Mobile MapReduce: Minimizing Response Time of Computing Intensive Mobile Applications

Mohammed Anowarul Hassan and Songqing Chen

Department of Computer Science, George Mason University
{mhassanb, sqchen}@gmu.edu

Abstract. The increasing popularity of mobile devices calls for effective execution of mobile applications. A lot of research has been conducted on properly splitting and outsourcing computing intensive tasks to external resources (e.g., public clouds) by considering insufficient computing resources on mobile devices. However, little attention has been paid to the overall users’ response time, where the network may dominate.

In this study, we set to investigate how to effectively minimize users’ response time for mobile applications. We consider both the impact of the network and the computing itself. We first show that outsourcing to nearby residential computers may be more advantageous than public clouds for mobile applications due to network impact. Furthermore, to speed up computing, we leverage parallel processing techniques. Accordingly, we propose to build Mobile MapReduce (MMR) to effectively execute outsource computing intensive mobile applications. Based on the original MapReduce framework, a new scheduling model is built in MMR that can always leverage the best computing resources to conduct computation with appropriate parallel processing. To demonstrate the performance of MMR, we run several real-world applications, such as text searching, face detection, and image processing, on the prototype. The results show great potentials of MMR in minimizing the response time of the outsourced mobile applications.

1 Introduction

Mobile devices are getting more and more popular. According to International Data Corporation, the total number of smartphones sold in 2010 is 305 millions [5], which is a 76% increase from the previous year, and there are already over 4.6 billion mobile subscribers in the world and the number is still growing [6].

Different from traditional mobile devices (e.g., cellphones) that are mainly used for voice communication, mobile devices today are typically equipped with much more powerful processor and more sensors. Such increasing power of mobile devices has enabled fast development of mobile applications, such as picture editing, gaming, document processing, financial tracking [8]. Recently, Amazon released SDK for Android users [1] to develop mobile applications using Amazon cloud such as uploading images and videos to the Amazon cloud storage, and sharing game, movies, and other data among users.
However, constrained by the size and weight, mobile devices’s processing power is still significantly lagging behind that of their desktop counterpart. Thus, many desktop applications, if running on mobile devices, can result in poor performance. For example, an OpenGL application on an Android phone can refresh slowly on the screen and drive the user away quickly. On the other hand, mobile devices are ultimately constrained by the limited battery supply and a computing-intensive application can quickly exhaust the limited battery power. Such a situation is worsened by the industrial trend to equip more sensors on mobile devices for a wider scope of non-traditional applications, such as environment monitoring [17], health monitoring [17], social applications [23, 22], which are often more computing intensive.

From the resource perspective, a lots of research have considered to outsource computing intensive tasks to external resources [10, 18, 26]. For example, the virtual machine-based cloning approach [12] has been explored to clone the entire mobile environment to the cloud without worrying about modifying the application or dividing the job. Similarly, Zap takes a full process migration [25] approach with resource and process consistency. On the other hand, a number of job partitioning strategies have been proposed [28, 14] to simplify the partitioning of the existing applications between the mobile device and the external computing resources.

While many existing schemes have focused on how to split the computing-intensive tasks and outsource to external resources, the impact of the network latency on outsourced applications has not been well investigated, which may be a dominant factor in the total response time to mobile users. For mobile applications, a minimal response time is not only critical to the users’ experience, but also important for preserving the limited battery power supply on mobile devices. This is particularly true for delay sensitive and interactive mobile applications. When partial tasks of such applications are outsourced, it is critical to reduce the total response time to the user in order to maintain the QoS of the application. In this paper, we aim to minimize the response time of mobile applications from the users’ perspective. Since outsourcing often involves both network transferring and computing, we first show that outsourcing to appropriate resources considering data affinity and network latency could be more advantageous than public clouds in reducing the overall response time. Furthermore, to speed up computing, we leverage parallel processing techniques. Accordingly, we design and implement Mobile MapReduce (MMR) based on the original MapReduce framework. In MMR, a new scheduling model is built that can always dynamically leverage the best computing resources, be nearby computers or public clouds, with the most appropriate parallelism considering the parallelization overhead [19] for any mobile application.

To demonstrate the performance of MMR, we have built a prototype and experimented MMR with several real-world applications, including text searching, face detection, and image processing. The results show that MMR not only outperforms on-device computing by 15 times and 20 times in terms of response time and the battery power consumption, respectively, but also outperforms public