Demo: Fully Automated UI Testing System for Large-scale Android Apps Using Multiple Devices

Taeyeon Ki, Alexander Simeonov, Chang Min Park, Karthik Dantu, Steven Y. Ko, Lukasz Ziarek
Department of Computer Science and Engineering
University at Buffalo, The State University of New York
{tki, agsimeon, cpark22, kdantu, stevko, lziarek}@buffalo.edu

Abstract

We demonstrate AutoClicker, a fully automated UI testing system for *large-scale* Android apps using *multiple devices*. It provides a way to quickly and easily verify that a large number of Android apps behave correctly at runtime in a repeatable manner.

1. INTRODUCTION

Recently, many researchers leverage app instrumentation techniques to achieve different goals, such as improving energy efficiency for always-on sensing [4], or providing mobile deep links [3]. With compiler validation, One can easily check that instrumented apps are correctly transformed at compilation time. However, verifying if a large number of instrumented apps behave correctly at runtime is still challenging and time-consuming. This is because researchers have to schedule many apps for testing and use multiple devices in parallel in order to facilitate testing. Moreover, researchers need to understand app testing APIs and libraries, and build a system using them. To address this issue, we have developed AutoClicker, which is a fully automated UI testing system that does not require any testing scripts or manual efforts. Only a set of apps to test and a set of real devices or emulators are required.

2. ARCHITECTURE OVERVIEW

AutoClicker takes a set of apps and a set of devices as input and generates logs and result plots. AutoClicker provides command-line options such as testing time, the number of runs, runtime type (Dalvik, ART), cleanup mode, etc. It uses `adb` (Android Debugging Bridge) to control devices connected to AutoClicker. AutoClicker has six core components as shown in the Figure 1. Device Controller detects all devices including real devices or emulators connected to it, and checks device conditions such as battery level, occupied status, and so on. From this information, APK distributor selects a device and an app that is not tested yet. It installs and launches the app on a device to start testing. UI Inspector leverages Android UI Automator [1], which can inspect all UI elements and their hierarchy for an app running on a device. Using this information, Random Event Generator generates random UI events such as button clicks and text input. In the background, Log Collector captures all logs that the app under test generates. To reduce the log size, it filters out unnecessary logs. The processes mentioned above are repeated until all apps are completely tested. Then, Graph Generator draws a result plot by analyzing the logs. For each run, AutoClicker restores a predefined image to clean up for testing consistency if the cleanup mode is enabled.

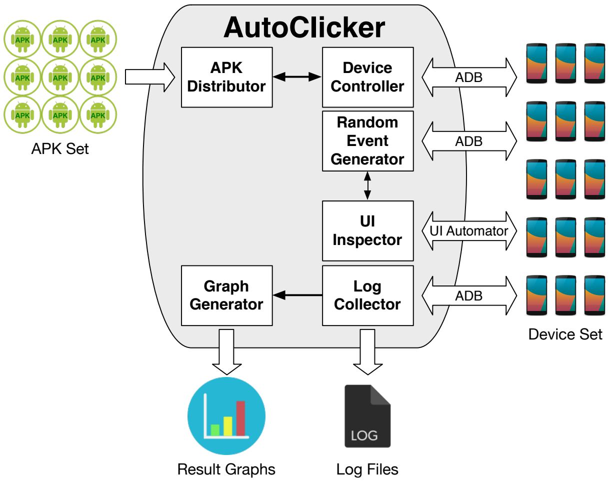

Figure 1: AutoClicker Architecture

3. DEMONSTRATION

In this demo, we will show the capabilities of AutoClicker on a normal desktop connected to fourteen Nexus 5 devices running Android 4.4 for Dalvik and Android 6.0.1 for ART. We will test 2,207 apps downloaded from Google Play 10 times for one minute. During testing, AutoClicker will verify if the runtime behavior of each app in the test set is correct. A demo video of AutoClicker can be found from [2].

4. REFERENCES

[1] Android UI Automator. https://goo.gl/xSxzoj.

[2] AutoClicker Demo. https://goo.gl/O4yweO.

[3] T. Azim, O. Riva, and S. Nath. ulink: Enabling user-defined deep linking to app content. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, 2016.

[4] H. Shen, A. Balasubramanian, A. LaMarca, and D. Wetherall. Enhancing Mobile Apps to Use Sensor Hubs Without Programmer Effort. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, 2015.

MobiSys'17 June 19-23, 2017, Niagara Falls, NY, USA

ACM ISBN 978-1-4503-4928-4/17/06.

DOI: http://dx.doi.org/10.1145/3081333.3089330