Computation Offloading Decision in Mobile Cloud Computing: Challenges of Mobile Devices

Mi Swe Zar Thu, Hsu Mon Kyi, Ei Chaw Htoon
University of Information Technology, Yangon, Myanmar
swezar@uit.edu.mm, hsumonkyi@uit.edu.mm, eichawhtoon@uit.edu.mm

Abstract

Functionality on mobile devices are ever richer in daily life. Mobile devices have limited resources like battery life, storage and processor, etc. Offloading some computing tasks from mobile devices to the cloud 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. This paper proposes an Enhanced Computation Offloading Algorithm, to extend the life time of mobile devices, partition the job as a graph and taking into account the CPU load, state of Charge, wireless network bandwidth, transmission data size. Based on the inputs, the system decide whether to offload the application to the cloud or not. Offloading is an effective method for extending the life time of mobile devices by executing some components of applications remotely.(E.g. processing applications on mobile cloud). Result will demonstrate that the 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;*

1. Introduction

In our daily life, mobile devices have become common entity. These mobile devices provide us with many more exciting applications which require large computing power, memory, network bandwidth and energy to run such applications. These devices' houses many sensors like accelerometer, proximity etc. used for different functions like multimedia, GPS navigation, real time games etc. which also adds energy consumptions constantly. As there is a great improvement in a battery technology, mobile devices still suffers from battery lifetime problem. 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 improve performance and save energy.

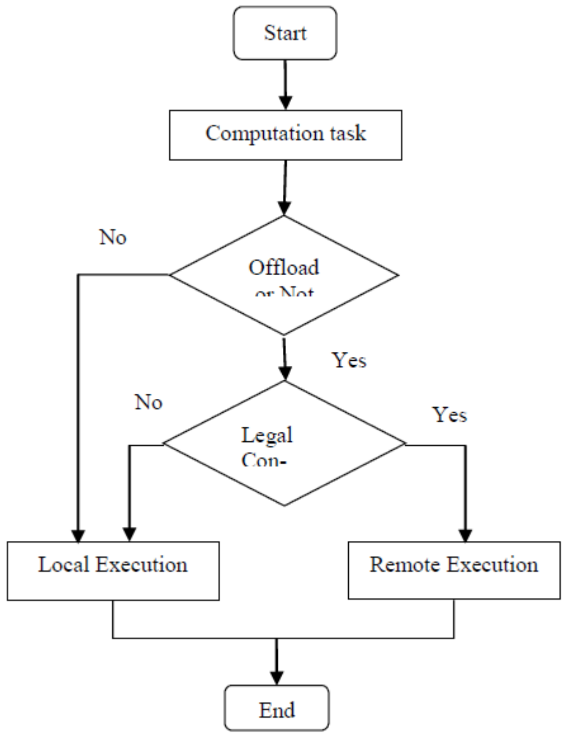

Fig. 1.1: Computation Offloading Process

There are two factors for limiting performance in existing mobile devices: the scale of integration of electronic devices and the capacity of the batteries. Devices use for different functions of application like multimedia, GPS navigation, real time games etc.

In this context, several recent studies proposed to encompass the limited capacity of the existing batteries by means of a transient offload of some of the applications from the mobile terminal to a remote server. An effective approach to alleviate the constraints of mobile devices is to offload resource-intensive/ compute intensive components of mobile applications to resource abundant/resourceful servers for processing: called Cyber Foraging / Computational Offloading. Challenges in offloading in mobile cloud environments exit. Offloading may not always achieve the lowest cost due to possible high communication and remote execution costs. Therefore, prolonging the battery life of mobile devices has become one of the top challenges.

Offload Types: Offloading can occur either at start time referred to as static offloading or at runtime called dynamic offloading. During static offloading, a middleware or programmers partition the program before execution. Thus, at expanded uniformity of network environments and surrogates, static offloading cannot ensure the best partitioning for all probable situations which could be beneficial. In contrast, dynamic offloading starts to offload tasks when the required resources for offloading is insufficient and partitions the program according to the availability of at runtime. Fig1.1 show the decision of offloading the mobile applications based on existing conditions and is therefore beneficial and more flexible.

The limitations of battery lifetime and computing capability of mobile devices can be alleviated by computation offloading, i.e., sending some heavy computation components of a mobile application to the resourceful servers on the cloud. To solve the limited of device, offloading decision algorithm is future for computation offloading.

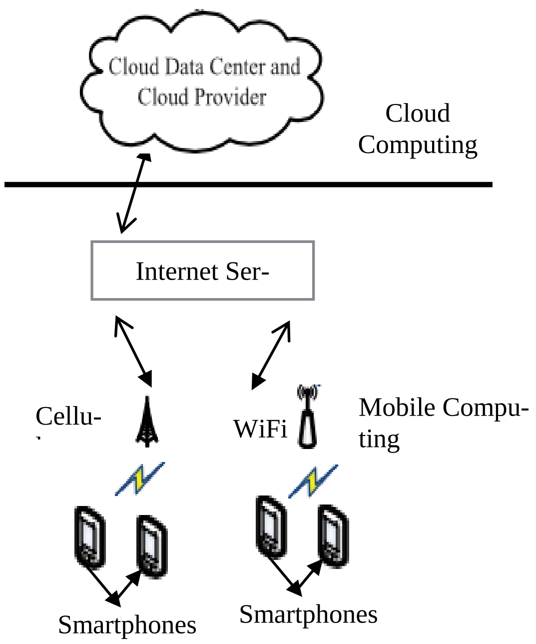

Fig 2.1: Mobile Cloud Computing

Mobile Cloud Computing system consists of 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. These mobile clients can be any kind of mobile devices with different physical characteristics like storage, power capacity, processing speed,etc. The proxy is responsible for recording information of mobile clients and scheduling tasks for mobile users as show in Fig 2.1.

2. BACKGROUND THEORY

Mobile and cloud are two of the most useful technologies today and it is only a matter of time before these technologies combine. These system are being tried to combine these two technologies. 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. Karthik Kumar [7] surveyed the common approaches used to make offloading decisions, and classified these approaches based on various factors, including (why to offload, when to decide offloading, what mobile systems use offloading, types of applications, infrastructures for offloading).

3. 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, thus decreasing 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. E.Cuervo[4] had proposed 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. Chun[2] had presented a technique known as Clone Cloud to reduce the burden of Mobile Devices. Clone Cloud is a system which automatically convert applications of the mobile devices by partially offload 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. Huaming Wu[5] proposed Path-based offloading partitioning(POP) Algorithm to determine which portions of the application tasks to run on mobile devices and which portions on cloud servers with different cost models in mobile environments. Long running applications on resource-constrained mobile devices can lead to software aging. Dejan Kovachev [3] proposed Mobile Augmentation Cloud Services (MACS) middleware which enables adaptive extension of Android application execution from a mobile client into the cloud .The middleware does the heavy lifting of adaptive application partitioning, resource monitoring and computation offloading.

Amal Ellouze [1] proposed 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 conditions 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. Yang Zhang [11] proposed MDP based offloading algorithm during the execution, the mobile user can dynamically decide to execute application phases locally on the mobile device or offload to nearby cloudlets as a Poisson point process (PPP).As a result, the user has to restore and execute the failed application phase again (either locally or remotely) in the next decision period). A given set of computational components which constitutes a mobile application, to decide which components should be offloaded to the cloud such that the application can be completed at the minimal execution cost. Yaling Tao [10] proposed optimal computation offloading algorithm to decide which components should be offloaded to the cloud such that the application can be completed at the minimal execution cost. Shuiguang Deng [8] solved mobile computation offloading problem where multiple mobile services in workflows can be invoked to fulfill their complex requirements and makes decision on whether the services of a workflow should be offloaded. S M Azharul 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. The system based on the inputs, decide whether to offload the application to the cloud or not. That algorithm saves more time, compared to a previous approach, and also reduces device energy usage by moving energy hungry processes to the cloud. Jessica Oueis [6] proposed 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 IV briefly introduces mobile cloud computing. The computation offloading decision is proposed in Section IV. Finally, the conclusion is Section V.

4. PROPOSED SYSTEM MODEL

The section illustrate the overall system model of offloading decision to mobile cloud is shown in Fig.5.1.Whether the system offload certain parts of an application to the cloud or not depends on the following factors: current CPU load ,State of Charge(SOC), wireless network bandwidth and transmission data size(AMOD).

Before the system considers these factors, the program to be partitioned as a Weight Object Rela-

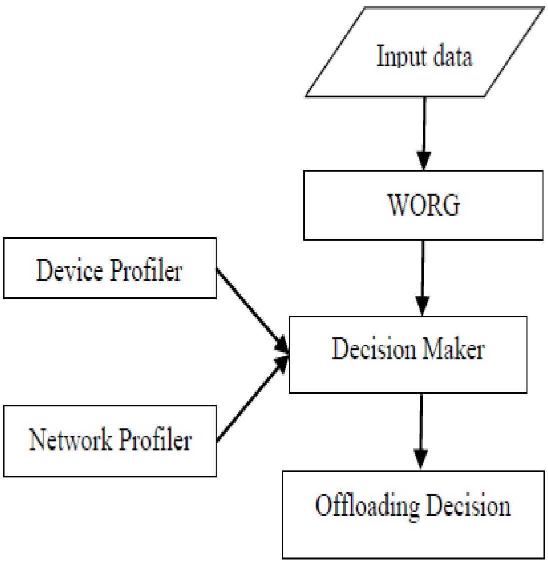

Fig.5.1: Overall Proposed System

tion Graph (WORG) constructed by using static analysis and dynamic profiling techniques according to the estimated computational and communication costs, and further derive a new *Enhance Computation Offloading Algorithm* designed especially for applications that can be sequentially executed.

The goal of the proposed system is to partition data size (AMOD) as a graph WORG = (V,E), with vertices set V and edges set E.

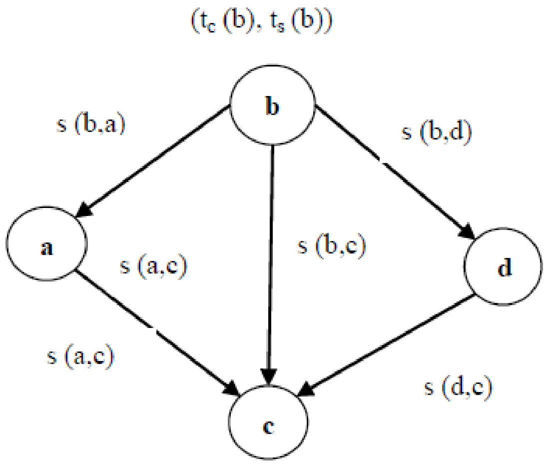

Fig.5.2: WORG of an Application

As shown in Fig.5.2, the weight of the vertex v is described as a 2 tuple ($t_c(v)$,$t_s(v)$), where $t_c(v)$ indicates the CPU execution time for each object running on the client, and $t_s(v)$ is that for each object running on the server.The round-trip time between the mobile terminal and the server is a key parameter that conditions the level of interactivity of the applications that can be offloaded.

After received the result of the graph (WORG), offload decision maker will process as shown in Fig.5.3.Enhanced computation offloading algorithms will consider based on these following data.

- Input from Weight Object Relation Graph (WORG)
- condition of Device profiler:
  - State of Charge (SOC)
  - CPU load
- condition of Network profiler:
  - depend on bandwidth(eg:3G,WiFi..)

Moreover, the propose system trigger quality of user experience (QOE).According to the conditions, the algorithm will determine to offload to the cloud server or not (run on mobile device).

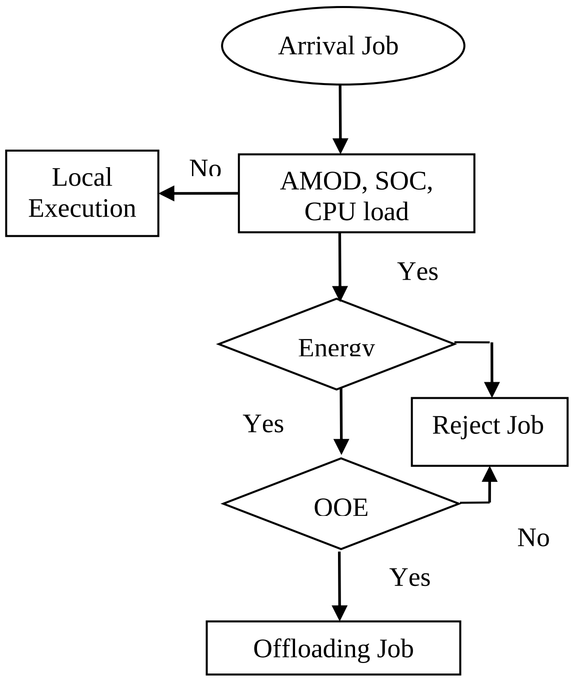

Fig.5.3: Flow of Offloading Decision Maker

5. Conclusion

Research objective is to optimize energy use in the device and execution costs. Computation offloading decision will base on some parameters, the system has decided whether to offload the application to the cloud or not. The major challenge in task offloading is to estimate accurately the energy consumed during the network activities of task offloading. The decision is optimized to achieve the lowest cost (e.g., computation and communication costs). The efficiency of the proposed algorithms is examined according to theoretic analysis and numerical experiments.

REFERENCES

[1]. Amal Ellouze, Maurice Gagnaire, Ahmed Haddad: “A Mobile Application Offloading Algorithm for Mobile Cloud Computing”, in 3rd IEEE Conference, 2015.

[2]. Chu, B., Ihm, S., Maniatis, P., Naik, M., Patti, A.: “CloneCloud: Elastic Execution between Mobile Device and Cloud.” In: ACM Workshop EuroSys, pp. 301–314 (2011).

[3].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”, ser. MobiSys 10. New York, NY, USA: ACM, 2010, pp. 4962.

[4]. Huaming Wu, Katinka Wolter,”Software Aging in Mobile Devices: Partial Computation Offloading as a Solution”,IEEE,2015

[5]. Jessica Oueis, Emilio Calvanese Strinati, Sergio Barbarossa: “Multi-parameter Decision Algorithm for Mobile Computation Offloading”, IEEE 2014.

[6]. Karthik Kumar, Jibang Liu, Yung-Hsiang Lu: “A Survey of Computation Offloading for Mobile Systems”, Springer, 2012.

[7]. Shuiguang Deng, Longtao Huang, Javid Taheri, Albert Y. Zomaya: “Computation Offloading for Service Workflow in Mobile Cloud Computing”, IEEE Transaction on (Volume: 26, Issue: 12), 2015.

[8].S M Azharul Karim, John Prevost: “Efficient Mobile Computation Using the Cloud”, 3rd IEEE Conference, 2015.

[9]. Yaling Tao, Yongbing Zhang, Yusheng Ji: “Efficient Computation Offloading Strategies for Mobile Cloud Computing”, IEEE 29th Conference, 2015.

[10]. Yang Zhang, Dusit Niyato, Ping Wang: “Offloading in Mobile Cloudlet Systems with Intermittent Connectivity”, IEEE Transaction on (Volume: 14, Issue: 12), 2015.