Demo: A Programming Cloud of Smartphones

A. Konstantinidis
Dept. of Computer Science
University of Cyprus
akonstan@cs.ucy.ac.cy

C. Costa
Dept. of Computer Science
University of Cyprus
cs07cc6@cs.ucy.ac.cy

G. Larkou
Dept. of Computer Science
University of Cyprus
glarko01@cs.ucy.ac.cy

D. Zeinalipour-Yazti
Dept. of Computer Science
University of Cyprus
dzeina@cs.ucy.ac.cy

1. ABSTRACT

In this demonstration we present *SmartLab*[1], an exciting experimental testbed of approximately 40+ real Android Smartphones, plus emulated devices, deployed at the Department of Computer Science building at the University of Cyprus. SmartLab provides a public, permanent testbed for the development and testing of smartphone network applications via an intuitive web-based interface. Registered users can upload and install Android executables (APKs) on a number of Android smartphones, capture their output, reboot the devices, create concurrent interactive jobs using MonkeyRunner scripts, interact with the remote devices and many other exciting features. SmartLab aims to facilitate research in smartphone network programming environments, communication protocols, system design, and applications.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Android Smartphones, Programming Cloud, System Design

2. SMARTLAB AND DEMONSTRATION

In this section, we overview the basic elements of our programming cloud, the basic steps of interaction as well as our demonstration scenario. SmartLab is inspired by both PlanetLab [1] and MoteLab [2].

Architecture: Our 40+ smartphones are located in an air-conditioned laboratory that features WiFi coverage and are remotely mounted to a VMWare VCenter located in the Computer Science building cluster-room. In particular, smartphone devices are connected with USB 2.0 cables, which provide sustained power, to a proxy PC that mounts our devices over TCP/IP to a virtualized Ubuntu Linux 10.10 servers.

Allocate: SmartLab supports real and emulated Android devices. For the former category, the user can browse through an intuitive web-based catalog in order to allocate devices for a certain amount of time (see web site). During sign up, a registered user is provided with credit hours that can be utilized to reserve *"1 smartphone for 1 hour"* (charges incur on a minute basis). Reservations are automatically released when the user budget is due. For the latter category, a user can allocate up to 5 emulated devices for an unlimited amount of time. SmartLab is free to the academic community, but the allocation scheme might be adjusted in the future depending on supply, demand and usage.

File-Transfer: A registered user is able to upload APKs and data to his allocated devices through an intuitive web-based drag-n-drop user interface. Subsequently, these files can be transferred to one or more devices through a simulated network share that is mounted on all devices rented by the user. Additionally, the user can perform any kind of interaction with the local file system of the allocated devices (i.e., both external and on-board flash).

Interact: SmartLab supports four (4) modes of user interaction: i) *Remote Control Terminals (RCT)*, which support our in-house implementation of an ajax-based web-based remote screen terminal for Android that can mimic user clicks and gestures such as sliding in order to unlock devices and conduct other functionalities; ii) *Remote Shells (RS)*, which supports our in-house implementation of an ajax-based web-based shell that can be utilized to issue a wide variety of known UNIX commands (e.g., `ls, ps, df, pwd, date`, etc.) to the Linux kernels that are located at the core of each Android device; iii) *Remote Scripting Environment (RSE)*, which allows users to author Android MonkeyRunner automation scripts (written in python) in order to quickly perform repetitive tasks on selected devices; and iv) *Remote Debug Tools (RDT)*, which provide web-based debugging extensions to the Android Debug Bridge (ADB) that is used during development.

We will start our demonstration out by overviewing the main elements of SmartLab. We shall then present the complete interaction workflow, i.e., allocate devices, transfer-files and initiate/control demonstration programs with our four modes of interaction (RCT, RS, RSE and RDT). We particularly plan to show how somebody can test both widely-known applications available through the market and interesting applications that we have developed in-house for crowd-sourced trajectory matching, optimal P2P search, distributed games, etc.

[1]Video and System available: http://smartlab.cs.ucy.ac.cy/

MobiSys'12, June 25–29, 2012, Low Wood Bay, Lake District, UK.
ACM 978-1-4503-1301-8/12/06.

3. REFERENCES

[1] Peterson L. et. al., "A Blueprint for Introducing Disruptive Technology into the Internet," *HotNets'02.*

[2] Werner-Allen G. et. al., "MoteLab: a wireless sensor network testbed," *IPSN'05.*