DroidLab: mobile sensing made easy, fast and cheap

Balázs Lajtha, Rolland Vida
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Email: {lajtha.balazs, vida}@tmit.bme.hu

Abstract—**We are living in a parallelized world. What was thought to be impossible for one can now easily be achieved by many. On one hand, we have the cloud: an army of obeying faceless machines. On the other hand we have the crowd: A colorful collection of volunteers. Clouds are built and sold but to control the crowd, each member of has to be won over.**

Keywords—

I. Introduction

"Crowdsourcing is the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call." This was the initial definition of crowd sourcing when first used in 2006 by Jeff Howe and Mark Robinson. Two main orthogonal problem sets can be solved using crowd sourcing: data gathering and data processing. When a data gathering problem is distributed among members of the crowd we use the term crowd sensing or participatory sensing, while for distributed computational tasks we use the general term crowd sourcing. Probably the most popular crowd sourced computational problem was Seti@Home, a project lasting several years and aimed to ??? starts and stuff, extraterestrial ???. This problem was divided into a large number of subtasks and distributed among the willing participants. Users could set a Seti screen-saver that download and solved these subtasks when the user was away. Seti's success came from it's innovative design and from the interesting task that it performed. From the point of view of the volunteer, this was a low cost, low gain activity. PCs back than had long startup times and high standby power consumption, once Seti was installed it didn't pose a visible overhead, and provided the user with insight of the performed work. Some early gamification elements were also implemented like leaderboards. A somewhat different example is the Google Image classification game. In this game two players cooperated to solve a simple puzzle. An image was displayed in both player's browsers. The first player had to find five words that described the image. Then the second player had to guess those words. This human image processing game allowed Google to improve Image Search results. The players were motivated to play the game for the game itself, and for added gamification elements.

Both examples involved utilization of computational resources owned by the users to perform a highly parallelized task that the users perceived as common good. In both cases, users received an input that they processed based on a given algorithm. The tasks were not personalized. The first task was solely performed by a machine, while the second task had to be solved by human users. In both examples users had other motivators.

Patricipatory sensing involves data collection outside of the system. Sensing can be automated or can involve the active participation of the user. Before smartphones, sensing was either done through expensive Internet-connected sensors or with the participation of the user through fixed personal computers. Sensors had to be deployed, and sensing involving the user had large delay and required considerable effort from the part of the user. The evolution of smartphones eased many new complex sensing applications. Smartphones have the connectivity that enable real time, high bandwidth communication. The devices can access precise location data, and are equipped with a wide range of sensors from cameras and microphones to accelerometers, gyroscopes and barometers. New mobile operating systems offer many seamless ways for the sensing system to prompt for user input.

To exploit these possibilities we offer DroidLab. DroidLab is a cloud backed mobile framework still under development, that aims to solve key challenges of mobile crowd sourcing. In this article we will present successful and promising mobile crowd sourcing application and crowd sourcing frameworks. We will identify the key challenges of mobile crowd sourcing and define the use cases for our framework. Then present the architecture and design of our solution and conclude with the evaluation of the current framework and define the future works.

II. Related work

We already mentioned two successful examples of crowd sourcing.

III. Use cases and challenges

There are many applications and even more proposed applications of mobile sensing and mobile crowd sourcing.

IV. Solutions

V. Conclusions

VI. Acknowledgment