Special Issue - 2015
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
NCETEMS-2015 Conference Proceedings

Android Based Cloud Server Services

Deepak Gupta[1], Satish Kumar[3]
[1,3]Department of Electronics & Communication Engineering,
Ganga Institute of Technology and Management, Kablana, Jhajjar, Haryana, India

Anshu Aggarwal[2]
[2]Department of Computer Science & Engineering,
Al-Falah School of Engineering & Technology, Faridabad, Haryana, India

***Abstract*— Smartphone users and mobile applications are growing rapidly. Though smart phones are expected to have PC-like functionality, hardware resources such as CPUs, memory and batteries are still limited. To solve this resource problem, many researchers have proposed architectures to use server resources in the cloud for mobile devices. The system proposed conceptual architecture of development of android cloud as a service, which enables multiple user Android applications on cloud server via network. Android is an open-source mobile OS initiated by Google. Android is open-source product and runs on an x86 CPU.Android is mainly designed for physical Smartphone; Android's features are useful to construct a server platform.**

***Keywords*:- *Server*, Android,Multi-tenant, Cloud, Cloud Server**

I. INTRODUCTION

According to a recent report, 45 million people in the U.S. own Smartphone's and 234 million people. The number of Smartphone users and mobile application are growing rapidly subscribe to the mobile phone application stores [1]. There are several mobile Operating Systems (OSs), such as Symbian, iOS, Android, and Windows Mobile. Because thousands of application developers construct many kinds of applications for these platforms, users can easily enjoy their individual Smartphone lifestyle. The architecture for remotely using mobile application on server is called Mobile Application Platform on Cloud Server that intends to handle not only user data but also user applications in a cloud server. This approach changes the application lifecycle as follows. "Write once, run everywhere. Install once, use everywhere."Android is an open-source mobile OS initiated by Google. The main reason to select Android as a server platform is that it is able to run not only for Smartphone but also for x86 processor.Multi-tenancy, this is defined as a feature where the software running on a server provides services to many users. It is one of the important features for cloud computing. From the viewpoint of both economy and ecology, it is beneficial to share hardware resources among users. Using a mobile OS would be more effective than

using a desktop OS because the resource requirements of mobile OSs are smaller.

II. MULTI-TENANT FOR ANDROID

Architecture: Multi-tenancy is achieved through different approaches. The first approach is by running multiple users Virtual Machines in a server via a hypervisor. This approach has the advantage of application usability and maintenance. From the viewpoint of application usability, every mobile application that can run on Android-x86 is usable because each Android OS runs only one application.

The second approach implements multi-tenant function in kernel-layer. This approach changes Android OS to run multiple user applications in separate processes. This approach is similar to an ordinary thin client server running multiple user applications in a server. The main challenge is that original Android supports only one display and keypad device since Android is mainly designed to work on a Smartphone.Another approach is to create a multi-tenant function at framework-layer, similar to existing Java-based multi-tenant framework. This approach remodels Android the framework and APIs to support multiple user applications. The main challenge is how to run existing Android applications in modified framework. (figure 2.1 Different Layer Approach)

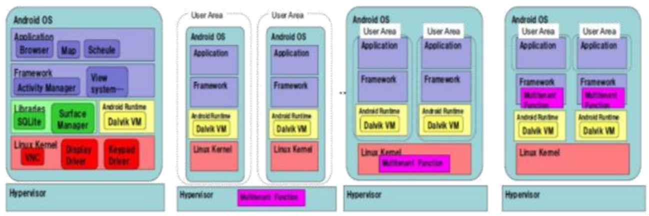

Figure 3: Overview of Android on a Server | Fig.3-(i) Hypervisor-layer approach. | Fig.3-(ii): Kernel-layer approach | Fig.3-(iii): Framework-layer approach

Fig 1

Implementation: In the functional overview of the architecture two new functions are defined for enabling multi-tenant for Android. The first function is the multiple application controller installed in an Android OS, and the second is the user area manager located in a host OS. The multiple application controllers enables running of multiple applications as if each application is running on independent physical Smartphone. It is important requirement to decrease implementation cost for Android OS because of maintenance about OS version up problem. The user area manager controls server resources and act as an interface between a terminal and the multiple application controllers.

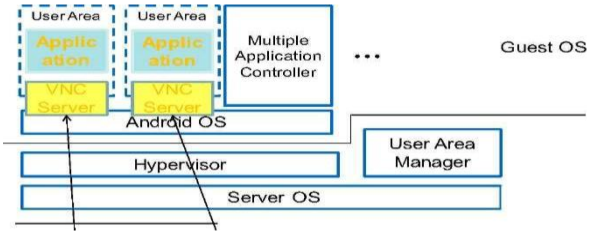

Fig. 2 Function overview of multi-tenant architecture for Android.

A common sequence of application launch is illustrated in the Figure. When user wants to use an application, the user terminal contacts the user area manager and order to launch application. The user area manager checks the server resources and select which guest OS to run application. The multiple application controllers launches the application based on an order from the user area manager. The user area manager returns VNC connection information such as IP address and port

III. INTEGRATION OF MOBILE DEVICE AND CLOUD

Researches have proposed integration between mobile devices and cloud computing. Satyanarayanan et al.[3] outlined their vision of allowing mobile users to seamlessly use nearby computers to obtain cloud-computing resources by instantiating a —cloudlet— that rapidly synthesizes virtual framework named —Ad Hoc cloud providers‖. At this framework, mobile devices can execute their jobs using other device resources around them as if it is executed on one cloud server. Our approach is closely related to that of Chun and Maniatis[5]. They proposed the creation of clone VMs to run mobile applications as if they were running on mobile devices. They recognized five categories of augmented execution to speed up mobile applications, namely Primary, Background, Mainline, Hardware, and Multiplicity, and presented a research agenda to bring the vision into reality. Their project homepage can be found in [6]. Our multi-tenant architecture for Android can be seen a specific study of Multiplicity.

IV. VIRTUAL SMARTPHONE OVER IP

In this prototype, Android-X86[11] is adopted on a mobile server OS running on a hypervisor. The client program installed on a physical Smartphone can remotely interact and control Android-x86 images. The client program transmits various events from the physical device not only the keyboard but also the touch screen and various sensors such as GPSs, accelerometer, and thermometers, to the mobile server OS and receives graphical screen updates from it via Virtual Network Computing (VNC). These programs enable to use server side virtual mobile OS applications as if it is running on a physical Smartphone. The performance evaluation using a common Smartphone and a server shows that the virtual Smartphone on a server is at least 10 times faster than on a physical Smartphone.

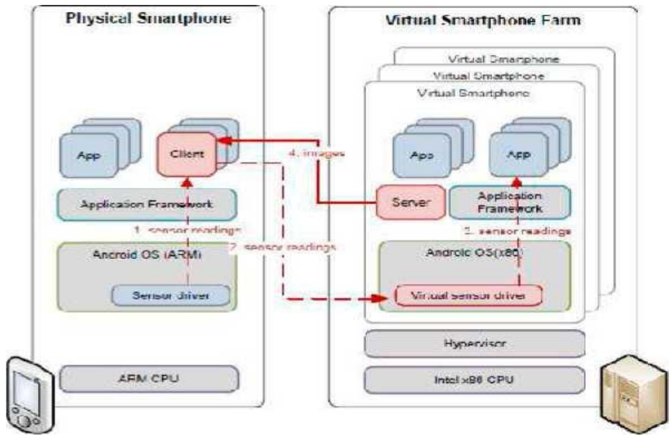

Fig 3: Prototype implementation of virtual Smartphone

Implementation

A pair of VNC-based server and client program is implemented. Server program resides in each Android -x86 image that run on top of VMWARE ESXi while the client program is installed in the physical Android device. The client program enables a user to remotely interact and control Anroid-x86 images. The client program transmits various events from the physical device to the virtual Smartphone and receives graphical screen updates from the virtual Smartphone.

A virtual sensor driver can be implemented in the Android-x86 image. Most modern Smart phone is equipped with various sensor devices such as GPS, accelerometer and thermometers. While VNC itself supports only keyboard and mouse as the primarily input devices, client program can be extended to transmit sensor readings (accelerometer, orientation, magnetic field and temperature etc) to the virtual sensor driver in the Android-x86 image. The virtual sensor driver can be implemented in such a way that the sensor readings from the physical Android device would appear to come from the Anroid-x86 images itself. 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.

V. SYSTEM OVERVIEW

Here SAAS (Software As A Service) is used, where the software can be requested and executed by multiple mobile users connected to a cloud server. The group of mobile clients share the service provided by cluster of computers connected via network

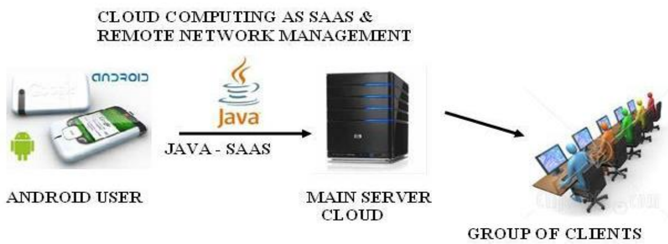

Fig.4: System Architecture

VI. PROBLEM ANALYSIS

Mobile Application Platform on Cloud Server: As a numbers of service providers such as Dropbox[12] and Zumodrive[13] provide online storage services, the architecture for remotely using mobile application on server has many benefits for users. This approach, called Mobile Application Platform on Cloud Server, intends to handle not only user data but also user applications in a cloud server [14]. This approach changes the application lifecycle as follows. —Write once, run everywhere. Install once, use everywhere.‖ Figure 2 illustrates an overview of the concept. By executing a mobile application in the cloud server, users and developers free from device limitation such as CPU power, memory, and battery, and from device software environment such as OS or version. Moreover, once a user installs an application on the cloud server, she/he can use the application anywhere,an any device.

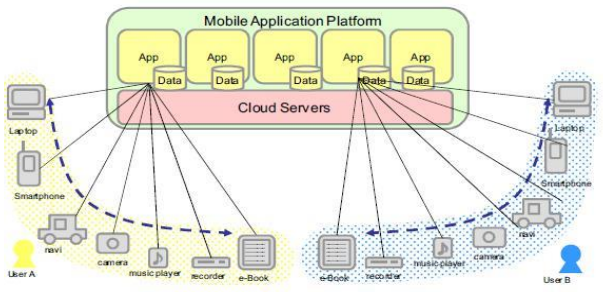

Fig 5

Multitenant for Android : Multi-tenancy, which means that software running on a server provides services to many users, is one of important features for cloud computing. From the viewpoint of both economy and ecology, it is beneficial to share hardware resources among users. Using a mobile OS would be more effective than using a desktop OS because the resource requirements of mobile OSs are smaller. However, to the best of our knowledge, there is still no service that uses Android as multi-tenant system. The proposed system discusses the multi-tenant architecture for Android and how to construct it.

Multi-tenant architecture for Android: This section discusses the process to construct multi-tenant architecture for Android based on related work. The proposed system discusses the three types of approach, hypervisor-layer, kernel-layer, and framework-layer, for multi- tenant architecture.

- Hypervisor: The hypervisor-layer approach uses the Virtual Smartphone over IP system as already stated in related work. Each user owns her/his Android OS image on a server and freely runs her/his application in a separate VM Multi-tenancy is achieved by running multiple users VMs in a server via a hypervisor.
- Kernel-layer : The second approach implements multi-tenant function in kernel-layer. This approach changes android Os to run multiple user application in separate processes. This approach is similar to an ordinary thin client server running multiple user application in a server. The main challenge is that original Android supports only one display and keypad device since android is mainly designed to work on a Smartphone.
- Framework-layer : Another approach is to create a multi-tenant function at framework-layer, similar to existing a Java-based multi-tenant framework. This approach remodels Android the framework and APIs to support multiple user application. The main challenge is how to run exiting Android application in modified framework.

VII. CONCLUSION

This paper, proposes to develop the Android based Server Platform system that enables the use of sharing server-side Android OS among multiple users. It is possible to develop a prototype system about proposed multi-tenant Android architecture. The proposed Android architecture is planning to develop a prototype system about multi-tenant. The system believes that proposed architecture shows high performance on virtual image-based virtualization for mobile application.

REFERENCES

[1] Android Phones Steal Market Share. http://www.informationweek.com/news/mobility/smart_phones/showArticle.jhtml?a rticleID=224201881.

[2] Naresh Kumar N (M.Tech), 2 Prof. Mohan K , —GCM Service Driven Communication With An Android Application In Cloud Computing‖ , ISSN: 2278-0181, May 2013.

[3] G. H-Canepa and D.Lee, —A Virtual Cloud Computing Provider for Mobile Devices‖ . San Francisco : MCS'10, 2010.

[4] B. G. Chun and P. Maniatias. —Augmented Smartphone Application through Clone Cloud Execution‖, ACM, Intel, Berkeley, Princeton, 2011.

[5] Clone Cloud Project at Intel Research,. http://berkeley.intelresearch.net/bechun/clonecloud/.

[6] Y. Royon, S. Frenot, and F. L. Mouel, —Virtualization of Service Gateways in Multi-provider Environments‖ , Heidelberg : CBSE 2006, 2006.

[7] C. P. Bezrmer and A. Zaidman, —Multi-tenant SaaS Application: Maintenance Dream or Nightmare? Antwerp‖ , Belgium : IWPSE-EVOL' 10,2010.

[8] Arsalan Shahid, Muhammad Naeem Ahmed Khan. , —Object-Relational Mapping Framework to Enable Multi-Tenancy Attributes in SaaS Application‖ , International Journal of Cloud Computing and Services Science, ISSN: 2089-3337, 2012.

[9] E. Y. Chen and M. Ito. , —Virtual Smartphone over IP. Montreal‖ , QC, Canada: IEEE WOWMOM, 2010.

[10] Android-x86 project – Run Android on your PC (Android-x86 – Porting Android to x86). http://www.android- x86.org/.

[11] Dropbox – Home – Online backup, file sync and sharing made easy. , http://www.dropbox.com/ .

[12] Zumodrive- Enjoy your media and documents from every device. , http://www.zumodrive.com/ .

[13] M. Toyama, S. Kurumatani, J. Heo, K. Terada, and Eric, Y.chen, . —Android as a Server Platfom‖ , Midori-cho, Mausashio-shi, Tokyo, 180-8585, Japan, IEEE Consumer Communication and Networking Confernence, 2011.