A 'Follow-Me' computing scheme based on virtual machine movement for QoS improvement in mobile cloud computing environments

Xu Gaochao[1], Ding Yan[1], Ou Shumao[2], Hu Liang[1], Zhao Jia[3*]

[1] *College of Computer Science and Technology, Jilin University, Qianjin Str. 2699, 130012 Changchun, China*

[2] *Department of Computing and Communication Technologies, Oxford Brookes University, Wheatley, Oxford OX33 1HX, United Kingdom*

[3] *College of Computer Science and Engineering, ChangChun University of Technology, Yan'an Str. 2055, 130012 Changchun, China*

Received 1 March 2014, www.tsi.lv

Abstract

Mobile cloud computing utilizes virtualized cloud computing technologies in the mobile Internet. To improve Quality of Service (QoS) and execution efficiency of mobile cloud applications, we propose a novel computing scheme called "Follow-Me" (FM), which is based on live wide-area virtual machine (VM) migration. In a virtualized mobile cloud environment based on the VMs of cloud side and mobile devices of user side, the purpose of the proposed FM scheme is to migrate the corresponding VM in real-time when a mobile device moves from one service area to another. FM obtains the current positions of mobile devices, estimates the next servicing areas, and finally migrates the VMs along with the mobile users' movement. The proposed FM scheme has been tested in an experimental environment by using the CloudSim platform. The experimental results demonstrate that FM evidently improves the QoS of mobile cloud computing compared with the existing approaches. FM achieves a better average service response time, a clearly smaller error rate and consumes less energy.

Keywords: Mobile Cloud Computing, Mobile Device, Virtual Machine, Area Localization, Live Wide-Area Migration

1 Introduction

Cloud computing [1, 2] is the development of parallel computing, distributed computing and grid computing. It distributes the computation tasks into a resource pool composed by great amounts of computing resources, and makes users on-demand obtainable computing power, storage space and information services. With the vigorous development of the mobile Internet, the cloud computing services based on mobile devices such as mobile phones, tablet PCs, etc. have emerged and been widely used for the fields of information sharing, mobile learning, e-commerce, home monitoring and mobile health etc. Mobile cloud computing is not only a kind of IT resources but also a delivery and utilization mode of information and services. It obtains the required infrastructures, platforms and software (applications) etc. in an on-demand and scalable manner through the underlying mobile Internet. Mobile cloud computing applies cloud computing technology into mobile Internet. Mobile devices are limited to its battery capacity and computing power, which has been always hampering the development of mobile computing. However, in mobile cloud computing system, mobile applications can be partitioned into several computation modules or components. The complicated computations of a mobile application will be performed on a VM, which is located in a cloud data center. It is envisioned that mobile cloud computing will play a crucial role in people's daily life in the near future. Figure 1 shows the differences between single-machine computing and mobile cloud computing where an application is partitioned and some parts are executed in a VM at a remote data center. The distributed execution at mobile devices and mobile clouds can be done automatically or semi-automatically.

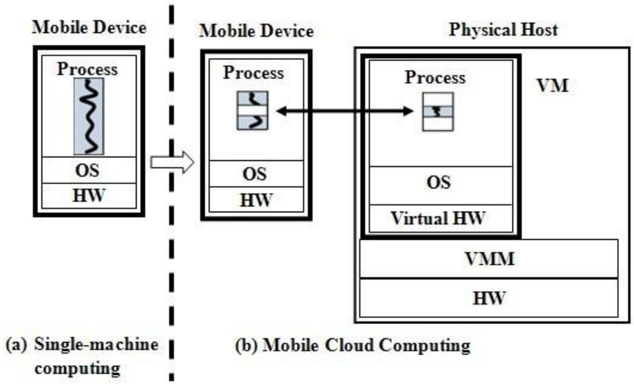

FIGURE 1 The system model of transforming a single-machine execution (mobile device computing) into a distributed execution (mobile device and cloud computing) (semi)-automatically

* ***Corresponding author*** e-mail: zhaiyj049@sina.com

The performance of mobile applications running on mobile cloud computing environments is mainly affected by three aspects: the computation in mobile device side, the computation in the corresponding VM side and the communication (mainly the wireless part) between the two sides. Since a cloud data center has large amount of resources, generally the computation cost of a mobile application in the corresponding VM is very little in comparison with it on the mobile device and normally negligible. The computation performance of mobile device side can also be improved by well-designed partition approaches, such as [3-5]. Due to the unreliability nature of the wireless communication, the mobile wireless network becomes the bottleneck of the performance of mobile cloud computing. To relieve this bottleneck, the transmission distance between a mobile device and its corresponding VM should be kept as short as possible. As shown in Figure 2, Mobile Device A in Area A is served by Cloud Data Center A, i.e. the corresponding VM of Mobile Device A in Cloud Data Center A is communicating with Mobile Device A. If Mobile Device A moves into Area B, which is far away from Area A, the wireless distance, for which it communicates with its VM located in Cloud Data Center A will become longer or even unreachable. This will seriously affect the QoS of mobile cloud applications and mobile users will take no advantage from mobile cloud computing. Assume that there are Cloud Data Center B and Access Point B in the nearby areas, and which are from the same provider. If the VM of mobile device A in Cloud Data Center A moves into Area B, it will be able to communicate with its VM locally by Access Point B and thus the reliability of the communication and to the QoS of the mobile cloud computing services can be much improved. To address this problem and achieve this goal, we propose a novel computing scheme called “Follow Me” (FM). It has abilities in efficiently moving the VM into the nearest cloud data center along with the mobile device’s movement. It can minimize the communication cost and greatly improve the QoS and finally provide better and more efficient mobile cloud computing services.

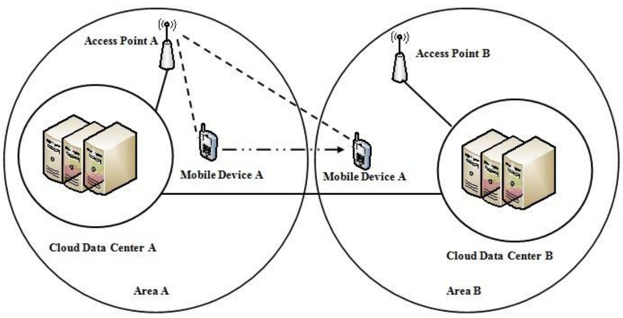

FIGURE 2 Traditional model of mobile cloud computing services

The rest of the paper is organized as follows. In Section 2, we present the related work. The reasonable prerequisites, the algorithm and implementation of the FM scheme are discussed in detail in Section 3. Section 4 reports the experimental results and analysis on CloudSim platform. Section 5 concludes the paper and lists our future work.

2 Related work

To the authors’ best knowledge, very little research work on the problem of moving the VM of a mobile device from a source cloud data center to a approaching target cloud data center has been done. . Most related research is focused on some similar problems such as improving the performance and QoS of mobile cloud computing.

The applications in mobile cloud computing systems run diverse workloads under diverse device platforms, networks and clouds. Traditionally these applications are statically partitioned between less-powerful devices and powerful clouds, thus their execution may be significantly inefficient in heterogeneous environments and with different workloads. Byung-Gon and Petros in [4] proposed an approach of dynamic partitioning of applications between weak devices and clouds and argued that dynamic partitioning is the key to addressing heterogeneity problems. The authors found that most of the mobile applications running on mobile devices can be restructured so that they can be statically partitioned between the weak device and a server running in the cloud.

Jung *et al.* in [6] proposed to exploit the potential of smart phones in proximity cooperatively, using their resources to reduce the demand on the cellular infrastructure, through a decision framework called RACE (Resource Aware Collaborative Execution). RACE enables the use of other mobile devices in the proximity as mobile data relays. RACE is a Markov Decision Process (MDP) optimization framework that takes user profiles and user preferences to determine the degree of collaboration. Both centralized and decentralized policies are developed and validated through simulation using real mobile usage traces.

In [7], Klein *et al.* proposed an architecture to provide an intelligent network access strategy for mobile users to meet the application requirements. The authors proposed a so called Context Management Architecture (CMA) which is responsible for acquiring, processing, managing, and delivering context information. Finally, the authors presented a context-aware radio network simulator (CORAS) that was able to model context availability, accuracy, and delay, thus enabling an evaluation of the impact of different levels of context relevance, confidence, and quality on simulation results.

Zhang *et al.* in [8] designed and constructed a multi-hop networking system named MoNet based on WiFi, and on top of which they designed and implemented WiFace, a privacy-aware geosocial networking service. For the situation without any infrastructure, they designed a distributed content sharing protocol, which can significantly shorten the relay path, reduce conflicts and improve data persistence and availability. A role strategy was designed to encourage users to collaborate in the network. Furthermore, a key management and an authorization mechanism were developed to prevent some attacks and protect privacy.

Kovachev *et al*. in [9] proposed a Mobile Community Cloud Platform (MCCP) as a cloud computing system that can leverage the full potential of mobile community growth. An analysis of the core requirements of common mobile communities is provided before they present the design of their cloud computing architecture that supports building and evolving of mobile communities.

Liang *et al*. in [10] proposed a Security Service Admission Model (SSAM) based on Semi-Markov Decision Process to model the system reward for the cloud provider. They first define system states by a tuple represented by the numbers of cloud users and their associated security service categories, and current event type (i.e., arrival or departure). They also derived the system steady-state probability and service request blocking probability by using the proposed SSAM. The approach provided strong security protection while achieving resource management for the maximum revenue.

3 The proposed FM scheme

3.1 PREREQUISITES

In this paper, we assume that all the target areas, which a mobile device may leave for have mobile cloud data centres owned by the same provider. The storages of all cloud data centres of a provider are distributed and shared [11]. This assumption can be easily relieved for the service areas, which belong to different providers, if certain level of trust can be built between the providers.

In the proposed FM approach, the disk-image of a migrated VM will also be migrated for the reason of efficient access and better QoS performance. In another word, the migration of a VM consists of the memory of VM, its running-time status [12] (including CPU, registers, I/O states), VM disk-image and the related information for network recovery.

We further assume that the network connections between the mobile cloud data centres are wired, high-speed and reliable (e.g. by optical fibers). In addition, we assume the mobile applications can tolerant short period of inaccessibility during VM migration. Like other research work, e.g. [11, 12], these assumptions are believed not to affect to the performance and efficiency of the proposed FM scheme.

3.2 THE FM SCHEME

The proposed FM Scheme is designed to improve the performance and efficiency of mobile cloud computing. Actually, FM is a placement selection policy of mobile devices' VMs and it is based on area localization and live wide-area VM migration, which involves the migration of memory and status, the memory of VM disk-image and the redirection of wide-area network. More Specifically, FM firstly performs a decision of area location of the mobile device according to some existing method [13]; secondly, it calculates the two distances from a mobile device to the cloud data centre, which its VM is located in and to the cloud data centre, which mobile device is approaching to; thirdly, FM compares the two distances. If the former is larger, FM does not do anything. Once the latter is larger, FM will perform a live wide-area VM migration of the corresponding VM located in source cloud data centre. To achieve the efficient live wide-area VM migration proposed in FM, we have employed our previously proposed live VM migration mechanism HMDC used for live local-area VM runtime states migration [14] as well as we have combined the composed image cloning (CIC) methodology and wide-area network redirection methodology with the HMDC approach to address the problems of wide-area VM migration and thus to achieve the fast and efficient live wide-area VM migration mechanism used in the proposed FM approach. After the VM has migrated and resumed running in the target cloud data centre, the mobile device will begin to be connected to the wireless access point located in target area and communicate with the target VM located in target cloud data centre, as show in Figure 3; finally, source VM is deleted. FM keeps on monitoring the signals from all mobile devices and repeats this process.

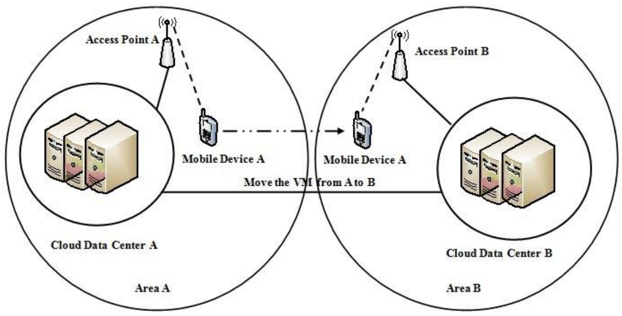

FIGURE 3 FM model of mobile cloud computing services

3.3 IMPLEMENTATION

The detailed processes of the FM scheme are described as follows:

Target area localization of mobile device A: mobile device A is moving from area A to target area B. Once the mobile device is connected to wireless access point A, the position of mobile device A will be obtained by FM. This can be done by searching a pre-defined table, for instance. The part is not the focus of this paper.

Distance Calculation: as show in Figure 4, after FM has obtained the position of mobile device A continuously, it would find that mobile device A is getting closer to cloud data centre B. FM calculates the distance l from mobile device A to cloud data centre B and the distance *h* from mobile device A to cloud data centre A which its VM is running in continuously. As long as it finds that *l* is smaller than *h*, FM will perform the follow steps. If *l* is bigger than *h*, FM will continue to monitor and calculate. Furthermore, if there are more cloud data centres available in the vicinity of mobile device A, FM will calculate the distance between the mobile device and each of the areas and once there is one distance being gradually smaller than the distance from mobile device A to current cloud data centre, FM will move the VM to that cloud data centre.

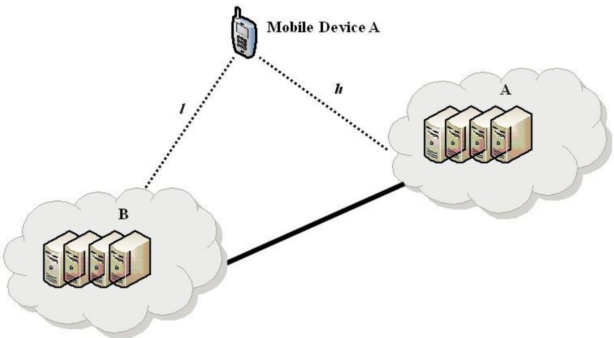

FIGURE 4 An example of mobile cloud computing model

Live wide-area VM migration: once the FM approach determines that a VM needs to be migrated, it begins to perform the live wide-area migration process of the VM. The live wide-area VM migration includes three main tasks: runtime states (memory and status) migration, VM disk-image migration and wide-area network recovery. The proposed FM approach assumes each VM is composed of two elements: the composed VM disk-image and the user work directory, both stored inside a NAS of the cloud where the VM is hosted [15]. Each cloud data centre holds its own VM components repository and NAS. The repository contains a set of VM components, whereas the NAS contains the storage of each VM (the composable and user data blocks). As mentioned above, cloud data centres are connected with each other through high-speed wired networks. When FM needs to migrate a VM, first of all it will send information to target cloud data centre in order to tell it which components the migrant VM is formed by. The target cloud checks in its own repository and notifies source cloud which components it does not hold. Thus, target cloud copies the components from the repository of source cloud, once it holds all the required VM components, it locally clones the composed VM disk-image of the migrant VM. Then, source cloud begins to migrate the memory and status with the proposed HMDC approach. At this point, we assume that target cloud has determined the target host by a specific placement selection policy.

Source host utilizes dynamic ballooning mechanism [16] to recycle idle pages of source VM to reduce total data transmitted. Source host opens up memory cache of source VM. The total memory is transmitted to target host with source VM running. During this process, if a page is to be overwritten, its original data will be copied to memory cache at first. All pages dirtied are marked to dirty_page bitmap and idle dirty pages are marked using a special mark during total memory copy. While total memory copy is completed, the source VM stops running.

After source VM stops running, source host looks up cache blocks of dirty pages marked in dirty_page bitmap. If the old version of a dirty page is cached in memory cache, the flag bit of the memory page is denoted by "1". If not, is denoted by "0". After all dirty pages have been checked, source host generates a new bitmap called cache_bitmap, which marks all dirty pages whose old versions are cached in source host. Then source host copies the two bitmaps to target VM. As illustrated in Fig.5, after target VM receives them, according to the number of dirty pages marked in cache_bitmap, target host opens up memory cache and caches the corresponding memory pages to memory cache. At the same time, an AMT (address mapping table) to maintain the cache mapping is created. Finally, it sets the lowest EPT (Extending Page Table) items of all dirty pages to "non-present" according to dirty_page bitmap. Memory cache of source and target will be recycled once its data has been used so that the approach does not generate extra space overhead. During source VM's stopping running, an IP tunnel [17] between the old IP address at the source and its new IP at the target will also be set with the help of *iproute2*. In addition, the other runtime data including CPU status and I/O states etc. will be transmitted to target host in the process.

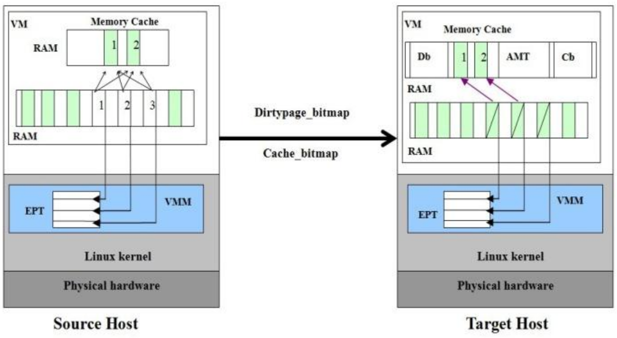

FIGURE 5 The process of bitmaps copy.

Target VM resumes running and source host begins pushing dirty pages periodically. The algorithm sets a timer. If the timer times out, according to dirtypage_bitmap, HMDC copies non-idle dirty pages to a pushing queue until it is full or dirty pages are exhausted. After a process of active push is completed, the timer restarts. If during timing receiving a page request, source host immediately suspends the timer and copies the page requested to the pushing queue. Subsequently according to the principle of locality, also copies its left and right neighbour dirty pages to the pushing queue until it is full or the dirty pages are exhausted. The timer resumes timing after sending the queue. If the dirty pages to be transmitted have the cache of old versions, the algorithm performs delta compression on the dirty pages. It firstly computes the delta page by applying XOR on the current and previous version of a page, and then get the delta compression page by compressing the delta page using XOR binary RLE algorithm [18]. Finally, a delta compression flag is set in the page header. Source host replaces dirty pages with their delta compression pages and sends the queue to target host. Finally, the corresponding cache blocks are recycled by VMM (virtual machine monitor). If the old versions of pages do not exist in memory cache, source host directly transmits dirty pages to target host. Delta compression should consume a minimum of CPU resources, both for cache hits and cache misses to not slow down the migration process or disrupt the performance of VM. Accordingly, a 2-way set associative caching scheme is employed.

At the same time when target VM resumes running, target host begins performing demand paging. While the pages marked in dirtypage_bitmap are accessed, memory access faults will occur and then be fallen into VMM kernel to be captured by the algorithm. If the page is not an idle dirty page, with target VM suspended a page request will be sent to source host. Source host transmits a set of pages, which include the requested page to target host as its response. Target host receives the response and updates memory pages correspondingly. Then target VM resumes running. If the requested page is an idle dirty page, target host does not send the request to source host but directly allocates a memory page to target VM from the local. Subsequently target VM immediately resumes running. During target VM running, target VM will receive dirty pages from source VM periodically. For target VM, both the response of a dirty page request and the active push of source host are checked whether they have delta compression flags. If yes, the algorithm firstly decompresses the pages to get delta pages and then rebuilds the new pages by applying XOR on the delta pages and the old versions cached in target host. The algorithm updates the memory pages using the new rebuilt pages and recycles the cache blocks to VMM. If no, the algorithm directly updates the pages. According to dirtypage_bitmap, while all dirty pages have been synchronized, memory migration will ends. Furthermore, after target VM resumes running, the algorithm begins forwarding all pockets, which arrive at source VM for the VM's old IP address to the target through the IP tunnel. In target host, the target VM has two IP addresses: its old one, used by existing connections through the tunnel, and its new one, used by new connections directly. The IP tunnel is torn down when no connections remain use the VM's old IP address. When the migration has completed and the VM can respond at its new network location, the dynamic DNS entry [19] for the services the VM provides will be updated in order to ensure that future connections are directed to the VM's new IP address [20]. It is notable that the packets, which arrive during VM downtime have to either be dropped or queued. The proposed FM approach will utilize *iptables* to drop them in order to avoid connections reset.

Although the proposed FM approach has exploited composed image cloning methodology to reduce the size of a VM disk-image and thus to make the algorithm need only to transmit zero or several small components through network before live migration of runtime states, the user data blocks which may be accessed at any time during VM running has not yet been migrated to target host. To address this problem of live migration of the user data blocks (the user wok directory), we have introduced a storage access mechanism into the proposed FM approach. It is composed of two main objects: a storage server and a proxy server of a block-level storage I/O protocol (e.g. iSCSI and NBD). FM has employed the NBD protocol due to its simple and plain. Source and target host nodes are connected to the storage and proxy server, respectively, by the NBD protocol, using TCP/IP. A VM accesses virtual disks via block device files (e.g. /dev/nbd0) on a host operating system. Before live migration, it works in the same manner as a normal NBD storage server, which redirects I/O requests from the VM to a disk image file. After FM starts live migration, the mechanism works together with memory migration. Once the target VM resumes running in target host during the live migration of memory, all I/O is performed at the target via the proxy server. When a user data block is accessed and it is not cached to the proxy server, it redirects the I/O requests to the storage server at the source and also caches the disk blocks to a local file at the proxy server node. This is similar to the demand paging of memory migration. The proxy server keeps on remote block copies through an NBD connection until all user data blocks are cached at the target. The background copying mechanism that copies the rest of the in-use blocks which still remain at the source is the same as the active push mechanism of memory migration. Afterwards, the proxy server terminates the NBD connection and the VM does not have to rely on the storage server at the source. It continues to work in the same way before migration [21].

After the downloading of small components, the live migration of runtime states, the redirection of wide-area network and the migration of the user data blocks have been completed, the whole live wide-area VM migration will be completed. When mobile device A needs mobile cloud computing services and communicate with its VM, it will be connected to access point B and interact with the target VM. The target VM provides mobile cloud computing services to mobile device A. The source VM is then deleted. FM continues to monitor all mobile devices and repeat the above algorithm process.

4 Evaluation

4.1 EXPERIMENTAL SCENARIOS

In this section, we will experimentally verify the proposed FM scheme. We conduct several simulations to study the performance and efficiency of the proposed FM scheme for improving QoS of mobile cloud computing services relying on effective VM migration. We implemented the proposed FM scheme and its algorithm by using CloudSim 2.0 [22]. 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 centre, etc [23-24]. In our experiments, the live wide-area migration process is ignored and the default migration mechanism in CloudSim is employed so it will not affect the comparison of the experimental results. The effectiveness and the efficiency of the proposed FM scheme are evaluated mainly on the average service response time (delay time), energy consumption and error rate, via comparing with the traditional models of mobile cloud computing services.

On CloudSim platform, we create two cloud data centres A and B. Each cloud data centre is composed of five hosts. There are two VMs running in each host at the beginning. Besides, we create an independent host A running Android OS to simulate our mobile device A. The host A can be regarded as mobile device A by adjusting its communication bandwidth with cloud data centres A and B. The communication bandwidth is larger and the distance from mobile device A to the cloud data centre is shorter. The mentioned bandwidth refers to available wireless bandwidth. With the distance gradually increasing, the available wireless bandwidth will decrease since the link path is longer and the loss of network signals is higher as well as the probability with which the bottleneck link is encountered is higher. In our simulation, we use the smooth curve of inverse proportion function to measure the relationship between the bandwidth and the distance.

$$Bandwidth = \frac{k}{Distance}, \quad (Distance > 0), \qquad (1)$$

where k is a positive degree coefficient. The cloud data centre A, cloud data centre B and host A have a coordinate attribute (i.e. (x, y)) respectively as shown in Table I to denote their positions.

TABLE 1 Position coordinates

Cloud data centre A	(0, 0)
Cloud data centre B	(x_1, y_1)
Host A	(x_2, y_2)

The position of cloud data centre A is set as the origin of a rectangular coordinate system. The position of cloud data centre B is initialized at random. The position of host A can be updated randomly and it is limited within the rectangle formed by cloud data centre A and cloud data centre B, as shown in Figure 6.

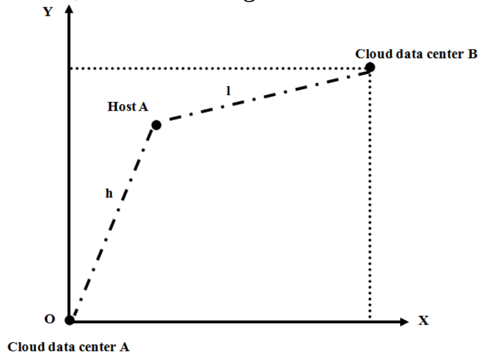

FIGURE 6 Positions in the rectangular coordinate system

In cloud data centre A, we set a VM as the source VM of mobile device A. It uses Android OS and executes a simulated mobile cloud service. The host A sends the service request to the VM periodically. After receiving a request, the VM will respond the request and send a response data to the host A. In CloudSim platform, we have implemented the simulation process of the proposed FM algorithm and processes. The host A sends simulated signals to the FM process periodically to make FM obtain its position. In addition, the traditional model is also implemented through the absence of the VM migration process.

4.2 EXPERIMENTAL RESULTS AND ANALYSIS

In the first set of experiments, we have verified the feasibility, effectiveness and efficiency of the proposed FM approach by comparing with the traditional approach on the average service response time (i.e. delay time). As shown in Figure 7, we can find that the average service response time of the proposed FM approach is shorter than that of the traditional approach all the way. The reason for the experimental result is that the proposed FM approach has abilities in making the VM providing mobile cloud computing services follow its mobile device. This will reduce intermediate links and indirectly increase the effective available bandwidth and throughput of their interaction with each other as well as to thus achieve a better average service response time finally.

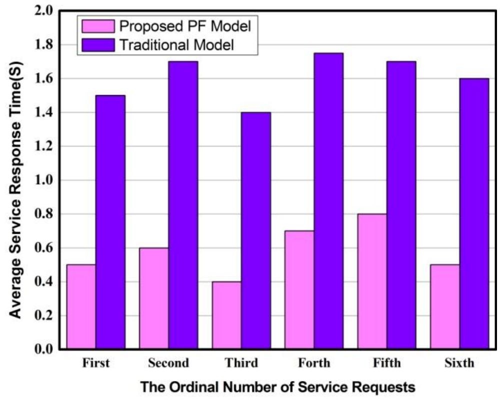

FIGURE 7 Comparison of average service response time

In the second set of experiments, we have compared the proposed FM approach with the traditional approach on the error rate of network communication. We analyze the proposed FM approach and the traditional approach by the statistics of error rate of every day on the communication data between mobile device (the host A) and its VM. Figure 8 shows the comparison of the traditional approach and FM approach in error rate. The experimental result indicates that compared with the traditional approach, FM has achieved a clearly smaller error rate. And as the time increases, the error rate of FM is more stable. This is because a better available network bandwidth, a better communications link and the absence of intermediate link will lead to a better communication quality and make the errors not easy to produce.

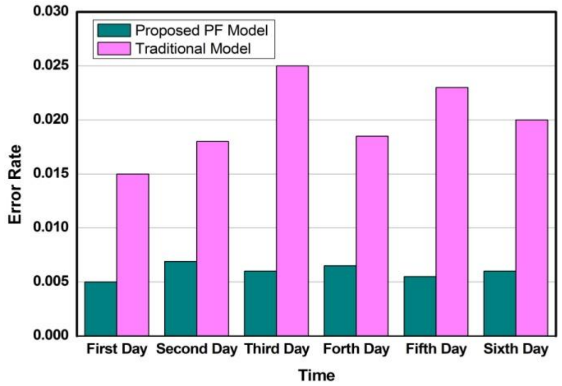

FIGURE 8 Comparison of error rate

In the third set of experiments, we have verified the proposed FM approach by evaluation of energy consumption of the whole system. The energy consumption of the two systems implementing the FM approach and traditional approach is compared to test the performance and efficiency of the FM approach. We analyze the two approaches by statistics of energy consumption of each hour in the whole system. The experimental result is as shown in Figure 9.

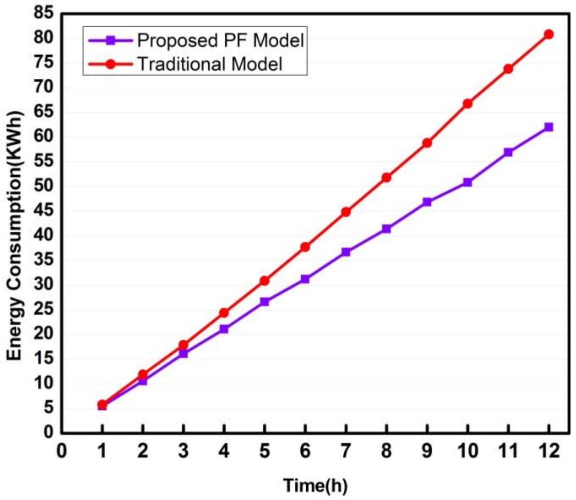

FIGURE 9 Comparison of energy consumption

It indicates that compared with the traditional approach, FM has a less energy consumption. In addition, as the time increases, the incremental energy consumption of FM is less than that of the traditional approach. The reason for the experimental result is that FM makes the interaction between mobile device and its VM not need more transmission of intermediate nodes. In other words, the FM approach has made the mobile device's communication with its VM achieved in a relatively shortest link. As a result, it consumes much less energy.

5 Conclusion and future work

In this paper, we have proposed an efficient "Follow Me" mobile computing scheme FM, which selects VM placements dynamically and migrate VMs effectively. FM scheme is designed for improving QoS and efficiency based on live wide-area VM migration in mobile cloud computing environments. Based on the mobile network features, in particular network bandwidth and network delay, the FM algorithm takes into account the distance between mobile device and its VMs. When a mobile device moves to other areas, the corresponding VM providing services to the mobile device will also be dynamically moved to that area in order to obtain better QoS. FM has achieved the live wide-area migration of VM based on obtaining the position of mobile device in real time and thus to make that the VM follows its mobile user come true. To achieve a 'Follow-Me' policy and the live wide-area VM migration based on it, FM has not only exploited the wireless positioning technology and proposed the method of comparing the distances to achieve the trigger mechanism of migration, but also combined live local-area migration mechanism HMDC with the composed image cloning (CIC) methodology and wide-area network redirection methodology to achieve the efficient live wide-area VM migration mechanism. The final experimental results demonstrate that FM evidently improves the performance, efficiency and QoS of mobile cloud computing with the user moving compared with the traditional approach. FM has the better average response time and error rate as well as has a less energy consumption. It makes the result of mobile cloud computing higher effective and more meaningful.

Aiming to further improve the performance of FM, we plan to study the robustness of FM in the future. In the cases of power outage or server crash, etc., FM should have the abilities of recovering the original VM and run-time data. As a future work, we will implement a prototype of the FM scheme to further study the energy efficiency aspect and the throughput of a FM scheme implemented mobile cloud network.

Acknowledgments

This work was supported by European Commission the 7th Framework Programme MONICA project under Grant No. PIRSES-GA-2011-295222.

References

[1] Gaochao Xu, Yan Ding, Liang Hu, Xiaodong Fu, Jia Zhao, Hao Yan, Jianfeng Chu 2013 *Journal of Convergence Information Technology* **8**(8) 341-8

[2] Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M 2009 *Communications of the ACM* **53**(4) 50-8

[3] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, Ashwin Patti 2011 *Proc. of the sixth conference on Computer systems* ACM New York: Salzburg 301-4

[4] Byung-Gon Chun, Petros Maniatis 2010 Dynamically partitioning applications between weak devices and clouds *Proc. of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond* ACM New York: San Francisco

[5] Lei Yang, Jiannong Cao, Yin Yuan, Tao Li, Andy Han, Alvin Chan 2013 *Performance Evaluation Review* **40**(4) 23-32

[6] Eric Jung, Yichuan Wang, Iuri Prilepov, Frank Maker, Xin Liu, Venkatesh Akella 2010 User-profile-driven collaborative bandwidth sharing on mobile phones *Proc. of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond* ACM New York: San Francisco

[7] Klein A, Mannweiler C, Schneider J, Hans D 2010 Access schemes for mobile cloud computing *Proc. of the 11th International Conference on Mobile Data Management (MDM)* IEEE Computer society: Kansas City 387

[8] Lan Zhang, Xuan Ding, Zhiguo Wan, Ming Gu, Xiang-Yang Li 2010 WiFace: a secure geosocial networking system using WiFi-based multihop MANET *Proc. of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond* ACM New York: San Francisco

[9] Kovachev D, Renzel D, Klamma R, Yiwei Cao 2010 Mobile Community Cloud Computing: Emerges and Evolves *Proc. 2010 Eleventh International Conference on Mobile Data Management (MDM)* IEEE Computer Society: Kansas City 393-5

[10] Liang H, Huang D, Cai L X, Shen X, Peng D 2011 Resource allocation for security services in mobile cloud computing *Proc. of Computer Communications Workshops (IEEE Conference on INFOCOM WKSHPS)* IEEE: Shanghai 191-5

[11] Jia Zhao, Liang Hu, Gaochao Xu, Yan Ding, Jianfeng Chu 2013 A survey on green computing based on cloud environment *International Journal of Online Engineering* **9**(3) 27-33

[12] Liang Hu, Jia Zhao, Gaochao Xu, Yan Ding, Jianfeng Chu 2012 A Survey on Data Migration Management in Cloud Environment *Journal of Digital Information Management* **10**(5) 324-31

[13] Cheng C, Jain R, van den Berg E 2003 Location prediction algorithms for mobile wireless systems *Wireless internet handbook* ed Borko Furht and Mohammad Ilyas: Boca Raton 245-63

[14] Liang Hu, Jia Zhao, Gaochao Xu, Yan Ding, Jianfeng Chu 2013 HMDC: Live Virtual Machine Migration Based on Hybrid Memory Copy and Delta Compression *Applied Mathematics & Information Sciences* **7**(2L) 639-46

[15] Celesti A, Tusa F, Villari M, Puliafito A 2010 Improving Virtual Machine Migration in Federated Cloud Environments *Proc. of 2010 Second International Conference on Evolving Internet* CPS: Valencia 61-7

[16] Hines M R, Gopalan K 2009 Post-Copy Based Live Virtual Machine Migration Using Adaptive Pre-Paging and Dynamic Self-Ballooning *Proc. of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments* ACM New York: Washington 51-60

[17] Perkins C 2003 IP encapsulation within IP *RFC 2003*

[18] Pountain D 1987 Run-length encoding *Byte*, 317-319

[19] Wellington B Secure DNS Dynamic Update *RFC 3007*

[20] Bradford R, Kotsovinos E, Feldmann A, Schioberg H 2007 Live wide-area migration of virtual machines including local persistent state *Proc. of the 3rd international conference on Virtual execution environments* ACM: San Diego 169-79

[21] Takahiro Hirofuchi, Hirotaka Ogawa, Hidemoto Nakada, Satoshi Itoh and Satoshi Sekiguchi 2009 A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds *Proc. of 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid* IEEE Computer Society: Washington 460-5

[22] CloudSim 2.0 http://www.cloudbus.org/cloudsim/ 2013

[23] Calheiros R N, Ranjan R, Beloglazov A, De Rose C A F, Buyya R 2011 CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms *Software: Practice and Experience* **41**(1) 23 – 50

[24] Buyya R, Ranjan R, Calheiros R N 2009 Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: challenges and opportunities *Proc. International Conference on High Performance Computing & Simulation* IEEE: Leipzig 1 – 11

Authors

Gaochao Xu, born in 1966, Xiaogan City, Hubei Province, China

Current position, grades: Professor and PhD supervisor of College of Computer Science and Technology, Jilin University, China.
University studies: Computer Science and Technology
Scientific interest: Distributed System, Grid Computing, Cloud Computing, Internet Things, etc.
Publications: 55

Yan Ding, born in 1988, China

Current position, grades: Changchun, Master
University studies: bachelor degree at Jilin University in 2011
Scientific interest: Virtualization, Cloud Computing, Mobile Cloud Computing
Publications: SCI 1

Shumao Ou, Changsha

Current position, grades: a senior lecturer in the Department of Mechanical Engineering and Mathematical Sciences at Oxford Brookes University, Professor
University studies: PhD degree in Electronic Systems Engineering and MSc degree (with distinction) in Computer Information Networks from the Department of Electronic Systems Engineering, University of Essex, Colchester, United Kingdom, in 2004 and 2007 respectively
Scientific interest: Wireless Networks, Vehicular Communications networks
Publications: SCI 12
Experience: Dr Shumao Ou is a member of IEEE. From the end of 2006, he worked in the School of Computer Science and Electronic Engineering, University of Essex as a Senior Research Officer. From June 2009, he took a lectureship in the School of Technology at Oxford Brookes University.

Liang Hu, born in 1968, Changchun

Current position, grades: Professor PhD supervisor of College of Computer Science and Technology, Jilin University, China
University studies: BS degree on Computer Systems Harbin Institute of Technology in 1993 and his PhD on Computer Software and Theory in 1999
Scientific interest: Network Security, Computer Network, Cloud Computing
Publications: SCI 12
Experience: As a person in charge or a principal participant, Dr Liang Hu has finished more than 20 national, provincial and ministerial level research projects of China.

Jia Zhao, born in 1982, Changchun City, Jilin Province, China

Current position, grades: Doctor
University studies: Computer Science and Technology
Scientific interest: Distributed System, Cloud Computing.
Publications: 11
Experience: Doctor of College of Computer Science and Technology, Jilin University, China.