ISSN 2348 – 8034
Impact Factor- 3.155

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SURVEY OF DISKLESS WORKSTATION

Abhijeet Daga*1, Manish Ghumnani2 and Prasanna Pawar3
*1,2,3Department of Computer Engineering, Vishwakarma Institute of Information & Technology, Kondhwa (BK), Pune, University of Pune, India

ABSTRACT

Enabling disk-less/cheap Android devices (i.e. with minimal internal disk or flash memory)
By minimal disk means, your Android phone would have 2GB disk as opposed to 8/16/32GB internal storage and no external SD card slots. Many devices already don't have external SD card slot (e.g. Google Nexus devices, Apple iPads etc.). The reason, as Google explained, is that it is confusing for users [1]. Apple/Google are trying to move users to Cloud. Tablets/Phones with larger internal flash memory cost more [2, 3]

Keywords— *NFS (Network File System), DVM (Dalvik Virtual Machine), Rootfs (Root File System), JDK (Java Development Kit).*

I. INTRODUCTION

The number of Smartphone users and mobile application offerings are growing rapidly. A Smartphone is 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 these limitations need to take into Consideration. In attempt to alleviate the limitations of Smartphone storages, we are proposing cloud computing environment. It allows to create virtual Smartphone images in the cloud and to remotely run their Mobile applications. They can choose to install their mobile applications either locally or in the cloud. Mainly it frees from the limit of processing power, memory and battery life of a physical Smartphone. We are proposing cloud computing environment. Proposed system provides remote mounting of Android file system for diskless/cheap devices. Not a single system is exploring design of remote mounting of Android file system for diskless/cheap devices. This will be advantageous for high battery and high privacy. Our system also has options by which the user can perform the selection for storage. Any user will have its need as it provides high privacy, battery, etc.

II. OVERVIEW OF THE SYSTEM

This project requires you to load Android directly from Cloud. Dropbox enables Cloud storage for user data/documents. [1] However, it is currently an Application that user downloads from the App Store. This project requires you to add support for Cloud Storage in a file system - something similar to NFS, but highly efficient by enabling intelligent caching [7]. This file system will use local disk/SD card as cache for the disk in the cloud. If the data requested is not found in local cache, then a trip to the Cloud (over the network) is made to fetch that data using Wi-Fi/3G/4G LTE, whatever technology is available. The challenge is that each time a Cloud network trip is made over 3G/4G, Cellular Data Service is used and the user gets charged by their Service Provider (e.g. Airtel, Reliance etc.). This also drains battery on the device. So this file system must be highly efficient so as to save network trips by intelligently caching files locally.

ISSN 2348 – 8034
Impact Factor- 3.155

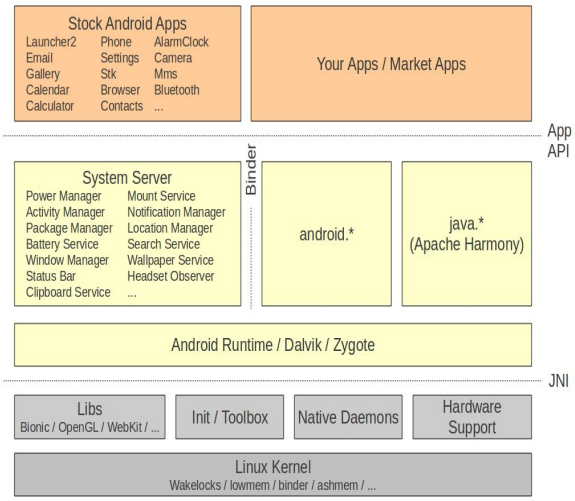

Fig.1 System Architecture

System Architecture

Architecture defines overall system. [2] Discovery module is responsible for carry out discovery of devices connected in the network. When discovery is completed, the results are stored in discovery database. This is secondary database which is compared with actual database for finding the discrepancies. Reconciliation module then takes those discrepancies & ask user to reconcile. If user performs the reconciliation then changes are updated in main database.

Linux kernel

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine Web Kit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. [6] This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

ISSN 2348 – 8034
Impact Factor- 3.155

Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications Contacts Books, Browser, Games.

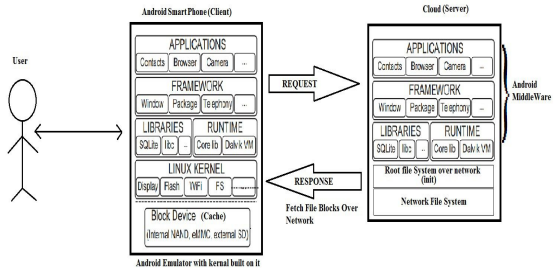

Fig 2. Proposed System

Sr.no	Title of Paper	Author	Points to discuss

The above figure shows the interaction of user through smartphone and further with cloud database.

INIT

User interacts with kernel at the time of booting of the device. Kernel get loaded at the time of booting of the device and the process INIT get called at startup. The kernel get aware about the partition since the rootfs is mounted over nfs [5]. Default file /root files will be available on device only and supporting files will be available on cloud which will be faced at time of retrieving of files.

INTERPROCESS COMMUNICATION

A special driver called "Binder" allow an efficient interposes communications (IPC)) in which allow references to objects are passed between processes. The real objects are stored in Shared Memory. This way the communication between the processes is optimized as less data must be transferred.

INTELLIGENT CACHE

The intelligent cache will keep eye on supporting files which will be faced from cloud and if retrieving of same files is most of the time then it will be remained in cache only will be fetched from cache only in order to reduce the time require to fetch files from cloud and also reduce the bandwidth and latency moron avoid the trip over the cloud [8] [9].

DALVIK VERTUAL MACHINE

Android apps are commonly written in java and compiled to byte code. They are then converted from JVM-compatible .class files to Dalvik compatible .dex (Dalvik Executable) files before installation on a device. The compact Dalvik Executable format is designed to be suitable for systems that are constrained in terms of memory and processor speed.

ISSN 2348 – 8034
Impact Factor- 3.155

1	**Clone Cloud: Elastic Execution between Mobile Device and Cloud**	ByungGonChun,Mayur Naik,AshwinPatti,Sunghwan Ihm	Partitions applications to identify what could be offloaded to cloud.
2.	**Virtual Smartphone over IP**	Eric Y. Chen Mistutaka Itoh	allows users to create virtual images of smartphones in the cloud and access these images remotely from smartphone.
3.	**A File is Not a File**	Tyler Harter, Chris Dragga, Michael Vaughn	we consider the possible effects of our findings on future file and storage systems
4.	**I/O Stack Optimization for Smartphones**	Sooman Jeong, Kisung Lee,Seongjin Le,Seoungbum Son, and Youjip Won	significant inefficiency originates from the fact that File system journals the database journaling activity.
None of the above is exploring design of remote mounting of Android file system for diskless/cheap devices. The issues that we are going to address in this project (privacy, battery, latency, etc.)			

Table.1 Literature review

III. CONCLUSION

Using the concept of mounting rootfs over nfs in Ubuntu, the same can be done for the android file system i.e. loading the android middleware on the cloud. By observing various characteristics such as latency, battery consumption a cost model can be prepared. Finally, the concept of an intelligent cache can be implemented so that the trips over cloud are minimized.

ISSN 2348 – 8034
Impact Factor- 3.155

REFERENCES

1. *"Dropbox - Home - Online backup, file sync and sharing made easy."* http://www.dropbox.com/.
2. *"Android Developers," http://developer.android.com/index.html.*
3. *Sooman Jeong1 , Ki sung Lee 2, and Youjip Won I/O Stack Optimization for Smartphone, USENIX annual Technical Conference 2013.*
4. *Eric Y. Chen and Mistutaka Itoh, Virtual smartphone over ip, NTT Information Sharing Platform Laboratories, Tokyo 180-8585, Japan 2011.*
5. *http://developer.android.com/tools/help/emulator.html*
6. *https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-12-04*
7. *http://elinux.org/Android_Architecture*
8. *Byung-Gon Chun and Petros Maniatis, Cloud based Mobile Augmentation, Intel Research Berkeley 2010.*
9. *Eduardo Cuervo and Aruna Balasubramanian MAUI - offloads stuff to the Cloud to save energy/battery, University of Massachusetts Amherst.*
10. *Mark S. Gordon and D. Anoushe Jamshidi, COMET - migrates apps to Cloud, USA, 2012.*
11. *Oumya Simanta and Kiryonh Ha, Cloudlets: Mobile Code Offload in, Hostile Environments, USA, 2007*

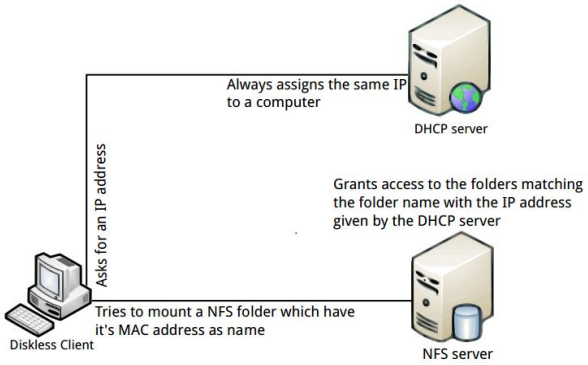

Fig. 3 A diskless workstation