IPASJ International Journal of Information Technology (IIJIT)
Web Site: http://www.ipasj.org/IIJIT/IIJIT.htm
A Publisher for Research Motivation ....
Email:editoriijit@ipasj.org
Volume 6, Issue 6, June 2018
ISSN 2321-5976

A Survey on Mobile Virtualization using Cloud Computing

Hind M. Said[1], A.A. Abdel-Aziz[2] and Abdel-Aziz Khamis[3]

[1]Dept. of Information Technology and Computing
Arab Open University
Cairo, Egypt

[2]Dept. of Information Systems
College of Computer and Information Sciences
Jouf University, Kingdom of Saudi Arabia

Dept. of Information Systems & Technology
Institute of Statistical Studies and Research
Cairo University, Egypt

[3]Dept. of Computer & Information Sciences
Institute of Statistical Studies and Research
Cairo University

ABSTRACT

With the fast growth of mobile technology, mobile devices have become an essential part of human life. But yet its hardware has faced lots of challenges due to the limitation of their resources, such as processing power, battery life, and storage. Mobile Cloud Computing (MCC) and virtualization have been introduced to integrate the cloud computing into the mobile environment to overcome the obstacles related to the performance of the mobile devices, and to enhance its resource utilization. In this paper, we present an overview of virtualization in terms of concepts, types, techniques, and approaches that allow users to create virtual mobile images into a cloud to free them from the limitations of their mobile devices' recourses and easily tap into the power of the cloud. Moreover, we highlight benefits and issues of some mobile virtualization approaches. This survey will support the future research and development work and raise the awareness about the presented approaches.

Keywords:MCC, Virtualization, Hypervisor

1. INTRODUCTION

Mobile devices are becoming an important part of human life, it has been moved from providing simple voice calls to support different various services like browsing websites, recording videos and many other services that make it the most useful tool used by ordinary people. All of the mobile hardware is limited in their resources such as storage, battery life [19]. Even though there are many service providers that offer online storage services to mobile device users to cope the storage limitation, yet still there is no service that present full computation facilities to mobile phone users.

To overcome the above-mentioned problem the term Mobile Cloud Computing was introduced. Before we discuss this term, let us describe the term Cloud Computing, it is a means of offering resources like storage and CPU that can be used by users through the internet. The Mobile Cloud Computing transfers the data processing and storage from the mobile devices to more powerful computing platforms located in the cloud that can be accessed via wireless connection [18]. It is similar to Cloud Computing, but the client side has replaced to make it visible for mobile devices, However, the major concept is still cloud computing. With the assistance of Cloud Computing and Virtualization, the mobile environment can improve its resource performance. Virtualization of resources is the main demand for the cloud provider, to create the fancy of infinite resources to the cloud user.

In this paper, we will describe Virtualization and how it can be applied in mobile phones to run multiple Operating System (OS) in one device, and the approaches used to virtualize the whole mobile OS to the cloud, and the approaches used to offload only mobile application to be executed to the cloud. Moreover, we will discuss the benefits and issues of Mobile virtualization approaches.

2. VIRTUALIZATION

Virtualization is a phrase that directs to the abstraction of computer resources to enhance its utilization [19]. Using Virtualization certain resource can be accessed and used by multiple applications or operations while oblivious to access to the same resource by others. Virtualization in Mobile devices can reduce its cost and increase its battery life, and assist to address safety and surety matters. Virtualization techniques are applied to generate numerous isolated partitions on an individual physical server and these techniques vary in the Virtualization solution methods and the degree of abstraction while offering similar traits and traveling towards the same end. The most popular virtualization techniques are:

A. Emulation

It is a virtualization technique which converts the behavior of the computer hardware to a software program and lies in the operating system layer which lies on the hardware. Emulation provides enormous flexibility to the guest operating system, but the speed of the translation process is low compared to the hypervisor and requires a high configuration of hardware resources to run the software [1].

B. Fullvirtualization:

Full virtualization means a complete machine is installed on another machine. That virtual machine provides all the functionality which exists on the original machine. It facilities when actual machine not free then user use the virtual machine [2]. Full virtualization differs from emulation in that operating systems and applications are designed to run on the same architecture as the underlying physical machine. Figure 1 shows the architecture of fullvirtualization.

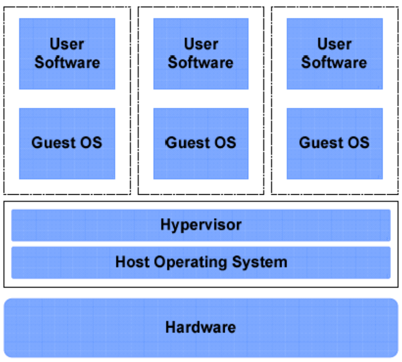

Figure 1. The Architecture of Full Virtualization.

C. ParaVirtualization

It is the method wherein a modified guest OS is able to speak directly to the hypervisor. This reduces translation time and overhead as the symbiotic relationship between the two is more efficient. However, unlike full virtualization, para virtualization requires changes to the virtualized operating system. This allows the VM to coordinate with the hypervisor, reducing the use of the privileged instructions that are typically responsible for the major performance penalties in fullvirtualization [3]. Figure 2 shows the architecture of paravirtualization.

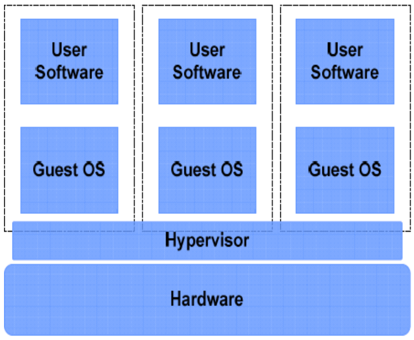

Figure 2. The Architecture of Para Virtualization

3. MOBILE VIRTUALIZATION

Mobile Virtualization is similar to Computer and Server Virtualization, however, because of the mobile hardware resources and energy limitations, a thin layer hypervisor can be used between the Operating System and the hardware that will demand less memory. The key concept of Virtualization stays the same but in terms of mobile technology.

3.1 Virtualization within Mobile Devices

[4] Xu et al claims that Mobile Virtualization demands are different from Computer Virtualization because of the power and resource limitation. He illustrates that in Fullvirtualization the computer is heavy due to the executed instructions will be replaced in real time, which will be incompatible with the mobile system, so with the high performance of Paravirtualization it can be the solution for the virtualization in mobile phones.

i. Multiple OS in one device

Virtualization allows more than one virtual smartphone to run on a single mobile device. [5] Presents Cells, an architecture that enables multiple virtual smartphones to run concurrently on a single device. It uses lightweight Operating System virtualization to supply virtual namespace that can run more than one Virtual Phones VPs in one Operating System that present only one application at a time. It introduces a usage pattern having one foreground VP that is presented and multi background VPs that are not presented at a certain time. Each virtual phone is isolated from each other to maintain security. Cells use a Paravirtualization technique which gives good performance in terms of usage. It has high compatibility and scalability. It supports moderate context switching it is weightless and provides less overhead.

[6] Proposed a (hypervisor) for smartphones called MobiVMM, to deal with the smartphone limitations that the present Virtual Machine Monitors (VMMs) doesn't address. MobiVMM schedule the display of the VMs by its priority, it gives heights priority to the real-time VM, and mark it as a non-real time to give a chance to another VM to run and display. To avoid preemption at the critical time, interrupt are enabled while there is sufficient processor time for interrupt processing. MobiVMM monitors the battery if it is low it will ask the user to disable certain services so that the main service keep running.MobiVMM works currently only with Linux as a guest OS, they are planning to support other operating systems in the future.

ii. Mobile Security

As mobile devices started to connect to the internet via wireless technologies, its security has become a serious issue, [7] proposed a layout of system Virtualization for ARM CPU architecture and describe a prototype called Xen on ARM using Xen hypervisor. Xen is a paravirtualization, they pick the ARM architecture because it is extensively used in mobile phones and pick Xen as it is the most known open source hypervisor. Xen is currently supporting IA-64, X86-64 and Power PC it cannot be ported to ARM CPU because its virtualization designed for desktop and server, they modified it to run mobile development platform. They isolated multiple VMs under the hypervisor to provide solutions to enhance security. The main encouragement is to isolate OS kernel from the user mode. Besides that, in order to deprivilege guest OS and allow full resource control to VMM, only VMM runs in the supervisor mode and guest OS runs in user mode together with the application. They provide a solution to improve security and fine-grained access control by isolating multiple virtual machines under the hypervisor.

3.2 Mobile Virtualization using Cloud Computing

Mobile virtualization using cloud allows users to create virtual smartphone images into the cloud; each image will meet different needs. Users will be able to tap freely into the power of the data center by installing the desired mobile applications remotely in one of these images. Because the mobile applications are controlled remotely, they are not constrained by the limit of processing power, memory and battery life of a physical smartphone.

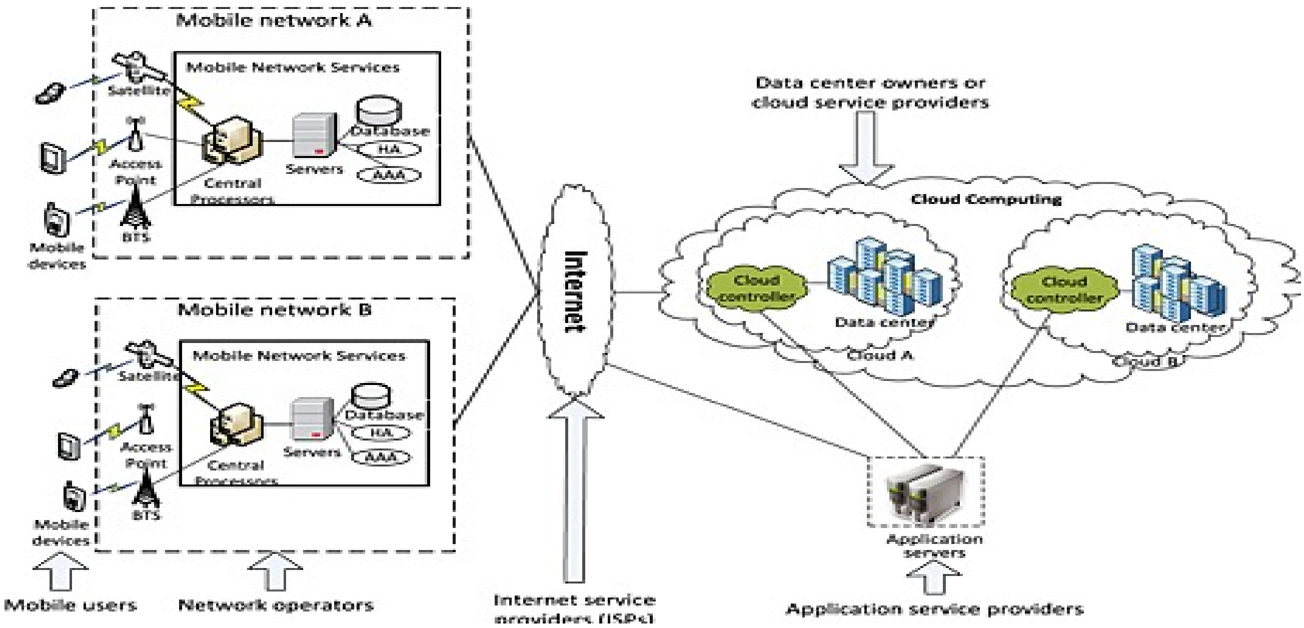

Figure 3. The Architecture of Mobile Cloud Computing

3.2.1 Mobile System Virtualization

3.2.1.1 Virtual Network Computing (VNC)

[8] In this paper the author presented a system called Virtual Smartphone over IP that creates a virtual image of mobile phones in the cloud that can be controlled remotely by the users from their physical mobile devices. Their system consists of VNC server runs in every the Smartphone image and a VNC client in the physical smartphone device. Using the VNC client the user can send and receive events from the mobile device to the virtual smartphone. As the VNC supports only Keyboard and mouse as input devices while all the smartphone devices are provided with sensor devices, the author implemented a virtual sensor driver in the virtual smartphone image, and allow their client program to send the sensor reading to those images. This is an important feature as it allows Android applications in an Android-x86 image to obtain sensor readings from the physical smartphone without any modification. And since only the graphic pixels of screen images are delivered to user's mobile phone, the actual data never leaves the secure data center. In this approach, the complete application was offloaded from the Android smartphones to the cloud. This architecture provides a viable solution to data leakage problem.

[9] Presented a mobile-to-mobile remote computing protocol for smartphones called Peek, Peek is an application that allows smartphone users to remotely interact with any application on another smartphone as if they were operating on their smartphone device. The Peek is device and Operating System OS separate that is server and client program can be implemented in for any OS and could be established in each of two physical device or virtual image in the cloud. Peek raises the ease of interaction compared to VNC because it raises the number of supported touch gestures. Using Peek on

the time taken to remotely perform certain actions on the server is reduced by 62.8%. Peek associates sensor context of a session, which allows users to experience a wide range of applications that use sensor input. Peek chooses a frame compression mode based on the server's CPU/memory load rate of change of screen pixels and network bandwidth. Using synthetic datasets, we show that Peek can potentially reduce the bytes sent over a network by over 30% compared to VNC.

3.2.1.2 Utilize Resources from Nearby Devices

Satyanarayanan et al. [10] Allows users to utilize close computers to gain the resources of cloud computing by instantiates a "cloudlet" that rapidly creates virtual machines on nearby infrastructure that can be accessed through WLAN. The author describes cloudlet as a trusted, cluster of computers that are connected to the internet and is ready for use by nearby mobile devices to overcome the limitation of their mobile devices by utilizing the cloudlet resource. The smartphone device acts as a thin client with respect to the service. This approach needs a cloudlet to be available around, if no cloudlet is available nearby, the mobile device can gracefully degrade to a fallback mode that involves a distant cloud or, in the worst case, solely its own resources. Full functionality and performance can return later when a nearby cloudlet is discovered.

The same approach has been proposed by [11] the author describes a possible way to use ad-hob networked collection of smartphones in a more constructive way. The proposed framework for this emulates a cloud provider using nearby mobile devices, so avert any external connections to a computer infrastructure based cloud. As smartphone has limitations in its resource, and these resources are mostly available for own usage by the smartphone's owner. The authors suggested a way to provide an external resource via the virtual cloud, by bringing together a collection of mobile devices to support large-scale distributed applications, using smartphones as the basis for a cloud computing infrastructure. The results from the paper suggest improvements in energy saving when compared with a single mobile device because it requires less time for the same computation no speedup factors are presented. As Hadoop is mainly designed for deployment on many Wired-Ethernet connected Servers, techniques, and tools used on big clusters cannot be directly ported to mobile clouds, due to several of limitations and attributes of mobile devices and mobile ad-hoc networks They have mentioned some Hadoop issues in creating an ad-hoc cloud using mobile devices, Hadoop suffers from low performance with small files since it is programmed to create a new JVM (Java Virtual Machine) per each map processing. Job creation process in Hadoop is heavy as it involves a lot of things. Had the input data being large, then we would see Hadoop doing better overall. Otherwise, the overhead of starting the cluster could be big enough to get better run-times.

3.2.1.3 Securing Mobile Cloud Computing Communications

Huang et al. [12] proposed an architecture called "MobiCloud", which enhance the function of the ad hoc network by treating mobile devices as service nodes. MobiCloud transforms each mobile node from the structured communication node to Service Node SN. Each SN is mirrored to one or more Extended Semi-Shadow Images (ESSIs) in the cloud. The ESSIs create a virtualized mobile ad hoc Networks (MANET) routing and maximize the availability of pervasive computing services for each mobile user. Many researchers proposed security mechanisms, but the existing mechanisms could not perform well due to issues like security, computation overhead, and poor accuracy. To overcome these problems, an IdM framework is designed to authenticate and authorize user's data access in the MobiCloud environment. The proposed IdM framework incorporates three mechanisms, namely, Key Based Mutual Authentication (KBMA), Pattern Based, User Authentication (PBUA) and Optimized Role-Based User Authorization (ORBUA). These three mechanisms are proposed to prevent the man-in-the-middle attack, masquerading attack, password guessing attack and shoulder surfing attack. Operation delay is the real-time issue with respect to the performance of MobiCloud. The applications that run in this cloud are to be designed carefully keeping the framework in mind, and thus they can have the desired features. The key focus of MobiCloud is to provide a security service architecture and they present 'Virtual Trusted and Provisioning Domain' (VTaPD), which is a service to handle information flows in various security domains, using programmable routing.

Currently, mobile security solutions mirror the traditional desktop model in which they run detection services on the device, this approach is complex and resource intensive in both computation and power. So Oberheide et al. [13] has suggested a way where whereby mobile antivirus functionality is moved to an off-device network service employing multiple virtualized malware detection engines. The core of this approach is expending bandwidth to reduce on-device

CPU andmemory resources and thereby save power. Security can also be taken care of by using cloud computing concept. Running an anti-virus application on mobile takes a toll on its resources. Their architecture involves a host agent which runs on mobile devices and sends files to the network for analysis, and network service which has multiple detection engines scans the file and sends back the result.

3.2.2 Mobile Application Virtualization

3.2.2.1 Offloading to cloud

Chun [14] designed and implemented CloneCloud, a system that transforms mobile applications to the cloud. The system divides and offloads part of the mobile application execution running in an application level to clones functioning in a compute cloud, and keeps the small workload to be executed locally on the device. The CloneCloud uses static analysis and dynamic profiling to a portioning application. The static analyzer determines which partition will be migrated to be executed on the cloud and which partition will be executed locally based on if the partition needs to recall resources in a mobile device will not migrate. The cost of the migration is registered by dynamic profiling. CloneCloud is limited in some respects by its inability to migrate native state and to export unique native resources remotely. Conceptually, if one were to migrate at a point in the execution in which a thread is executing native code, or has native heap state, the Migrator would have to collect such native context for transfer as well.

While [15] Presents MAUI, a system that allows fine-grained code offload to the infrastructure rather than depending heavily on a programmer to partition an application, or migrating full process (full VM). The purpose of this is to maximize energy saving and minimize the programmer load. The MAUI fulfil these advantages by creating two versions of a mobile application, one of them runs on the mobile phone and the other runs in the infrastructure as well as by recognizing the portability methods and extract only the program state needed by those methods, and by using serialization to determine its network shipping and CPU costs to construct a linear programming formulation of the code offload problem. The MAUI Offloading decision relies on three factors the smartphone device energy consumption features, the application features such as run time and the resources needed by individual methods and the network features of the wireless environment such as the bandwidth, latency, and packet loss, MAUI is currently designed only to support applications written for the Microsoft .NET Common Language Runtime (CLR).

[21] Proposed a framework called ThinkAir, that address scalability issues by adjusting the online method of offloading and mobile cloud improves the power of mobile cloud computing by parallelizing the method execution using multiple virtual machine (VM) images. Using parallelization, they were able to reduce execution time and battery consumption. They also used past data to achieve energy conservation and prioritize energy usage. Like MAUI, ThinkAir uses code annotation. A part of the code is to be considered for remote execution, a programmer simply annotates it with @Remote. This gives the developer more control over what he wants or does not want to send for remote execution.

As well as [16] proposed an approach that minimizes over the head by portioning the offloading among mobile devices and cloud nodes, and allow this migration to occur at the application level. This approach avoids frequently checking migration state on the application level after every function return, so it differs from other approaches that it does not enforce major overhead on execution while there is no offloading occurred. Additionally, their proposed approach migrates only the top portion of the runtime stack between mobile devices and cloud nodes, instead of migrating the entire processes or threads. This design takes advantage of the temporal location of stack-based execution, in which the most recent execution state always sits on the top segment of a stack, by partial stack migration this approach can decrease the migration cost.

3.2.2.2 Offloading to nearby Devices

[17] This paper examines the ability to offload mobile computation and transfer the user interface to more powerful devices available around. The author claims that people prefer to work "on the go", so they can take their mobile devices to public locations such as libraries and offload their applications to devices available around like personal computers, smart-TV or projectors to free them from the limitations of their mobile devices. They formulate a stochastic optimization problem to optimize the allocation of user applications to equipment despite. The scenario examines an extensive case where users can migrate application between locations.

4. ADVANTAGES AND DISADVANTAGES

In the table below, we mentioned the advantages and disadvantages of some of the mobile virtualization approaches mentioned in this paper.

Table 1. Advantages and Disadvantages of some Mobile Virtualization Approaches

Virtualization within the Mobile device			
	Advantage	***Disadvantage***	***Year***
MobiVMM	Light-weight with minimal memory footprint. It monitors the battery condition and will prompt the user to shut down some functionalities in low battery, so that the core services keep running.	Support only Linux as a guest OS.	***2008***
Xen	It has very high performance and low context switching. This architecture provides maximum security to the mobile device.	Overlapping memory scheme was devised to eliminate the TLB flushes with domain ID tagged TLB entries. However, non-overlapping memory partitions offer inflexible design of virtualized memory systems.	***2008***
Cells	Cells is a lightweight virtualization architecture which operates on all the open source platforms and It has high compatibility and scalability.	The TCB necessary for ensuring security is potentially larger than a bare metal hypervisor. High Virtual Phone switching overhead, due to remapping and frame buffer copying.	***2011***
Mobile Virtualization using Cloud Computing			
Cloudlet	This technology can help mobile users overcome the limitations of CC due to wide area network latency and low bandwidth.	Trust and security for cloudlet are other issues in implementing this idea because adversaries can create a fake cloudlet to steal the user's information.	***2009***
Virtual Smartphone over IP	The architecture provides a viable solution to data leakage problem the actual data never leaves the secure data center.	The latency of sending sensor data to the cloud wouldn't make an app very responsive.	***2010***
MobiCloud	IdM framework is designed to authenticate and authorize user's data access in the MobiCloud environment.	Operation delay is the real-time issue with respect to the performance of MobiCloud.	***2010***
MAUI	MAUI can maximize the potential for energy savings through the fine-grained code offloading while minimizing the changes required for applications.	MAUI allows a fine-grained offloading mechanism on the level of single methods, experiments result shows that several methods should be combined to achieve benefits	***2010***
Clonecloud	Have a 'cost model' that analyses the cost involved in migration and execution in the cloud and compares the cost against a monolithic execution.	Inability to migrate native state and to export unique native resources remotely.	***2011***
Peek	npression scheme used is optimized for mobile device operating conditions such as the application demands on CPU load, memory usage, and available network bandwidth.	Increasing the number of stored key frames results in better compression. However, this reduction is not prominent and considerable benefits can be achieved by using just 2 key frames.	***2016***

5. CONCLUSIONAND FUTURE WORK

In this paper, we presented a detailed discussion of various approaches, architectures, and techniques of mobile virtualization using cloud computing, nearby devices, virtualization within mobile devices and mobile application virtualization. Also, we highlighted the benefits and drawbacks of those approaches and techniques. In the future, we aim to utilize and merge some of the existing mobile virtualization approaches into one to phase the performance of mobile cloud computing to a higher level.

References

[1] BhuvneshPurohit, Tushar Sharma, ShreyanshJarged,“Virtualization Techniques In Cloud Computing”, in Proceedings with Imperial Journal of Interdisciplinary Research (IJIR), 2016.

[2] Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC Load Balancing in Cloud Computing Rajwinder Kaur1 and Pawan Luthra2 1 SBS State Technical Campus/M.tech ,CSE,Student,Ferozepur, India Email: rajwindersandhu40@gmail.com 2 SBS State Technical Campus/M.tech, CSE, Assistant Professor, Ferozepur, India Email: pawanluthra81 @gmail.com.

[3] C. Yang, K. Wang, H. Cheng, C. Kuo et al., “Green power management with dynamic resource allocation for cloud virtual machine,” in Proc. HPCC, IEEE, 2011, pp. 726-733.

[4] Xu, Y., Bruns, F., Gonzalez, E., Traboulsi, S., Mott, K. and Bilgic, A. (2010), Performance evaluation of para-virtualization on modern mobile phone platform, in `Proceedings of the International Conference on Computer, Electrical, and Systems Science, and Engineering'.

[5] Jeremy Andrus, ChristofferDall, Alexander Van’t Hof, Oren Laadan, and Jason Nieh. "Cells: A Virtual Mobile Smartphone Architecture." Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP 2011). Portugal, Cascais. 173–87. Web.

[6] Yoo, S., Liu, Y., Hong, C.H., Yoo, C., Zhang, Y.: MobiVMM, A Virtual Machine Monitor for Mobile Phones. In: Proceeding MobiVirt Proceedings of the First Workshop on Virtualization in Mobile Computing. ACM, New York (2008).

[7] J.-Y. Hwnag et al., “Xen on ARM: System Virtualization Using Xen Hypervisor for ARM-Based Secure Mobile Phones,” *IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, Jan. 10–12, 2008, pp. 257–261. J.-Y. Hwnag . “Xen on ARM: System Virtualization Using Xen Hypervisor for ARM-Based Secure Mobile Phones,”.In : IEEE Consum. Commun. Netw. Conf.; Las Vegas, NV, USA. Jan. 10–12, 2008; p. 257–261. 10.1109/ccnc08.2007.64.

[8] Chen, Eric Y., and MistutakaItoh. "Virtual smartphone over IP." World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a. IEEE, 2010.

[9] Moravapalle, Sivakumar “Peek: A Mobile-to-Mobile Remote Computing Protocol For Smartphones And Tablets” ,2016 International Conference on Computing, Networking and Communications, Mobile Computing and Vehicle Communications.

[10] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, “The Case for VM-based Cloudlets in Mobile Computing,” IEEE Pervasive Computing, 2009.(The Case for VM-based Cloudlets in Mobile Computing).

[11] Barca, Cucu, “A Virtual Cloud Computing Provider for Mobile Devices ”ECAI 2016 - International Conference – 8th Edition Electronics, Computers and Artificial Intelligence 30 June -02 July, 2016, Ploiesti, ROMÂNIA.

[12] D. Huang, X. Zhang, M. Kang, and J. Luo, “Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication,” in Proceedings of 5th IEEE International Symposium on ServiceOriented System Engineering, 2010.

[13] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, “Virtualized in-cloud security services for mobile devices,” in Proceedings of the First Workshop on Virtualization in Mobile Computing. ACM, 2008, pp. 31–35.

[14] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: Elastic execution between mobile device and cloud,” in Proceedings of the sixth conference on Computer systems. ACM, 2011, pp. 301–314.

[15] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl, Maui:making smartphones last longer with code offload, Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys’10, ACM, New York, NY, USA, 2010, pp. 49–62.

[16] R. Ma and C.-L. Wang, “Lightweight application-level task migration for mobile cloud computing,” in Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on, March 2012, pp. 550–557.

[17] J. Chase, D. Niyato, and S. Chaisiri, "Bring-Your-Own-Application (BYOA): Optimal stochastic application migration in mobile cloud computing," in *Proceedings of IEEE GLOBECOM*, San Diego, CA, USA, 6-10 December

[18] M. Satyanarayanan, 1996. "Fundamental challenges in mobile computing," in Proceedings of the 5th annual ACM symposium on Principles of distributed computing, pp. 1-7

[19] White Paper. Mobile Cloud Computing Solution Brief. AEPONA, 2010.

[20] JyotiprakashSahoo, SubasishMohapatra, Radha Lath, "Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues," Computer and Network Technology, International Conference on, pp. 222-226, 2010 Second International Conference on Computer and Network Technology, 2010

[21] Kosta, Sokol, AndriusAucinas, Pan Hui, Richard Mortier, and Xinwen Zhang: Thinkair: Dynamic resource allocation and parallel execution.