Understanding the Challenges in Mobile Computation Offloading to Cloud through Experimentation

Padmaja Joshi *, Ashwin Nivangune*, Ranjan Kumar*, Sathish Kumar*,
Rakesh Ramesh*, Sushant Pani*, Arif A. Chesum*
*Centre for Development of Advanced Computing (C-DAC)
Gulmohar Cross Road No.9, Juhu, Mumbai
India

***Abstract*—With increase in network connectivity and possibility of low cost cloud resources, dynamic task offloading to cloud seems to be an ideal solution to improve performance of mobile devices along with saving on battery consumption. Mobile Cloud Computing (MCC) allows extending limited resources available on mobile devices to execute complex and rich mobile applications. This paper brings out the findings of the experiments carried out to understand the impact of application characteristics, cloud end architecture and the android emulator used, on application performance when the application is augmented to cloud.**

I. Introduction

Devices such as mobile phones and tablet computers, wearable devices such as smart-watches, etc., are the new medium through which users carry out a substantial number of daily activities. Tipped to completely replace desktop computers in the years to come in terms of utility, mobile systems lag behind the former in terms of resources. Mobile systems have limited resources such as battery life, storage capacity and processor performance. This is where the concept known as computation offloading can be used for sending heavy computational activities within large applications to resourceful servers and receiving the results from these servers, thus, executing the applications on the mobile devices rather seamlessly. Many issues related to offloading have been investigated in the past decade. This paper first provides an overview of the techniques, systems and research areas for offloading computation, and then tries to understand the applicability of these techniques with respect to the type of applications, architecture followed, number of concurrent accesses, etc.

II. Literature

A lot of work has been done to study and develop computation offloading [1], [2], [3]. A lot of work has been done on partitioning applications and various architectures [5], [6].

III. Experimentation

Figure 1 shows the architecture followed for computation offloading [4]. An offloadable application will have a device component, which has the original application along with proxies for offloadable classes, and a cloud component which has all offloadable classes. *profiler* monitors various parameters such as available memory, CPU utilization, bandwidth, etc., and based on the profiler response an *Execution Controller* makes the computation offliading decision The *Proxy* forwards a function call to the remote cloud server if the function is offloaded to the cloud else it is executed on the mobile client. This offloading can be synchronous where mobile devices waits/polls cloud-end application component till results are available or it can be asynchronous where the application continues executing other tasks and receives notification from the cloud component whenever the results are ready. The connection to the cloud end can be through GSM /WCDMA (2G/3G) network or a WIFI hotspot connected to back-end wired network.

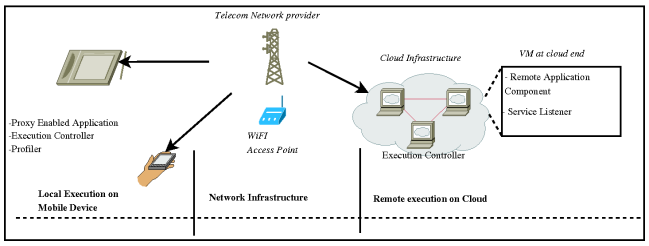

Fig. 1: Generalized Architecture for Computation Offloading

Scalability

In scalability related experiments we try to explore the impact of multiple clients trying to offload the computation. In this experimental set-up a VM is dedicated to an application and each mobile device acts as a client requesting execution on this application server. The client server architecture used in the experimentation even supports the dynamic offloading of application execution.

Application Selection

Available mobile applications are mainly client-server in nature, and hence, have fixed partitions of code. The code on server cannot run on client and vice-versa. As mobile devices are touted to be successors of desktop computers, stand-alone

978-0-7695-5566-9/15 $31.00 © 2015 IEEE
DOI 10.1109/MobileSoft.2015.43

applications can be developed for mobile platforms, which would run by utilizing dynamic offloading. Such applications are sparse in play store. Experiments were carried to identify suitability of applications for offloading based on its compute intensiveness, device interactions and coupling or interactiveness among different modules of the application.

IV. Observations

- *Emulator Impact on Offloading* - From Table I it is evident that the emulator has an impact on performance of an app when offloaded to cloud, and Genymotion provides better performance as compared to AVD, because unlike AVD, Genymotion runs on top of x86 architecture through virtual box.

TABLE I: N-Queen execution on different Emulators

N Value	AVD	Genymotion
10	1	0
11	7	0
12	39	0
13	218	4
14	611	29
15	4023	209
16	30000	1663

- *Impact of Offloading*- Application performance improves as the mobile device memory is increased. The compute intensive applications such as N-Queen, crash or take a lot of time on real or virtual mobile devices for higher values of N even with good memory such as 1 or 2 GB. However, when the compute intensive portion is offloaded to the cloud set-up, the applications are executed within seconds. Thus, it can be concluded that performance of suitable applications can be drastically improved when compute-intensive portions of the applications are offloaded to the cloud.
- *Application Suitability for Offloading* - Though compute intensive applications' performance improve with offloading, not all applications benefit from the same. The experimentation proves that the performances of applications that are GUI-intensive, have less computations, and are interactive degrade when offloaded. It is because for these applications execution time is less as compared to network latency.
- *Impact of Multiple Clients* - When multiple clients access the server at the same time, the performance has been seen to be reduced, resulting in each client taking more time than expected. As can be observed from the Table II, multiple client connections were not hampering the performance expected until the chosen hardware could not handle the requests. The performance can be considered similar to those of client-server applications.

TABLE II: Effect of Multiple Clients

N	1 Client	2 Clients	3 Clients	4 Clients	5 Clients
12	1	1	1	1	1
13	3	3	3	4	4
14	19	19	20	21	22
15	131	133	159	165	198
16	972	972	974	1184	1336

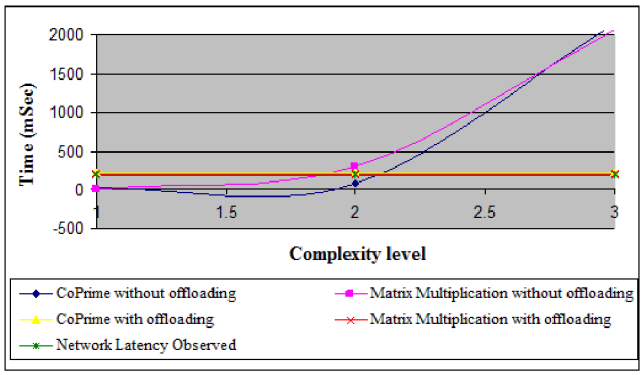

Fig. 2: Optimal Condition for Offloading

- *Offloading Performance* - With offloading, all devices should be able to give equal computational performance for an application as the same back-end infrastructure is working for all. The mobile device will act as only thin client depending upon the profilers output. With different approaches in existence for defining the cloud-end architecture, the scalability could impact the performance of applications after offloading.
- *Optimal Condition for Offloading* - Application performance degrades with offloading if the computation is less in whereas it improves drastically for higher computing requirements. Figure 2 shows how to get the optimal condition for offloading. This can be one of the input for the controller that decides "to offload or not to offload". OR The optimal condition for offloading has already been implied in Application Suitability for Offloading. This can be one of the input for the controller that decides "to offload or not to offload".

References

[1] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: Elastic execution between mobile device and cloud. In *EuroSys, Salzburg, Austria*, pages 80–86, April 2011.

[2] Byung-Gon Chun and Petros Maniatis. Augmented smartphone applications through clone cloud execution. In *HotOS 2009*, 2009.

[3] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: Making smartphones last longer with code offload. In *Proceedings of the The 8th Annual International Conference on Mobile Systems, Applications, and Services (ACM MobiSys)*, pages 49–62, 2010.

[4] Abhishek Dwivedi, Padmaja Joshi, and Abhay Kolhe. Mobile stand-alone application code off-loading: Architecture and challenges. *International Journal of Computer Applications (0975-8887)*, 94:22–27, May 2014.

[5] Jiwei Li, Kai Bu, Xuan Liu, and Bin Xiao. Fast dynamic execution offloading for efficient mobile cloud computing. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing (ACM SIGCOMM)*, pages 39–44, 2013.

[6] Sehoon Park, Youngil Choi, Qichen Chen, and Heon Y. Yeom. Some: Selective offloading for a mobile computing environment. In *IEEE International Conference on Cluster Computing*, pages 588–591, 2012.