Terminal Virtualization for Mobile Services

Tao Zheng, Song Dong
Orange Labs International Center
Beijing, China
e-mail: {tao.zheng, song.dong}@orange.com

***Abstract*–Terminal virtualization focuses on applying Information Technology (IT) virtualization technology to the terminals to realize the full or parts of terminal functions extension or migration to other devices on the network, such as resource reducing, information sharing, data synchronization, etc. It is becoming increasingly clear that more and more features of terminal virtualization and mobile computing on the edge will be used in practice. However, some issues are raised with terminal virtualization, such as security, privacy, Quality of Service (QoS), efficient transmission, computation/functions offloading management, etc. In this paper, after analyzing above issues, a mobile terminal virtualization framework is proposed and considered to be implemented in terminal Operating System (OS) and transparent to users.**

Keywords-terminal virtualization; mobile cloud computing; computation offloading; QoS

I. INTRODUCTION

With the explosive growth of mobile terminals in recent years, user preferences have shifted from traditional cell phones and laptops to smartphones and tablets. In recent years, there are abundant applications in various categories, such as entertainment, health, games, business, social networking, travel and news, running at mobile terminals. The burden of computation on the terminals has been raised rapidly and more functions and sensors are required to be applied to them. Mobile cloud computing and terminal virtualization are proposed to handle these issues, which are able to provide tools to the user when and where it is needed irrespective of user movement, hence supporting location independence. Indeed, "mobility" is one of the characteristics of a pervasive computing environment where the user is able to continue ones work seamlessly regardless of the movement.

Advances in the portability and capability of mobile terminals, together with widespread Long Term Evolution (LTE) networks and WiFi access, have brought rich mobile application experiences to end users. Undoubtedly, mobile broadband terminals, such as smart phones, tablets, wireless dongles and some data-intensive apps have been an exponential increase in mobile Internet Protocol (IP) data usage, which will used up the mobile bandwidth. The demand for ubiquitous access to a wealth of media content and services will continue to increase, as indicated in a report by Cisco [1]: the Compound Average Growth Rate (CAGR) of global IP traffic from mobile terminals is 61% from 2013 to 2018, which is triple CAGR from fixed Internet.

In addition, the resource-constrained mobile terminals, especially with limited battery life, have been a barrier to the improvements of mobile applications and services. While new smart phones with bigger screens, faster Central Processing Units (CPUs), and larger storage are launched continually, and the bandwidth of wireless networks has increased hundreds of times in just a few years, the development of batteries has lagged far behind the development of other components in mobile terminals. In fact, faster CPUs, larger displays and multimedia applications consume more battery energy. The limitations of computation resources and sensors are other stumbling blocks for services development. Mobile cloud computing and terminal virtualization can help to resolve this issue.

Mobile cloud computing and terminal virtualization have been the leading technology trends in recent years. The increasing usage of mobile computing is evident in the study by Juniper Research, which states that the consumer and enterprise market for cloud-based mobile applications is expected to rise to $9.5 billion by 2014 [2]. Mobile cloud computing/terminal virtualization is introduced to resolve the conflicts mentioned above, in which the cloud serves as a powerful complement to resource-constrained mobile terminals. Rather than executing all computational and data operations locally, mobile cloud computing/terminal virtualization takes advantage of the abundant resources in cloud platforms to gather, store, and process data for mobile terminals. Many popular mobile applications have actually employed cloud computing to provide enhanced services. More innovative cloud-based mobile applications like healthcare monitoring and massively multiplayer online mobile games are also under development.

The objective of the paper is to introduce the concept of terminal virtualization and study the related issues and research status. On this basis, a proposal terminal virtualization framework is finally presented.

This paper is organized as follows. In Section 2, we introduce the concept and current status of terminal virtualization. In Section 3, capabilities and functions extension are analyzed. In Section 4, a terminal virtualization framework for mobile networks is presented. Some implementing issues about this framework are discussed in Section 5. Finally, Section 6 summarizes the conclusions.

II. TERMINAL VIRTUALIZATION

Terminal virtualization helps to relief the local resource-constrained problem through offloading some tasks to the cloud and utilizing capabilities and functions in the cloud. First, the scope of terminal virtualization needs to be clarified.

A. The scope of terminal virtualization

From the points of view of virtualization, mobile cloud computing can be as a kind of terminal virtualization scenario. There are two scenarios as following:

- Full Virtualization Scenario

The requirement for full terminal virtualization mainly comes from some enterprises. In these enterprises, employees are buying their own terminals and want to connect to the enterprise network so that they can do their work with greater flexibility. However, the employees also don't want to give up user experience and freedom at the cost of complex IT security policies. In order to achieve this goal, terminal virtualization is becoming a very attractive choice because it offers flexibility and addresses the concerns over privacy of personal data while also delivering the security requirements of the enterprise. On the other side of the ecosystem, the terminal makers and carriers will benefit from terminal virtualization because they are able to more easily replicate the features found in various terminals and also deliver more features at a lower cost.

Full terminal virtualization is not an ordinary schema for public mobile customers. In general way, the terminal is sold with a pre-determined OS and customers can use services based on this OS.

- Partial Virtualization Scenario

Broadly speaking, mobile cloud computing can be as one kind of partial terminal virtualization, a part of terminal computation powers and functions can be virtualized into the remote networked cloud. Terminals can get local experience through running remote apps or some information located in the remote cloud.

This scenario is more practiced and popular in present. Some applications employ this method to add extending functions or improve user experience. Even cloud phone appears and is deeply merged with networking services for user convenience. In this paper, we mainly focus on the partial virtualization scenario.

B. Drivers and Benefits of Terminal Virtualization

Terminal virtualization facilitated the fusion of mobile terminal and cloud service that provides a platform wherein some computing, storing and data abstraction tasks are performed by the cloud and mobile terminal simply seeks an access to them. Following shows the drivers and benefits of terminal virtualization:

- Limitless Storage Space

Now, instead of memory cards for more space, the cloud storage can provide limitless space for applications, even with the help of terminal virtualization framework/middleware they don't need to care about the location of the storage.

- Improved Processing Facility

The price of a mobile terminal is largely dependent on its CPU's speed and performance. With the help of terminal virtualization, all the extensive and complex processing is done at the cloud level. The vital computations, encryption and decryption, everything can be handled by the cloud thereby enhancing the mobile terminal's performance.

- Save Radio Access Network (RAN)/Access Bandwidth & Resources

With the tremendous increase in mobile bandwidth consumers and user's throughput, RAN/access resources have become more valuable than before. When some functions and computation tasks are offloaded into cloud, the result instead of the original metrical is sent to the terminal, so the RAN/access bandwidth can be saved for other use.

- Enhanced Battery Life

Terminal virtualization lends a very strong helping hand to battery life of terminals. With most of the processing handled by the cloud, the battery life is enhanced, thereby making the most optimum use of the remaining recharge cycles.

- Improved User Experience

The above mentioned features will improve the end-user experience substantially, especially the experience from low-end terminals.

- Economic Factors

For the consumers, terminal virtualization can bring some new functions and improved capabilities to the old terminal without spending one penny. For operators, the benefits come from saved network resources and flexible service deployment by terminal virtualization.

- Reserving for Upcoming Technologies

Terminal virtualization is adapt with the tremendous pace of developing technology and works most efficiently with the upgrades. Through separating the implementation from the function body, upcoming technologies can be easy to be introduced to the terminals.

C. Challenges

In this section, we discuss that the issues have not been sufficiently solved in terminal virtualization.

- Energy-efficient Transmission

Wireless networks are stochastic in nature: not only the availability and network capacity of access points vary from place to place, but the downlink and uplink bandwidth also fluctuates due to weather, building/geographical shields, terminal mobility, and so on. Measurement studies [3] show that the energy consumption for transmitting a fixed amount of data is inversely proportional to the available bandwidth.

Computation/Data offloading can save energy only if heavy computation is needed and a relatively small amount of data has to be transferred. Energy efficiency can be substantially improved if the cloud stores the data required for computation, reducing data transmission overhead. Bandwidth allocation and admission control mechanisms in cellular base stations and access points may guarantee network connectivity to a certain extent, but cannot eliminate the stochastic nature of wireless links. An alternative approach is to dynamically adjust application partitioning between the cloud and mobile terminals according to network conditions, although it is challenging to quickly and accurately estimate the network connectivity with low overhead.

Energy-efficient transmission is also critical when exploiting the cloud to extend the capabilities of mobile terminals. Frequent transmissions in bad connectivity will overly consume energy, making the extended capabilities unattractive, as battery life is always the top concern of mobile users. A solution called eTime [4] is to adaptively seize the timing opportunity when network connectivity is good to pre-fetch frequently used data while deferring delay-tolerant data.

- Security

There are several aspects of terminal virtualization security, including antivirus, authentication, data protection, and digital rights management. Security vulnerability can cause serious problems, including property damage, cloud vendor economic loss, and user distrust. Since mobile terminals are resource-constrained, locally executed antivirus software can hardly protect them from threats efficiently. A current solution is to offload the threat detection functionality to the cloud. Nevertheless, since a pure cloud antivirus relies on cloud resources, it is difficult to deal with malware that can block the terminal's Internet connection.

Besides, authentication is critical for access to sensitive information, such as bank accounts and confidential files. With constrained text input on mobile terminals, users tend to use simple passwords, making mobile applications more vulnerable to authentication threats. To solve this issue, Chow et al. [5] builds up an authorization platform where users are identified by their habits (e.g., calling patterns, location information, and web access). The platform routinely records user behavior information. When a server receives an authorization request, it redirects the request to an authorization engine, which uses the aggregated behavior information and an authorization policy to decide whether to accept the request or not.

- Privacy

Since mobile terminals are usually personal items, privacy must be considered when leveraging the cloud to store and process their confidential data remotely.

A secure data processing framework [6] can be used into terminal virtualization, in which critical data are protected by the unique encryption key generated from the user's trusted authority and stored in an area isolated from the public domain. Even when storage is breached in the cloud, unauthorized parties including the cloud vendor cannot obtain the private data.

Another particular privacy issue for mobile users is the leakage of personal location information in location-based services. To address the issue, a method called "location cloaking" [7] makes user location data slightly imprecise before submitting them to the cloud. But the imprecise data sometimes cannot provide relevant or satisfactory results to users in certain applications. Therefore, location cloaking should be adaptively tuned to balance the trade-off between privacy and result accuracy.

- Real-time Requirements and Service QoS

When terminal virtualization and mobile computing are applied, QoS will become more important. How to guarantee the related data or stream to be transmitted in time determinates the services' failure or success.

While different applications offer different functionality to end users, the primary service Key Quality Indicators (KQIs) across the application's customer facing service boundary for end users of applications generally include service availability, service latency, service reliability, service accessibility, service throughput, and application specific service quality measurements.

III. Computation Offloading & Functions Extension

Terminal virtualization enables enhanced mobile experiences that were previously impossible on resource-constrained and function-constrained mobile terminals. Many commercial mobile applications use the cloud to bring about rich features. They usually employ a client-server framework that consists of two parts, which run on the mobile terminal and the cloud, respectively. Essentially, cloud computing helps extend the capabilities and functions of mobile terminals in some aspects.

A. Capabilities & Functions extension

Through terminal virtualization, the capabilities and functions can be reallocated between terminal and cloud, as shown in the following examples:

- Computation-intensive Task

At present, many applications nowadays support speech/picture/video recognition. The models for recognition and high-quality synthesis must be trained with millions of samples in thousands of examples. This computation-intensive task is infeasible on a mobile terminal and should be offloaded to the cloud.

- Remote Sensors/Inductors

Because of the limitation from terminal itself (low-end model lacking some sensors or inductors) or other conditions (e.g., distance exceeding the maximum length of sensors), some services cannot work well. However, these services can work through getting and storing related information from mobile cloud platform. From the terminal point of view, its functions are extended.

- Application Portability

It allows for the rapid transfer of applications (which may occur on the fly), providing the freedom to optimize, without the constraints of the location and required resources of the virtual appliances. The precise but extensible definition of the

 ISBN: 978-1-61208-404-6

services provided by the application platform is the key to ensuring application portability.

B. Computation offloading Decision

To overcome resource constraints on mobile terminals, a general idea is to offload parts of resource-intensive tasks to the cloud (centralized server or other peers). Since execution in the cloud is considerably faster than that on mobile terminals, it is worth shipping code and data to the cloud and back to prolong the battery life and speed up the application. This offloading procedure is illustrated in Fig. 1. Several technologies to realize the runtime environment in the cloud, the major differences between offloading techniques lie in the offloading unit and partitioning strategies.

- Client–Server Communication Mechanism

In the Client–Server Communication, process communication is done across the mobile terminal and cloud server via protocols, such as Remote Procedure Calls (RPC), Remote Method Invocation (RMI) and Sockets. Both RPC and RMI have well supported APIs and are considered stable by developers. However, offloading through these two methods means that services need to have been pre-installed in the participating terminals.

Spectra [8] and Chroma [9] are the examples of systems that use pre-installed services reachable via RPC to offload computation. Hyrax [10] has been presented for Android smartphone applications which are distributed both in terms of data and computation based on Hadoop ported to the Android platform. Another framework based on Hadoop is presented by in [11], for a virtual mobile cloud focusing on common goals in which mobile terminal are considered as resource providers. Cuckoo [12] presents a system to offload mobile terminal applications onto a cloud using a Java stub/proxy model. The Mobile Message Passing Interface (MMPI) framework [13] is a mobile version of the standard Message Passing Interface (MPI) over Bluetooth where mobile terminals function as fellow resource providers.

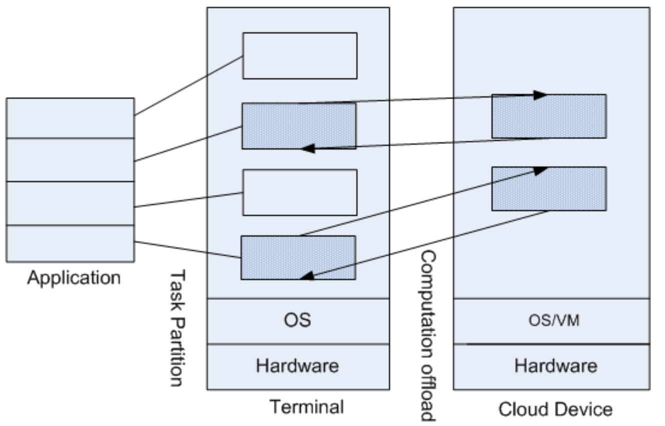

Figure 1. The procedure of computation offloading

- Mobile Agent

Scavenger [14] is another framework that employs cyber-foraging using WiFi for connectivity, and uses a mobile code approach to partition and distribute jobs. Using its framework, it is possible for a mobile terminal to offload to one or more agents and its tests show that running the application on multiples in parallel is more efficient in terms of performance. However, the fault tolerance mechanism does not be discussed and since its method is strictly about offloading on agents and not sharing, it is not really dynamic. Also its agents are all desktops and it is unclear if Scavenger is too heavy to run on mobile phones.

- Virtualization/Virtual Machine (VM) Migration

The execution can't be stopped when transferring the memory image of a VM from a source terminal to the destination server [15]. In such a live migration, the memory pages of the VM are pre-copied without interrupting the OS or any of its applications, thereby providing a seamless migration. However, VM migration is somewhat time-consuming and the workload could prove to be heavy for mobile terminals.

VM migration is used by a majority of frameworks, including Cloudlets [16], Maui [17], CloneCloud [18], and MobiCloud [19]. Virtualization greatly reduces the burden on the programmer, since very little or no rewriting of applications is required. However, full virtualization with automatic partitioning is unlikely to produce the same fine grained optimizations as that of hand coded applications, although rewriting each and every application for code offload is also not practical. Maui actually does not rely on pure VM migration as done in CloneCloud and Cloudlets, but uses a combination of VM migration and programmatic partitioning. However, in cases where the mobile terminal user is within range of an agent terminal for a few minutes, using VM migration may prove to be too heavyweight, as is pointed out by Kristensen [14] which uses mobile agents in light of its suitability in a dynamic mobile environment.

C. Applications

The following lists the current applications using terminal virtualization concept, from functions extension to complex computing tasks.

- Mobile Cloud Phone

Mobile cloud phone differs from other smart phones in that it doesn't need to download and store apps and content on the phone; it instead accesses personal information and runs programs stored on remote network servers, via the cloud.

YunOS 3.0 [20], developed by Alibaba, debuts officially with cloud-based service for movie, taxi and other reservations on October 20th, 2014. It comes with the brand new service Cloud Card, which runs entirely in the cloud and offers the user the option to select movie tickets, taxi services and more.

- Cloud Storage and Video Adaption

Through terminal virtualization, some part of data which is stored in the cloud instead of being stored on the terminal can be treated as local data. And video stored in cloud platform can be adapted to appropriate format and code streaming fitting for the terminal when the terminal requests this video.

- Image and Natural Language Processing

For this kind of applications, the complex computation jobs which are difficult for local operating OS should be offloaded to the cloud platform and the mobile terminal just holds some interface functions. Image and voice recognition, search, adaption, natural language translation, and Artificial Intelligence (AI) machine conversation, etc., which belong to this kind of applications, can be implemented on some low-end phones with the help of computation offloading.

- Augmented Reality (AR)

Algorithms in augmented reality are mostly resource and computation-intensive, posing challenges to resource-poor mobile devices. These applications can integrate the power of the cloud to handle complex processing of augmented reality tasks. Specifically, data streams of the sensors on a mobile device can be directed to the cloud for processing, and the processed data streams are then redirected back to the device. It should be noted that AR applications demand low latency to provide a life-like experience.

IV. PROPOSED FRAMEWORK FOR MOBILE SERVICES

This section proposes a mobile terminal virtualization framework based-on mobile OS. The framework locates in the middleware layer between OS kernel and applications, as shown in Fig. 2.

A. The framework overview

The framework illustrated in Fig. 3 is composed of four processing modules: application virtualization module, computation virtualization module, storage virtualization module and network virtualization module, and a management module.

In the framework, processing modules are in charge of receiving and responding the callings from applications to OS. Management module is used to manage the framework, including security management, configuration management, network and cloud service monitoring, etc.

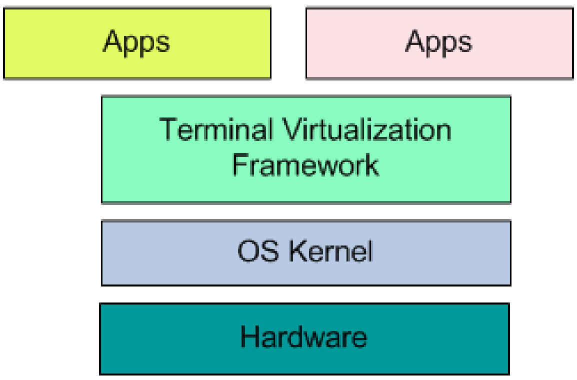

Figure 2. Hierarchical structure of the framework

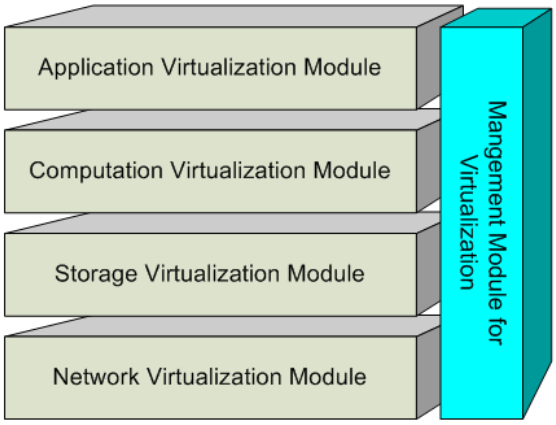

Figure 3. The overview of the framework

B. Component Modules

As shown in Fig. 4, the processing modules are able to choose the best method to process the calling according to the current status of mobile network bandwidth, local resources, terminals hardware limitation and remote cloud resources. Meanwhile, the framework provides local calling responses to the applications and shields the actual calling responses.

- Application Virtualization Module

This module is in charge of processing the functions extension of applications. When the application accesses the terminal's hardware, for example one kind of sensor, this module will check if it is available. If not, this module is responsible for finding a same remote available sensor in the cloud to satisfy the application's demand and providing the response with the result from the remote sensor to the application.

- Computation Virtualization Module

This module takes charge of monitoring the terminal's computation resource and analyzing the computation request from applications. When the computation resource is constrained (for example, the CPU usage is more than 85%) or some applications with sophisticated computing power are run (for example, virtual reality service, language processing, etc., it can be configured in advanced), this module will offload some computation tasks to the remote cloud server and provide the processing result to the applications.

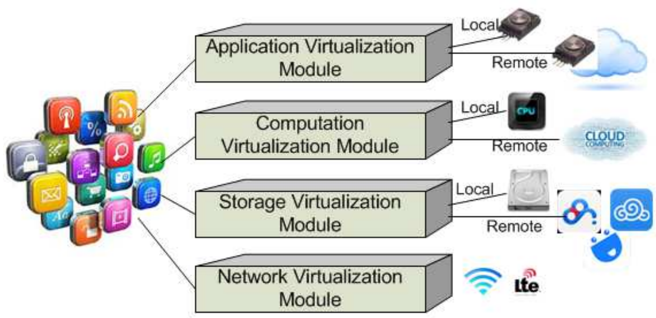

Figure 4. The functions of processing modules

- Storage Virtualization Module

This module is in charge of monitoring the terminal's storage resource and providing the remote cloud storage to applications. Through this module, the local applications can use cloud storage providing by different Service Providers (SPs), such as Baidu, Tencent, Huawei, etc., as using a local storage. At the same time, this module monitors the speed and status of the remote cloud storages and provides the best one to applications.

- Network Virtualization Module

This module is in charge of monitoring the terminal's network status and providing the best one or binding different network accesses to increase the data throughput according to the applications' demand.

- Management Module

The management module is responsible for managing the framework.

The security function including network security and resources security is an important function in this module. Other management functions include all kinds of configuration management, for example, some resources and offloading thresholds, and remote resource monitoring, for example, all kinds of cloud services, network status.

V. Considerations for Framework Implemention

We are implementing an early-phase prototype based-on Android OS according to the proposed framework. The implemented modules include network virtualization module and storage virtualization module, which are relatively easier to be implemented than other three modules in the framework.

The network virtualization module employed a method [21] to implement the network access independence, for example, using multiple access paths simultaneously, switching between access paths according to the current network environment, and recovering the access path automatically. A local proxy in terminal and a remote proxy in network cooperate to implement the functions of network virtualization module. And the applications are able to automatically adapt the change of network and not affected by it.

The storage virtualization module added online storage services to the local storage as a directory, which can be accessed by the applications as a local one. When the directory is accessed, the storage virtualization module will automatically exchange the data with the online storages. Baidu and Huawei online storage services are currently supported and chosen by the module.

To implement application virtualization, some operating system calls to local hardware need to be intercepted and rewritten. The functions to access online hardware resource will be added into the application virtualization module.

In the computation virtualization module, we plan to take different approaches according to the type of tasks. For example, for the tasks requiring sophisticated computing power defined in advance, RPC/RMI method will be used; for the independent tasks undefined in advance, virtual machine migration will be used.

To implement the security function in the management module, the data communication between the terminal and cloud platform is encrypted. In addition, security function also helps protect the terminal and remote resources from being abused by applications on the terminal.

VI. Conclusion and Future Work

In this paper, we analyze some aspects of current terminal virtualization, highlight the motivation for it, and present its functions, applications and some challenges. Terminal virtualization has overlapped with other areas, such as mobile peer-to-peer computing, application partitioning, and context-aware computing, but it still has its own unique challenges. These are still a long way to go in terminal virtualization.

Because more and more cloud resources can be made available to the mobile terminal (via the mobile cloud facility), we proposed a terminal virtualization framework for mobile services. In this framework, four processing modules and one management module are employed to handle the resource requests from apps and shield the details for accessing cloud services. In the future work, we consider completing the prototype of this framework and analyzing its performance.

References

[1] Cisco Visual Networking Index: Forecast and Methodology, 2013–2018, http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html, March 2015.

[2] "Mobile Cloud Applications & Services", Juniper Research, 2010.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications," Proc. ACM IMC, 2009, pp. 280-293.

[4] P. Shu, F Liu, H. Jin, M. Chen, F. Wen, and Y. Qu, "eTime: Energy-Efficient Transmission between Cloud and Mobile Devices," Proc. IEEE INFOCOM, 2013, pp. 195-199.

[5] R. Chow et al., "Authentication in the Clouds: a Framework and Its Application to Mobile Users," Proc. ACM Cloud Computing Security Wksp. 2010, pp. 1-6.

[6] D. Huang, Z.Zhou, L. Xu, T. Xing, and Y. Zhong, "Secure Data Processing Framework for Mobile Cloud Computing," Proc. IEEE INFOCOM, 2011, pp. 614-618.

[7] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar, "Preserving User Location Privacy in Mobile Data Management Infrastructures," Proc. Wksp. Privacy Enhancing Technologies, 2006, pp. 393-412.

[8] J. Flinn, S. Park, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing", Proc. IEEE ICDCS 2002, pp. 217-226.

[9] R. Balan, M. Satayanarayanan, S. Park, and T. Okoshi, "Tactics-based remote execution for mobile computing", Proc. ACM Mobisys, 2003, pp. 273-286.

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

[11] G. Huerta-Canepa, and D. Lee, "A virtual cloud computing provider for mobile devices", Proc. ACM MCS 2010, article No. 6.

[12] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones", Proc. ACM Mobisys, 2010, pp. 59-79.

[13] D. C. Doolan, S. Tabirca, and L.T. Yang, "Mmpi a message passing interface for the mobile environment", Proc. ACM Mobisys, 2008, pp. 317-321.

[14] M. Kristensen, "Scavenger: transparent development of efficient cyber foraging applications", Proc. IEEE PerCom, 2010, pp. 217-226.

[15] C. Clark, et al., "Live migration of virtual machines", Proc. of the 2nd conference on Symposium on Networked Systems Design & Implementation, USENIX Association, 2005, vol 2, pp. 273-286.

[16] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM –based Cloudlets in Mobile Computing,", IEEE Pervasive Computing, vol. 8, no. 4, 2009.

[17] E. Cuervo et al., "Maui: Making Smartphones Last Longer with Code Offload," Proc. ACM MobiSys, 2010, pp. 49-62.

[18] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic Execution Between Mobile Device and Cloud," Proc. ACM EuroSys, 2011, pp. 301-314.

[19] D. Huang, X. Zhang, M. Kang, and J. Luo, "MobiCloud: Building Secure Cloud Framework for Mobile Computing and Communication," Proc. IEEE SOSE, 2010, pp. 27-34.

[20] http://www.yunos.com, YunOS, March 2015.

[21] T. Zheng, and D. Gu, "Traffic Offloading Improvements in Mobile Networks", ICNS 2014, pp. 116-121.