Sustainable Computing: Informatics and Systems 8 (2015) 38–46

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Partial mobile application offloading to the cloud for energy-efficiency with security measures

Salwa Adriana Saab, Farah Saab, Ayman Kayssi*, Ali Chehab, Imad H. Elhajj

Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon

ARTICLE INFO

Article history:
Received 30 September 2014
Received in revised form 16 June 2015
Accepted 30 September 2015
Available online 26 October 2015

Keywords:
Offloading
Mobile cloud computing
Free sequence protocol
Energy efficiency
Power consumption

ABSTRACT

Mobile applications are becoming computationally intensive nowadays due to the increasing convenience, reliance on, and sophistication of smartphones. Nevertheless, battery lifetime remains a major obstacle that prohibits the large-scale adoption of such apps. Mobile cloud computing is a promising solution whereby apps are partially processed in the cloud to minimize the overall energy consumption of smartphones. However, this will not necessarily save energy if there is no systematic mechanism to evaluate the effect of offloading an app onto the cloud. In this paper, we present a mathematical model that represents this energy consumption optimization problem. We propose an algorithm to dynamically solve the problem while taking security measures into account. We also propose the free sequence protocol (FSP) that allows for the dynamic execution of apps according to their call graph. Our experimental setup consists of an Android smartphone and a Java server in the cloud. The results demonstrate that our approach saves battery lifetime and enhances performance. They also show the effects of workload amount, network type, computation cost, security operations, signal strength, and call graph structure on the optimized overall energy consumption.

1. Introduction

Smartphones are becoming a major part of human lives. With the fast advancement of applications, people are enjoying a large variety of useful and exciting services offered by smartphones. However, smartphone applications are becoming more demanding in terms of computation and storage because of the advanced algorithms that are developed to satisfy complex user requirements. Due to the slow development of battery technology, limited battery lifetime raises a major concern. To this end, researchers are focusing on software solutions from basic development of algorithms and applications, to receiving aid from external resources such as cloud computing, which as we will show in this paper, is very promising.

Cloud computing is a booming technology that provides on-demand unlimited virtual resources that can be accessed via the Internet [1]. The offered applications are known as Software-as-a-Service (SaaS) where hardware and software elements in the hosting datacenters are known as the cloud. SaaS users can benefit from the provided services without having to own and maintain servers themselves.

Cloud computing can offer great opportunities to mobile phones. Since smartphones are typically connected in a permanent fashion to the Internet, the cloud becomes a feasible extension to limited phone resources. Combining cloud resources with mobile devices is known as mobile cloud computing (MCC) [2].

The limitations of smartphones in terms of storage and computational capacities can be resolved by offloading application processing to the cloud. Offloading the entire application may not always be the best solution since the resulting energy consumption due to communication might exceed that due to local execution. Therefore, a systematic algorithm that accounts for all influencing factors is needed to generate the best partitioning schema of any application. Local execution, full remote processing, and partial offloading should all be considered to minimize energy consumption depending on the given circumstances. The possibility that partial offloading might supersede both remote and local execution requires a well-defined approach that allows the mobile application to be executed in any combination at the mobile and cloud sides.

In this paper, we analyze MCC as a software solution for the problem of energy consumption in smartphones. A mathematical model representing this energy optimization problem is developed. We adopt a minimum-cut algorithm and extend it to account for

* Corresponding author. Tel.: +961 3965258.
E-mail addresses: adrianasalwa.saab@gmail.com (S.A. Saab), fws02@aub.edu.lb (F. Saab), ayman@aub.edu.lb (A. Kayssi), chehab@aub.edu.lb (A. Chehab), ie05@aub.edu.lb (I.H. Elhajj).

http://dx.doi.org/10.1016/j.suscom.2015.09.002
2210-5379/

security measures. We also propose a new protocol – free sequence protocol (FSP) – that allows a mobile application to execute in any possible combination between the client and server sides. A full system consisting of a profiler, decision engine, and an FSP-based mobile app is developed on an Android platform. Experiments are performed with Amazon EC2 cloud to study the impact of different factors on the offloading decision and to evaluate using such a solution to minimize energy consumption using real energy measurements.

The rest of the paper is organized as follows: Section 2 discusses related work. We present our system design in Section 3 and system architecture in Section 4. Section 5 examines the implementation of the system. Results and discussions are provided in Section 6. We conclude with remarks and potential future work in Section 7.

2. Related work

Several researchers have discussed static and dynamic partitioning. MCC was initially aimed at minimizing the delay in executing mobile applications. It then developed to become a potential solution for the limited lifetime of batteries. In this section, we describe the most prominent solutions of MCC and highlight their main shortcomings.

Cuckoo is a framework that allows for computation offloading for Android devices in typical client-server architecture. The application contains services that are offloaded once from the mobile to the server where they become always available on the cloud. The heuristics used to decide the execution location of a service are simple: if remote resources are available, services shall be offloaded and executed on the cloud. Such a decision, however, may increase the energy consumption and is not considered optimal [3].

Chun and Maniatis [4] introduced dynamic partitioning as a better alternative to static partitioning which is not always efficient due to the diversity in workloads, networks, and types of handset. They formulated the dynamic partitioning problem to minimize total execution time. Their basic goal was to avoid offloading modules if offloading introduces more delay to the system. However, power consumption was not taken into consideration. The work was then extended by Chun et al. [5] who suggested the CloneCloud architecture based on cloning the smartphones in the cloud to partially offload execution to these clones. This approach only considers limited input conditions in the offline pre-processing and needs to be bootstrapped for every new application built. They also did not address security concerns or connection problems. Moreover, a high overhead is initially imposed since the cloud and the mobile must be synchronized.

In the MAUI system [6], parts of the application code are offloaded to the server. The authors model the offloading problem to provide an optimal partitioning at runtime. They discuss factors that affect the battery life of an application when executed remotely. However, MAUI only supports applications developed for Microsoft .Net common language runtime, thus introducing a portability issue. Also, the authors did not consider security concerns.

ThinkAir [7] extends MAUI and CloneCloud by exploiting the auto-scaling feature offered by the cloud. It addresses MAUI's lack of scalability by creating virtual machines of a complete smartphone system on the cloud. And it eliminates restrictions on input conditions that CloneCloud induces by adopting online method-level offloading. It can perform on-demand resource allocation and exploit parallelism by dynamically forming and terminating virtual machines in the cloud when needed. The authors assume a trustworthy cloud server execution environment. Security concerns such as authentication and confidentiality are mentioned in their work but not yet implemented.

In [8], the authors present their mobile augmentation cloud services (MACS) middleware which is inspired by Cuckoo and MAUI systems, but performs more profiling and resource monitoring of applications. Also, it adapts the partitioning decision at runtime. The authors define three metrics for decision making, the available memory, energy consumption, and execution time. They only use WiFi in their evaluations so it is difficult to say if offloading would be beneficial when 3G is used. Also parallelization of the offloaded services is yet to be implemented.

Miettinen and Nurminen [9] discussed energy efficiency in MCC. They analyzed the factors affecting the power consumption of offloading and processing the application on a server. Their evaluation is based on processing the whole application remotely. They note that MCC does not always reduce energy consumption. Kumar and Lu [10] showed that cloud computing can save energy for mobile users by offloading the applications. Their discussion is limited to offloading the whole application to the cloud without partitioning.

Han et al. [11] suggested migrating components of the application onto a regular client-server architecture to achieve energy savings. Their evaluation was only based on simulations. Moreover, their discussion does not include cloud computing scenarios and does not show how to achieve dynamic partitioning of the application in any order.

Liu and Lu [12] analyzed the energy savings of mobile systems in privacy-preserving computation offloading. They used homomorphic encryption to protect an image retrieval application sent to the cloud for processing. They proved that energy savings can be achieved and thus additional security does not necessarily compromise those gains. Their solution however, does not react dynamically to the conditions of the surrounding environment. Wolski et al. [13] argue that the bandwidth of the network between the client and the server is the critical factor that determines the decision for offloading.

Satyanarayanan et al. [14] discuss how a wide range of issues can be transformed into trivial tasks using mobile real-time cognitive assistance where sensing capabilities of mobile devices are combined with compute-intensive processing in the cloud. Cloudlets are presented as central elements representing the middle tier of a three-tier hierarchy (mobile device – cloudlet – cloud), that can be viewed as datacenters whose goal is to bring the cloud closer, thus improving latency. Such an approach requires widespread deployment of servers to make it feasible, which is not yet a reality. The authors of [15] suggest the use of tiered-clouds to improve performance and scalability of smartphones. They consider delay, energy consumption, and user cost to find an optimal partitioning scheme between phones and tiered cloud architectures. They propose an efficient heuristic algorithm, MuSIC that is implemented and tested on a two-tier mobile cloud and rich mobile apps. They show that MuSIC supports scalable operation and ensures high performance.

In [16], a cloud-hosted middleware layer was introduced to improve user interaction with the cloud. This layer is viewed as the interface between the cloud and the phone. It enables additional options such as message optimization, protocol transformation, and request/response augmentation. Choi et al. [17] proposed the use of "cloud parallel process" technology to deal efficiently with large volumes of unstructured data. The effectiveness and convenience of their suggested model were validated through the design, implementation, and experimental evaluation of a mobile cloud computing model that utilizes a distributed computing framework based on open sources.

Others researchers achieved better results by executing certain functions on the cloud. Houmansadr et al. [18] propose a cloud-based intrusion detection and response system for mobile phones. Their intrusion detection system copies smartphones to virtual machines in the cloud through a proxy. Incoming traffic is copied