An Adaptive Cloudlet Placement Method for Mobile Applications over GPS Big Data

Haolong Xiang[1], Xiaolong Xu[1], Haoquan Zheng[2], Shu Li[1], Taotao Wu[1], Wanchun Dou[1,*], Shui Yu[3]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]State Grid Electric Power Science Research Institute, Nanjing, China
[3]School of Information Technology, Deakin University, Melbourne, Australia

{hlx6700, njuxlxu, babblishu, wutaotaoxpy}@gmail.com, douwc@nju.edu.cn, shui.yu@deakin.edu.au

***Abstract*—Mobile cloud computing provides powerful computing and storage capacity on managing GPS big data by offloading vast workloads to remote clouds. For the mobile applications with urgent computing or communication deadline, it is necessary to reduce the workload transmission latency between mobile devices and clouds. This can be technically achieved by expanding mobile cloudlets that are moving co-located with Access Points (APs). However, it is not-trivial to place such movable cloudlets efficiently to enhance the cloud service for dynamic context-aware mobile applications. In view of this challenge, an adaptive cloudlet placement method for mobile applications over GPS big data is proposed in this paper. Specifically, the gathering regions of the mobile devices are identified based on position clustering, and the cloudlet destination locations are confirmed accordingly. Besides, the traces between the origin and destination locations of these mobile cloudlets are also achieved. Finally, the experimental results demonstrate that the proposed method is both effective and efficient.**

Keywords*—big data; mobile device; mobile cloud computing; cloudlet placement

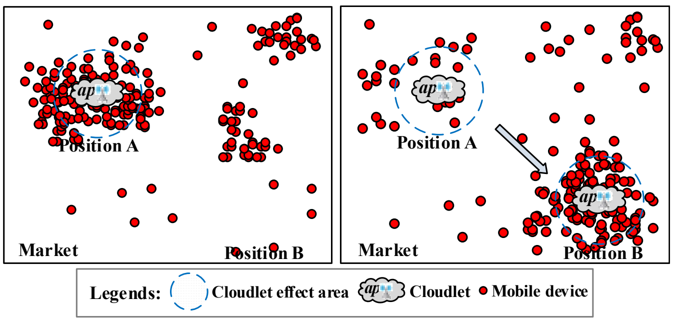
I. INTRODUCTION

With the rapid development of mobile devices and wearable technology, an increasing amount of mobile data is distributed among mobile devices. In particular, these mobile data not only have big capacity, but also have high latency in processing. In order to sufficiently manage above mobile big data, mobile cloud computing is applied to improve the computing capacity of mobile devices by offloading the vast workloads to remote cloud [1]. Although mobile cloud computing can help to overcome limitations of mobile devices in particular of the processing power and data storage [2], the clouds are geographically far away from mobile users. Thus communication delays between the clouds and mobile devices may be long, which is intolerable for some mobile applications, such as natural language processing, face recognition, interactive gaming [3], [4].

In order to overcome the long latency, cloudlets are deployed close to users to provide rich computing and storage resources for mobile applications. Cloudlets are usually classified into two types: one is offered by the network operators (NOs), which is self-managing and can be accessed through wireless Access Points (APs) [5]; the other is self-organized between mobile devices by P2P network [6]. Thus mobile users can offload their tasks to local cloudlets for processing rather than to the remote clouds. As a result, mobile applications can benefit from such physical proximity offloading process with lower transmission latency.

In practice, the cloudlet is so scarce that we could not deploy it everywhere. Accordingly, how to cost-effectively deploy the capacitated cloudlets to enhance the cloud services for dynamic context-aware mobile applications is a meaningful problem. Formally speaking, given a certain area with specific distribution of mobile devices, we need to deploy a limited number of mobile cloudlets to maximize the whole system utility. In this paper, we take the pressure of network traffic as a specific measurement of our system utility.

Current researches mainly focus on capacitated cloudlets placement to cover most network nodes and these cloudlets are often fixed after their placement [7], [8], [9]. These stationary cloudlets cannot keep efficient in response to the application requirements for frequently moving devices. Movable cloudlets introduced in this paper are capable to move adaptively to enhance the cloud service for mobile users.



(a) before cloudlet movement (b) after cloudlet movement

Fig. 1 A motivating example of adaptive cloudlet placement.

Here, a motivating example is presented to illustrate the problem investigated in this paper. A number of mobile devices, located through GPS, are randomly distributed in the market and a cloudlet with APs is in position A, just as illustrated in Fig. 1(a). The cloudlets are small servers, such as the laptops, provided privately to enhance the public network with lower latency. In order to cover the most mobile devices, the cloudlet co-located with wireless APs are placed on the densest field at first. Nevertheless, the crowd moves frequently in a market, which leads to a small coverage of mobile devices if the cloudlet is still in position A. In such condition, the resources are wasteful and most users can not enjoy the benefit of this cloudlet. On the contrary, if the cloudlet can move to position B, as shown in Fig. 1(b), this problem goes away and the cost is saved as well. What's more, movable cloudlets are easy to achieve, such as applying a small server co-located with wireless AP into a movable robot.

Based on these observations, it is still a challenge to place the movable cloudlets efficiently to enhance cloud services for the mobile devices whose positions are constantly changing.

In view of this challenge, an adaptive cloudlet placement method for mobile applications over GPS big data is proposed in this paper. Specifically, the gathering regions of the mobile devices are identified based on K-means algorithm through GPS positioning and the cloudlet destination locations are confirmed accordingly. Besides, the traces between the origin and destination locations of these mobile cloudlets are also technically achieved through our cloudlet movement principle. 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 movement principle is presented in Section 2. Section 3 elaborates an adaptive cloudlet placement method for mobile applications over GPS big data in detail. In Section 4, experimental evaluations are conducted to demonstrate the validity of our method. Some related work is described in Section 5. Section 6 concludes the paper and gives an outlook on possible future work.

II. CLOUDLET MOVEMENT PRINCIPLE

In this section, formalized concepts are given to facilitate our further discussion on adaptive cloudlet placement for mobile applications over GPS big data. To simplify the discussion, key terms used in our cloudlet movement principle are summarized in Table I.

TABLE I. KEY TERMS AND DESCRIPTIONS

Terms	Description
A	The set of the points in device activity area, $A=\{(x, y)\mid 0\le x\le W, 0\le y\le H\}$
D	The set of mobile devices, $D=\{d_1, d_2,\ldots, d_N\}$
d_n	The n-th ($1\le n\le N$) mobile device in D.
$dp_n(t)$	The position of d_n at time t, $dp_n(t)= (dpx_n(t), dpy_n(t))$
L	The set of the cloudlet, $L=\{l_1, l_2,\ldots, l_M\}$
P	The set of the AP, $P=\{p_1, p_2,\ldots, p_M\}$
l_m	The m-th ($1\le m\le M$) cloudlet.
$lp_m(t)$	The central position of l_m at time t, $lp_m(t)= (lpx_m(t), lpy_m(t))$
r_m	The coverage radius for l_m.
$dc_m(t)$	The device collection of lp_m at time t.
ρ	The density threshold for cloudlet placement judgment.
$TN(t)$	The number of covered devices by all cloudlets.
$dc_{m,P}(t')$	The device collection of lp_m placed in position P(x, y) at time t'

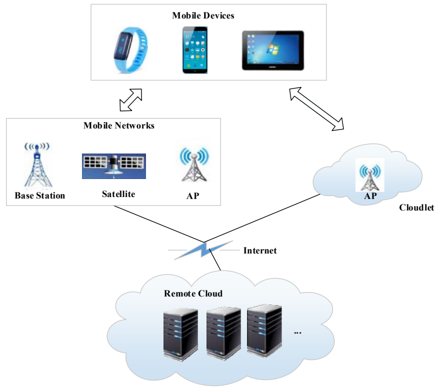


Fig.2 Mobile cloud infrastructure with cloudlet over GPS big data.

Currently, cloudlets play a key role in the mobile cloud infrastructure over GPS big data as shown in Fig. 2. In this infrastructure, the mobile device clients can access the mobile cloud service through wireless networks or directly access the cloudlets through APs. The remote cloud provides data storage and computing service, likewise, the cloudlets co-located with APs provides data storage and computing service. Compared to the traditional client-server communication model [10] without cloudlet, this infrastructure can contribute to reducing access latency. That is because those mobile devices can quickly access nearby cloudlets through APs to get direct cloud storage and computing resources.

Cloudlet placement with AP can help to enhance the cloud service for dynamic context-aware mobile applications. Generally, to make the mobile users use cloud services smoothly, the cloudlets are provided in a rectangle activity area in this paper. What's more, other shaped areas can be assembled by multiple rectangles of different sizes.

Definition 1 (Device Activity Area). Device activity area is defined by the x-y plane with definite ranges of x-axis and y-axis, denoted as $A = \{(x, y) \mid 0 \le x \le W, 0 \le y \le H\}$.

Within the device activity area, there are a number of mobile devices distributed randomly, denoted as $D= \{d_1, d_2,\ldots, d_N\}$, where N is the number of devices in A. Each mobile device has a position to show their locations.

Definition 2 (Mobile Device Position at Time *t*). The n-th ($1 \le n \le N$) mobile device d_n is located through GPS at some positions at time t that is a 2-tuple in A, denoted as $dp_n (t) = (dpx_n (t), dpy_n (t))$, where $dpx_n (t)$ and $dpy_n (t)$ are the x-axis value and the y-axis value of d_n at time t, respectively.

To serve the device user better, the cloudlets equipped with AP can help to enhance the service quality. Suppose there are M cloudlets used in A is denoted as $L= \{l_1, l_2,\ldots, l_M\}$. Each cloudlet is equipped with an AP, thus there are also M APs, denoted as $P= \{p_1, p_2,\ldots, p_M\}$. As in this paper we focus on the cloudlet placement, the central positions of the cloudlets are necessary to be defined to place the AP.

Definition 3 (Cloudlet Central Position at Time *t*). The m-th ($1 \le n \le M$) cloudlet l_m in L has a central position in which AP is placed, and its location is a 2-tuple in A, denoted as $lp_m(t) = (lpx_m(t), lpy_m(t))$, where $lpx_m(t)$ is the x-axis value and $lpy_m(t)$ is the y-axis value of l_m at time t, respectively.

The cloudlet can benefit users within AP coverage area with service enhancement. As the cloudlet is placed at a position with high-density mobile devices, so it is important to detect the device collection of all the cloudlet central positions. Suppose the coverage radius for l_m is denoted as r_m, the device collection of a cloudlet central position is employed to identify the covered devices.

For the m-th cloudlet l_m, the corresponding device collection at time t is denoted as $dc_m(t) = \{d_n(t) \mid dis(dp_n(t), lp_m(t)) \leq r_m, 1 \leq n \leq N\}$, where $dis(dp_n(t), lp_m(t))$ is calculated by

$$dis\left(dp_n(t), lp_m(t)\right) = \sqrt{\left(dpx_n(t) - lpx_m(t)\right)^2 + \left(dpy_n(t) - lpy_m(t)\right)^2} \text{ . (1)}$$

Definition 4 (Cloudlet Placement Principle). If cloudlet l_m with the central position $P(x, y)$ is working to serve the users with mobile devices at time t, the number of mobile devices covered by $l_m(t)$ at this time should satisfy the condition that $dc_m(t) \geq \rho$, where ρ is a density threshold for cloudlet placement judgment.

In the whole device activity area A, we aim to maximize the total number of covered devices by the provided cloudlets, which is calculated by

$$TN(t) = \left| \bigcup_{m=1}^{M} dc_m(t) \right| . \quad (2)$$

Assume cloudlet l_m is placed at P(x, y) at time t, and during the time range $(t, t']$, some mobile devices move in A randomly. To keep the cloudlets working efficiently, a cloudlet movement principle is proposed as movement reference.

Definition 5 (Cloudlet Movement Principle). After movement at time t', if the device collection of $P(x, y)$, denoted as $dc_{m,P}(t')$ and another position $P'(x', y')$ satisfy the placement principle of l_m, i.e., $dc_{m,P'}(t') \geq dc_{m,P}(t')$, and there are no other cloudlets placing around P' at time t' within radius r, in this situation l_m will move from P to P'.

III. Adaptive Cloudlet Placement Method for Mobile Applications over GPS Big Data

In this section, an adaptive cloudlet placement method for mobile applications over GPS big data is proposed for the mobile cloudlet placement. This method consists of the three steps specified in detail by Fig. 3.

A. Central position Identification

In this part, consider a set of mobile device positions $DP = \{dp_1, dp_2, \ldots, dp_N\}$, located through GPS, which is in an uneven distribution over the device activity area. Since the cost of mobile cloudlets is expensive, the number of cloudlets for mobile applications is limited and only parts of the activity area can be covered by cloudlets. To get the best usage of mobile cloudlets, it is appropriate to find the device gathering place for cloudlet placement. Therefore, the mobile device positions are clustered by using the well-known K-means algorithm to identify device gathering central positions.

Step1: Central position Identification. To achieve the best usage of cloudlets, the central position of device gathering place is identified based on K-means clustering algorithm. Then a path graph is employed to adjust these positions.

Step2: Cloudlet location confirmation. Undesired positions are filtered out by the cloudlet placement principle. Based on this, the specific location, chosen from the left positions, for deploying each cloudlet is confirmed according to the distances between them.

Step3: Adaptive cloudlet placement. In this step, the moving trace of mobile cloudlets is generated according to the area structure. A path graph is utilized to help the obstacle avoidance and the moving trace generation. Moreover, to achieve energy savings, the cloudlets having no destination locations will stay still.

Fig. 3 Specifications of adaptive cloudlet placement method for mobile applications over big data.

Moreover, considering the safety and efficiency of path searching in the Step3, all the cloudlets can only stay in several available positions. Such positions, denoted as $V = \{v_1, v_2, \ldots, v_U\}$, and the paths between adjacent position in V, denoted as $E = \{e_1, e_2, \ldots, e_U\}$, form a path graph in device activity area, which is defined as follow.

Definition 6 (Path Graph in Device Activity Area). A path graph is employed to adjust central positions of the clustering and help generating moving traces of the mobile cloudlets, denoted as $G = (E, V, W)$, where V represents the available positions for cloudlets, E represents the movable path between adjacent positions in V and W represents the weights of all positions.

Based on such path graph, the device central positions are adjusted to positions in V depending on the distance between them.

Algorithm 1 Center Position Identification(*DP*, *V*)

Input: A set of mobile device positions DP and a set of cloudlet available positions V.
Output: A set of device central positions.

```
Centers ← KMeansCluster (DP, N);
  // k - means(k = N) clustering algorithm and N is the number of cloudlets
CP ← ∅;
For i = 1 to N do
    MinDis ← dis(cp_i, v_1);
    Pos ← v_1;
    For j = 1 to U do
        If dis(dp_i, sp_i) < MinDis then
            MinDis ← dis(cp_i, v_i);
            Pos ← v_i;
        End if
    End for
    Add Pos to CCP;
End for
Return CCP;
```

Algorithm 1 specifies the identification process of center position and these positions are candidates for cloudlet central positions.

B. Weight path graph confirmation

To facilitate the placement of mobile cloudlets, several device central positions are obtained in Step1. Dense mobile devices surround such positions so that the cloudlets set on them can achieve high utilities. According to the mobile device distribution, some device central position may not be surrounded by considerable mobile devices. If cloudlets move to such positions, the utility of cloudlet resource is low. To get rid of such bad situation, a cloudlet placement principle defined in section II is presented.

After filtering the undesired device center positions, the destination position of each cloudlet is required. Given the origin locations of all cloudlets, a pairing function can be conducted for this selection according to the distance between cloudlet origin locations and device central positions in *CP*.

Algorithm 2 Center Position Filtering(*DP, CCP*)

Input: A set of mobile device positions *DP* and A set of device clustering central positions *CCP*.
Output: A set of cloudlet central positions.
For cp_i in *CCP* **do**
  $dc_i \leftarrow \emptyset$;
  For j = 1 to M **do**
    If $dis(cp_i, dp_i) < r$ **then**
      Add dp_i to dc_i
    End if
  End for
End for
$CP \leftarrow \{ cp_i \mid cp_i \in CCP$ and $\mid dc_i \mid > \rho\}$
//ρ is the density threshold for cloudlet placement judgment

Algorithm 3 Pairing Function(*CP, L*)

Input: A set of cloudlet central positions *CP* and a set of cloudlet *L*
Output: pairs of cloudlets and their central position
For i = 1 to Z **do**
  $pcp_i \leftarrow L$;
  Sort pcp_i in the increasing order of the distance between cp_i and cloudlets.
End for
While not all CPs have selected a cloudlet **do**
  For i = 1 to Z **do**
    If cp_i have not selected a cloudlet **then**
      Add cp_i to c_j that l_j is cp_i's nearest central position in pcp_i;
    End if
  End for
  For each c_i that is not empty **do**
    If l_i have not selected a CP or a nearer CP selected cl_i this time **then**
      Get l_i's nearest CP cp_j in c_j;
      cl_i and cp_j select each other;
      For each a that cp_a is in c_i and is not selected **do**
        Delete cl_i from pcp_a;
      End for
    End if
  End for
End while

Algorithm 2 specifies the process of filtering out undesired device central positions. For each central position, a traversal of all mobile device positions is conducted.

Algorithm 3 specifies the process of the selection between device central positions and cloudlet origin locations.

C. Adaptive cloudlet placement

In this section, we present a cloud service enhanced method for the mobile cloud infrastructure with cloudlets. We use the weight path graph provided by Step2 to determine the moving trace of each cloudlet.

In this purpose, a fact should be noticed that all these locations, including origin and destination, belong to available positions *V* of path graph *G*, which is defined in section 3.1. This transforms the moving trace generation problem to a shortest path problem in undirected graph *G*. Therefore, we employed the Dijkstra algorithm for moving trace generation and then the moving traces will be transmitted to corresponding mobile cloudlets.

Algorithm 4 Moving Trace Generation(*G*)

Input: A path graph *G* with no weights.
Output: The moving trace of all cloudlets .
1: **For** t = 1 to t_{max}
2: Get *CCP* by **Algorithm 1** $CPI(DP(t), V)$;
3: Filter *CP* by **Algorithm 2** $CPF(DP(t), CCP)$;
4: Get the pairs of CLs and their selected CPs by **Algorithm 3**;
5: **For** i = 1 to M **do**
6: **If** l_i have selected a CP **then**
7: Get the shortest trace tr_i from cl_i to its selected CP by the Dijkstra Algorithm $Dijkstra(G, l_j)$;
8: Transmit tr_i as the moving instruction to l_i;
9: **End if**
10: **End For**
11: **End for**

Algorithm 4 specifies the process of cloud service enhancement, where Algorithm 1 is used to get candidate device central positions (line 2) and Algorithm 3 is used to form the weight path graph.

IV. EXPERIMENT EVALUATION

In this section, we present the performance evaluation and comparison analysis of our proposed method and DBSCAN based method.

A. Experiment Settings

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

The shape of device activity area used for our experiment is set to square and the value domain of x-axis and y-axis of this area are both set to [0m, 180m]. Within this area, a set of mobile device location data is randomly generated. For better

TABLE II. THE EXPERIMENT CONTEXT

	Client	**HANA Cloud**
Hardware	Lenovo ThinkpadT430 machine with Intel i5-3210M 2.50GHz processor, 4GB RAM and 250GB Hard Disk.	Master/Slave: HP Z800 Workstation Intel(R) Multi-Core X5690 Xeon(R), 3.47GHz/12M Cache, 6cores, 2 CPUs, 128GB (8 × 8GB+4 × 16GB) DDR3 1066MHz 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

simulating real world application, the randomly generated location data defers to the *Gaussian Distribution*. The specific value domain of each parameter is shown in Table III.

B. Performance Evaluation

In this section, performance evaluations on distributions of the movable cloudlets and cloudlet coverage values are presented.

To display the distributions of mobile devices and cloudlets, three sequential record instants, i.e. t_0, t_1 and t_2 are selected in our experiments, as shown in Fig. 4. In Fig. 4, the moving mobile devices, notated by blue points, and stationary mobile devices, notated by red points, scatter in the device activity area. Meanwhile, the big green points notate the mobile cloudlets and the blue dashed circle represents the radio range of each cloudlet. The distributions at time t_0 are presented by Fig. 4(a). At time t_1, the positions of moving mobile devices are not changed much, as shown by Fig. 4(b), and the cloudlets stay still. At time t_2, considerable changes are taken by the distribution of mobile devices, two cloudlets move to new locations, as shown in Fig. 4(c).

Fig. 5 indicates the evaluation results by our method and DBSCAN with different cloudlet numbers. Please note that with the assumptions above, every simulation is conducted for 50 times initiated by different set of mobile device locations and the average coverage value is leveraged as final result.

From Fig. 5, we can find that our method has the higher coverage value compared with DBSCAN based method. Besides, in Fig. 5, we can also find that once the number of mobile device is fixed, the coverage value is affect by the

TABLE III. PARAMETER SETTINGS

Parameter item	**Domain**
The maximum x-axis value W	180 m
The maximum y-axis value H	180 m
The density threshold ρ for cloudlet placement judgment	30
The radio range r of all mobile cloudlets	30m
The number of APs M	{1,2,3,4,5,6,7}
The number of mobile devices N	{200,300,400}
The time period of experiments T	60min

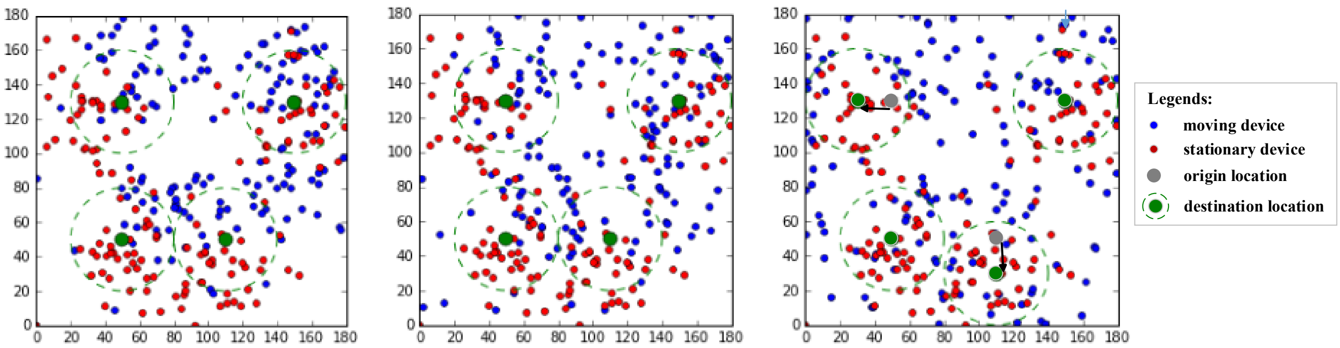
number of cloudlets. Furthermore, the coverage value of our method increase faster than the DBSCAN based method. Therefore, our method provides better performance than the DBSCAN based method.

V. RELATED WORK AND COMPARISON ANALYSIS

Mobile cloud computing can greatly improve the computing capacity of mobile devices by offloading the vast workload to clouds. And there have been many studies on mobile cloud computing [2], [11]. However, the clouds are geographically far away from mobile users, which lead to a long latency. In view of this challenge, Cloudlet is applied to get the shorter response time and reduce the energy consumption of mobile devices by alternating the offloading destinations. They have been fully investigated in [12], [13], [14], [15], [16], to name a few.

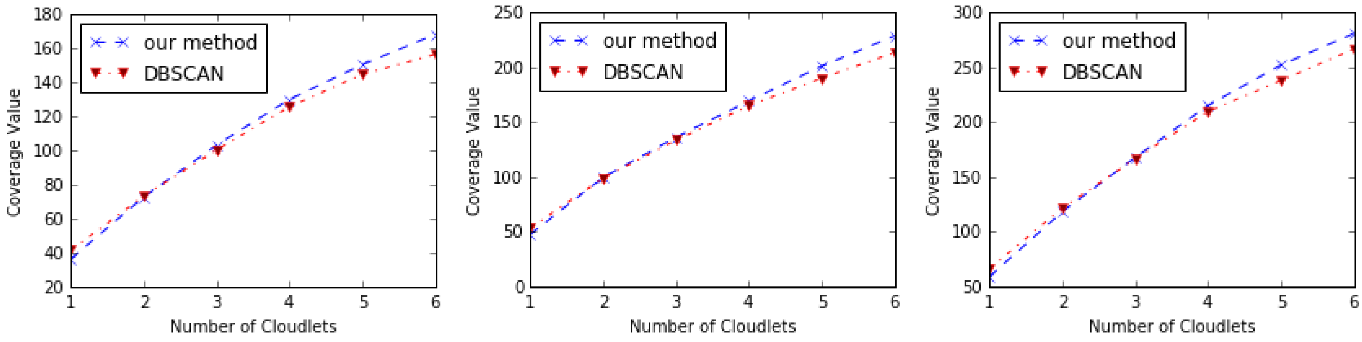
In [12], the author proposed a system Clone Cloud that combined static analysis and dynamic profiling to automatically transform mobile applications. Hoang *et al.* [13] formulated an optimal solution for dynamic resource sharing of mobile users in cloudlets as a semi-Markov decision process (SDP). Xia *et al.* [15] devised an efficient online algorithm to solve an online location-aware offloading problem in a combined mobile cloud computing environment. Gelenbe *et al.* [16] formulated the optimization problem between a local cloud and a remote cloud, which incorporates energy consumption and quality of service criteria. Despite the increasing momentum of cloudlet research, the limit of cloudlets' coverage has largely been overlooked.

Current researches mainly focus on capacitated cloudlets placement to cover most network nodes through efficient algorithms. Xu *et al* [5] studied the cloudlet placement problem in a large-scale Wireless Metropolitan Area Network (WMAN) consisting of many wireless Access Points (APs). Jia *et al* [9] devised an algorithm for cloudlets, placed at user dense regions



(a)distributions at t_0 (b) distributions at t_1 (c) distributions at t_2

Fig. 4 Distributions of mobile devices and cloudlets at three record instants, i.e. t_0, t_1 and t_2.



(a) Number of mobile devices = 200 (b) Number of mobile devices = 300 (c) Number of mobile devices = 400

Fig. 5 Comparison of coverage value with our method and DBSCAN.

in a wireless metropolitan area network, to balance their workload of the WMAN. Generally speaking, the dense regions of mobile devices may change greatly, such as in a market. We need to determine the new dense region to place the limited cloudlets adaptively rather than assigning mobile devices to the cloudlets. Thus the movable cloudlets are proposed in this paper, which can achieve higher resource utilization compared to the above researches.

VI. CONCLUSION AND FUTURE WORK

In this paper, an adaptive cloudlet placement method for mobile applications over GPS big data has been proposed. Concretely, the gathering regions of the mobile devices are identified based on K-means algorithm and the cloudlet destination locations are confirmed accordingly. Besides, the moving traces between the origin and destination locations of these mobile cloudlets are also technically achieved through our cloudlet movement principle. At last, the experimental results demonstrate the validity of the proposed method.

Based on the work done in this paper, we plan to integrate our adaptive cloudlet placement method for mobile applications over GPS big data into real-world cloudlet placement in the future. We will investigate the energy consumption of our method. And we also will look at the holistic response time of mobile applications after applying our method.

ACKNOWLEDGMENT

This research is partially supported by the National Science Foundation of China under Grant No. 91318301 and 61672276, the Key Research and Development Project of Jiangsu Province under Grant No. BE2015154 and BE2016120, the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

REFERENCES

[1] X. Chen. Decentralized computation offloading game for mobile cloud computing. *IEEE Transactions on Parallel Distributed Systems*, vol. 26, no. 4, pp. 974-983, 2015.

[2] H. Yin, X. Zhang, T. Zhan, Y. Zhang, G. Min, and D. Wu. NetClust: a framework for scalable and pare to-optimal media server placement. *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2114-2124, 2013.

[3] J. Cohen, Embedded speech recognition applications in mobile phones: Status, trends, and challenges, *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5352-5355, 2008.

[4] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, Cloud-vision: Real-time face recognition using a mobile cloudlet-cloud acceleration architecture in, *Proc. IEEE Symposium on Computers and Communications (ISCC)*, pp. 59-66, Jul. 2012.

[5] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo. Efficient Algorithms for Capacitated Cloudlet Placement. *IEEE Transactions on Parallel and Distributed Systems*, in press, 2015.

[6] Y. Li, and W. Wang. Can Mobile Cloudlets Support Mobile Applications?. *Proc. IEEE Conference on Computer Communications (INFOCOM)*, pp. 1060-1068, Apr. 2014.

[7] X. Wang, J. Wang, X. Wang, and X. Chen. Energy and delay tradeoff for application offloading in mobile cloud computing. *IEEE Systems Journal,* in press, 2015.

[8] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo. Capacitated cloudlet placements in Wireless Metropolitan Area Networks. *Proc. IEEE 40th Conference on.Local Computer Networks (LCN)*, pp. 570-578, Oct. 2015.

[9] M. Jia, J. Cao, and W. Liang. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, in press, 2015.

[10] H. T. Dinh, C. Lee, D. Niyato, and P. Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, vol. 13, no.18, pp. 1587-1611, 2011.

[11] M. Barbera, S. Kosta, A. Mei, and J. Stefa. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. *Proc. IEEE Conference on Computer Communications (INFOCOM)*, pp. 1285-1293, Apr. 2013.

[12] B.G Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: elastic execution between mobile device and cloud. *Proc. EuroSys '11 Proceedings of the sixth conference on Computer systems*, pp. 301-314, 2011.

[13] D. T. Hoang, D. Niyato, and P. Wang. Optimal admission control policy for mobile cloud computing hotspot with cloudlet. *Proc. IEEE Wireless Communications and Networking Conference*, pp. 3145-3149, Apr. 2012.

[14] X. Zhao, Y. Zhang, L. Wei, and M. Lu. Computational and communication resource allocation for mobile cooperative cloudlet computing systems. *proc. International Conference on Wireless Communications & Signal Processing (WCSP)*, pp. 1-5, 2015.

[15] Q. Xia, W. Liang, Z. Xu, and B. Zhou. Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. *Proc. IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pp. 109-116, 2014.

[16] E. Gelenbe, R. Lent, and M. Douratsos. Choosing a local or remote cloud. *Proc. IEEE 2nd Symposium on Network Cloud Computing and Applications*, pp. 25-30, Dec. 2012.