

# JVM Based Cloud Computing For Android

Shruti Kulkarni

**Abstract**— The number of smartphone users and mobile application offerings are growing rapidly. A smartphone is often expected to offer PC-like functionality. In this paper, we present “JVM Based Cloud Computing for Android” system that allows users to deploy applications in the cloud system and to customize each application to meet different needs. Users can easily and freely tap into the power of the data center by installing the desired mobile applications remotely in cloud. 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.

**Index Terms**— Smartphone, Android, Virtualization, Cloud.

## 1 INTRODUCTION

THE number of Smartphone users and mobile application offerings are growing rapidly. Smartphone's are often expected to offer PC-like functionality, which requires powerful processors, abundant memory and long-lasting battery life. However, their hardware today is still very limited and application developers are forced to take these limitations into consideration.

A number of service providers such Drop box and Zumodrive provides online storage services to Smartphone users in attempt to alleviate the limitations of Smartphone storages. However, to the best of our knowledge, there is still no service that offers full computation resources to Smartphone users. In this project, we propose “JVM Based Cloud Computing for Android”, which provides cloud computing environment specifically tailored for Smartphone users. It allows users to create virtual Smartphone images in the cloud and to remotely run their mobile applications in these images as they would locally. The motivation is to allow Smartphone users to more easily tap into the power of the cloud and to free themselves from the limit of processing power, memory and battery life of a physical Smartphone. Using our system, Smartphone users can choose to install their mobile applications either locally or in the cloud.

Running applications remotely in the cloud has a number of advantages, such as avoiding untrusted applications from accessing local data, boosting computing resources, continuing to run applications on the background and opening up new ways to use smartphones.

This paper presents the design and implementation of “JVM Based Cloud Computing for Android”. Section II describes the basic design of our

this paper.<sup>1</sup>

## 2 LITERATURE SURVEY

Our project idea, ‘JVM Based Cloud Computing for Android’ comes from Drop box & Zumodrive software. Most of the knowledge about the project we got from Microsoft documents and IEEE paper of Implementation of “Virtual Smartphone Over IP”. Various algorithms related to Virtual Smartphone Over IP system were known from the PDF documents downloaded from internet.

### 2.1 Existing Systems

There are various service providers like Drop box and Zumodrive; they provide online storage of data which is easily accessible from any machine irrespective of platform as they are web based. Here, only storage is given to the user which is retrieved later. Other services like Google Doc, Gmail are implemented through cloud computing they use SAAS flavor of cloud computing.

### 2.2 Basic Idea of System

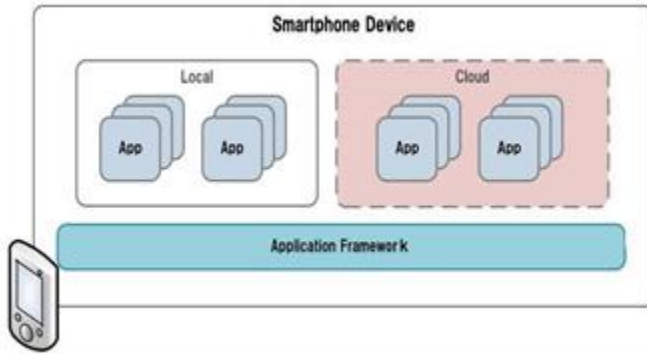
As mentioned above there are services offered through cloud computing but there are no services which offer customized/particular user applications to run on cloud, personal computers cope with this problem by local installation of applications due their high end processors, large abundant memory and constant power supply, but when smartphone are considered this is a major problem because of processor speed and memory constraints.

---

• Shruti Kulkarni is currently pursuing degree program in computer engineering in Pune University, India, PH-09860339395. E-mail: shruti.samii.kulkarni@gmail.com

system and Section III describes a proof-of-concept prototype that we have implemented. Section IV concludes

### 3 BASIC DESIGN



Our Project system adopts an architecture similar to ones commonly used by server hosting providers. As illustrated in Figure, the system is composed of a number of external smartphone clients, a front-end server, and a network file system (NFS).

The front-end server admits service requests from smartphone users across the Internet and establishes remote sessions. The front-end server also allows smartphone users to create, configure and destroy sessions.

The front-end server admits service requests from smartphone users across the Internet and establishes remote sessions to the appropriate virtual smartphone images. The front-end server also allows smartphone users to create, configure and destroy virtual smartphone images. Once a remote session is established, the user can install and run mobile applications on one of these images instead of his own physical smartphone.

The network file system is used by virtual smartphones for all persistent file storage, in much the same way that an SD card holds data for physical smartphones. Since the NFS is easily scalable, it practically provides each virtual smartphone an unlimited file storage.

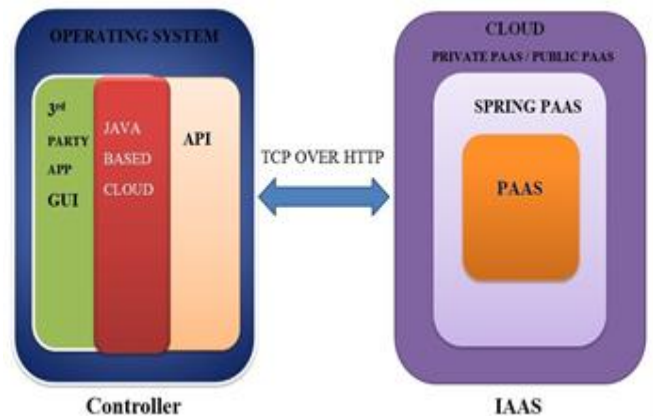
### 4 IMPLEMENTATION



Fig.1. Sample Android Phone used in Implementation  
We have implemented a proof-of-concept prototype using Android, an open-source mobile OS initiated by Google.

The main reason behind our choice is that Android OS is open source and gives more degree of freedom during implementation of system.

Our Deployment is depicted in Figure we have implemented a client and server based programs. The server program resides in Micro Cloud Foundry that runs on top of VMWARE Player while the client program is installed in the physical Android device. The client program enables a user to remotely interact and control applications. The client program transmits various events from the physical device to the virtual smartphone and receives graphical screen updates from the virtual smartphone.



Our prototype allows applications running in the cloud to appear like local applications on the physical device, Our prototype also features remote shortcuts to remote applications in the virtual smartphone that minimize the number of steps required for users to launch remote applications, Furthermore, each short-cut can point to a different virtual image, and thus allowing users instant access to remote applications.

### 5 APPLICATIONS

Our system allows users to customize each image to meet different needs. To create a new smartphone image in the cloud, the user can simply select from a number of preconfigured image templates to get up and running immediately.

The following are some examples of how our system can be used.

#### 5.1 Remote sandbox

A virtual smartphone image can be used to execute unknown mobile applications from unverified third parties. This environment is conventionally called a sandbox because applications do not run natively on the physical device and can access only a tightly controlled set of remote resources visible from the virtual smartphone image. Network and data access is heavily restricted to minimize possible negative impacts of potentially harmful applications. A sand-box is particularly useful for

Android users who would like to install the less-trusted applications obtained outside the official Android Market.

[11] "Spring "- Application development framework for enterprise Java"  
<http://www.springsource.org/>

## 5.2 Data leakage prevention

Our system can also be used as a viable solution against data leakage if the data is stored in the data centre and accessible only through one of the virtual smartphone images. Since only the graphic pixels of display images are transferred to the physical smartphone, corporate data is securely contained within the data centre and never stored in the physical smartphone. This allows employees to work with the data remotely and securely without retaining it on their devices.

## 5.3 Other possibilities

There are many other ways to use our system. It can be used to archive the less frequently used applications and free up storage space on physical smartphones. It can help users prevent their local device from accumulating unwanted residual files from trial applications. Android applications sometimes leave residual files even after they have been uninstalled by the user.

## 6 CONSTRAINTS

Our project is dependent on the network technologies if network fails operations are blocked. If cloud failure is there then whole system will go down. It is compatible with only Android smartphones.

## 7 CONCLUSION

In this paper we presented JVM based cloud computing for Android which helped to develop system which helps to optimize mobile applications and which can be used at large scale in mobile field. The prototype we implemented integrates the remote environment with the local environment and allows users to run remote applications as they would locally. Through our prototype, mobile applications installed in the cloud can access sensor readings on the physical smartphone. Our prototype also boosts the performance of mobile applications by providing virtually unlimited computing resources at user's fingertips, without draining the device battery.

## REFERENCES

- [1] "Dropbox - Home - Online backup, file sync and sharing made easy."  
<http://www.dropbox.com/>.
- [2] "ZumoDrive - Enjoy your media and documents from every device,"  
<http://www.zumodrive.com/>.
- [3] "Android Developers,"
- [4] <http://developer.android.com/index.html>.
- [5]
- [6] "Android-x 86 Projects - Run Android on Your PC  
(Android-x86 - Porting Android to x86),"
- [7] <http://www.android-x86.org/>.
- [8]
- [9] "Cloud Deployment"
- [10] <http://www.cloudfoundry.com/>