

Mobile Application Platform on Cloud Server

V.L.Divya⁺

Affiliated to Anna University, Anand Institute of Higher Technology

Abstract. The number of Smartphone users and mobile applications are growing rapidly. The major problem with Smartphone is that 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. This paper proposes a conceptual architecture where mobile application platform share the software as a service among multiple users on cloud server via network

Keywords: Android, Multi-tenant, Cloud

1. Introduction

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.

2. Multi-Tenant for Android

2.1. 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.

⁺ Corresponding author. Tel.: +08122215503.
E-mail address: divibiju07@gmail.com.

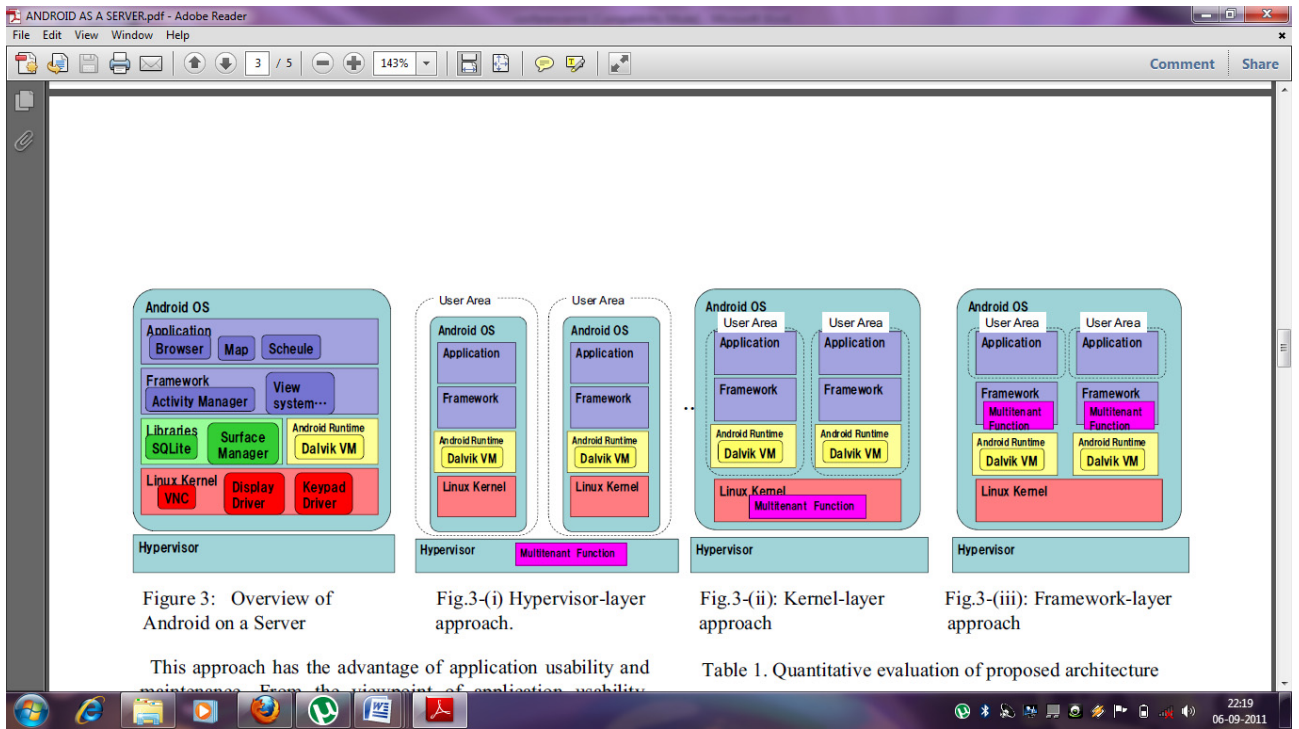


Fig 2.1: Different Layer Approach

2.2. 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 controller 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 controller.

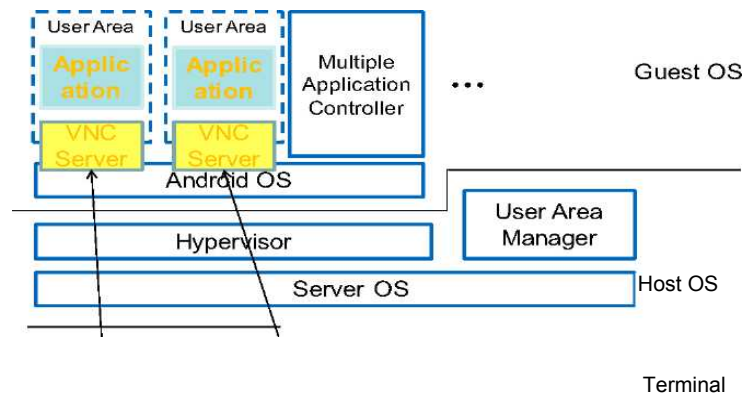


Fig. 2.2: Function overview of multi-tenant architecture for Android.

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 controller 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

3. Integration of Mobile Device and Cloud

Mobile devices are integrated with cloud computing. Mobile users can seamlessly use nearby computers to obtain cloud-computing resources by instantiating a "cloudlet" that rapidly synthesizes virtual machines on a nearby infrastructure that can be accessed through a Wireless LAN. Cloudlet can be defined as resource rich computer or cluster of computers that provide services to multiple users. The creation of clone VMs enable to run mobile applications as if they were running on mobile devices. Augmented execution is used to speed up mobile applications, namely Primary, Background, Mainline, Hardware, and Multiplicity, and presented a research agenda to bring the vision into reality.

Augmented execution is performed in four steps. Initially clone of the Smartphone is created in the single machine into a distributed execution. Second step is that state of the primary and clone is synchronised. In the next step is that application augmentation is executed in the clone on request. Finally results from the clone execution are integrated back into the Smartphone state.

4. 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.

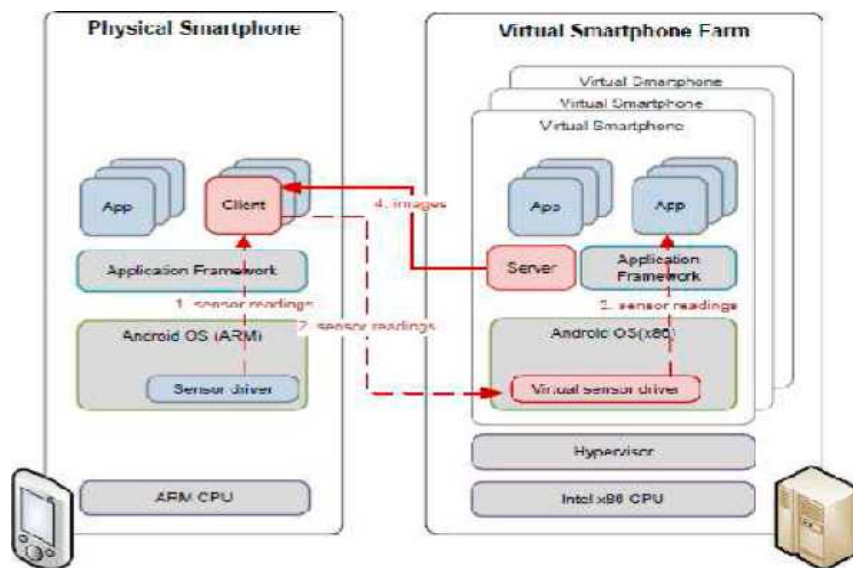


Fig .4.1: Prototype implementation of virtual Smartphone

4.1. 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.

5. 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.5.1: System Architecture

6. Conclusion

This paper, proposes Android as a 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. It is believed that proposed architecture shows high performance on virtual image-based virtualization for mobile applications.

7. Acknowledgements

I am really thankful to the Almighty. I am also thankful to my guide Mrs. Roselin Mary, HOD of Computer Science Department, AIHT, who is helping me for various researches in the project. I also convey my thanks to all the staff members of AIHT for helping me out in this project.

8. References

- [1] M. Satyanarayanan, V.Bahl, R. Caceres, and N. Davies, The Case for VM-based Cloudlets in Mobile Computing. : IEEE Pervasive Computing, 2009.
- [2] G.H-Canepa and D.Lee A Virtual Cloud Computing Privoder for Mobile Devices. San Francisco : MCS'10, 2010.
- [3] B.G. Chun and P. Maniatis. Augmented Smartphone Applications Through Clone Cloud Execution.
- [4] Y.Royon, S.Frenot, and F.L.Mouel Virtualization of Service Gateways in Multi-provider Environments. Heidelberg : CBSE 2006, 2006.
- [5] C.P.Bezemer and A.Zaidman Multi-Tenant SaaS Applications: Maintenance Dream or Nightmare? Antwerp, Belgium : IWPSE-EVOL' 10, 2010.
- [6] E.Y.chen and M.Ito. Virtual Smartphone over IP. Montreal, QC, Canada: IEEE WOWMOM, 2010.
- [7] Android-x86 Project - Run Android on Your PC(Android-x86 - Porting Android to x86). <http://www.android-x86.org/>.
- [8] Application Fundamentals | Android Developers. <http://developer.android.com/guide/topics/fundamentals.html>