Enhanced Battery Life of Mobile Device by Computation Offloading Decision Algorithm

Mi Swe Zar Thu,Hsu Mon Kyi

University of Information Technology
Yangon, Myanmar
swezar@uit.edu.mm,hsumonkyi@uit.edu.mm

Abstract

Functionality on mobile device is ever richer in daily life. Mobile devices have limited resources like battery life, storage and processor, etc. Nowadays, Mobile Cloud Computing (MCC) bridges the gap between the limited capabilities of mobile devices and the increasing user demand of mobile applications by offloading the computational workloads from local devices to the remote cloud. Deciding to offload some computing tasks or not is a way to solve the limitations of battery life and computing capability of mobile devices. Application offloading is energy efficient only under various conditions for determining where/which code should be executed. This paper presents a Computational Offloading Decision Algorithm (CODA) , to save the battery life of mobile devices, taking into account the CPU load, state of charge, network bandwidth and transmission data size. The system can take decision which method should be offloaded or not based on different context of the mobile device to obtain minimum processing cost. Numerical study is carried out to evaluate the performance of system. Experimental result will demonstrate that the proposed algorithm can significantly reduce energy consumption of mobile device as well as execution time of application.

Keywords - Mobile Cloud Computing, Computation Offloading, Wireless Network Bandwidth, Energy, Computation Offloading Decision.

1. Introduction

In our daily life, mobile devices have become common entity. However, the battery lifetime is still a major concern of the modern mobile devices. From the users' perspective, they need better performance of their mobile devices, which reflects on longer battery life and shorter processing time of any kind of services. These mobile devices provide us with much exciting applications which require large computing power, memory, network bandwidth and energy to run applications as multimedia, GPS navigation, real time games etc. which also adds energy consumptions constantly. Energy or Battery is the only resource in mobile devices that cannot be restored immediately and needs external resources to be renewed.

Computation offloading is a way to overcome this obstacle. Many works have been done in Mobile Cloud Computing (MCC), mainly focus on the code partitioning and offloading techniques, assuming a stable network connection and sufficient bandwidth. However, the context of a mobile device, e.g. network conditions and locations, changes continuously as it moves throughout the day. To tackle the issues mentioned above and improve the service performance in mobile cloud computing, we propose a Computation Offloading Decision Algorithm (CODA) that takes the advantages of both nearby cloudlet, local mobile device cloud, and public cloud computing services in the remote to provide an adaptive and seamless mobile code offloading service. The objective of the proposed system is to take offloading decision into the account of total execution cost of each method, device profiler and network profiler to provide better performance and less battery consumption.

2. Related work

There have been many attempts to improve energy and CPU efficiency in mobile devices. These approaches enable to reduce application execution time on mobile devices and decrease the energy consumption of CPU. These attempts could be classified into two approaches: fine-grained and coarse grained tasks offloading schemes.

The first one relies on application developers to modify the code to handle partitioning, state migration, and adaption to various changes in network conditions. C.Eduardo [4] proposed Mobile Assistance Using Infrastructure (MAUI) profiler that measures the device characteristics at initialization time. It continuously monitors the program and network characteristics. Because these can often

change and a state measurement may force MAUI to make the wrong decision.

The second approach assumes that the full process/program or full Virtual Machine (VM) is migrated to the remote servers. Then programmers do not have to modify the application source code to take advantage of computation offloading. B.G.Chun [2] presented a technique known as Clone Cloud to reduce the burden of Mobile Devices. Clone Cloud is a system which automatically converts applications of the mobile devices by partially offloading it into the virtual clone (phones) present in the cloud. The author test Clone Cloud in HTC G1 device. As a result, Clone Cloud technique is helpful in reducing execution time and consumption of energy on mobile devices. A.Ellouze [1] presented Mobile Application Offloading (MAO) algorithm triggered by two conditions: the current CPU load and State of Charge (SoC) of the battery, assess its performance in terms of rejected jobs and the amount of energy savings achieved. To reduce application execution time on mobile devices, the round-trip time between the mobile terminal and the server is a key parameter that contains the level of interactivity of the applications that can be offloaded. The users and cloudlets may change their locations and become disconnected from each other. This will cause offloading failure. S M A.Karim [9] proposed an intelligent and dynamic algorithm to offload computation to the cloud focus on offloading computation based upon the communication topology, device energy and user inputs. That algorithm saves more time, compared to a previous approach, and also reduces device energy usage by moving energy hungry processes to the cloud. J.Oueis [7] discussed offloading algorithm which incorporates a multitude of parameters in the offloading decision process while reducing the mobile handset energy consumption and keeping a good user quality of experience. The purpose of this paper is to study how to deploy an offload_able application in a more optimal way, by dynamically and automatically determining which parts of the application tasks should be processed on the cloud server and which parts should be left on the mobile device to achieve a particular performance target (low energy consumption, low response time, etc.).

The remainder of this paper is organized as follows. Section 3 presents background theory of mobile cloud system. The overview of proposed system is presented in section 4. Then, proposed system architecture is explained in section 4.1 and followed by MAUI solving model is presented in section 4.2. This paper discusses Computational Offloading Decision Algorithm (CODA) in Section 4.3. After that, section 5 presents performance evaluation result. Finally, section 6 concludes the paper.

3. Background Theory

Computation offloading, as one of the main advantages of MCC, is a paradigm/solution to improve the capability of mobile services through migrating heavy computation tasks to powerful servers in clouds. Computation offloading yields saving energy for mobile devices when running intensive computational services, which typically deplete a device's battery when are locally run.

Nowadays, there has been a significant amount of research focusing on computation offloading because virtualization techniques enable cloud computing environments to remotely run services for mobile devices. These research themes are mostly related to explore ideas/ways to make computation offloading feasible, to draw optimal offloading decisions, and to develop offloading infrastructures. There are many factors that can adversely affect the efficiency of offloading techniques, especially bandwidth limitation between mobile devices and servers in cloud and the amounts of data that must be exchanged among them. Many algorithms have already been proposed to optimize offloading strategies to improve computational performance and/or save energy. These algorithms and techniques mostly analyze a few system parameters including network bandwidths, computation capability, available memory, server loads, and the amounts of exchanging data between mobile devices and cloud servers to propose offloading strategies. The real world performance benefits and battery gains are achieved by offloading certain amount of computational work from a mobile device (An android device in our case) to a cloud server. The system found a clear gain in terms of the load on the CPU of the device as well as the battery life consumption.

The concept of Mobile Computational Offloading provides a solution for the execution of resource-hungry applications. Computation Offloading occurs at the code level in which an application is partitioned or analyzed before its development. While offloading computation to a cloud server, there are two important factors to keep in mind. The first factor is the size of the computation being performed and the second factor is the amount of data that needs to be sent and received for the computation to be successful.

Mobile Cloud Computing system consists of

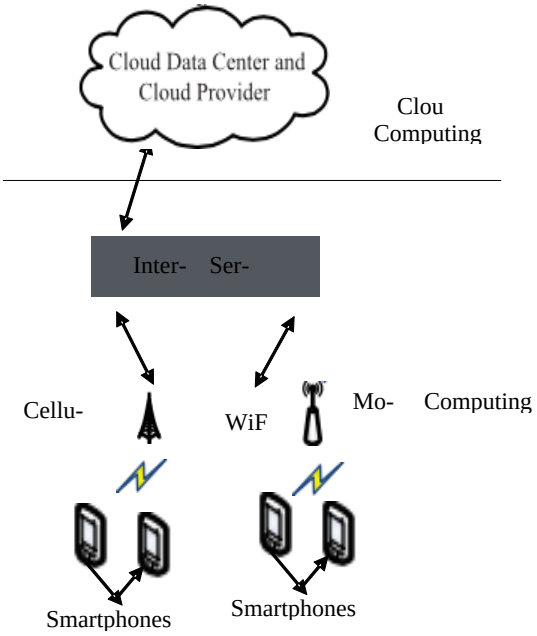

Figure 1. Mobile Cloud Computing

three parts, cloud, mobile clients, and wireless network. Cloud offers applications as services. Mobile clients access application services through proxies using wireless communication and proxies communicate with the cloud servers over fixed wired network.

4. System Overview

In this section, the components of proposed offloading system architecture are presented in detailed. This system is built on MAUI architecture to perform the offloading on method level. According to its three main components, the framework of proposed system is considered as in the following steps.

- **Offloading monitor:** is responsible for collecting real-time information on mobile devices, mobile networks and cloud servers.
- **Offloading planner:** is the decision making component of our framework. According to the collected information stored in the offloading monitor, it decides what services must be run locally and what services must be offloaded.
- **Offloading engine:** is responsible to execute mobile services based on decisions made by the offloading planner.

4.1. Proposed System Architecture: The proposed system is considered according to the conditions of the cloud and devices, such as CPU load, available memory, remaining battery power on devices, bandwidth between the cloud and devices.The system decides the offloading method of application which is run on local device or remote cloud based on above conditions. The system components design and relations between each component are described below

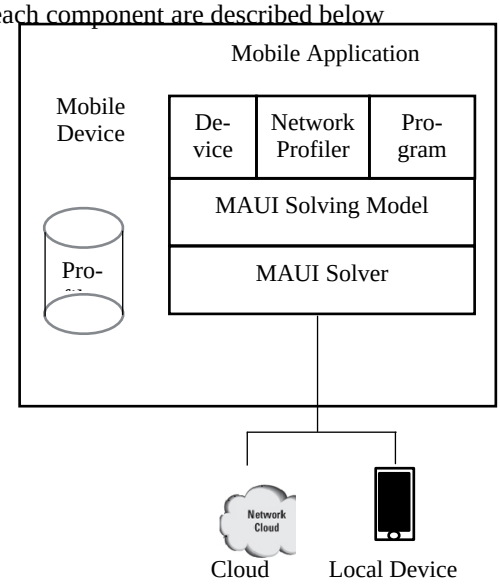

Figure 2. Proposed system architecture

- **Device Profiler:** The device profiler includes factors like energy consumption and other processing factors related to mobile devices. The profiler gathers the hardware information of device and passes it onto the MAUI solver for prediction. The factor includes 1) the average CPU usage 2) memory usage and 3) battery level.
- **Network Profiler**: The network profiler includes network properties like latency, bandwidth and so on. It also monitors: (1) cellular connection state and its bandwidth, (2) WiFi connection state **MAUI Solver:** The MAUI solver uses the informatory report sent by the profilers. Based on the data in the report, MAUI solver uses the MAUI solving model to formulate a mathematical formula which is solved to give either 0 or 1 as an answer. This answer determines the partitioning strategy for method offloading. For each method invocation, a problem is solved, and it is offloaded to the cloud if the answer comes out to be 1, otherwise it is processed locally on the mobile device.

$$S_k = \begin{cases} 0 & mobile \\ 1 & cloud \end{cases}$$

This objective function is defined for each mobile device and the offloading decision of certain parts of an application to the cloud or not, depends

on the following factors: total preprocessing time, current CPU load, State of Charge (SoC) and network bandwidth.

4.2. MAUI Solving Model: In this section, the details of MAUI solving model and the Computational Offloading Decision algorithm (CODA) are presented. Before presenting the MAUI solving model, the preprocessing step of the system is explained.

In the preprocessing step of the system, we calculate the total computing cost of application on mobile device. The decision problem is to find a solution of selecting where to execute the task and how to offload so that the overall execution time and energy consumption. Let us suppose that we have n number of methods which can be offloaded, m_1, m_2 ... m_n.

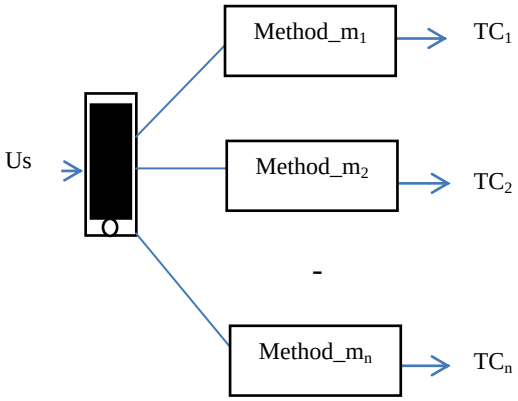

Figure 3. Preprocessing step for execution time of each method

Before presenting the MAUI solving model, the notations and symbols are described in Table 1.

Table 1. Notations and symbols used in MAUI solving model

Symbol	Description
W_n	WiFi network state
C_n	Cellular network state
SoC	State of Charge
TC	Total Execution Time of method
$m_1, m_2, \ldots m_n$	Methods of Application
m_{i_mem}	Memory usage of method i
m_{i_CPU}	CPU load of method i
m_{i_code}	Code size of method i
$E_{Transfer}$	Round-trip time on remote execution
$T_{Offload}$	Total Cost on offloading application
T_{Local}	Total Cost on local device
S	State of Offloading

The system offloads each of the methods based on several properties i.e. for specific method i, its memory cost m_{i_mem} , CPU load m_{i_CPU} and Code size m_{i_code} . Moreover, the system must be considered the offloading transfer costs for remote execution. Transfer cost includes local device to cloud side cost m_{i_send} and send back to the local device $m_{i\_reecive}$. Then, the model decides whether the method i is executed locally (m_i = 0) or remotely (m_i = 1).

The cost function is represented as follows:

(i) Round Trip Time Cost for Remote Execution

$$E_{Transfer} = m_{i\_send} + m_{i\_execution} + m_{i\_receive} \quad (1)$$

(ii) Total Cost for Offloading method m_i on Remote Cloud Side

$$T_{Offload} = \alpha_{tr} E_{Transfer} + \alpha_{mem} m_{i\_mem} + \alpha_{CPU} m_{i\_CPU} \quad (2)$$

where α_{tr} , α_{CPU} and α_{mem} the weight factors of each cost that the system can adjust the portion of the cost to different scenarios.

(iii) Total Cost for method m_i on Local Device

$$T_{Local} = \alpha_{tr} m_{i\_execution} + \alpha_{mem} m_{i\_mem} + \alpha_{CPU} m_{i\_CPU} \quad (3)$$

(iv) Minimum Cost function of the MAUI solving model

$$\min_{m_i \in 1,2\ldots,n} Cost(T_{local}, T_{Offload}) \quad (4)$$

4.3 Computational Offloading Decision Algorithm (CODA): In this section, we explain Computational Offloading Decision Algorithm (CODA) of the system as shown in below.

Algorithm 1: Computational Offloading Decision Algorithm (CODA)

Input: Set of Context parameter are WiFi network, Cellular network, State of Charge and Total execution time of method.

tasks - method m_1, m_2 ... m_n

Output: minimize energy consumption of mobile device

1: procedure GetDecision (context, tasks)
2: para[] ← context
3: task[] ← tasks
4: local cost ← estimate execution cost on client device
5: check TC
6: if TC is not valid then
7: check network state
8: else if network is Cn then
9: return decision← minCost(local, offload)
10: else if network is Wn then
12: check Soc
13: if Soc if less 20% then
14: return decision (local execution, null)

15: else
16: return decision← minCost(local, offload)
21: else
22: return decision (local execution, null)

In the proposed offloading decision algorithm, all the context parameters and profiles are collected from the system components at runtime to provide offloading decision making policies. In preprocessing step, estimate execution cost on client device. Firstly, the algorithm will check the total execution time (TC) of each method is getting from preprocessing step, if time cost is more than system's constraints the algorithm also need to check the network bandwidth. There are different types of bandwidth which is depended on user choice. As soon as the SoC of the battery gets below 20% of the total capacity of the battery the algorithm is activated to test if the current application can be offloaded. If it is not the case, the application is run on mobile.

5. Performance Evaluation

In this section, the performance of offloading scheme in MCC is demonstrated. Table 2 shows the hardware specification of mobile device. Local device is based on this specifications and cloud side simulation use the cloudSim simulator.

Table 2. Hardware specifications of mobile device

Hardware Components	Specification
Android type and OS	Samsung Galaxy S (Android 2.3)
Memory	512 MB
CPU	1 GHz Cortex-A8

In this experiment, we evaluate the performance of the system based on the CODA algorithm. The input workload size is changed from 10 MB to 50 MB.

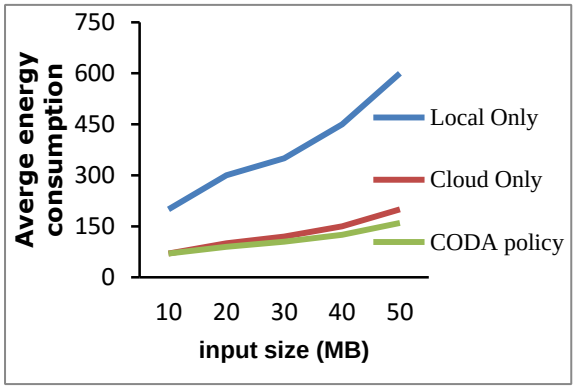

Figure 4. Average energy consumption on various workloads

The output energy consumption level is changed depending on the input size.

We apply CODA algorithm to select the best decision under the current context such as workload size, the device information, network state and execution time of each method to obtain the minimum energy consumption. In this experiment, assume that the network condition is stable. Figure 4 shows the average energy consumption while the input workload size changes. The X-axis represents input size and Y- axis shows an average energy consumption of different situation as shown in Fig.4.

6. Conclusions

The mobile cloud computing is one of the mobile technology trends in the future because it combines the advantages of both Mobile Computing and Cloud Computing, thereby providing optimal services for mobile users. A vast number of works have been done on the mobile code offloading process which is one of the vital parts of MCC. Even though it's numerous existing frameworks availability, there is much scope which is left for enhancing the offloading process to make it more feasible and attractive. In this paper, we proposed a Computation Offloading Decision Algorithm (CODA) for better performance and less battery consumption.

References

[1] A.Ellouze, M.Gagnaire, and A.Haddad,"A mobile application offloading algorithm for mobile cloud computing",, *3rd IEEE International Conference on Mobile Cloud Computing, Service and Engineering* , 2015.

[2] B.G.Chun, "Clonecloud: elastic execution between mobile device and cloud", *6th ACM conference on Computer System*, 2011, pp.301-314.

[3] B.Zhou, A.V.Dastjerdi, "A context sensitive offloading scheme for mobile cloud computing service", *8th IEEE International Conference on Cloud Computing*, 2015.

[4] C.Eduardo, "MAUI: making smartphones last longer with code offload", *8th ACM International Conference on Mobile System, Application and Services* , 2010, pp.49-62.

[5] D.Kovachev, and R.Klamma. "Framework for Computation Offloading in Mobile Cloud Computing" *International Journal of Interac-*

tive Multimedia and Artificial Intelligence, 2012, pp .6-15.

[6] F.Mehmeti, and T.Spyropoulos, "Performance analysis of mobile data offloading in heterogeneous networks", *IEEE Transaction on Mobile Computing*, vol. 16, no 2, 2017, pp. 482-497.

[7] J.Oueis, E.C.Strinati, and S.Barbarossa, "Multi-parameter decision algorithm for mobile computation offloading", *IEEE Wireless Communications and Networking Conference*, 2014, pp. 3005-3010.

[8] M.S.Z.Thu, H.M.Kyi, E.C.Htoon, "Computation Offloading Decision in Mobile CloudnComputing: Challenges of Mobile Devices", *15th International Conference on Computer Applications*, 2017, pp.23-27 .

[9] S.M.A.Karim, and John J.Prevost,l "Efficient mobile computation using the cloud.", *3rd IEEE International Conference on Future Internet of Things and Cloud* , 2015.

[10] T.Truong-Huu, C.K.Tham, and D.Niyato, "To Offload or to Wait: An Opportunistic Offloading Algorithm for Parallel Tasks in a Mobile Cloud", *6th IEEE International Conference on Cloud Computing Technology and Science*, 2014.