Testing iOS Apps with HadoopUnit

Rapid Distributed GUI Testing

Synthesis Lectures on Software Engineering

The Synthesis Lectures on Software Engineering publishes 75-150 page publications on all aspects of software design, engineering, and process management.

Testing iOS Apps with HadoopUnit: Rapid Distributed GUI Testing
Scott Tilley and Krissada Dechokul
December 2014

Hard Problems in Software Testing: Solutions Using Testing as a Service (TaaS)
Scott Tilley and Brianna Floss
August 2014

Model-Driven Software Engineering in Practice
Marco Brambilla, Jordi Cabot, Manuel Wimmer
September 2012

Testing iOS Apps with HadoopUnit: Rapid Distributed GUI Testing
Scott Tilley and Krissada Dechokul

ISBN: 978-3-031-01420-8 print
ISBN: 978-3-031-02548-8 ebook

DOI 10.1007/978-3-031-02548-8

A Publication in the Springer series
SYNTHESIS LECTURES ON SOFTWARE ENGINEERING #3

Series ISSN 2328-3319 Print 2328-3327 Electronic

Testing iOS Apps with HadoopUnit

Rapid Distributed GUI Testing

Scott Tilley
Florida Institute of Technology

Krissada Dechokul
Suwat Dechokul Part., Ltd.

SYNTHESIS LECTURES SOFTWARE ENGINEERING #3

ABSTRACT

Smartphone users have come to expect high-quality apps. This has increased the importance of software testing in mobile software development. Unfortunately, testing apps—particularly the GUI—can be very time-consuming. Exercising every user interface element and verifying transitions between different views of the app under test quickly becomes problematic. For example, execution of iOS GUI test suites using Apple's UI Automation framework can take an hour or more if the app's interface is complicated. The longer it takes to run a test, the less frequently the test can be run, which in turn reduces software quality.

This book describes how to accelerate the testing process for iOS apps using HadoopUnit, a distributed test execution environment that leverages the parallelism inherent in the Hadoop platform. HadoopUnit was previously used to run unit and system tests in the cloud. It has been modified to perform GUI testing of iOS apps on a small-scale cluster—a modest computing infrastructure available to almost every developer.

Experimental results have shown that distributed test execution with HadoopUnit can significantly outperform the test execution on a single machine, even if the size of the cluster used for the execution is as small as two nodes. This means that the approach described in this book could be adopted without a huge investment in IT resources. HadoopUnit is a cost-effective solution for reducing lengthy test execution times of system-level GUI testing of iOS apps.

KEYWORDS

software testing, iOS, apps, Hadoop, HadoopUnit, cloud computing, cluster

Contents

Figures

Tables

Foreword

Software quality has never been as important as it is today. A tidal wave of new software development is seen in mobile applications, which are quickly becoming ubiquitous. They are used for many purposes, from enjoyable entertainment to safety critical applications used by first responders.

At the same time, the bar to entering the mobile application marketplace is very low. Anyone with the desire to create a mobile application can do so with freely available tools in a rather short amount of time. A quick browse through the Apple App Store reveals that many applications fall short of user expectations: there are many apps in the store with three stars or fewer.

Creating a mobile application is one thing. Testing it to ensure that it works correctly, is secure, and achieves high usability factors is another thing altogether.

First, there is the plethora of mobile device platforms and appliances—each of which has its own configuration and behavior differences.

Second, the time required to test a moderately featured application just one time on one platform is substantial. Manual testing can be tedious and automated testing simulating user actions can relieve some of the tedium, but it still requires time to design and perform functional tests.

Third, mobile applications undergo constant change. The constant march of change fixes bugs and introduces new features—and new defects. Therefore, testing should include not only testing bug fixes and new features, but also regression testing of the unchanged features. However, this all takes precious time.

Finally, consider that many mobile application developers consist of small teams, perhaps just an individual, with limited funds and resources for testing. So, the customers get to be the "testers," except they use the applications for important real-world tasks like arranging travel, managing finances, tracking severe weather, and navigating roads. The customers don't consider themselves as testers. They are customers and even though they might not have paid any money for an app, they don't like to waste their time on defective ones. Also, when mobile applications fail during important tasks, defects are more than a mere inconvenience—they impact lives in a negative way.

While developers may perform bug fixes, customers often abandon apps quickly due to a bad experience. It is very easy to delete an app from an iOS device, so developers who wish to have a successful app in the App Store need to understand the detrimental impact of defects on their success.

That's why this book is important for mobile application developers and testers. HadoopUnit offers a solution for testing mobile apps on the iOS platform that is not only free, but reduces test

time dramatically. There will still be a need for usability and compatibility testing, but the more troubling defects are those that impact mobile application reliability.

Scott Tilley and Krissada Dechokul have done an excellent job with this book in describing in detail how to design and implement tests in HadoopUnit. My hope is that everyone involved in developing iOS applications will read this book and create more reliable and robust applications that get five-star reviews.

Randall W. Rice
CSTE, CSQA, CTAL
Founder, Principal Consultant, and Vice-President of Research and Development
Rice Consulting Services, Inc.

Preface

The app ecosystem is enormous. We have grown dependent on smartphone apps for almost every aspect of our lives. When the apps don't perform as expected, the consequences can range from midly irritating to life-threatening. For this reason, testing smartphone apps is very important—particularly the GUI that is the main window into the app's functionality. Unfortuanately, GUI testing can be a time-consuming process, which leads to fewer tests being run, further exacerbating the app's quality problems.

This book focuses on the specific problem of GUI testing for iOS apps found on Apple's products such as the iPhone and the iPad. Apple provides developers with a testing framework called UI Automation, but its capabilities are limited in terms of speeding up the testing process. The result is that GUI testing of complex iOS apps can take many hours.

The solution proposed here is to leverage the parallelism inherent in the Hadoop distributed platform to provide an environment for concurrent test execution. The approach builds upon an existing system, called HadoopUnit, that was previously used to reduce regression testing time of large JUnit test suites. HadoopUnit has been customized to drive UI Automation test cases in a manner that is easy for developers and testers to adopt, yet provides measurable improvement in test case execition times.

WHAT IS UNIQUE ABOUT THIS BOOK?

This book represents the continuation of research that began in 2009 on addressing the problem of execution times for large regression test suites. The first of this work were the HadoopUnit distributed exection environment and the SMART-T migration decision framework [60]. The work also led to the creation of a new community of researchers and practicioners interesting in software testing in the cloud (STITC) [59][62].

The STITC project evolved to examine the applicability of testing as a service (TaaS) to hard problems in software testing (HPST) [61][58]. TaaS is a promising new development that offers a service-oriented interface to the testing capabilities provided by an environement like HadoopUnit. The number one problem found in the HPST project was education & training, and it's currently an open question where TaaS may help alleviate this timeless challenge.

GUI testing of iOS apps is a timely update and specific instance of the classic regression testing problem, and one that HadoopUnit is well suited to address. The research reported in this book is unique is its application of an advanced environment such as HadoopUnit for concurrent

testing in manner that is accessible to almost all developers and testers—even if they are of modest means. A simple two-node cluster is all that is needed to realize significant testing benefits.

WHO SHOULD READ THIS BOOK?

Anyone who is involved in GUI testing of iOS apps will find the material presented in this book valuable. This is particularly true for testers, but developers, managers, and even end-users can benefit from understanding the challenges faced when using the UI Automation framework and the possible benefits of using the customized HadoopUnit to address these challenges.

Modern software engineering—and app development in particular—involves the use of sophisticated IDEs and integrated coding platforms. Many of these tools are moving to the cloud. An understanding of how Hadoop can be used in the domain of GUI testing provides valuable insight into the power of the MapReduce programmign paradigm.

OUTLINE OF THE BOOK

Chapter 1 discusses the challenges of GUI testing with UI Automation, outlines prior results for rapid testing using HadoopUnit, and outlines related work in the areas of GUI testing tools and distributed testing platforms. Chapter 2 provides background information on the challenges of software testing in general, and GUI testing of iOS apps with UI Automation in particular, and summarizes the Hadoop platform and the HadoopUnit distributed test execition environment. Chapter 3 outines how HadoopUnit can be used to drive the UI Automation framework to faciliate parallel test execution. Chapter 4 details three experiments in rapid GUI testing of iOS apps using the customized HadoopUnit. Lastly, Chapter 5 summarizes the main results, objectives, and contributions of this work and outlines possile avenues of further investiation.

The book also contains two appendices. Appendix A describes how to set up a HadoopUnit cluster on Mac OS X. Appendix B provides source code samples for HadoopUnit, suitably modified for iOS GUI testing with UI Automation.

Scott Tilley
Melbourne, FL

Krissada Dechokul
Bangkok, Thailand

October 2014

Acknowledgments

We are indebted to everyone who helped develop HadoopUnit—the platform upon which this research is built: Tauhida Parveen, Eric Bower, colleagues at Yahoo!, collaborators at SAP, and members of the global STITC community.

Our thanks to Apple for making the excellent UI Automation framework available for free as part of the Xcode development environment.

We appreciate the the invaluable comments privided by the book's reviewers. Their suggestions helped improve the text. Any remaining errors or omissions are ours alone.

We are grateful to the Florida Institute of Technology for supporting this research.

Finally, our gratitude to Morgan & Claypool for their guidance and patience in helping us publish the results of our work.

Dedication

To Miel
— Scott Tilley

To my parents
— Krissada Dechokul