Spectrum Aware Cloud Offloading for Mobile Devices

I. Introduction

Recent years have seen a significant increase in the development of sophisticated wireless mobile applications for augmented reality, natural language translation, speech recognition, 3D interactive gaming, activity inferencing, context awareness, healthcare sensing and analysis etc. This uptick in sophisticated mobile apps has not only increased the computational demand on the end device itself, but has also created an increasing load on the carriers core networks and put enormous pressure on spectrum resources. Although offloading computation to the more resource-strong cloud is a solution (and has existed in various forms in related areas) this will also place a further burden on the spectrum resources since data will have to be transferred between the mobile devices and the cloud in order to successfully deliver the application on the mobile device. Parallely, in the wireless networking world, the trend has shifted to multi-radio enabled communications. *Hence, it is imperative to study cloud offloading strategies in the context of both spectrum efficiency of the underlying end-to-end network (spectrum access + core network) as well as energy efficiency within the mobile device itself.* This is the uniqueness of the problems we consider. We seek access to cloud services provided by the Chameleon Cloud and CloudLab to solve the computationally heavy optimization problems that arise in the course of achieving this goal. Specifically our needs are to use the Chameleon Cloud and CloudLab as workhorses for our cloud enabled mobile offloading solutions.

II. Problem Statement

We propose to use the cloud services provided by the Chameleon Cloud and CloudLab for the following research problems. Experiments will be conducted in stages starting from the single radio scenario to the advanced multi-radio case. HTC Vivid smartphone with a 1.2 GHz dual core processor with two wireless interfaces: LTE and WiFi will be used as the mobile devices.

A. Single Radio Offloading:

First we consider the single radio case. We consider applications that can be decomposed into components with general dependencies (as opposed to simple chain dependencies) between components as shown in Figure 1. The goal is to process the application within the specified amount of time while minimizing the energy expenditure of the mobile device. In such problems, there needs to be a trade-off between the amount of energy spent in processing and data transfer against the allotted time limit to fully process the application.

We formulate the problem as a 0-1 integer linear program using time-indexed variables x_{pjt} that is assigned the value 1 when component j is processed in time period $[t, t+1)$ on either the cloud ($p = 1$) or the mobile device ($p = 0$). Such a formulation affords us the scope to easily expand the problem in our future studies to consider additional constraints as dynamic spectrum allocation in multi-radio enabled communications. Additionally, solutions to the 0-1 integer linear program are guaranteed to satisfy all the constraints of the problem as well as serve as a guide to designing time-efficient heuristics.

The cloud services provided by the Chameleon Cloud and CloudLab will be used for two purposes: to solve the 0-1 integer program as well as to emulate the execution of the application by the mobile device to calculate the actual energy expense and time expense. Our current experiments use the Amazon Cloud server and our emulations are run on very small example applications. To thoroughly investigate our techniques and heuristics and to study their scalability, we seek to test our techniques on significantly larger applications.

In particular, we propose to use the cloud resources to carry out the following experiments:

1) Evaluate our techniques and heuristics on applications with general precedence constraints vs. series-parallel constraints. As it is well known, the underlying scheduling problems for the general-precedence constraint case is conjectured to be as hard to approximate as the vertex cover problem.
2) Investigate scalability by considering larger applications - vary number of components in the applications from 100 to 1000. This will also increase the size of the underlying integer programming formulation significantly.

B. Multi-Radio Offloading:

In the second stage we will advance to scenarios where more than one radio network is available to the mobile user as shown in Fig. 2. Experiments in this case will be broken down into stages with increasing complexity.

1) **Stage 1:** In this first case, we consider an application with simple dependencies (chain rather than graph) between the components. The problem here is to optimally decide which components of an application to offload and which to execute locally, while simultaneously optimizing the percentage of data (associated with this offloading) to be sent via each radio

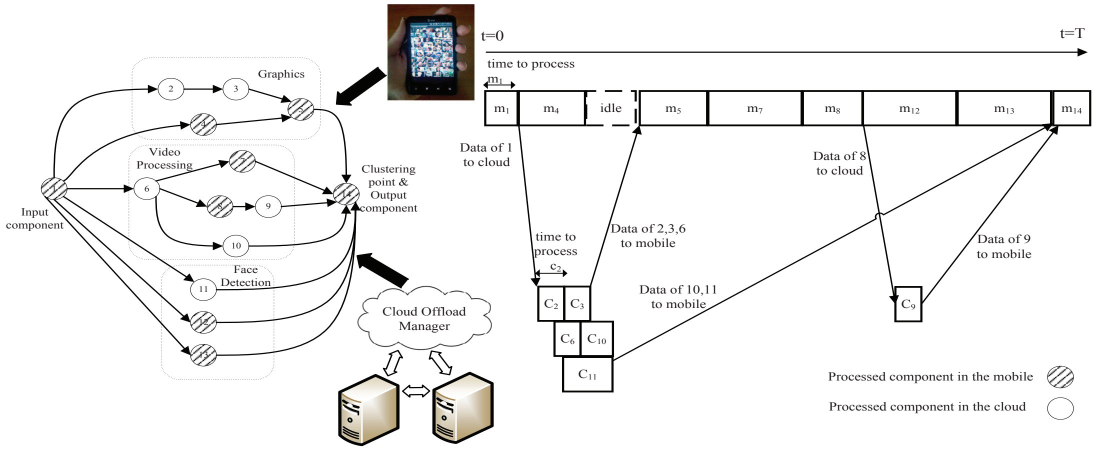

Fig. 1. Scheduling Model for Cloud Computing in a 14-Component Mobile Application.

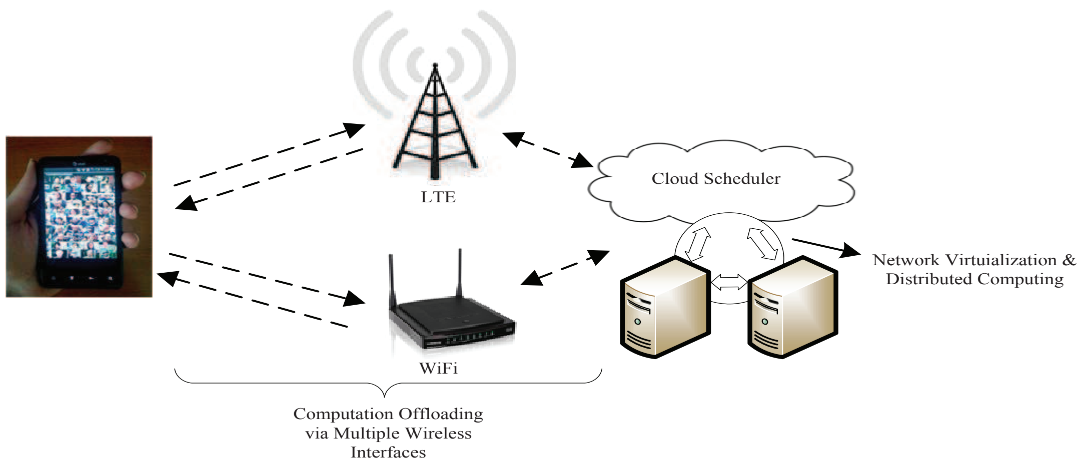

Fig. 2. Spectrum Aware Cloud Computing Infrastructure.

interface. This computation offloading problem is set up as a joint optimization to minimize the energy consumed on the device while at the same time maximizing the radio resources available to the device, under two constraints: (1) the total run time deadline of the application and (2) the maximum flow rate constraint on the radio resources. Since this optimization problem is non-linear and hence computationally intense, we also propose an iterative algorithm that converges to a local optimum. To keep things simple at this stage, we will only optimize the uplink as opposed to both uplink and downlink transfers. We will use Chameleon and CloudLab to experiment on complex applications with several components.

2) **Stage 2:** In this second scenario, we will increase the complexity of the problem to add the optimization of the downlink transfers as well. Experiments on the CloudLab and Chameleon will now involve running larger apps with more “expensive” optimization algorithms.
3) **Stage 3:** Putting the pieces together. In this phase, we will include apps with more complex inter-component relationships (similar to the ones studied in II-A) and add multiple radio interfaces to the scenario. Both uplink and downlink optimizations will be considered.