Report-2
Saving energy on Mobiles by Computation offloading on Cloud: Are we there yet?
(Application oriented study of Offloading)

Aditya Khune

August 28, 2015

Contents

Abstract

Computation Offloading or Cyber foraging is a decade-old concept, which is today being widely considered for saving battery power in Compute Intensive Smartphone applications. Researchers say that with the right offloading decision the power consumed by CPU can be saved by offloading much of its processing either partially of fully to a Cloud. Huge amount of work is already published in this area, which shows positive results for offloading with right offloading decisions. But critiques doubt the idea of saving battery power with offloading because of various complexities involved in the process.

We believe that it is important to critically examine each aspect which is involved in overall smartphone battery problem. In this paper we have done application oriented study of Offloading in varying real-time network scenarios (3G, 4G, WiFi). We have presented both positive and negative sides of offloading solution with the help of experimental results obtained with a large set of applications on 3 different Smartphones (Amazon Fire phone, Samsung S3 and LG Nexus). While we do believe that Cloud applications have great advantages; but when it comes to energy savings our results indicate that Offloading does not provide clear benefits over local processing always and we argue that use of offloading should be restricted to a small set of applications.

We have surveyed various applications which are likely to benefit from Offloading as suggested by important publications. We have found applications mentioned in most of the publications are as follows: natural language translators, speech recognizers, optical character recognizers, image processors, image search, online games, video processing and editing, navigation, face recognition, augmented reality, etc. These applications consume large mobile battery, memory, and computational resources. Out of these we take three applications to show why offloading is a bad idea when it comes to energy savings on Smartphones. On the other hand we demonstrate specific applications which are found to be benefiting from the offloading on cloud.

Chapter 1

WHY COMPUTATION OFFLOADING IS A BAD IDEA WHEN OUR AIM IS TO SAVE BATTERY POWER ON SMARTPHONES

There are many components in smartphone which are responsible for the overall poor battery performance. CPU, GPU, LCD screen, Wi-Fi, GPS, Camera, Various sensors, speakers etc. Out of all these offloading mainly focuses on the CPU processing. In offloading we are using a network component WiFi/3G/4G while trying to compensate on CPU processing energy. With the advances in networking technology we have 4G available to us, but it is seen that 4G consumes more energy than 3G and WiFi.

- Network Inconsistency: Most important research work in offloading Decision Engines require a consistent network performance for offloading. However, such consistency is difficult to achieve because of frequent mobile user movements and unstable network quality. the power consumed by the radio interface is known to contribute a considerable fraction of the total device power. With recent advent of 4G LTE networks, there has been increased interest in the offloading domain, but Research shows that LTE is as much as 23 times less power efficient compared to WiFi, and even less power efficient than 3G [1].

- Smartphones: Most modern-age smartphones have powerful processors, up to 1 GB of memory and ample secondary storage, such users are less likely to require frequent mobile cloud support as compared to users with feature phones.

- Application: if the data size is too large and application data is unavailable in the cloud, the mobile side computation is encouraged. coz this scenario involves higher execution time and consumes high energy in terms of communication which may negate the benefits of offloading.

In the next section we have done experimentations with three types of applicatins namely: Cloud-based Web Browser, Mobile Gaming, Voice Recognition. We have used 3 real smartphones(Samsung Galaxy S4, Amazon Firephone and LG Nexus) to obtain the results on the varying network (3G, 4G and Wi-Fi) while offloading the computation and data on cloud.

1.1 EXPERIMENTS AND RESULTS

Cloud-based Web Browser:

Cloud-based Web browsers([2], [3], [4], [5]) use a split architecture where processing of a Mobile web browser is offloaded to cloud partially; It involves cloud support for most browsing functionalities such as execution of JavaScript (JS), image transcoding and compression, parsing and rendering web pages. Research shows that CB does not provide clear benefits over device-based browser (e.g. Local Processing) either in energy or download time. Offloading JS to the cloud is not always beneficial, especially when user interactivity is involved [6]. Already there are a number of cloud-based mobile web browsers that are available in the industry e.g. Amazon Silk [2], Opera Mini [4], Chrome beta [3] etc.

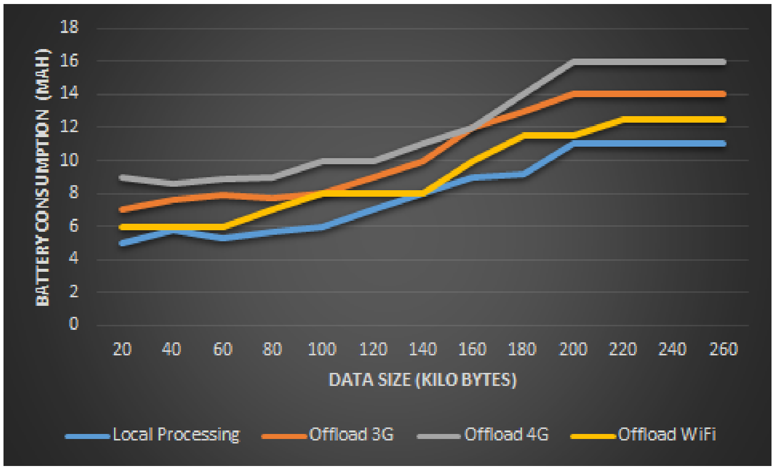

Figure 1.1: MobileBrowserDataImpact

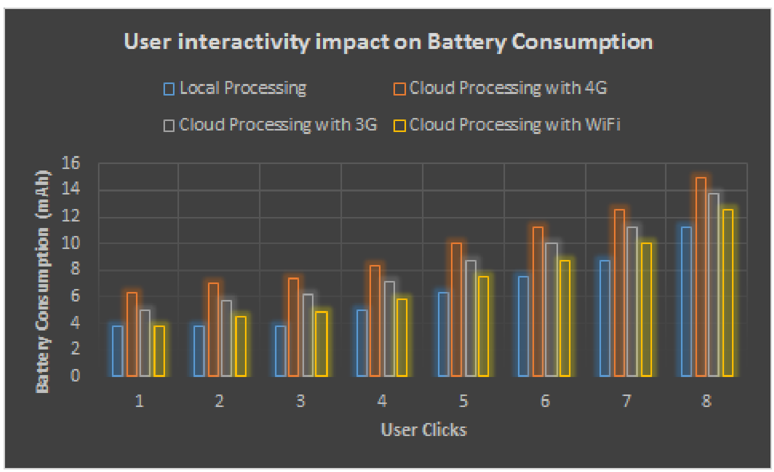

Figure 1.2: MobileBrowserUserInteractivity

Cloud-based Mobile Gaming:

Gaming-on-Demand is an emerging trend of the gaming industry which uses Cloud. Video games are executed in the private cloud servers, and the gaming video frames are transmitted over the Internet to desktop PCs, Smartphones or interactive televisions. Game players interactions are sent to the cloud server over the same network. For our experiments we have used an open-source cloud gaming platform called GamingAnywhere [gamingeverywhere] also we have done experimentations with commercial cloud gaming solutions by companies such as Gaikai [7], G-Cluster [8]. During screen-on periods, the CPU on average, is Idle for 80.0% of time, signifying that user activities do not use a lot of computational power. On average, the total CPU busy time during screen-on and screen-off periods is 10.2%

Our results show that while playing high end Video games on Mobile as much as 43% of battery will be used by the Mobile Screen and around 15% of battery by CPU. By offloading we might lower the percentage of CPU to 8-10% but then we need extra energy for the network components. So there is no significance amount of battery savings can be achieved by Cloud gaming if any. There are chances offloading may cost higher energy.

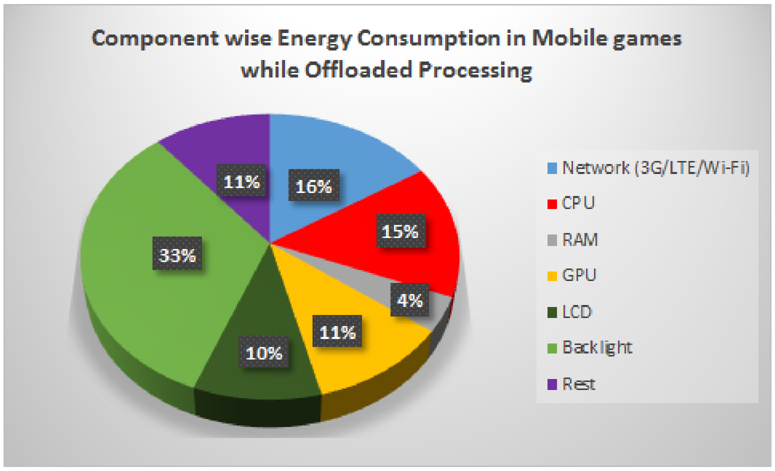

Figure 1.3: MobileGamesOffloadedProcessing

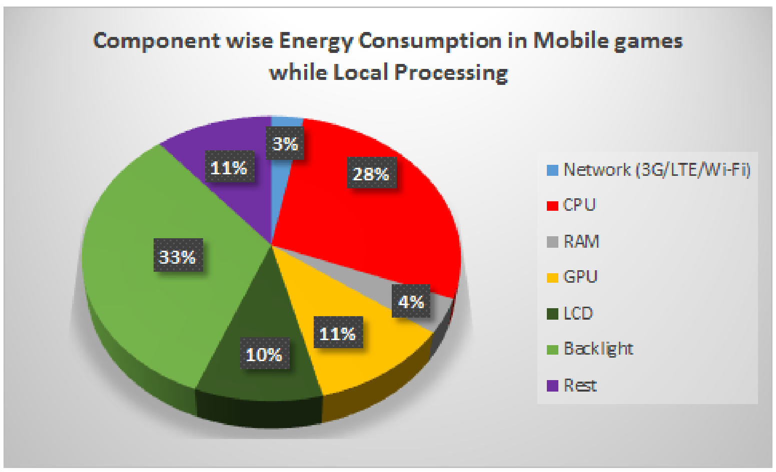

Figure 1.4: MobileGamesLocalProcessing

Voice Recognition

Results show that Voice Recognition can be done on local system efficiently. Google translate is one of the app which uses cloud to do the voice recognition. It also has an offline translation mode which does local processing on the device with a Neural Network. In the figure 1.5 we have shown the impact of increasing number of words on the voice recognition energy consumption of the device.

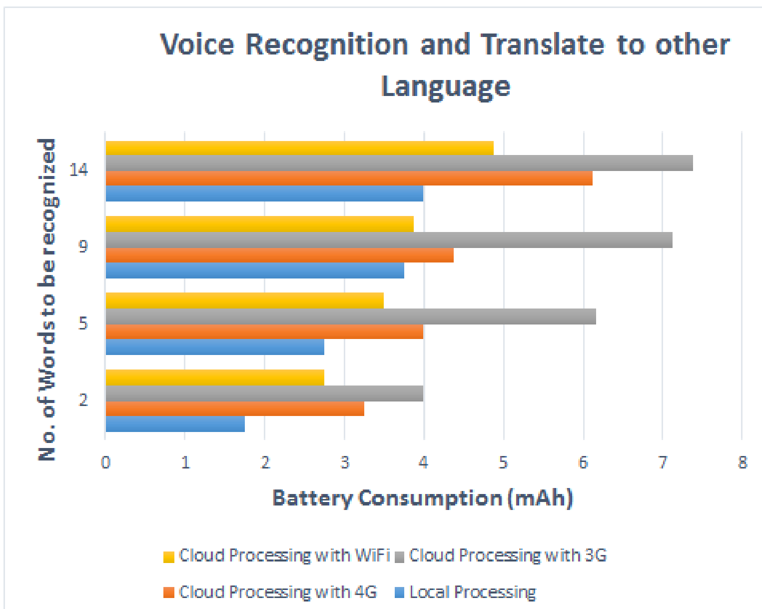

Figure 1.5: MobileVoiceRecognitionTranslateEnergy

Chapter 2

WHERE COMPUTATION OFFLOADING CAN BE USEFUL

We have studied that the Offloading is useful when an application is compute intensive at the same time not data intensive because then the data need to be transferred on cloud costing more energy to the device. To test this claim we have used offloading for Matrix calculator app which calculates inverse of a matrix. In this section we have shown how the cloud computing can benefit for the applications which may not be compute intensive but data intensive for example Torrent downloads. On the other hand offloading can be beneficial for the compute intensive and data intensive applications such as Machine Learning (for Face Recognition, Health monitoring applications). Image processing applications like face recognition require large data sets to train the learning models, which costs energy. Therefore such applications are very likely to benefit from the offloading process, when the data is already present in the cloud. Image Searching on Cloud Data is such another task which might need large computation given the huge image data collected by the users is already on the cloud, for instance Google Photos which claims to give the users unlimited image and video storage.

2.1 EXPERIMENTS AND RESULTS

Torrents:

Kelenyi et al. [9] proposed a strategy to save energy of handheld devices by offloading the download processing on the Cloud. In their strategy the cloud servers are used as a BitTorrent client to download torrent pieces on behalf of a mobile handheld device. While the cloud server downloading the torrent pieces, the mobile handheld device switch to sleep mode until the cloud finishes the torrent processes and upload the torrent file in one shot to the handheld device. This strategy saves energy of smartphones because downloading torrent pieces from torrent peers consumes more energy than downloading a one burst of pieces from the cloud.

Matrix Calculator:

This application calculates values of an Inverse Matrix. Figure 2.2 we can see the battery consumption of CPU increases as the size of Matrix increases, this is because the number of floating point operations increase. This application calculates Matrix inverse using Adjoint Method. Offloading the processing for matrix calculation on Cloud saves energy as shown in the figure.

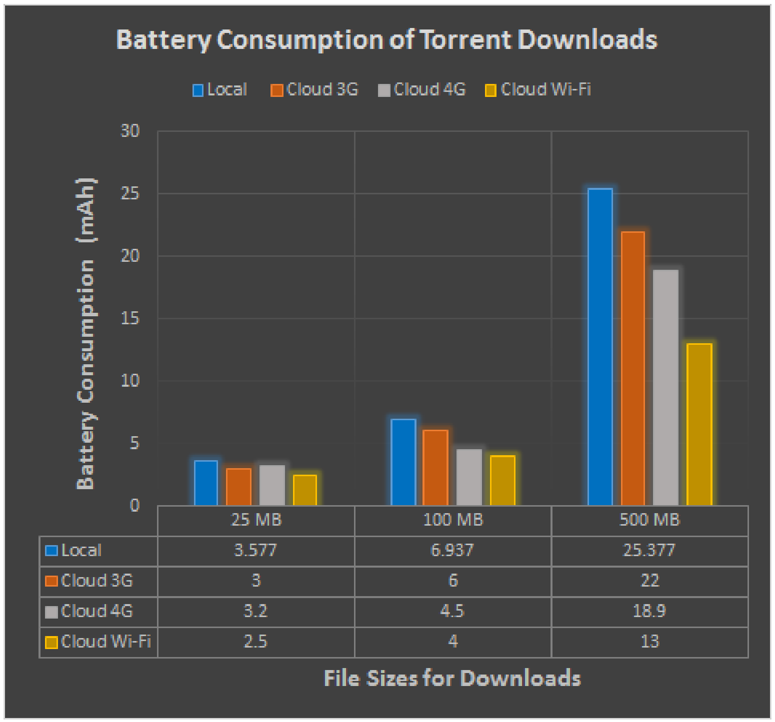

Figure 2.1: Energy Consumption for Torrent app

Zipper:

The Zipper app is used to compress the files. We can see that the data to be processed and transferred can be larger. The processing of zipping the files will be done either locally or on the cloud as directed by the Decision engines. In figure 2.3 and figure 2.4 we have given a comparison of energy consumption and Response Time while doing Local Processing and Offloaded Processing with varying file sizes.

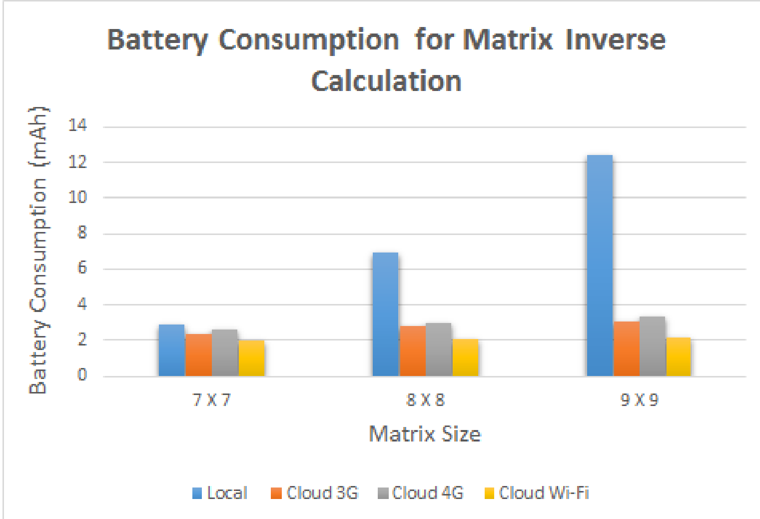

Figure 2.2: Matrix Calculator app Energy consumption

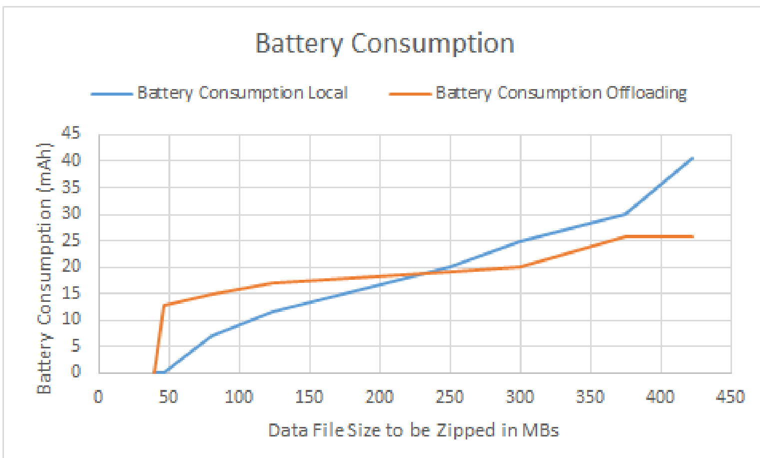

Figure 2.3: Battery Consumption for Zipper app

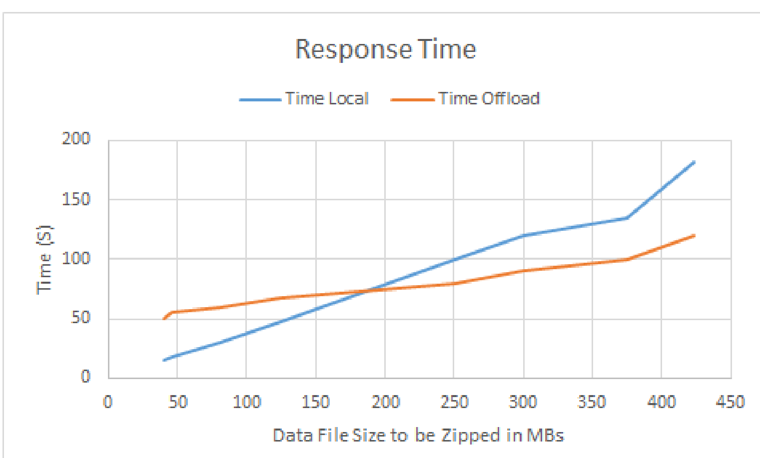

Figure 2.4: Response Time for Zipper app

Chapter 3

Conclusion

We have presented both positive and negative sides of offloading solution with the help of experimental results obtained with a large set of applications on 3 different Smartphones (Amazon Fire phone, Samsung S3 and LG Nexus). While we do believe that Cloud applications have great advantages; but when it comes to energy savings our results indicate that Offloading does not provide clear benefits over local processing always and we argue that use of offloading should be restricted to a small set of applications. We support this claim with the help of three types of applications namely: Cloud-based Web Browser, Mobile Gaming, Voice Recognition.

In the previous research works we have studied that the Offloading is useful when an application is compute intensive at the same time not data intensive because then the data need to be transferred will cost more energy to the device. In our experiments we have shown how the cloud computing can benefit for the applications which may not be compute intensive but data intensive for example Torrent downloads. On the other hand offloading can be beneficial for the compute intensive and data intensive applications such as Machine Learning (for Face Recognition, Health monitoring applications). Image processing applications like face recognition require large data sets to train the learning models, which costs energy. Therefore such applications are very likely to benefit from the offloading process, when the data is already present in the cloud. Image Searching on Cloud Data is such another task which might need large computation given the huge image data collected by the users is already on the cloud, for instance Google Photos which claims to give the users unlimited image and video storage. .

Bibliography

[1] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 225–238, ACM, 2012.

[2] "Amazon silk split browser architecture.." `https://s3.amazonaws.com/awsdocs/AmazonSilk/latest/silk-dg.pdf`.

[3] "Data compression proxy in android chrome beta.." `https://developers.google.com/chrome/mobile/docs/data-compression`.

[4] "Opera mini architecture and javascript.." `http://dev.opera.com/articles/view/opera-mini-and-javascript/`.

[5] X. S. Wang, H. Shen, and D. Wetherall, "Accelerating the mobile web with selective offloading," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pp. 45–50, ACM, 2013.

[6] A. Sivakumar, V. Gopalakrishnan, S. Lee, S. Rao, S. Sen, and O. Spatscheck, "Cloud is not a silver bullet: A case study of cloud-based mobile browsing," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, p. 21, ACM, 2014.

[7] "Gaikai." `https://www.gaikai.com/`.

[8] "G-cluster global." `http://www.gcluster.com/eng/`.

[9] I. Kelényi and J. K. Nurminen, "Cloudtorrent-energy-efficient bittorrent content sharing for mobile devices via cloud services," in *Proceedings of the 7th IEEE on Consumer Communications and Networking Conference (CCNC)*, vol. 1, 2010.