

Android Applications Offloading Using Cloud Computing

¹Harsh Bandhu Parnami, ²Mayank Arora, ³Deepika Khokhar

¹Dept. of Computer Science, Pursuing M.Tech from Indo Global College of Engineering,

Abhipur, Mohali, Punjab

²Dept. of Computer Science, Asst. Professor at Chandigarh College of Engineering
and Technology, Chandigarh

³Dept. of Computer Science and Information Technology, Asst. Professor at Indo Global College of
Engineering, Abhipur, Mohali, Punjab

Abstract

The usage of smart phones has increased rapidly over the last few years. Due to their mobility and good connectivity, smartphones are increasing thrice as compared to PC's. However they are still constrained by limited processing power, memory and battery. In this paper we propose a framework for making the applications of these smartphones autonomous enough, to offload their compute intensive parts automatically from the smart phone to the virtual image of the smartphones on the cloud thus using the unlimited resources of the cloud. By using this framework the application developers will be able to enhance the capabilities of the smart phones.

Keywords

Smartphone, offloading, cloud, android

I. Introduction

Cloud can be viewed as a great service which hosts everything may it be the data, applications or any other running programs. A cloud could be seen as an amorphous collection of computers and servers that could be accessed through the internet. Cloud computing has emerged as a great technology in terms of scalability and portability. It has changed our view of carrying data and communication. Cloud services are also very much indulging into mobile networks as most of the smartphones have the capability to support cloud computing environment.

Smartphones are mobile phones with advanced computing capability, connectivity and rich set of functionality. In a nutshell, a smartphone combines the functionalities of a phone, personal digital assistant (PDA) and a small computer. With the increasing popularity and a large number of developers developing applications for smartphone, the users of these phones have started using them for high end 3D gaming, to handle their finances i.e. internet banking and as their health and wellness managers (e.g. eat this, not that app for android). These new applications could be very resource exhaustive and the phones have a limited memory, computational power and battery life. That's why it makes good sense to offload the heavy applications to the virtual smartphone running on the cloud, thus saving the actual phone's precious resources.

A number of techniques have been proposed to offload the applications of smartphones to the cloud, including complete offloading of the applications as well as partial offloading of the applications. In these techniques used for offloading, the application is partitioned at the binary level and thus making this partitioning transparent for the application developer. But this process is compute intensive itself and to make changes at the binary level of an application needs changes in the application loader also which is difficult as well as leads to security vulnerabilities. Furthermore, in the proposed techniques an application called the application partitioner or offloader needs to be installed on the smartphone which makes the partitions and offloads the appropriate partition of other applications to the cloud. The application offloader makes the offloading decision for all the applications in the phone and

thus become an overhead on the phone's resources. In this paper we propose a framework for offloading an application partially i.e. only the compute intensive, non interactive part of an application is offloaded. The partitioning is done by the application developer and the offloading decision is taken by the application itself thus eradicating the need of making changes at the binary level and the need of application partitioner or offloader.

II. Motivation

The applications and features of smartphones are increasing day by day because the usage of these feature rich phones is increasing. People are replacing their laptops and personal computers with these smartphones, thus the demand for processing and memory is increasing. These phones use a battery as their power source which has a limited capacity as compared to plug in devices like personal computers.

Some of the major problems faced by the smartphone users now-a-days are listed as follows. Firstly, the applications using heavy graphics, memory or CPU result in a lot of battery drainage. Secondly, due to the small size of the phones the processing power, memory and battery are limited and these phones are not able to perform computer intensive tasks which our laptops or desktops could perform. The solution to these problems is either to increase the size of battery, processing power of the CPU and the size of memory which in turn results in increased size and cost of the phone or to use the resources of the cloud to execute the heavy applications thus saving the phone's scarce resources.

Cloud computing on the other hand provides computing resources (hardware and software) as a service through internet.

The major motivation for this paper is to use the computing resources provided as a service by the cloud to run the resource exhaustive applications of the smartphones connected to the cloud through internet. In the past couple of years some techniques have been proposed to partially or completely offload the applications onto the cloud. We will be discussing those techniques in the next section and the challenges faced by the offloading techniques.

III. Review of Proposed Architecture in The Related Research

Quite a few approaches have been proposed for offloading applications from a smartphone to the cloud which includes offloading the complete application, offloading an application partially. Related work in the field of offloading applications from android phones to the cloud have been discussed below.

In 1998 Alexey Rudenko et al. proposed a scheme to enhance the battery life of a laptop through remote processing of power costly tasks. They proposed that the battery life of a laptop could be increased by shifting the power costly tasks onto a server through wireless connectivity of the internet. Thus powerful server will perform the tasks as required and send back the results to the laptop saving the laptop from processing the task itself and in the meanwhile the laptop will keep on performing the less power costly tasks. This research gave birth to a new idea of remote processing of power costly tasks of the smartphones using the resources of the cloud.

Year 2009 witnessed the proposal of augmented smartphone applications through clone cloud execution by Byung Gon Chun and Petros Maniatis. This research proposed to augment the smartphone's capabilities by offloading an application partially or completely to a clone smartphone. A clone is a virtual system on the cloud running the same operating system as that of the phone using hardware from the cloud's pool of hardware. The application is offloaded partially because only the part of the application which is compute intensive is to be offloaded and thus reducing the load on the smartphone. While the compute intensive part is being executed by the clone the actual phone executes the remaining application. After the clone is finished with the execution of the compute intensive part of the application it returns the result to the actual phone. The phone processes the results as required and provides the user with the results. The proposed architecture includes a controller and a replicator installed in the actual phone and an augmenter and replicator installed on the clone. The replicator synchronize the changes in the phone software and state to the clone. The controller offloads the application from the smartphone to the clone and merges back the results from the clone to the phone. The augmenter running in the clone manages the local execution, and returns a result to the actual phone.

Following the vision provided by the above research, Byong Gon Chun et al. in the year 2011 implemented an architecture named clone cloud for offloading an application partially to its clone in the cloud. This scheme uses a partition analyzer which partitions the application to be offloaded for remote execution. The partition analyzer has a static analyzer which discovers the possible migration points and the constraints for migration and a dynamic profiler to build a cost model for execution and migration. The partition analyzer helps the migration unit to migrate and re-integrate the application at the chosen points. The migration unit comprises of a migratory, node manager and a partition database. The migratory provides the part of application to be migrated to the node manager which migrates the part to the clone and an entry is made to the partition database which helps in re-integrating the partitioned application.

In the year 2011, a new approach of offloading the applications from android smartphone to the cloud was introduced by Eric Y. Chen and Mistutaka Itoh named virtual smartphone over IP. In this approach the complete application was offloaded from the android smartphone to the cloud. In this approach it was proposed

to provide cloud computing environment specially tailored for smartphone users. This architecture allows users of smartphones to create virtual smartphone images in the cloud and install and run their applications in these images remotely. The user can create a number of smartphone images using a dedicated server for each user.

In 2010 Georgios Portokalidis et al. proposed a new scheme named paranoid android to provide security to android phones by applying security checks on remote security servers that host exact replicas of the smartphones in virtual environments. As the remote servers are not constrained with battery or processing power, multiple detection schemes could be applied simultaneously. On the phone, a tracer records all information needed to accurately replay its execution. The recorded execution trace is transmitted to the cloud over an encrypted channel, where a replica of the phone is running on an emulator. On the cloud, a replayer receives the trace and faithfully replays the execution within the emulator.

Another system named MAUI was introduced in 2010 by Eduardo Cuervo et al. MAUI enables fine grained energy aware offload of mobile's code to a cloud infrastructure. MAUI uses code portability to create two versions of a smartphone application, one of which runs locally on the smartphone and the other runs remotely in the infrastructure. Managed code enables MAUI to ignore the differences in the instruction set architecture between today's mobile devices (which typically have arm-based CPUs) and servers (which typically have x86 CPUs). It uses programming reflection combined with type safety to automatically identify the remote able methods and extract only the program state needed by those methods. MAUI profiles each method of an application and uses serialization to determine its network shipping costs (i.e. the size of its state). MAUI combines the network and CPU costs with measurements of the wireless connectivity, such as its bandwidth and latency to construct linear programming formulation of the code offload problem.

IV. Challenges

The main problem which is identified in the previous architectures is that offloading a compute intensive application partially can improve the battery life of a smartphone but the offloading system will incur some overhead on the phone especially if the offloading decision is taken for a large number of applications when only a few number of applications are required to be offloaded.

The architectures discussed in the above section need to make changes in the binary of the application at the time of execution, to make it offloadable. To make changes at the binary level, the program loader has to be changed which may result in security vulnerabilities and to analyze the binary of an application could be compute intensive thus imposing overhead on the phone.

V. Proposed Architecture

In this paper we present a framework for automated offloading of compute intensive applications of android smartphones to the virtual image of the smartphone on the cloud. An offloading framework is proposed which if used by the developers of the application, will empower the application to offload its compute intensive, non interactive parts based on static analysis to the smartphone image on the cloud. The static analysis is done to make the decision making more fast and light than the previous techniques.

This framework proposes to make an application autonomous for offloading itself from the smartphone to the smartphone image

on the cloud. An application will be divided into two major parts i.e. a user interactive part which takes the inputs from the user and provides the output to the user and a compute intensive non interactive part which does the computations as shown in the figure 1. This framework will empower the application to offload its compute intensive part to the cloud via internet after analyzing the cost of offloading over the cost of running the application on the phone itself. The analysis will be done using parameters like input size and internet connectivity. By using this framework the developers will empower the applications to offload themselves without the need of some other application.

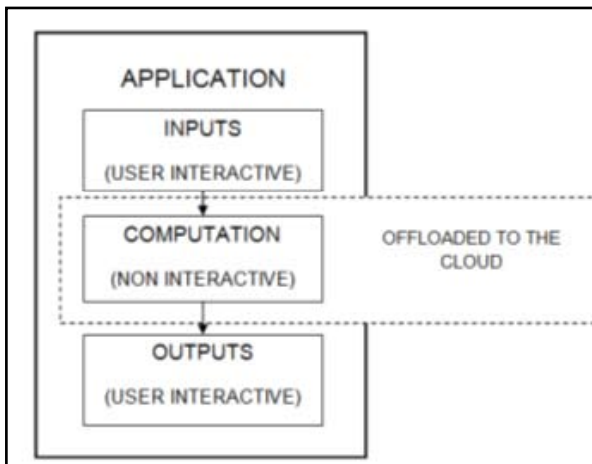


Fig. 1: Partitioning of Applications into Interactive and Non-interactive part

The figure 2 shows the working of the proposed architecture where the offloading decision is taken by the application itself after analyzing the parameters discussed above. According to the offloading decision the application is partially offloaded to the virtual phone on the cloud. And the computation is done by the cloud and thus saving the resources of the phone.

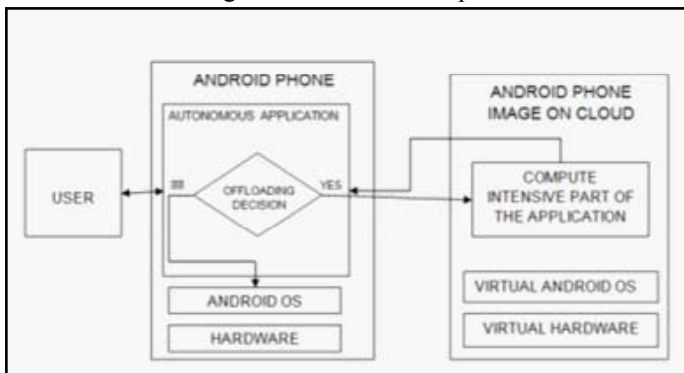


Fig. 2: Framework of Autonomous Computation Offloading Application for Android Phones.

The major advantage of using this framework is that the offloading decision will be taken by the application itself. This decision making will have to be inserted in the source code while writing the applications which is advantageous over the previous approaches in the following ways. Firstly, the approaches discussed earlier modified the applications at binary level using modifications in the loader and thus increasing the security vulnerabilities. Secondly, they analyzed each application's binary to offload it, but only a few compute intensive applications need to be offloaded thus causing an overhead on the phone's resources. By using this technique the offloading decision will be taken only for the

compute intensive applications which need to be offloaded and the application developers will not use this offloading framework for small applications. This technique will need to modify the applications at the development stage and will not modify the applications binary thus eradicating the need of changing the application loader.

VI. Conclusion And Future Work

In this paper, we explored the design of a framework which makes an application autonomous to offload its compute intensive part to the cloud thus saving the resources of the android phone. This framework makes changes in the application at the development time thus eradicating the need to make changes in the application's binary. The application will make the offloading decision using static analysis.

Our future work includes implementation and evaluation of this design, adding dynamic analysis to support the offloading decision and comparing the performance of both frameworks using static analysis and the one using dynamic analysis. We are also planning to incorporate security, privacy and trust related models in the proposed framework.

References

- [1] Health and wellness – Phone Applications University of California. <http://wellness.ucr.edu/wellness%20Apps%20Resources.pdf>
- [2] A. Rudenko, P. Reiher, G.J. Popek and G.H. Kuenning : Saving portable computer battery power through remote process execution. In *MCCR'98 – ACM SIGMOBILE Mobile Computing and Communications Review Newsletter*. Vol. 2, no. 1, 19-26(1998).
- [3] B.G.Chun, P.Maniatis : *Augmented Smartphone Applications Through Clone Cloud Execution*. In *Proceedings of 12th Conference on Hot Topics in Operating Systems*. 8-8(2009).
- [4] B. G. Chun, S. Ihm, P. Maniatis, M. Naik and A. Patti : *Clone Cloud: Elastic Execution between Mobile Device and Cloud*. In *Proceedings of 6th Conference on Computer Systems*, 301-314(2011).
- [5] E. Y. Chen and M. Itoh : *Virtual Smartphone over IP*. In *Proceedings of IEEE International Conference on World of Wireless Mobile and Multimedia Networks*, 1-6(2010).
- [6] E. Chen, S. Ogata and K. Horikava : *Offloading Android Applications to the Cloud*. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 788-793 (2012).
- [7] G. Portkalidis, P. Homburg, K. Anagnostakis and H. Bos. : *Paranoid Android : Versatile Protection for Smartphones*. In *Proceedings of the 26th Annual Computer Security Applications Conference*, 347-356 (2010).
- [8] E. Cuervo, A. Balasubramanian, D. K. Cho, A Wolman, S. Saroiu, R. Chandra and P. Bahl : *MAUI : Making Smartphones Last Longer with Code Offload*. In *Proceedings of the 8th International Conference on Mobile Systems, Applications and Services*, 49-62 (2010).
- [9] A. Saarinen, M. Siekkinen, Y. Xiao, J. K. Nurminen, M. Kemppainen and P. Hui, *Smartdiet : Offloading Popular Apps to Save Energy*. *ACM SIGCOMM Conference*, 297-298 (2012).
- [10] W. Song and X. SU.: *Review of Mobile Cloud Computing*.

- In IEEE 3rd International Conference on Communication Software and Networks, 1-4 (2011).*
- [11] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang and Q. Li : *Comparison of Several Cloud Computing Platforms. In Proceedings of IEEE International Conference on Information Science and Engineering, 23-27 (2009).*
- [12] National Institute of Science and Technology : *The NIST Definition of Cloud Computing. <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf> (2010).*
- [13] R. Buyya, J. Broberg and A. Goscinski : *Introduction to Cloud Computing. In Cloud Computing : Principles and Paradigms. Wiley Press [Online], 1-44. <http://media.johnwiley.com.au/productdata/excerpt/90/04708879/0470887990-180.pdf> (2009).*
- [14] *Android Open Source Project. Philosophy and Goals, Google. <http://source.android.com/about/philosophy.html> (2012).*
- [15] S. Singh and S. Bawa : *A Privacy, Trust and Policy based Authorization Framework for Services in Distributed Environments, International Journal of Computer Science, Vol. 2, no. 1, 85-92 (2007).*
- [16] S. Singh and S. Bawa : *A Framework for Handling Security Issues in Grid Environment using Web Services Security Specifications. In Second International Conference on Semantics, Knowledge and Grid, 2006, SKG'06, Guilin, China. 68 (2008).*
- [17] G. Singh and S. Singh : *A Comparative Study of Privacy Mechanisms and a Novel Privacy Mechanism [Short Paper]. In ICICS'09 Proceedings of the 11th International Conference on Information and Communication Security, Beijing, China. 346-358 (2009).*
- [18] S. Singh : *Trust Based Authorization Framework for Grid Services. Journal of Emerging Trends in Computing and Information Sciences. (2011). Vol. 2, no. 3, 136-144.*