algorithm. In the stage where the number of mobile cloudlets is small, the performance of the proposed method is roughly the same as that of the DBSCAN method. When the number of mobile cloudlets increases, the number of covered mobile devices obtained proposed by this method is gradually larger than the DBSCAN method. The number of covered devices increases with the increase of the number of cloudlets, but it can be seen that the increase amplitude of covered devices decreases as the number of cloudlets increases. So, when the number of cloudlets increases to a certain number, it will affect its use efficiency, resulting in energy waste.

As we can see in Fig. 6, after time period t, in the situation that locations of mobile devices location change, with the number of devices to cover increases, the number of cloudlets also substantially increases. In addition, the experiment compares the situation of fixed location cloudlets and the moving cloudlets based on the method proposed in this paper, and we found that in order to cover the same number of mobile devices, ECPM uses less cloudlets, proving that ECPM is a green energy. As the number of covered mobile devices continues to increase, the ECPM method proposed in this paper is significantly better.

5 Related work

As cloud resources have better availability, security, and scalability, it has become an attractive method to enhance the performance of mobile terminals by using cloud resources. We call this method a computational migration in a mobile cloud computing environment [37–41].

Qian et al. [39] designed the Jade system, which can add complex energy-aware computing migration functions for Android applications. Chen et al. [40] proposed a design pattern to enable applications to perform computational

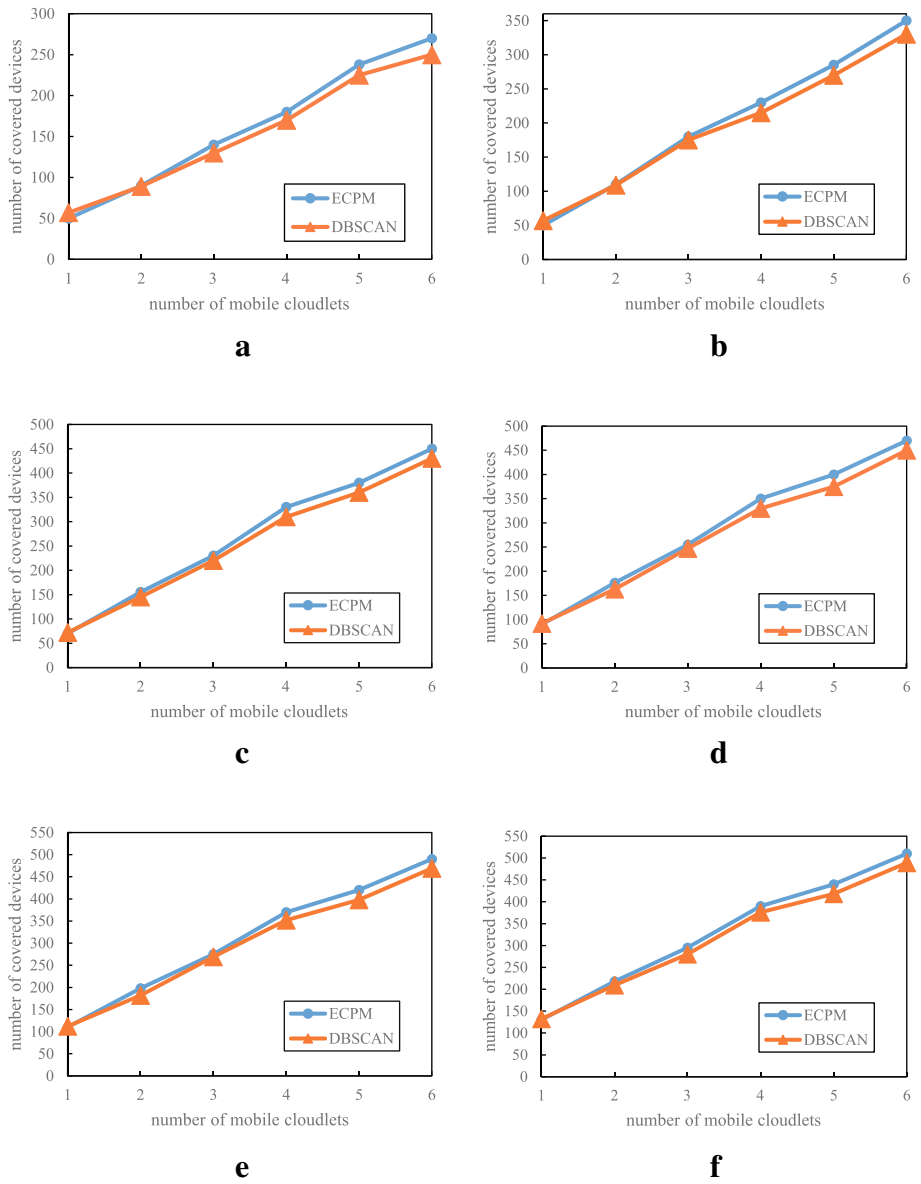

Fig. 5 Comparison of the covering equipment between the ECPM and DBSCAN. **a** 300 mobile devices. **b** 400 mobile devices. **c** 500 mobile devices. **d** 600 mobile devices. **e** 700 mobile devices. **f** 800 mobile devices

migration on demand and proposed an estimation model to automatically select cloud resources for migration. Shi et al. [41] designed the COSMOS system to provide computational migration as a service to mobile devices. COSMOS effectively manages the cloud resources used to migrate requests, improving mobile device migration performance and reducing the cost of sending each request to the provider.

In order to enhance the computing and storage capabilities of mobile devices, mobile applications and data are often migrated to the cloud side to be processed, and this has been studied much [1, 2, 4, 42–44]. However, the cloud and the device side are usually very far; the migration between applications and the equipment side always results in a high migration delay. In order to address this challenge, the concept of "cloudlet" is proposed and studied, for example, [17, 45–48], all describe cloudlet computing and related technologies.

In [17], the nearest cloudlet acts as a proxy server, which selects the optimum cloudlet among its nearby cloudlets, with respect to minimum power consumption or minimum latency or both. Cao et al. [45] presented a full distributed computation offloading algorithm based on machine learning, which can solve the multi-users

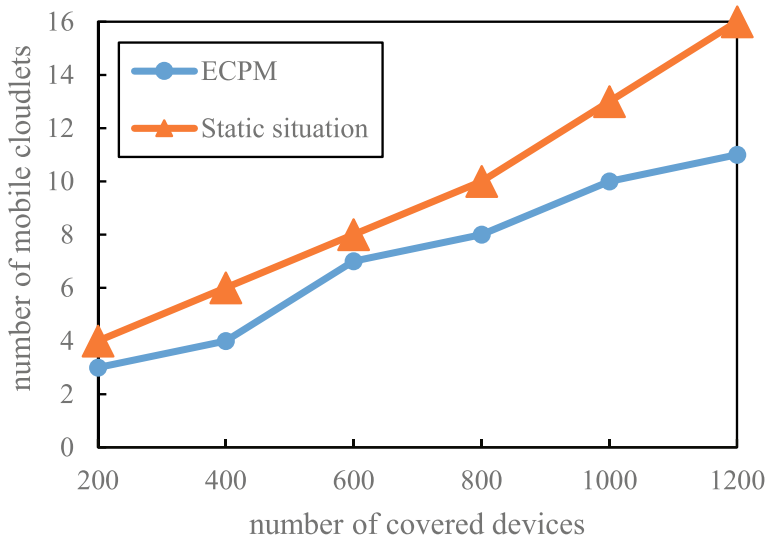

Fig. 6 The number of required cloudlets after a time period of *t*

computation offloading problem based on cloudlet in multi-channel wireless contention environment. In [46], authors address the problem of choosing the appropriate position to place cloudlet to reduce the user's access delay. For service provider, it is always very costly to deploy cloudlets; thus, how many cloudlets should be placed in a WMAN and how much resource each cloudlet should have is the challenging issue in [46]. In [47], the author optimizes the relationship between the local cloud and the remote cloud, maintaining a balance between energy consumption and QoS. Roy et al. [48] proposed an application-aware selection strategy in the cloudlet scenario, which can promote the energy cost of mobile devices and the communication delay of mobile application.

6 Conclusion and future work

In this paper, we focus on the cloudlet enhancement service and propose an adaptive method of mobile cloudlet to save energy consumption and improve the utilization efficiency of the cloudlet. In particular, the K-means algorithm is employed to obtain the dense gathering center of the mobile device, which is the best position for the cloudlet. Then, according to the mobile cloudlet placement strategy, we match the current location of the cloudlet and the nearest gathering center location. Finally, according to the proposed mobile cloudlet movement strategy, we can get the real-time cloudlet movement trajectory. The experiments verified that the proposed method proposed is energy efficient.

In the future, we hope that this green energy-saving cloudlet placement method will be applied to the real-life environment to do the field detection and remove the adverse impact factors. And we hope to be able to detect the overall response time of mobile applications in ensuring the premise of green energy, to meet the actual response needs.

Acknowledgements
This research is supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund and Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAEET).

Availability of data and materials
The dataset supporting the conclusions of this article is available, which can be downloaded at https://drive.google.com/open?id=1pThqWafKfniPwqCcFqe-JoIZPlVrEW2r.

Authors' contributions
CS, SX, and SF conceived and designed the study. CS and SF performed the simulations. CS wrote the paper. All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China. [2]School of Computer and Software, Jiangsu Engineering Center of Network Monitoring, Nanjing, China. [3]School of Computer Science and Technology, Silicon Lake College, Suzhou, China.

Received: 2 March 2019 Accepted: 25 April 2019
Published online: 31 May 2019

References

1. M. Jia, J. Cao, W. Liang, Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. IEEE Trans. Cloud Comput. **PP**(99), 1–1 (2017)
2. Z. Xu, W. Liang, W. Xu, M. Jia, G. Song, Efficient algorithms for capacitated cloudlet placements. IEEE Trans. Parallel Distrib. Syst. **27**(10), 2866–2880 (2016)

3. M. Satyanarayanan, in *Acm Workshop on Mobile Cloud Computing & Services: Social Networks & Beyond*. Mobile computing: the next decade, (2010)
4. K. O. Renkonen, M. Seppala, Edge analytics in the internet of things. IEEE Pervasive Comput. **14**(2), 24–31 (2015)
5. J. Cohen, in *IEEE International Conference on Acoustics*. Embedded speech recognition applications in mobile phones: status, trends, and challenges, (2008)
6. W. Liu, J. Cao, X. Qiu, J. Li, in *IEEE International Conference on Cloud Computing Technology & Science*. Improving performance of mobile interactive data-streaming applications with multiple cloudlets, (2014)
7. X. Xu, Y. Li, T. Huang, Y. Xue, K. Peng, L. Qi, W. Dou, An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks. J. Netw. Comput. Appl. **133**, 75–85 (2019)
8. S. Wang, L. Zhang, Y. Zhang, J. Sun, C. Pang, G. Tian, N. Cao, Natural language semantic construction based on cloud database. CMC-Comput. Mater. Contin. **57**(3), 603–619 (2018)
9. S. Wang, A. Zhou, R. Bao, W. Chou, S. S. Yau, Towards green service composition approach in the cloud. IEEE Trans. Serv. Comput. (2018)
10. K. A. Khan, Q. Wang, C. Grecos, C. Luo, X. Wang, in *IEEE International Conference on Electronics*. Meshcloud: integrated cloudlet and wireless mesh network for real-time applications, (2014)
11. Y. Xu, L. Qi, W. Dou, J. Yu, Privacy-preserving and scalable service recommendation based on simhash in a distributed cloud environment. Complexity, 2017 (2017)
12. Y. Li, W. Wang, in *IEEE Infocom*. Can mobile cloudlets support mobile applications? (2014)
13. X. Xu, S. Fu, L. Qi, X. Zhang, Q. Liu, Q. He, S. Li, An IoT-oriented data placement method with privacy preservation in cloud environment. J. Netw. Comput. Appl. **2018**(124), 148–157 (2018)
14. X. Wang, L. T. Yang, X. Xie, J. Jin, M. J. Deen, A cloud-edge computing framework for cyber-physical-social services. IEEE Commun. Mag. **55**(11), 80–85 (2017)
15. T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian, Y. Chen, B. Wang, Data collection from WSNs to the cloud based on mobile fog elements. Futur. Gener. Comput. Syst. (2017)
16. M. Jia, W. Liang, Z. Xu, M. Huang, in *IEEE International Conference on Computer Communications*. Cloudlet load balancing in wireless metropolitan area networks, (2016)
17. A. Mukherjee, D. De, D. Roy, A power and latency aware cloudlet selection strategy for multi-cloudlet environment. IEEE Trans. Cloud Comput. **PP**(99), 1–1 (2016)
18. K. Zhang, Y. Wang, An h-tensor based iterative scheme for identifying the positive definiteness of multivariate homogeneous forms. J. Comput. Appl. Math. **305**, 1–10 (2016)
19. X. Wang, L. T. Yang, L. Kuang, X. Liu, Q. Zhang, M. J. Deen, A tensor-based big-data-driven routing recommendation approach for heterogeneous networks. IEEE Netw. **33**(1), 64–69 (2018)
20. S. Wang, A. Zhou, M. Yang, L. Sun, C.-H. Hsu, et al., Service composition in cyber-physical-social systems. IEEE Trans. Emerg. Top. Comput. (2017)
21. X. Xu, Y. Xue, L. Qi, Y. Yuan, X. Zhang, T. Umer, S. Wan, An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Futur. Gener. Comput. Syst. (2019)
22. X. Xu, Q. Liu, Y. Luo, K. Peng, X. Zhang, S. Meng, L. Qi, A computation offloading method over big data for IoT-enabled cloud-edge computing. Futur. Gener. Comput. Syst. (2019)
23. S. Lian, Y. Duan, Smoothing of the lower-order exact penalty function for inequality constrained optimization. J. Inequalities Appl. **2016**(1), 185 (2016)
24. J. Q. Cai, H. Li, An implicit degree condition for relative length of long paths and cycles in graphs. Acta Math. Applicatae Sin. Engl. Ser. **32**(2), 365–372 (2016)
25. S. Wang, A. Zhou, F. Yang, R. N. Chang, Towards network-aware service composition in the cloud. IEEE Trans. Cloud Comput. (2016)
26. Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, C.-H. Hsu, User allocation-aware edge cloud placement in mobile edge computing. Softw. Pract. Experience (2019)
27. T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, Q. Jin, A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing. IEEE Internet Things J. (2018)
28. T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, W. Jia, Fog-based computing and storage offloading for data synchronization in IoT. IEEE Internet Things J. (2018)
29. F. Ma, G. Sheng, Y. Yin, A superlinearly convergent method for the generalized complementarity problem over a polyhedral cone. Journal of Applied Mathematics. **2013** (2013)
30. A. Bahtovski, M. Gusev, in *International Convention on Information & Communication Technology, Electronics & Microelectronics*. Multilingual cloudlet-based dictionary, (2014)
31. Q. Bai, X. Xu, J. Xu, D. Wang, Coordinating a supply chain for deteriorating items with multi-factor-dependent demand over a finite planning horizon. Appl. Math. Model. **40**(21-22), 9342–9361 (2016)
32. T. Wang, G. Zhang, M. Z. A. Bhuiyan, A. Liu, W. Jia, M. Xie, A novel trust mechanism based on fog computing in sensor–cloud system. Futur. Gener. Comput. Syst. (2018)
33. G. Wang, X. Yang, T. Cheng, Generalized Levitin-Polyak well-posedness for generalized semi-infinite programs. Numer. Funct. Anal. Optim. **34**(6), 695–711 (2013)
34. C. Wang, C. Ma, J. Zhou, A new class of exact penalty functions and penalty algorithms. J. Glob. Optim. **58**(1), 51–73 (2014)
35. B. Liu, B. Qu, N. Zheng, A successive projection algorithm for solving the multiple-sets split feasibility problem. Numer. Funct. Anal. Optim. **35**(11), 1459–1466 (2014)
36. J. Cai, An implicit sigma (3) type condition for heavy cycles in weighted graphs. Ars Combinatoria. **115**, 211–218 (2014)
37. S. Li, Y. Zhang, On-line scheduling on parallel machines to minimize the makespan. J. Syst. Sci. Complex. **29**(2), 472–477 (2016)
38. J. Cai, H. Li, A new sufficient condition for pancyclability of graphs. Discret. Appl. Math. **162**, 142–148 (2014)
39. H. Qian, D. Andresen, in *Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems*. Extending mobile device's battery life by offloading computation to cloud (IEEE Press, 2015), pp. 150–151
40. X. Chen, S. Chen, X. Zeng, X. Zheng, Y. Zhang, C. Rong, Framework for context-aware computation offloading in mobile cloud computing. J. Cloud Comput. **6**(1), 1 (2017)
41. C. Shi, K. Habak, P. Pandurangan, M. Ammar, M. Naik, E. Zegura, in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Cosmos: computation offloading as a service for mobile devices (ACM, 2014), pp. 287–296
42. S. Wang, Y. Zhao, J. Xu, J. Yuan, C.-H. Hsu, Edge server placement in mobile edge computing. J. Parallel Distrib. Comput. (2018)
43. M. V. Barbera, S. Kosta, A. Mei, J. Stefa, in *IEEE INFOCOM*. To offload or not to offload? The bandwidth and energy costs of mobile cloud computing, (2013)
44. Y. Wang, X. Sun, F. Meng, On the conditional and partial trade credit policy with capital constraints: a stackelberg model. Appl. Math. Model. **40**(1), 1–18 (2016)
45. H. Cao, J. Cai, Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: a game-theoretic machine learning approach. IEEE Trans. Veh. Technol. **67**(1), 752–764 (2018)
46. L. Ma, J. Wu, C. Long, in *IEEE/ACM International Symposium on Cluster*. DOTA: delay bounded optimal cloudlet deployment and user association in WMANs, (2017)
47. E. Gelenbe, R. Lent, M. Douratsos, in *Second Symposium on Network Cloud Computing & Applications*. Choosing a local or remote cloud, (2012)
48. D. G. Roy, D. De, A. Mukherjee, R. Buyya, Application-aware cloudlet selection for computation offloading in multi-cloudlet environment. J. Supercomput. **73**(4), 1672–1690 (2017)