ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A quick-response framework for multi-user computation offloading in mobile cloud computing

Zhikai Kuang [a], Songtao Guo [a,*], Jiadi Liu [a], Yuanyuan Yang [a,b,*]

[a] Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing, 400715, China

[b] Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

HIGHLIGHTS

- We propose an agent-based mobile cloud computing framework.
- We formulate the offloading optimization problem into the maximum energy saving problem.
- We propose a Dynamic Programming After Filtering (DPAF) algorithm.

ARTICLE INFO

Article history:
Received 22 March 2017
Received in revised form 25 August 2017
Accepted 22 October 2017
Available online xxxx

Keywords:
Computation offloading
Energy saving
Task filtering
Completion time constraint
Mobile cloud computing

ABSTRACT

The execution of much sophisticated applications on the resource-constrained mobile device will lead to the fast exhaustion of the battery of mobile device. Therefore, mobile cloud computing (MCC) is regarded as an energy-effective approach by offloading tasks from mobile device to the resource-enough cloud, which cannot only save energy for mobile devices but also prolong the operation time of battery. However, it still remains a challenging issue to coordinate task offloading among mobile devices and get offloading results quickly at the same time. In this paper, we propose an agent-based MCC framework to enable the device to receive offloading results faster by making offloading decision on the agent. Moreover, to get an offloading strategy, we formulate the problem of maximizing energy savings among multiple users under the completion time and bandwidth constraints. To solve the optimization problem, we propose a Dynamic Programming After Filtering (DPAF) algorithm. In the algorithm, firstly, the original offloading problem is transformed to the classic 0–1 Knapsack problem by the filtering process on the agent. Furthermore, we adopt dynamic programming algorithm to find an optimal offloading strategy. Simulation results show that the framework can more quickly get response from agent than other schemes and the DPAF algorithm outperforms other solutions in energy saving.

1. Introduction

Nowadays, smartphones are gaining enormous popularity due to that versatile mobile applications satisfy the needs of the users. Most computation-intensive applications, such as face recognition, augment reality, are complex and widely installed on smart devices, which causes high energy consumption [1]. However, hardware resource (e.g., CPU computation capacity, storage, etc.) and energy supply are constrained on mobile devices. In particular, energy supply is still the primary bottleneck for mobile devices [2]. Thus, how to run such complex application on the energy-limited mobile device remains a challenge.

Mobile cloud computing (MCC) is envisioned as a promising method to address such a challenge. Computing capabilities of smart mobile devices can be augmented by migrating computation tasks via wireless access to the resource-rich cloud, referred to as computation offloading. To achieve computation offloading, the computation-intensive application needs to be partitioned into many tasks, which can be separated into two categories. One is the non-offloadable tasks, which can be executed on mobile devices due to the dependencies among tasks or hardware dependencies. The other is the offloadable task, which can implement remote execution by offloading [3]. Obviously, offloadable tasks selectively executed on clouds can accommodate mobile devices to run sophisticated applications. The advantage of this approach is that it can alleviate the high computation workload on mobile devices. Moreover, the computation offloading technique is beneficial to save energy for mobile devices and prolongs the operation time of

* Corresponding authors.
E-mail addresses: songtao_guo@163.com (S. Guo), yuanyuan.yang@stonybrook.edu (Y. Yang).

https://doi.org/10.1016/j.future.2017.10.034
0167-739X/

battery. Some works [4–6] focused on virtual machine migration, which show that such migration can save energy. The works in [7–9] were dedicated to fine-grained granularity. These works indicate that the mobile devices can benefit from offloading.

Although mobile devices are allowed to take advantages of cloud computing to alleviate computation workload as well as prolong battery lifetime, it still remains challenging to coordinate offloading tasks among mobile devices and achieve quick response for mobile users at the same time. In most existing frameworks [4,7,10,11], mobile devices send the computation offloading requests to the cloud, and then the offloading decision made by the cloud will be sent back to the mobile devices. However, mobile users will waste long time for receiving offloading decision from the cloud without considering whether they can benefit from offloading. Especially for invalid requests, they not only suffer offloading failures from clouds but also prolong the execution time of the tasks. Besides, most existing works consider either the completion time constraint [7,12] or the bandwidth constraint [13,14] to achieve energy savings. However, it is necessary to take both of the constraints into consideration to reduce energy consumption for all users. Therefore, we consider two issues in our work: (1) How to design a framework in case of waiting for a long time to retrieve the reply of computation offloading request? (2) How to make an efficient computation offloading strategy among mobile devices under the constraints of completion time and limited bandwidth?

To address the above issues, in this paper, we propose an agent-based quick-response computation offloading framework to ensure that the mobile user can get the response of the offloading request more quickly. In addition, we propose Dynamic Programming After Filtering (DPAF) algorithm to solve the multi-user computation offloading problem.

The framework consists of distant cloud, agents as well as multiple mobile devices. In the proposed framework, we adopt the request filtering on the device and agent so as to effectively achieve the quick response. Our proposed framework has the following advantages. First, mobile devices only send beneficial offloading requests to the agent, which not only can avoid the energy consumption from invalid offloading requests, but also reduce the computation workload on the agent and alleviate the communication burden of the network. Second, compared with sending computation offloading requests to the remote cloud, mobile devices retrieve the offloading decision more quickly and shorten the waiting time for the offloading result.

More specifically, our proposed framework aims to provide an optimal offloading strategy to achieve multi-user maximum energy saving. To that end, we first formulate the computation offloading problem as an optimization problem. Then, we convert the original optimization problem into the classic 0–1 Knapsack problem by employing filtering process based on task completion time constraint. The main contribution of this paper can be summarized below.

- First, we propose an agent-based computation offloading framework aiming to shorten the delay of computation offloading request for mobile users, alleviate communication overhead in the network and avoid energy consumption of transmitting invalid requests.
- Second, we formulate the offloading optimization problem into the maximum energy saving problem under constraints of task completion time and network bandwidth.
- Third, to solve the optimization problem, we design a DPAF algorithm to provide a policy of computation offloading selection among mobile devices.
- Finally, we demonstrate the energy saving performance of the proposed algorithm by numerical results and evaluate the quick-response feature of the agent-based framework.

The rest of this paper is organized as follows: Section 2 introduces related work on offloading problem. Section 3 gives system framework and its operation procedure. Section 4 proposes the problem statement and a detailed algorithm. Section 5 shows experimental result and performance evaluation. Section 6 concludes this work and makes prospects to the future.

2. Related work

Related works are introduced in this section, which are separated into two aspects. One is the computation offloading scheme and the other is offloading policies. There have been a lot of works adopting the computation offloading model [7,10,11]. Cuervo et al. provided a method-level dynamic offloading framework MAUI to maximize the energy saving in [7]. Only if the method is remoteable in the mechanism, MAUI collects the parameters required by offloading and sends the requests to center cloud. The solver on the cloud decides whether to execute the task remotely. In [15], Yang et al. studied multi-user partitioning and offloading problem where PaaS middleware of the cloud makes the partitioning decisions. These works adopted a traditional offloading framework that the cloud makes decisions according to the information sent by mobile devices. However, there are two drawbacks of such a framework. One is that the response to offloading request from the cloud to device needs a long latency, and the other is that mobile devices will upload requests containing offloading parameters to the cloud while not considering whether the tasks can benefit from offloading, which causes extra energy consumption.

To shorten the latency between distant cloud and mobile device, the agent-based framework was proposed. Liu et al. in [16] proposed a BWRS scheme to improve the QoS of real-time streams and the overall performance of MCC network in the mobile agent-based architecture. In [14], Nir et al. investigated a scheduler model on the broker node which solves tasks assignment to minimize total energy consumption. The advantages of this framework are introduced in [17,18]. First, it can reduce information interaction and data delivery delay. Second, the agent can periodically apply for a collection of resources to satisfy multiple users instead of accessing cloud resources by every user.

Compared with the traditional framework, the agent-based framework can make offloading decisions on the nearby agents instead of uploading the requests to the center cloud for decisions. First, the agent receives the requests from mobile devices and then requires the information from the cloud and makes offloading decision finally. Obviously, the place for making decision just moves from distance cloud to nearby agent. That is why the agent-based framework reduces transmission delay. However, the number of uploading requests is still not changed in both frameworks. Thus, the agent with limited computation capacity has to deal with a great number of requests in the agent-based framework, which may cause the heavy computation workload. In our framework, however, only the beneficial offloading requests are send to the agent by checking the energy constraint, which cannot only reduce the number of the invalid requests but also save the energy of uploading requests. Various energy-based offloading policies are also studied [12,13,19–22]. In [19], Kumar and Lu proposed a simple energy model to quickly estimate the energy saved from remote execution for the single user. However, the completion time constraint is not considered, which may lead to an incorrect offloading decision. In [20], Liu et al. proposed an offloading decision method to minimize the energy consumption of mobile device with an acceptable time delay and communication quality. Besides, in [21], the authors studied to schedule the offloading tasks to minimize the energy consumption of mobile devices for one application under the constraint of total completion time. Moreover, in [13,22], the authors aimed to minimize both energy