Demonstration Abstract: CrowdMeter - Predicting Performance Of Crowd-Sensing Applications Using Emulations

Manoj R. Rege, Vlado Handziski, Adam Wolisz
Telecommunication Networks Group
Technische Universität Berlin
Email: {rege,handzisk,wolisz}@tkn.tu-berlin.de

Abstract—Predicting performance of crowd-sensing applications at large scale, in the pre-deployment phase, represents a significant challenge for developers. We demonstrate a solution to this problem in the form of a cloud-based emulation platform called CrowdMeter. Our platform emulates mobile devices and access network links, models human factors in crowd-sensing, and leverages virtualization through cloud infrastructure-as-service resources to model large scale crowd-sensing. In this demo we exhibit the capabilities of CrowdMeter by deploying VideoQuest, a simple crowd-sensing application, on hundreds of emulated mobile devices, and by measuring its performance.

Keywords—Crowd-Sensing, Emulation, Performance

I. INTRODUCTION

Evaluating performance of crowd-sensed applications in pre-deployment phase has been largely based on small scale measurements with recruited volunteers. Although this approach is close to reality, it validates just the basic application functionality, ignores scaling, and lacks control and repeatability due to human participation. In this paper, we present a cloud-based emulation platform called CrowdMeter that can be utilized for studying impact of variety of micro-level design decisions on crowd-sensing application performance at large-scale. These include decisions related to application code design, and crowd-sensing architecture such as data sensing, performing local processing of sensed data, temporary on-device data storage, computation offloading to cloud-based services, and data uploading. Further, impact of factors like type and quality of the mobile-cloud communication links, user mobility, user behavioral dynamics and usage patterns etc. on the end application performance at large-scale can also be analyzed using CrowdMeter.

II. CROWDMETER

A. Supported Crowd-Sensing Elements

Mobile Device - CrowdMeter enables the application-developers to emulate mobile devices by imitating their different hardware components such as CPU, memory, external storage, radios, etc. in software. CrowdMeter leverages existing high-fidelity Android platform emulators provided by different device vendors. A developer can reuse and run the real application code seamlessly on the emulators, that are deployed on top of virtual machine (VM) instances in a public cloud platform.

Human Factors - CrowdMeter supports modeling of human factors that can potentially influence the crowd-sensed application performance. Factors such as user activity and context, mobility, and user interaction with the mobile device etc. can be modeled as activity-event graphs. The activity represents the state of mobile device defined by users context and the events trigger change in the state. An activity can be represented at different granularity by defining different sensors and traces, and the mobile device events.

Mobile-Cloud Communication - Using CrowdMeter, the developers can emulate communication links between the mobile devices and the cloud-based application server(s). The network link emulation is performed on top of the real network interconnecting the emulated mobile device and the application server(s) by controlling the underlying real link behavior. For each emulated mobile device, a developer can select a wireless access technology link to be emulated using the latency, bandwidth, and packet loss parameters.

Cloud Back-End - CrowdMeter in itself does not emulate the cloud back-end. It enables integration of the emulated mobile devices that use emulated communication links with a real cloud back-end application server(s) that run real services.

Large-Scale - It leverages infrastructure-as-service resources from the public cloud platform to deploy and replicate large number of instances of the emulated mobile devices and links for enabling large-scale crowd-sensing.

B. Architecture

The CrowdMeter tiered architecture in Figure 1 reflect the crowd-sensing application elements. The three tiers on the right represent the emulated mobile devices, emulated mobile-cloud communication links, and the real cloud back-end respectively. In the mobile device tier, each VM host can support multiple emulator images. The mobile device tier of CrowdMeter is based on top of the Amazon EC2 cloud platform [1] and supports emulation of Android based mobile devices. CrowdMeter uses three different types of VM instances, namely Micro, Small, and Medium (as defined by Amazon EC2 based on varying degree of CPU and memory performance) for hosting mobile device emulator images. The mobile-cloud communication links are emulated using the DummyNet [2] network link emulator. Further CrowdMeter implementation details can be found in our earlier work [3].
The control tier on the left enables control and execution of emulation experiments remotely. The dispatcher sends messages including commands (configuration, sensing, and device events associated with modeled human factors) to each emulated mobile device. Each emulated mobile device has its own event handler on the VM host. The received messages are queued as events in the local event queue. The worker threads handle the events upon their time of expiry and fire them on the emulated mobile device. This event queuing mechanism is implemented using the Celery [4] distributed task queue framework. The control tier also monitors the state of each emulated mobile device and the underlying VM host. A database is used for storing information related to emulation configuration, state, and the collected traces. CrowdMeter uses MongoDB [5] database. CrowdViz GUI enables the developers to configure, setup, deploy, and visualize emulations.

III. Demo Description

The capabilities of CrowdMeter are demonstrated in three phases:

Setup and Configuration - In the first phase, we exhibit important aspects of emulation experiment setup such as: creation of cloud configuration (selection of physical location of the cloud, and type of VM instances for deploying emulators), configuration of emulator image parameters (CPU, sensor, radios, memory etc.), set up of mobile-cloud communication link emulation, and integration with real crowd-sensing application back-end services. Demo visitors would also be able to define models for human mobility and context, with different sensors and traces, using activity-event graphs.

Application Performance Evaluation - In this phase, we demonstrate the utility of CrowdMeter, using a simple Android-based crowd-sensed application called VideoQuest (inspired by Micro-Blog [6]). VideoQuest provides on-demand crowd-sensed image/video services using a XMPP based publish-subscribe framework. VideoQuest subscribers (users and web-services) create image/video subscriptions based on a certain geographical area of interest. The mobile device users periodically publish their locations and are notified by image/video query whenever they enter the geographic area of interest. Users can answer the queries by taking an image/video and publish them to the back-end services that notify the respective subscribers.

We demonstrate how VideoQuest can be deployed on CrowdMeter, and how our platform can be used to measure VideoQuest’s performance by answering the following questions:

- What is the best effort latency of publishing location data, and notification of sensing queries for XMPP Publish Subscribe framework used in VideoQuest?
- How does scaling the number of publishers impact notification latency of sensing queries?
- How does notification latency, and mobility of publishers impact coverage of area of interest, as defined in subscriptions?

User Interface - In the last phase, we demonstrate CrowdViz, our rich map-based GUI supporting emulation visualization, monitoring, and trace collection.

ACKNOWLEDGMENT

The work has been partially supported by EIT ICT Labs activity “From WSN Testbeds to CPS Testbeds” and the “Social Sensor Cloud” project under ERDF/ProFIT.

REFERENCES