A Virtual Machine Placement Algorithm in Mobile Cloud Computing Environment by Considering Network Features

Chaitra Sathyampet
M.E. Scholar
Department of Computer Science & Engineering
APPA Institute Of Engineering & Technology, Gulbarga
csathyampet@gmail.com

***Abstract*—by taking into consideration network features of a wireless mesh network environment where mobile cloud computing (MCC) system is deployed, this paper proposes an effective virtual machine (VM) placement algorithm. This algorithm tends to place a newly created VM for a cloud service to a cloud server, which itself is also a mobile device, that occurs lowest network delay when accessing to application data stored in wireline networks. The aim is to reduce cloud service response time as much as possible in order to deliver better quality of experience to cloud end users over more noisy and less robust wireless networks. In addition, the proposed deployment of cloud storage and cloud computational resources onto wireless networks is also a contribution of the paper. The simulation results have shown the effectiveness and the efficiency of the proposed VM placement algorithms both in terms of cloud service response time.**

***Keywords*—mobile cloud computing (MCC); virtual machine placement; network-awareness; wireless networks; service response time**

I.INTRODUCTION

With advances on mobile devices and wireless communication technologies, the demand for mobile devices to run heavier applications (like which run on desktop PCs) is on increase. Since mobile devices, even for the modern smart phones, are constrained on size and weight, their resources for computation and communication are limited comparing with their desktop siblings [1]. Therefore, it makes more sense for heavy mobile applications to run remotely in more powerful machines. The latter is exactly the essence of cloud computing.

Cloud computing [2] represents a computing paradigm where computing resources are not physically present at user's location. These resources, usually collectively called clouds, are owned and managed by cloud service providers and can be accessed by end users remotely via the Internet. The past few years have witnessed a rapid shift of computing from the desktop to the cloud. With the rapid advancement of both wireless network technologies and mobile smart phones, there is an increasing need for cloud services to be provided to mobile users via mobile wireless networks. This new research field is called mobile cloud computing (MCC) [3-6].

The aim of this paper is to contribute towards this young but vibrant field of MCC by investigating a key problem for cloud computing, i.e., virtual machine (VM) placement. VMs are the key component of a cloud and they provide the elasticity boasted by cloud computing. When the admitted service gets more demanding on computational resources a VM can be automatically created by the cloud to offload some balance to keep the cloud users satisfied. Then there is the issue of VM placement.

There is much work done on VM placement [7], mainly focusing on fixed networks. Most of them devise a function of resource utilizations of individual resource types to decide where to place a VM. For example, [8] makes VM placement decisions by a weighted sum of resources while [9] proposes a more complex mathematical function of resources which is similar to multi-dimensional bin packing problem. In addition, some other approaches focus on saving power usage [10, 11].

One common feature of this work is that none of them has considered the effect of network on cloud service performance. A new created VM may be located on another physical machine and this physical machine may not be in the same location as the other machines serving the same cloud service. To guarantee cloud users' requirement on the cloud service, such as maximum response time, it is essential to place the VM to a network that occurs less delay. Therefore network features shall be taken into consideration when carrying out VM placement.

Piao *et al* [12] carries out some investigation in this aspect. It proposes a strategy that places a VM on the physical machine with the smallest data transfer time to required data with consideration of network speed. However, the concerned network is wired network for data centers. Namely the users are high-end enterprise users rather than mobile users with their mobile devices of limited computational and networking resources. In our paper the network cloud servers run on is wireless network. In particular it is a kind of mesh network where both wireless base stations or access points (APs) exist supporting both point to multi-point communications and ad hoc point to point communication.

In this paper we consider a cloud computing environment where both storage resource and computation resource exist.

Namely there are both storage cloud and computation cloud and they are physically separate and interconnected via wireless networks. In particular storage cloud is deployed next to the wireless APs and is connected to the APs using wireline network. In

contrast, computation clouds are deployed on end user's mobile devices. With increasingly more memory and more powerful processors installed in mobile devices modern smart phones or pads are able enough to host tasks from other mobile devices by setting up virtual machines (VMs). This is to say that this paper considers a mobile cloud environment where VMs are deployed on mobile devices that are interconnected via wireless networks. Section II will present more about the concerned MCC networks.

Here in this paper a cloud application or a cloud service is a mobile application or a mobile service that runs part or all its processing in the cloud remotely. In particular we concern cloud services that also engage access to a large amount of data. When a new cloud application or service is initiated on a mobile device, a new VM needs to be created to serve this application. On which cloud server to place this new VM is a key issue and thus the focus of this paper. The new cloud service running on a cloud server (on a mobile device) needs to get access to the data stored in storage cloud. Due to instability of wireless links and the distance and thus channel quality and data rate between the cloud server (i.e., the mobile device where the VM is to run) and the AP the data is stored next to, it is essential to consider network features of this link when choosing a cloud server to place this VM. The most important feature to be considered is the link bandwidth as it is also the major impacting factor for delay.

In summary, the aim of this paper is to propose a VM placement algorithm to minimize the service response time which is a direct factor of the quality of experience of cloud end users. The major contributions lie in the following two aspects. Firstly, the proposed deployment of cloud storage and cloud computational resources onto wireless networks is relatively new, which considers both infrastructure-based and ad hoc network architecture and is sensible to the nature of cloud data storage and cloud computation resource respectively. Secondly, the proposed VM placement algorithm effectively works on the above new MCC network environment and is efficient in terms of cloud service response time.

The remainder of this paper is organized as follows. Section II further clarifies the MCC environment, based on which a detailed problem statement in a more technically formal format is presented. Section III describes the proposed VM placement algorithm where admission control is also presented. Performance evaluation and analysis are reported in Section IV before the paper concludes at Section V.

II.PROPOSED MCC ARCHITECTURE AND PROBLEM STATEMENT

A. Proposed MCC System Architecture

This paper bases itself on a cloud system where cloud resources are deployed within or next to mobile networks. This is different from the conventional assumption where cloud resources reside within the core network. In the latter case the way cloud services are provided to mobile devices is indifferent from the way they are offered to office devices such as PCs. Our proposal endeavours to push cloud resources closer to end mobile users aiming to provide a truly mobile cloud system that is genuinely tailored to mobile end users.

The MCC system architecture proposed in this paper is depicted in Fig. 1. There are two types of clouds: storage cloud and compute cloud. Storage cloud is composed of multiple storage nodes (SN), which provide massive data storage. Compute cloud is composed of multiple physical computational nodes (CN) which host VMs to carry out user applications.

As far as network is concerned, there are also two types: wired networks and wireless networks, which are assumed in this paper to be implemented by Ethernet and WiFi (IEEE 802.11g/n) respectively. The deployment of the two types of clouds on the above networks is as follows. Data cloud is deployed next to the WiFi access points (APs) and the connection between storage nodes and APs is wireline links such as Ethernet. In contrast, compute nodes are deployed on mobile devices within the wireless networks. This becomes increasingly feasible thanks to the performance boosting of modern smart phones. On the other hand, this is also necessary for situations where there is no connection to fixed networks. The self-organized infrastructure-less mobile networks can be the only choice to provide better-than-nothing services, e.g., in the case of disaster events. In this case the data should also be stored on mobile devices. Actually there are already some research outcomes on mobile-device-based cloud computing and services [1].

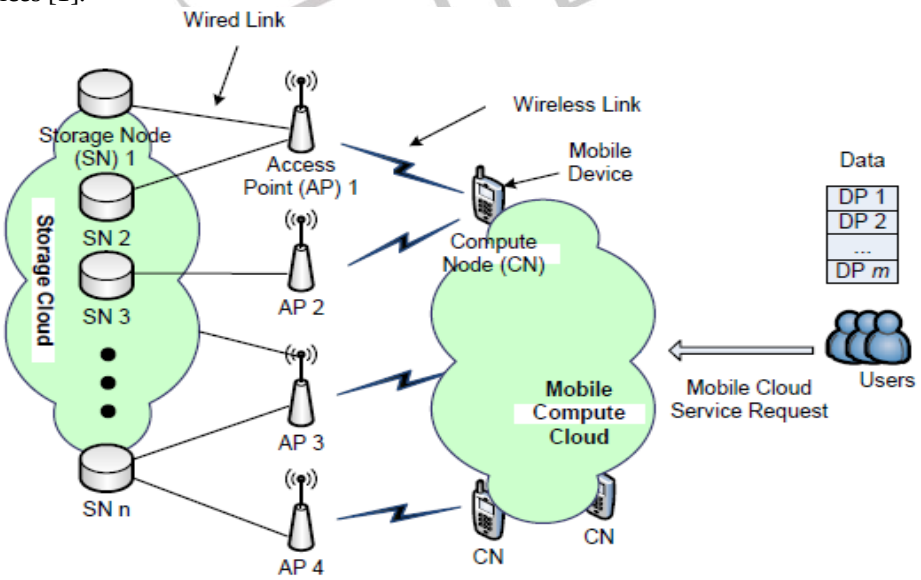

Fig. 1 Proposed MCC System and Network Architecture.

User data is stored on more than one storage node. The problem is: how to find a best CN to host a mobile cloud service with access to the corresponding application data on storage cloud while minimizing the response time of the mobile cloud service. Since a service is provided by a VM in cloud computing, the problem now is how to place a VM to an appropriate CN, namely the VM placement problem. But in this paper mobile network features are considered when designing such a VM placement algorithm, as to be detailed in the next sub-section.

B. Problem Statement

The network parameters to be considered include: delay, jitter, throughput, packet loss ratio, etc. To start with, this paper stays focused on delay and investigates the effect of delay on the performance of MCC system in terms of cloud service response time. Assuming that the network bandwidth between an SN and a CN (i.e., the mobile device that provides service) is known, the corresponding cloud service response time of an application can be obtained in the following manner.

Nowadays, most data are stored distributed on the network. The same assumption is taken in this paper. We assume the data for a mobile application is broken into *m* pieces which are called data pieces (DP) and these DPs are stored on a subset of *n* storage nodes. The data distribution can be represented as a matrix $D_{DP,SN}$.

$$D_{DP,SN} = \begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{pmatrix}$$

Here, *dij* represents the amount of data piece *i* stored on storage node *j*.

The proposed approach uses another matrix $B_{CN,SN}$ to denote the network bandwidth between these *k* computational nodes in compute cloud and the *n* storage nodes. So the value of the element *bij* in the matrix represents bandwidth between mobile device *i* and storage node *j*.

$$B_{CN,SN} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{k1} & b_{k2} & \cdots & b_{kn} \end{pmatrix} \qquad (1)$$

With data sizes across SNs and bandwidth between CNs and SNs in $D_{DP,SN}$ and $B_{CN,SN}$ respectively, the service response time matrix $T_{DP,CN,}$ which represents the response time from each mobile device to the related data, can be acquired by the following formula:

$$T_{DP,CN} = D_{DP,SN} \Big/ B_{CN,SN} = D_{DP,SN} \times A_{SN,CN} \qquad (2)$$

Here, $A_{SN,CN}$ is the inverse of matrix $B_{CN,SN}$. As a result, the potential response time of a mobile cloud service running on the mobile device *CN* can be calculated as:

$$t_{CN} = \sum_{DP=1}^{m} t_{DP,CN} \qquad (3)$$

Then essential idea of VM placement is to select a CN with minimum service response time to host the application, according to (3).

III.PROPOSED VM PLACEMENT ALGORITHM

A. Admission Control

Admission control is performed to make sure there is enough physical resource to accommodate the newly arrived mobile cloud services [13]. VM placement takes place after a service is admitted to the system. In this paper, an admission control mechanism is introduced to facilitate the management of resources allocation for VMs.

Upon receipt of a cloud service hosting request, this admission control mechanism checks whether there exists a list of CNs whose idle computation resources satisfy the requirement of this new service before placing a VM to one of the CN in the list. In this paper, we use the resource set

$$R = \{P, M, B\}$$

To denote the total computation capacity of a CN, including processor, memory and bandwidth respectively of a mobile device. *P* indicates the CPU resource, *M* the memory resources and *B* is used to represent the bandwidth.

$$R_o = \{P_o, M_o, B_o\}$$

Similarly, the set *Ro* above denotes computing resources that have been occupied by the currently running VMs on the mobile device. So, obviously, the available resource set is:

$$R_a = \{P_a, M_a, B_a\} = \{P - P_o, M - M_o, B - B_o\}$$

Suppose the requested resources by the new mobile cloud service are:

$$R_{req} = \{P_{req}, M_{req}, B_{req}\}$$

Then a VM to serve this cloud service could be placed on this mobile device only if the following condition is satisfied:

$$R_a > R_{req} \qquad (4)$$

This means all these three prerequisites $P_a > P_{req}$, $M_a > M_{req}$ and $B_a > B_{req}$ must be satisfied simultaneously.

Afterwards, if the VM is allocated successfully, the occupied and available computation resources of the mobile host should be updated via

$$R_o = \{P_o + P_{req}, M_o + M_{req}, B_o + B_{req}\} \qquad (5)$$

$$R_a = \{P_a - P_{req}, M_a - M_{req}, B_o - B_{req}\} \qquad (6)$$

B. VM Placement Algorithm

Based on the above discussion, a network-aware VM placement algorithm that is specifically for the MCC infrastructure depicted in Figure 1 has been proposed, which pseudo code is in Algorithm 1. Essentially, an upcoming VM is placed on a mobile host which has a minimum service response time.

The proposed algorithm needs data distribution matrix *DDP,SN*, network bandwidth matrix *BCN,SN*, and the requested data pieces of an arriving application *Reqset* as input parameters. The main steps of the algorithm are as follows:

At first, we calculate the service response time matrix *TDP,CN* based on *DDP,SN* and *BCN,SN* in Line 1. Then the column index *i* of *TDP,CN* is initialized to 0. *hostId* that represents the ID number of the selected mobile host is initialized to -1 and *minTime* that denotes the minimum service response time for the application is initialized to a large number. Lines 2-4 complete this initialization.

Line 5 starts the main loop of the algorithm that traverses matrix *TDP,CN* to find the mobile host with minimum service response time for the cloud service. Above all, check whether available resources on a mobile host meet the computation capacity requirement of the VM (Line 6). If this condition is satisfied then the algorithm computes the time needed to access the requested data pieces from this mobile host and compares it with *minTime* (Line 7). Line 8 and 9 update *minTime* and *hostId* if the computed value is smaller. The above procedure is repeated until all mobile devices are tested.

Following the above process, the algorithm allocates the VM on the selected mobile host with the ID number *hostId* to serve the cloud service (Line 14). Then the algorithm updates the available resources of this mobile host on the basis of the amount of resources that the VM requires, such as CPU resources, if the VM has been created successfully (Line 15 - 17).

Algorithm 1: Network-aware VM Placement Algorithm

Input: $D_{DP,SN}$; $B_{CN,SN}$;
Req_{set} -- requested data set of the cloud service
Output: chosen mobile device *hostId* to host the VM

1: $T_{DP,CN} \leftarrow D_{DP,SN} / B_{CN,SN}$
2: $i \leftarrow 0$
3: $hostId \leftarrow -1$
4: $minTime \leftarrow$ MAXVALUE
5: **while** $i < CN$ **do**
6: **if** $R_a > R_{req}$ **then** //admission control
7: **if** $\sum_{r \in Req_{set}} T_{r,i} < minTime$ **then**
8: update the value of *minTime*
9: $hostId \leftarrow i$
10: **end if**
11: **end if**
12: $i \leftarrow i + 1$
13: **end while**
14: Place VM to mobile host *hostId*
15: **if** VM placement is successful **then**
16: calculate Eq. (5) and Eq. (6);
// update resource status on mobile host *hostId*
17: **end if**

IV.PERFORMANCE EVALUATION AND DISCUSSION

A. Simulation Setup

In this Section, we conduct several simulations to study the performance of the proposed VM placement algorithm. We implement the algorithm using CloudSim 2.0. CloudSim is an extensible simulation toolkit that enables modeling and simulation of cloud computing environment and supports modeling and creation of VMs on a simulated node of a data center, etc [14], [15]. The effectiveness and the efficiency of our proposed approach are evaluated mainly on the average service response time, via comparing with that of the default VM placement algorithm of CloudSim which is known as *VmAllocationPolicySimple* [12]. Its strategy is to place a VM on a host with the least processor units in use.

The simulation environment is configured as follows. The storage cloud is composed of two storage nodes: *SN0* and *SN1*. Data pieces consist of three files: *DP1* (200MB), *DP2* (100MB) and *DP3* (500MB). Moreover, *DP1* and *DP2* are stored on *SN0* while *DP3* is stored on *SN1*. Furthermore, a data center named *datacenter0* is created as the compute cloud which has three mobile hosts as the compute nodes. These three mobile hosts are denoted as *CN1*, *CN2* and *CN3* respectively. The network bandwidth between these mobile hosts and storage nodes are denoted as a matrix $B_{CN,SN}$ which has been described in Section II

$$B_{3,2} = \begin{pmatrix} 3.2 & 16 \\ 6 & 7.2 \\ 2.4 & 4 \end{pmatrix}$$

Here, the values of the matrix are set according to the network features of wireless network. In addition, three VMs will be created on these mobile hosts according to the VM placement algorithm. Mobile cloud services are submitted to these VMs in the format of *cloudlets* in the CloudSim terms.

B. Cloud Service Response Time

We compare the proposed VM placement algorithm with VmAllocationPolicySimple in three simulation groups. In each group, the number of tasks submitted to the VMs is increased gradually from 5 to 40.

Firstly, we present the results of the first group of simulation where all tasks request *DP1* in Fig. 2. The average task response time increases as the number of tasks increases for both algorithms. As our proposed algorithm always selects the mobile host that has the best network connectivity as its first choice, its service response time increases gradually as better mobile hosts are used up. But in any case, our proposed network-aware VM placement outperforms the benchmark one.

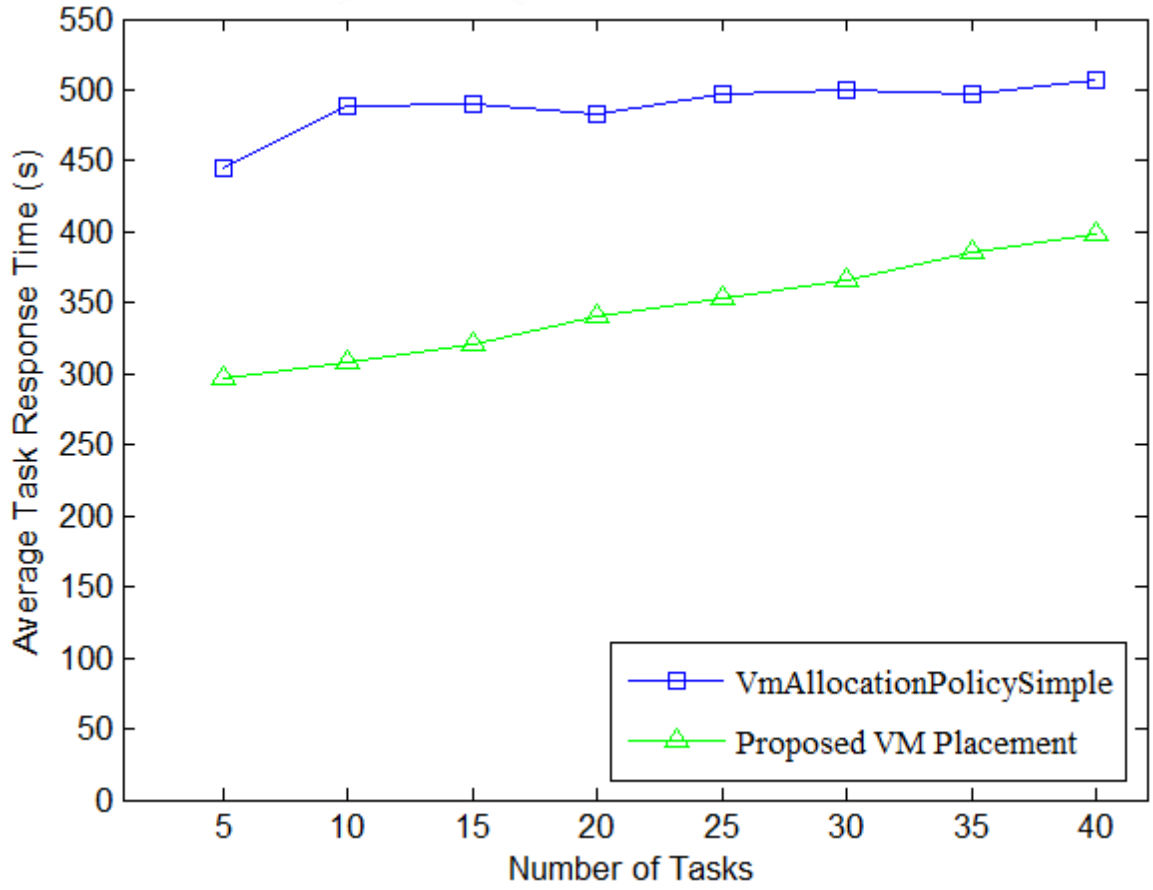

Fig. 2. Service Response Time for Tasks Requesting *DP*1.

In the second group of simulation, all tasks have requested DP2. As shown in Fig. 3, the service response time of the proposed algorithm is shorter as well, although the difference between these two algorithms is not as significant as the first group.

As to the third simulation group, which results are shown in Fig. 4, all tasks have requested DP3. Obviously, the performance gain of the proposed algorithm is more significant than these of the first and second simulations. This is largely because of the size of the files employed. Since the size of DP3 is the largest, the time saved via better network condition of our approach plays a greater role in improving the performance.

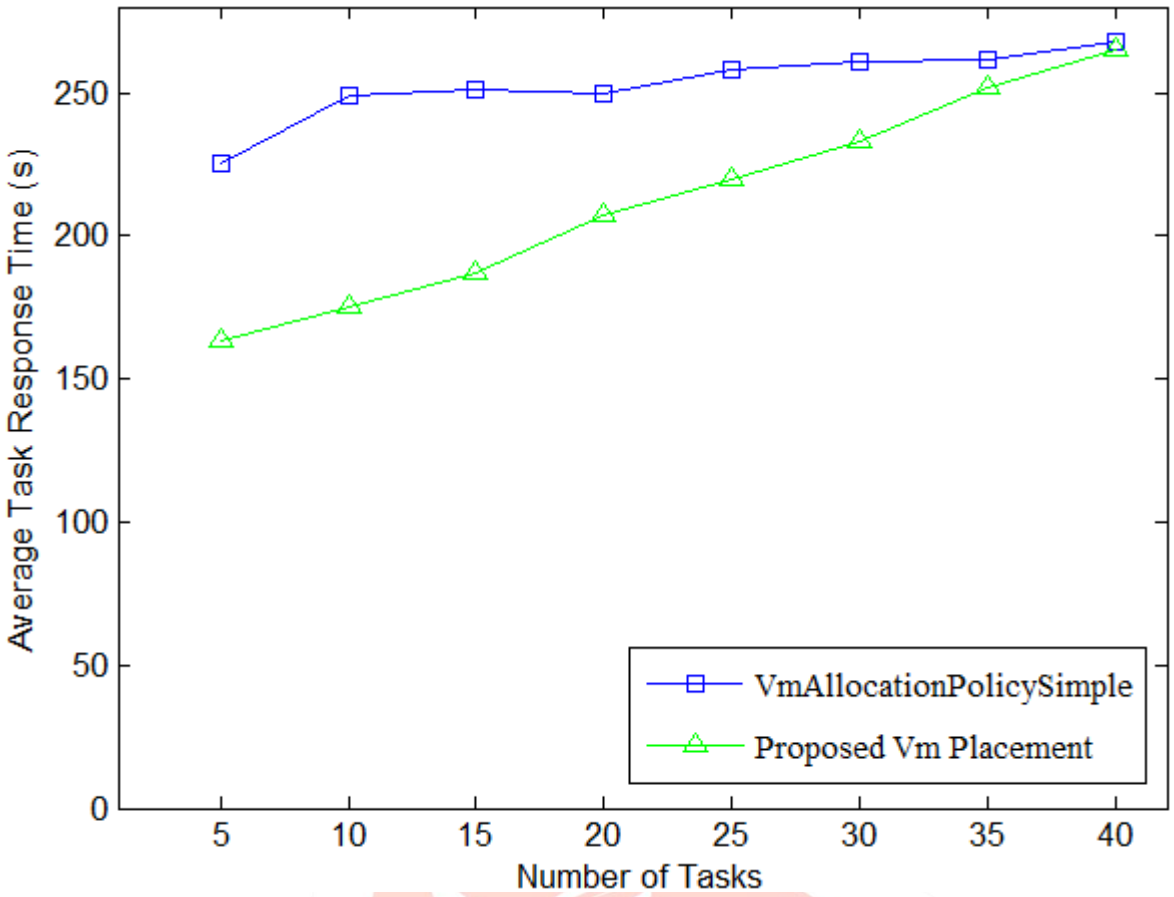

Fig. 3. Service Response Time for Tasks Requesting *DP*2

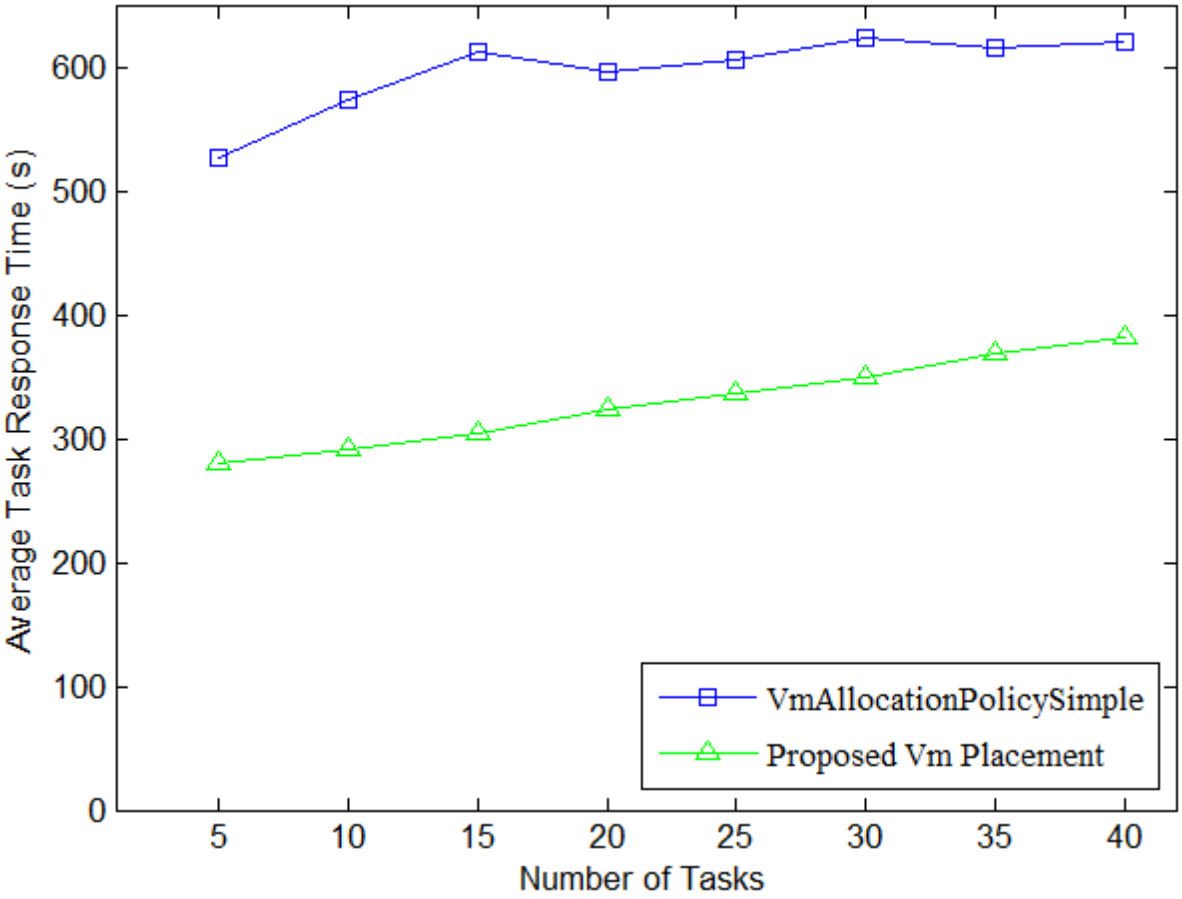

Fig. 4. Service Response Time for Tasks Requesting *DP*3

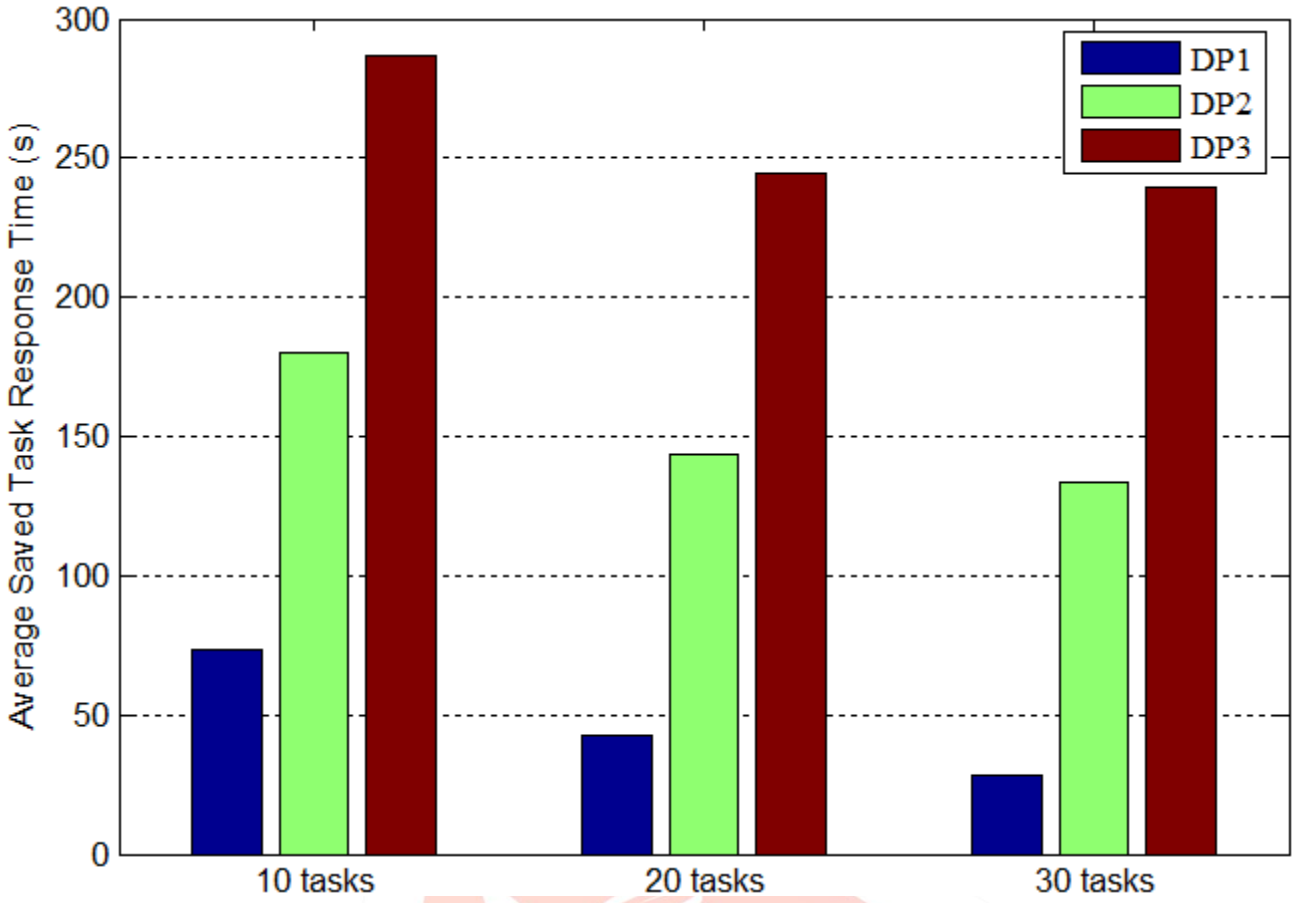

Fig. 5. Effect of Requested Data Size on Service Performance.

To further elaborate the effect of the requested data size on service response time, another set of simulation is designed. In this simulation, data pieces are made up of three files: DP1 (100MB), DP2 (200MB), DP3 (300MB). Another difference from the previous simulation is that all these files are stored on SN0 to ensure the consistency of network conditions. Then we submit 10, 20 and 30 tasks to the VMs respectively and measure the saved task response time of the proposed VM placement algorithm over VmAllocationPolicySimple. Through the simulation results as shown in fig. 5, it can be observed that the proposed algorithm benefits more in terms of task response time as the size of requested data pieces increases.

Overall, the simulation results described above have demonstrated that the proposed algorithm is more efficient for VM placement and is able to achieve better user experience in terms of service response time.

V.CONCLUSION AND FUTURE WORK

This paper has proposed an efficient VM placement algorithm for a mobile cloud computing environment that includes both cloud storage and cloud computation resources. This algorithm takes into account the network features, in particular network bandwidth and network delay, when choosing a cloud compute server to host a newly created VM. The simulation results have shown the effectiveness and the efficiency of the proposed VM placement algorithms in terms of cloud service response time.

The immediate future work is to consider VM migration after the VM has been placed in order to satisfy cloud service's quality requirement when network status dynamically changes, e.g., degradation of a wireless link due to interference or node mobility, etc. Following on this, we will look into energy efficiency aspect of the MCC system and algorithms.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing", Communications of the ACM, vol. 53, no. 4, 2010, pp. 50 – 58.

[2] L. Guan, X. Ke, M. Song, and J. Song, "A survey of research on mobile cloud computing", 2011 IEEE/ACIS 10th International Conference on Computer and Information Science, May 2011, pp. 387 – 392.

[3] W. Song and X. Su, "Review of mobile cloud computing", 2011 IEEE 3rd International Conference on Communication Software and Networks, May 2011, pp. 1 – 4.

[4] 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, 2011, DOI: 10.1002/wcm.1203.

[5] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce", Carnegie Mellon University, Sep. 2009.

[6] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing", 2012 IEEE Fifth International Conference on Cloud Computing, June 2012, Hawaii, USA. pp. 794 – 802.

[7] T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy. Predicting Application Resource Requirements in Virtual Environments. 2008.

[8] M. Mishra and A. Sahoo, "On theory of VM placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach", Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, July 2011, pp. 275 – 282.

[9] A. Kochut and K. Beaty. On strategies for dynamic resource management in virtualized server environments. In *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS '07. 15th International Symposium on*, pages 193–200, Oct. 2007.

[10] H. W. Choi, H. Kwak, A. Sohn, and K. Chung. Autonomous learning for efficient resource utilization of dynamic vm migration. In *ICS '08: Proceedings ofthe 22nd annual international conference on Supercomputing*, pages 185–194, New York, NY, USA, 2008. ACM.

[11] William Voorsluys" Cost of Virtual Machine Live Migration in Clouds"in 2008

[12] E. Arzuaga and D. R. Kaeli, "Quantifying load imbalance on virtualized enterprise servers". Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering, 2010, pp. 235 – 242.

[13] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration", Proceedings of 4th USENIX Symposium on Networked Systems Design & Implementation, 2007, pp. 229 – 242.

[14] M. Nelson, B. Lim, and G. Hutchins. Fast Transparent Migration for Virtual Machines. In Proc. USENIX 2005.

[15] W. Shi and B. Hong, "Towards profitable virtual machine placement in the data center", Proceedings of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing, 2011, pp. 138 – 145.

[16] B. Zhang, Z. Qian, W. Huang, X. Li, and S. Lu, "Minimizing communication traffic in data centers with power-aware VM placement", Proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2012, pp. 280 – 285.

[17] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing", 2010 9th International Conference on Grid and Cooperative Computing (GCC), Nov. 2010, pp. 87 – 92.

[18] D. Niyato, P. Wang, E. Hossain, W. Saad, and Z. Han, "Game theoretic modeling of cooperation among service providers in mobile cloud computing environments", Proc. of IEEE Wireless Communications and Networking Conference (WCNC) 2012. April 2012, Singapore. pp. 3128 – 3133.