ISSN:0975-9646

Joshi Rakhi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5284-5287

Enhancing Storage and Computational Performance of Android Phones Using Virtual Machine in Cloud

Joshi Rakhi, Prof. V.B. Maral

Department of Computer Engineering, KJ College of Engineering and Management Research, Pune, India

Abstract—**Worldwide Android becomes the fastest-growing mobile OS. Millions of new Android devices are activated worldwide every single day. Even so, it is the fact that Android smart phones have limited resources, such as battery charge capacity, network bandwidth utilization, storage capacity, and processor performance. These restrictions may be relieved by computation off loading: sending heavy computation to resourceful servers and receiving the results from these servers. Several issues related to offloading have been investigated in the past decade. Implementing Mobile Cloud Computing and providing Virtual Machine for Mobile on the same platform level as of Smart phone. In this paper we present offloading through implementation Virtual machine. The Mobile device will interact with Virtual Machine and will offload all the processing, storage and computations to VM. The mobile will be alleviated from all the computations and storage. The major activity on the mobile will be accessing the interface provided by the VM for interaction and results. The application state will be maintained on VM. The processing on Cloud based VM will bring heavy computational activities to the servers. Elasticity of the Cloud computing will help in dynamically growing VM's and due to Server computation capabilities will not be limited to quad-core or primary memory in one or two Gigabytes.**

Keywords: **Android, Cloud Computing and Client Server Model, Smart phones, storage manager ,Thin clients, Task manager, computation offloading, Virtual Machine**.

I. INTRODUCTION

The rapid development of multimedia and communication technology has led to increased demands for Smart phones.[1] Today's Smart phones had significant development in processing capacity and memory. Cloud computing plays a vital role in the development progress of the smart phone from user to user in a significant way. [1][2] Due to the ease of handling Android development and android based phones are in increasing demand. Thousand of applications are launched and user faces the limitation of resources on their mobile devices like battery capacity, network bandwidth utilization, large processing time for heavy data, insufficient storage space.[3] Offloading is the one of the leading technique among them. Sending computation to another machine is not a new idea now days. The client-server computing model enables mobile users to launch Web browsers, search the Internet, and shop online. Virtualization distinguishes cloud computing from the existing model. Instead of service providers managing programs running on servers like client-server, virtualization allows cloud vendors to run arbitrary applications from different customers on virtual machines. Computational and storage offloading helps to take off all the computation from user device to the server in cloud computing. Offloading boosts the energy performance of the smart phones. [4][5][6][7][8] In past decades several energy analysis studies for computation offloading in mobile devices focuses on whether to offload computation to a server or not ?[9][10][11][12]

However the most challenging scenario for Android based Smart phone is fragmentation: The major key difficulty to update the advanced Android version in mobile devices. Due to the limitation of hardware resources, there are still 2.3 (Ginger Bread) version based mobile devices in market where-as Android version 4.4 (Kit Kat) is available. This paper propose to bring VM for Android which will take out the hardware resources to Cloud computing and free the Smart phone from installing newer version of Android. Client module will be installed on user's smart phone to access the VM which will offer the version of Android and application he had chosen from remote cloud server. User will get interface to the Android version he had selected and will be installing apps on it. This will take out the computation and memory needs from Smart phone of user to Cloud.[13][14]

II. PROPOSED WORK

The figure 1 below shows the broad categorization of the modules in two components. The *thin client*: A thin client installed on Smart phone for accessing the VM on Cloud. The client process will provide rendering for the Cloud Server results and will work as interface for operations. For example, through my Android 2.3 (Gingerbread) device, once executed it will give the user a screen of Android 4.4, limitation will be the buttons provided by the Smart phone manufacturer.

The Server module: This will host a user configured VM of smart phone OS selected by User. This server process will be collection of process for handling connection failure, preparation of render-able output, Tasks management, computation job submitting, accessing memory etc. The computational operations and storage will be offloaded on Cloud Server. Thin client will provide the interface and user will interact through it with the Cloud Server. The interaction will need internet access for accessing the service offered by the Server module. User will install Thin Client on the Smart phone and configure the Server for the Android version and storage capacity.

User will get connected and full screen application interface will be brought to the Smart phone. The interaction process will read in user commands and then present it to the Server module. This will include touch commands like tap, swipe, zoom etc. The output Generation process will make the VM screen render able to the Smart phone client. Result will be the screen from VM on the Smart phone.

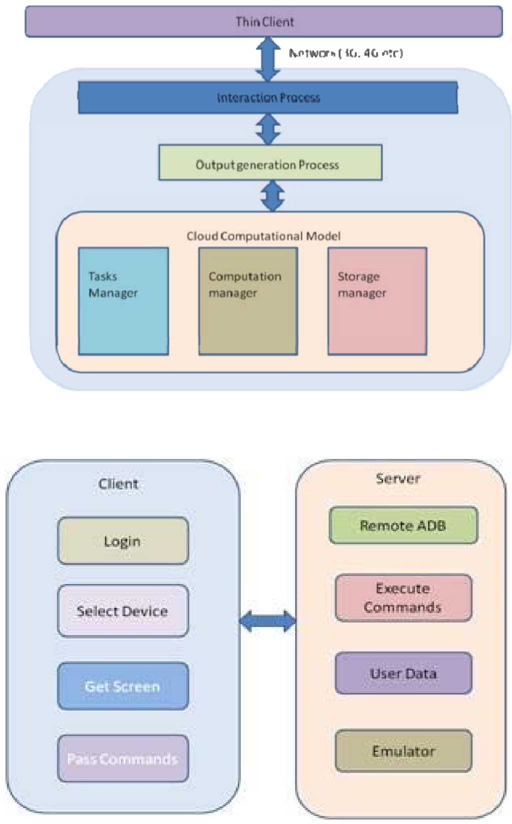

Figure1: Architecture of Cloud based Virtual Mobile

The challenges are as following:

a. Continuous connection to internet is required.
b. The hardware/predefined handset buttons will not be overridden, instead VM will offer buttons for operations.

Interaction Process:

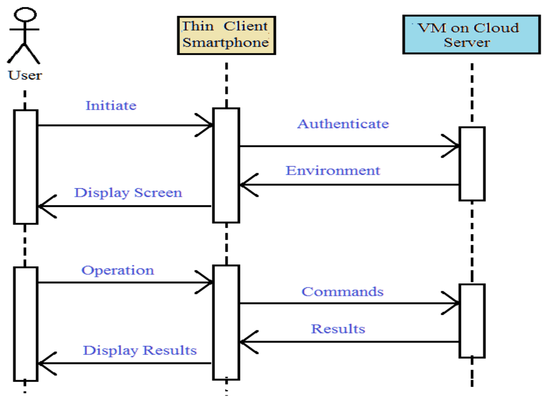

Figure 2: Interaction of User with the Application

The user will install Client application on the Android Smart phone just as any other application. User interacts with Client and remote sever through the Android Debug manager. Based on the Client ID the Server will authenticate the user and will return Android Environment set for the user. This Android environment will be display on screen of User's mobile device. The Pre-installed apps on user device will be processed on remote server. This will lead the user device free from computation and will make it more energy efficient. However the hardware buttons which are by default provided by the device will be not be overridden by the Android VM window. This limitation stands due to difference of Versions of Android and Devices. Instead a new set of buttons will be displayed on screen through VM window.

Output Generation: Output will be generated by the Client installed on Smart phone. The output for first operation in above figure will be home screen for the Android version selected. After each operation done by the User appropriate results will be displayed on screen by Thin Client.

User will be under the impression that the Android VM is installed on his device. Where-in all the operations will be happening on the VM server. On user device default buttons will not be used; instead of selected Android VM environment on server they will be appeared. This is due the limitations of hardware resources of the mobile devices.

III. Mathematical Model

This section presents Mathematical model of Android VM and communication. Objective of this Mathematical model is to demonstrate advantage of isolating Computational and Memory resource from user device.

We start with a Smart phone device (D); there exists a set of devices

$$D = \{D1, D2, D3 \dots Dn\}$$

The Resources (R) on Device D are:

$$Rd = \{P, M\}$$

Where, P = Processor and M = Memory.

At any given instance the Resources (P & M) are predefined for Device D and presented as

$$(P, M) \in D$$

There are set of Applications (A) represented as

$$A = \{A1, A2, A3 \ldots\}$$

In our project the Client (C) will be installed on the device such that combination

$$D\,(P, M) + C$$

There are set of VM (V) on Physical server (S) such as

$$S = \sum_{i=0}^{n} Vi \qquad (1)$$

And Cloud is set of such Physical Servers (S)

$$C = \sum_{k=0}^{m} Sk \qquad (2)$$

Scenario 1 without VM and client application:

As Number of Applications are installed on Devices

Memory availability decreases with number of applications.

This is represented as following:

$$\text{Avail Memory} = \frac{\text{Total Memory}}{\sum_{i=0}^{n} Mi} \qquad (3)$$

Where Mi is memory taken by Application Ai

The memory allotted to applications should not exceed the maximum capacity of Device:

D (Mi) size $\leq$ 100, with this:

$$D(Mi)size = \sum_{k=0}^{n} (100 - (\text{Application Memory})k) \qquad (4)$$

Computational efficiency decreases due to sharing between applications.

$$\text{Computation Efficiency} = \frac{\text{Computation Resource}}{\sum_{i=0}^{n} Ai} \qquad (5)$$

Scenario 2 with VM and client Application: With usage of VM and client application installed on the Device. Memory availability is on VM. With high usage of Applications a new VM can be spawned for other Application needs. When VM on physical server in Cloud gets exhausted, then using elasticity on Cloud a new physical server can take up charge.

$$\text{Resource Available} = \sum_{i=0}^{n} Si\left(\sum_{j=0}^{m} Vj\left(\sum_{k=0}^{p} Ak\right)\right) \qquad (6)$$

So, effectively the Device Memory will be in utilized nearly in constant value Ck:

$$Mi \approx \text{Client Memory} \approx Ck \qquad \textbf{(7)}$$

Computational efficiency will decrease due to sharing between applications, but new VM can be spawned. When VM are exhausted then new Physical server can be added dynamically at Cloud level. Therefore, Usage of D(P, M) of scenario 1 > Usage of D (P, M) of scenario 2.

The proposed system is developed using Android Development Kit with Java and all the results are approximates collected on Emulator and Android Device.

IV. Performance Analysis and Result

Consider a device with the following capabilities:

Device	ROM (GB)	Supported ips	Network speed (kbps)
D1	2	1500	24576

Table 1

Device D1 has 2 GB ROM with computational speed of 1500 instructions per second and network speed of 3 MBps.

Approximate data for Apps is as following Table 2:

Apps	Size (MB)	Computation (ips)	Network (kbps)
App1	**300**	**300**	**100**
App2	400	400	125
App3	200	250	200
App4	350	180	250
App5	450	200	150

Table 2

The size of the Apps is in Mega-bytes, Computation is in Instructions per second and network usage is in kbps.

Scenario 1 without VM and Client:

After installation of above Apps the device resources available will be:

Device	ROM (GB)	Suppor ted ips	Netwo rk 3G
D1	300	170	24576

Scenario 2 with VM and Client:

Using VM the apps will be installed on Cloud Server. This will leave our device with the following resources.

Device	ROM (GB)	Suppor ted ips	Netwo rk 3G
D1	1600	1400	24176

The result will be VM accessed on device. For example the image below shows Android version 4.4 Kit Kat on Android version .3 Gingerbread Devices.

V. CONCLUSION AND FUTURE WORK

We intend to bring a Virtual Mobile to smart phone for isolating the Environment, computation and storage to Cloud. This will free user from upgrading smart phone on regular basis due to memory and computational inefficiencies. Also, it will remove the current dependencies on battery limitations by isolating Computational and Memory requirements to Virtual Machine. Also, user can have multiple configurations based on his credentials to access different operating environments for mobile. User can opt for new VM when application limits exceed. Availability and fail-over mechanism will be added advantage for the user. In case of network failure the state of the user will be maintained on the Cloud server. This will help user to recover his operations from the last stage Elasticity of Cloud will help in dynamically adding servers in case of Server are exhausted of VM.

ACKNOWLEDGMENT

The authors would like to thank the management and staff of "KJ College of Engineering and Management Research" and "Trinity College of Engineering and Research" for constant support and cooperation during work.

REFERENCES

[1] Xiaoqiang Ma, Yuan Zhao, Lei Zhang, and Haiyang Wang, Limei Peng,"When Mobile Terminals Meet the Cloud: Computation Offloading as the Bridge", 0890-8044/13/$25.00 © 2013 IEEE

[2] "Cloud Computing in Smart phone: Is offloading a better-bet?",Milindkumar H. Tandel, Vijay S. Venkitachalam,CS837-F12-MW-04A

[3] "Toward a Unified Elastic Computing Platform for Smart phones with Cloud Support",Weiwen Zhang and Yonggang Wen,Jun Wu,Hui Li,0890-8044/13/$25.00 © 2013 IEEE

[4] Karthik kumar and yung-hsiang lu,"cloud computing For mobile users: can offloading computation save energy?",0018-9162/10/$26.00 © 2010 IEEE

[5] Karthik Kumar · Jibang Liu · Yung-Hsiang Lu ·,"A Survey of Computation Offloading for Mobile Systems",Mobile Netw Appl DOI 10.1007/s11036-012-0368-0,© Springer Science+Business Media, LLC 2012

[6] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa, "To offload or not to offload? the bandwidth and energy costs of mobile cloud computing," in Proc. of IEEE INFOCOM, 2013.

[7] Bart Dhoedt,and Piet Demeester,"Cloud-Based Desktop Services for Thin Clients",089-7801/12/$31.00 © 2012 IEEE.

[8] X. Gu, K. Nahrstedt, A. Messer, I. Greenberg, and D. Milojicic, "Adaptive Offloading Inference for Delivering Applications in Pervasive Computing Environments," in Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom 2003). Dallas- Fort Worth, TX, USA: IEEE, 2003, pp.107–114.

[9] Muhammad Shiraz1, Abdullah Gani2."Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines", 2012 International Conference on Information and Computer Networks (ICICN 2012) IPCSIT vol. 27 (2012) © (2012) IACSIT Press, Singapore.

[10] M. Satyanarayanan, P. Bahl, R. C´aceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[11] Antti P. Miettinen,Jukka K. Nurminen , Nokia Research Center "Energy efficiency of mobile clients in cloud computing".

[12] B.-G. Chun and P. Maniatis, "Augmented Smart phone Applications Through Clone Cloud Execution," in Proceedings of the 12th Workshop on Hot Topics in Operating Systems (HotOS XII). Monte Verita, Switzerland: USENIX, 2009.

[13] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making Smart phones Last Longer with Code Offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services (ACM MobiSys '10). San Francisco, CA, USA: ACM, 2010, pp. 49–62.

[14] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the Cloud: Enabling Mobile Phones as Interfaces to Cloud Applications,"in Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middle ware (Middle ware '09). Urbana Champaign, IL, USA: Springer, Nov. 2009, pp. 1–20

[15] J. Jing, A. S. Helal, and A. Elmagarmid, "Client-server Computing inMobile Environments," ACM Computing Surveys (CSUR), vol. 31, no. 2, pp. 117–157, 1999.