Contents lists available at ScienceDirect

Pervasive and Mobile Computing

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

Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing

Mati B. Terefe, Heezin Lee, Nojung Heo, Geoffrey C. Fox, Sangyoon Oh*

Department of Computer Engineering, Ajou University, 701 Paldal-Hall, Ajou University, Suwon, 443-749, South Korea

ARTICLE INFO

Article history:
Received 11 September 2014
Received in revised form 7 August 2015
Accepted 1 October 2015
Available online xxxx

Keywords:
Multisite
Offloading
MDP
Mobile cloud

ABSTRACT

Mobile systems, such as smartphones, are becoming the primary platform of choice for a user's computational needs. However, mobile devices still suffer from limited resources such as battery life and processor performance. To address these limitations, a popular approach used in mobile cloud computing is computation offloading, where resource-intensive mobile components are offloaded to more resourceful cloud servers. Prior studies in this area have focused on a form of offloading where only a single server is considered as the offloading site. Because there is now an environment where mobile devices can access multiple cloud providers, it is possible for mobiles to save more energy by offloading energy-intensive components to multiple cloud servers. The method proposed in this paper differentiates the data- and computation-intensive components of an application and performs a multisite offloading in a data and process-centric manner. In this paper, we present a novel model to describe the energy consumption of a multisite application execution and use a discrete time Markov chain (DTMC) to model fading wireless mobile channels. We adopt a Markov decision process (MDP) framework to formulate the multisite partitioning problem as a delay-constrained, least-cost shortest path problem on a state transition graph. Our proposed Energy-efficient Multisite Offloading Policy (EMOP) algorithm, built on a value iteration algorithm (VIA), finds the efficient solution to the multisite partitioning problem. Numerical simulations show that our algorithm considers the different capabilities of sites to distribute appropriate components such that there is a lower energy cost for data transfer from the mobile to the cloud. A multisite offloading execution using our proposed EMOP algorithm achieved a greater reduction on the energy consumption of mobiles when compared to a single site offloading execution.

1. Introduction

These days, the number of smartphone users is increasing rapidly. According to a Gartner press release from February, 2014 [1], worldwide sales of smartphones to end users totaled 968 million units in 2013, an increase of 42.3% from 2012. These numbers illustrate that smartphones are becoming the primary platform of choice for communication and computational needs. Moreover, there are an enormous number of mobile applications available for smartphones. However, smartphones still cannot match their desktop counterparts when performing complex multimedia operations such as image and video processing, object or face recognition, and augmented reality applications [2]. This is because mobile device systems still operate with limited resources such as battery life, network bandwidth, storage capacity, and processor

* Corresponding author. Tel.: +82 31 219 2633; fax: +82 31 219 1621.
E-mail address: syoh@ajou.ac.kr (S. Oh).

http://dx.doi.org/10.1016/j.pmcj.2015.10.008
1574-1192/© 2015 Elsevier B.V. All rights reserved.

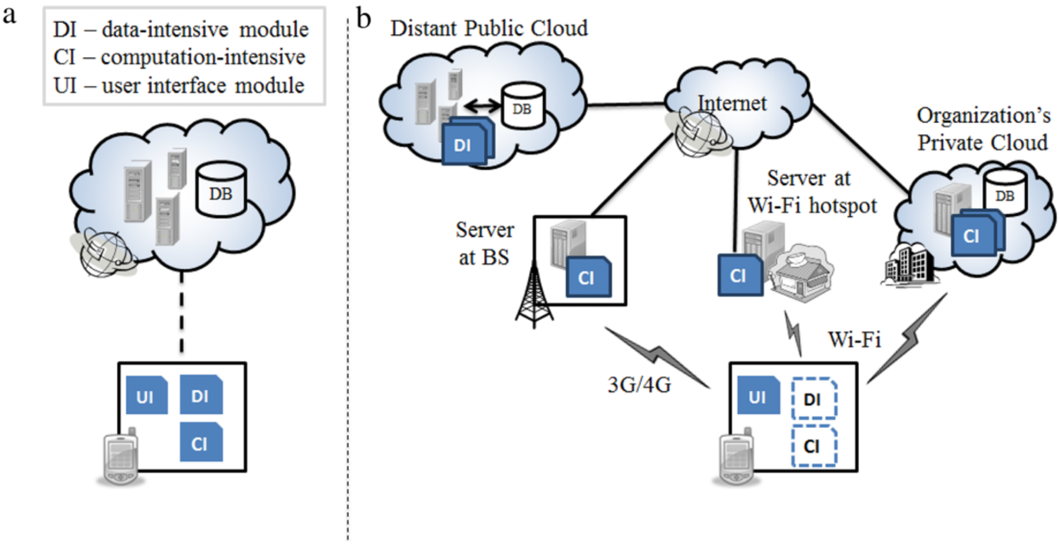
a
DI – data-intensive module
CI – computation-intensive
UI – user interface module
DB
UI
DI
CI
b
Distant Public Cloud
DB
DI
Internet
Organization's Private Cloud
DB
CI
Server at Wi-Fi hotspot
CI
Server at BS
CI
3G/4G
Wi-Fi
UI
DI
CI

Fig. 1. (a) Native execution of a mobile application. (b) Mobile application execution in a multisite offloading environment.

performance [3]. Therefore, augmenting the capabilities and prolonging the battery life of mobile devices have become one of the top areas of research.

In recent years, there has been a significant amount of research performed on computation offloading [4–9]. Computation offloading is a major approach in mobile cloud computing to augment mobile capabilities by migrating computation to more resourceful computers (i.e., servers) [10]. The current cloud computing infrastructure provides mobiles with an abundance of and easy access to public cloud computing resources. Therefore, there are several cloud computing providers that use public clouds to solve mobile computing problems. For instance, Apple's iCloud provides a service to its customer by hosting their applications and data in public clouds (i.e., Amazon EC2 and Microsoft Azure). However, for some mobile applications, such as perception and multimedia applications, the network latency of public clouds could make it difficult to achieve the desired performance [11]. Thus, mobiles devices prefer to access servers that have low latency for computation offloading. These servers include those that are accessed with a small number of network hops, such as servers in a Wi-Fi hotspot or within an organization's private cloud [12]. Moreover, there are situations where organizations might wish to store their private or proprietary data in their own private cloud. In such cases, mobile devices that manipulate these data often offload their computations to private cloud servers rather than public cloud servers.

Many researchers have proposed computation offloading algorithms that increase the performance and extend the battery life of mobile devices by migrating the energy-intensive part of a computation to a server. However, most studies propose a limited form of offloading. First, most of these schemes are limited to a single server as the offloading site [4,5,9]. Because there is an environment where mobile devices can access multiple cloud providers, it is possible for mobiles to offload computations to multiple servers. This approach is desirable because it distributes the appropriate application components among servers to save even more energy and increase mobile performance [13]. Second, in most cases, the schemes make offloading decisions based on profiling information that assumes a stable network environment [7,8]. However, this assumption is not always correct because the mobility of a user could create a dynamic bandwidth between the mobile and server. As a result, if the network profile information does not match the actual post-decision bandwidth, the offloading decision could lead to a critical Quality-of-Service (QoS) failure.

In this paper, we consider an environment where there are multiple heterogeneous servers for executing application components, and the network bandwidth between mobiles and servers is stochastically dynamic. In this environment, a multisite offloading is implemented to perform data- [8] and process-centric offloading. Hence, this technique is a suitable for mobile applications consisting of both data-intensive (DI) and computation-intensive (CI) modules. Real-time multimedia applications, especially augmented reality applications, are some applications that benefit from this technique. A mobile augmented reality application that extracts sets of features from a scene image is a good illustrative example. It uses feature descriptors to retrieve similar-looking entries in a database [14]. In this case, the feature extraction module is a CI module and the feature matching module is a DI module. It is clear that running these modules on mobile device will consume a large amount of energy and bandwidth (Fig. 1(a)). Therefore, it is a good approach to offload both modules to cloud servers to save mobile energy and increase performance. Hence, in this scenario, multisite offloading is an effective offloading technique because mobiles can offload the feature extraction module to a server that is close to the mobile, and the feature matching module can be offloaded to a server that is close to the database [8] (Fig. 1(b)). This approach is efficient because it has low network latency and results in a higher quality of object recognition [2].

In this paper, we propose an efficient multisite offloading approach that minimizes the energy consumption of a mobile while meeting the execution deadline. Our approach differentiates the DI and CI components of an application and performs