

SpringerBriefs in Computer Science

For further volumes:
<http://www.springer.com/series/10028>

Scott Tilley · Tauhida Parveen

Software Testing in the Cloud

Migration and
Execution

Scott Tilley
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL
USA

Tauhida Parveen
Independent Consultant
Melbourne, FL
USA

Additional online material available under www.stitc.org and www.springer.com/978-3-642-32121-4

ISSN 2191-5768 ISSN 2191-5776 (electronic)
ISBN 978-3-642-32121-4 ISBN 978-3-642-32122-1 (eBook)
DOI 10.1007/978-3-642-32122-1
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2012943956

ACM Computing Classification (1998): D.2, K.6

© The Author(s) 2012

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my parents

—Scott Tilley

To mom, dad, and grandpa

—Tauhida Parveen

“It’s testing, Jim, but not as we know it”

Foreword

This book fills a much-needed gap in research and guidance for advanced practitioners and scholarly investigators. It focuses on the neglected, but important topic of software testing in the cloud. Testing in the cloud is far more complex than standard software testing because resources constantly shift for different executions, many of the resources are not controlled by the consuming organization, and the business case for success is significantly different for cloud providers and consumer organizations. However, it is precisely for these reasons that testing of potential cloud applications can make or break the business case for migration to the cloud.

Testing with the use of cloud resources can provide substantial potential benefits because of the potential for large-scale economies of scale and the ability to access state of the practice computer resources without a substantial IT investment. However, it is not a solution for every organization; nor are its benefits automatic. Because of these factors, testing in the cloud requires a sober strategic and technical analysis that fully identifies goals of the effort, total costs, potential risks, and expected outcomes. This type of analysis, while still not commonplace, can demonstrate its worth by enabling decisions to be made with facts and logic rather than emotion. This book introduces a set of methods and tools to begin systematic testing in the cloud.

During the 15 years that I have been associated with Prof. Scott Tilley, he has consistently tackled hard problems with a rare combination of tight discipline, pragmatic application, and strong technical insight. This book is no exception. Scott recognized the potential of cloud computing when it was merely a speck on the horizon, and he began to organize a community of interest around the topic of testing in the cloud. Scott's collaboration with Dr. Tauhida Parveen has been fruitful in the writing of this book and the development of a community of interest. Tauhida has co-developed, implemented, and refined the SMART-T decision framework and the HadoopUnit execution environment over the past several years.

Each of the three major sections of the book illustrate how the co-authors routinely improve on ideas, methods, and tools and then apply them to novel situations. The SMART-T framework provides the decision-making backbone for

the rest of the book. This method extends the SMART framework that several colleagues and I developed at Carnegie Mellon's Software Engineering Institute. The extensions to cloud testing provide a unique foundation for making informed business, technical, and operational decisions.

While the SMART-T framework enables rational decision-making, HadoopUnit builds on Apache Hadoop, which is a framework for supporting large-scale distributed applications. HadoopUnit provides a distributed environment for concurrent execution of testing on a cluster of machine using cloud resources. This enables empirical verification of the decisions that are made, detailed data on whether the anticipated economies of scale in the use of cloud testing are being realized, and early feedback on whether mid-course corrections are needed.

The five case studies represent the third leg of this tightly woven book. They demonstrate promising early results and provide practical lessons learned in migrating testing to the cloud through a combination of SMART-T and HadoopUnit. These cases studies, which were developed to vary both in complexity and in the nature of the application, measure the relative performance improvement of using the cloud in testing as opposed to single machine testing. I found the following lessons learned to be particularly compelling for advanced practitioners:

- the wisdom of factoring in sufficient time to master the complexities of the current testing environment,
- the advisability of planning for partial and phased migrating testing to the cloud,
- the importance of the selection of test cases to take advantage of the performance improvement from the use of HadoopUnit with large test suites,
- the need to plan for the challenges of selection, configuration, and automation of software because of the lack of visibility into key infrastructure elements that is inherent in testing in the cloud,
- the necessity of detailed analyses to determine the causes for failures in the execution of test cases because of the multitude of factors that may be causing the failures, such as bandwidth, network latency, and other configurations.

For researchers, this book offers an intellectual foundation that provides a set of building blocks for future research programs. For practitioners considering a migration to the cloud, this book, with its focus on strong intellectual grounding and practical applications, should be required reading. The empirical applications serve the interests of both the practitioner and research communities and provide a much-needed practical basis for the methods and tools that are introduced.

Dennis Smith
Software Engineering Institute
Carnegie Mellon University

Preface

This book is focused on one of the most important and long-standing problems facing the software testing community: lengthy test execution times. For a large, complex system, the amount of test cases in a regression test suite can range from a few hundred to several thousands, which can take hours or even days to execute. Regression testing also requires considerable resources that are often not readily available, precluding their use in an interactive setting, further contributing to an inefficient testing process.

Traditional approaches to reduce the execution time for regression testing typically focus on excluding selected tests from the suite that need to be run after a change is made to the system. This research provides an alternate solution that does not require test case exclusion: leveraging the resources provided by cloud computing infrastructure to facilitate the concurrent execution of test cases. Cloud computing offers the use of virtualized hardware, effectively unlimited storage, and software services that can aid in reducing the execution time of large test suites in a cost-effective manner.

This book presents a decision framework called SMART-T to support migration of software testing to the cloud, a distributed environment called HadoopUnit for concurrent execution of test cases in the cloud, and a series of case studies illustrating the use of the framework and the environment. Experimental results indicate a significant reduction in test execution time is possible when compared with a typical sequential environment.

What Is Unique About This Book?

Our work in software testing in the cloud began in 2009 with a collaborative project with a large industrial partner. In April 2010, we organized the workshop on the topic at the IEEE ICST conference in Paris [117]. The interest generated from this event resulted in the edited volume, *Software Testing in the Cloud: Perspectives on an Emerging Discipline* [118].

Our own focus has been primarily on test execution. To facilitate this, we also examined the precursor issue of migrating testing to a cloud-based environment. Accordingly, the unique work described in this book has three primary objectives:

1. Create a decision framework that facilitates the disciplined migration of software testing to the cloud.
2. Develop a prototype distributed execution framework for concurrent test execution in the cloud.
3. Demonstrate the viability and evaluate the effectiveness of the approach using case studies carried out in real world settings.

This work has the potential to significantly impact the three major areas of software testing, cloud computing, and system migration. The results reported in the book affirm that a new area of research can benefit from a judicious blending of best practices from the others. We illustrate the need for a decision framework for migrating testing to the cloud and provide the testing community a sample decision framework that they can use. We also demonstrate that Hadoop, typically used for distributed data processing at scale, can also be beneficial when used in software testing.

The intellectual merit of this research is the advancement of knowledge and understanding of the nascent field of software testing in the cloud. A creative blend of the underlying areas of software testing, cloud computing, and system migration was used to develop this potentially transformative area of research. Our results show that when migrating software testing is desirable from a business point of view, and feasible from a technical point of view, the results can be acceptable from an operational point of view. For example, as described in [Chap. 4](#), a 165x performance improvement was realized after migrating test cases for GCC to the cloud.

Who Should Read This Book?

Software testing in the cloud is a subject that should interest advanced practitioners, academic researchers, and those in education and training. For advanced practitioners, the issue of cloud computing and its impact on the field of software testing is becoming increasingly relevant. For academic researchers, this is a subject that is replete with interesting challenges; there are so many open problems that graduate students will be busy for years to come.

The topic of software testing in the cloud is an excellent vehicle for educational exploration. For example, it builds upon core knowledge of software testing while introducing new concepts such as cloud computing into the course material. This topic was used in a graduate course on software engineering with considerable success: the students seemed to enjoy the exposure to a leading-edge research program, and the instructors benefits from the contributions of the students in the

assignments and group project—which closely followed the main themes presented in this book [116].

To disseminate the results of this work, a community of interest was created called “Software Testing in the Cloud” (www.STITC.org). This community has been active in working together and sharing ideas in migrating testing to the cloud, performing testing in the cloud, and testing of the cloud itself. We encourage you to contact us and get involved in this exciting new area.

Outline of This Book

The book is structured according to the three key elements of our research: migrating software testing to the cloud using SMART-T, executing test cases in the cloud using HadoopUnit, and performing case studies to illustrate the use of the approach in a variety of problem settings.

SMART-T: [Chapter 2](#) describes SMART-T, a decision framework for migrating testing in the cloud. The descriptive framework has three parts: (1) addressing business drivers (why migrate to the cloud); (2) technical factors (when to migrate to the cloud); and (3) operational results (how to migrate to the cloud). Each part of the framework ends in a decision point related to the desirability (business), feasibility (technical), and acceptability (operational) of the migration to the cloud.

HadoopUnit: [Chapter 3](#) explains HadoopUnit, a distributed environment for concurrent execution of test cases in the cloud. HadoopUnit is built upon Hadoop, the open source implementation of the MapReduce programming model. Several techniques were illustrated to migrate test cases from a legacy environment to the cloud-based environment, such as wrapping test cases using JUnit interfaces.

Case Studies: [Chapter 4](#) presents several case studies where SMART-T and HadoopUnit has been applied. Experimental results indicate a significant reduction in test execution time is possible when compared with a typical sequential environment. The results also showed that migrating testing to the cloud is not always the best solution to the problem; sometimes a different approach might be needed, such as reengineering the test cases to identify shortcomings before any migration is attempted.

The book concludes with [Chap. 5](#), which provides a summary of the results from the case studies and summarizes our lessons learned.

Acknowledgments

We are extremely grateful for the wonderful personnel at Springer who helped make this book a reality. In particular, we would like to thank Ralf Gerstner for his patience and encouragement.

We gratefully acknowledge the invaluable cooperation of our many collaborators and members of the STITC community. This work began as a joint project and it resulted in the creation of a new area of research.

Melbourne, FL, May 2012

Scott Tilley
Tauhida Parveen

Contents

1	Introduction	1
1.1	Software Testing	1
1.1.1	Regression Testing	2
1.1.2	Unit Testing	4
1.1.3	Distributed Testing	5
1.2	Cloud Computing	5
1.2.1	Cloud Services	7
1.2.2	Cloud Providers	8
1.2.3	Cloud Types	10
1.3	System Migration	11
1.3.1	Legacy Systems	11
1.3.2	Migration Techniques	12
1.3.3	SMART	13
1.4	The Problem of Lengthy Test Execution Times	15
1.5	Software Testing in the Cloud	16
2	SMART-T: Migrating Testing to the Cloud	19
2.1	SMART-T	19
2.2	Why Migrate Testing to the Cloud	21
2.2.1	Making Testing Faster	22
2.2.2	Making Testing Cheaper	23
2.2.3	Making Testing Better	24
2.3	When to Migrate Testing to the Cloud	26
2.3.1	Characteristics of the Application	27
2.3.2	Types of Testing	28
2.3.3	Cloud-Based Test Execution Environment	30
2.4	How to Migrate Testing to the Cloud	31
2.4.1	Trial Migration	32
2.4.2	Actual Migration	33
2.5	Summary	34

3	HadoopUnit: Test Execution in the Cloud	37
3.1	Hadoop	37
3.1.1	Hadoop Distributed File System	39
3.1.2	MapReduce	40
3.1.3	How Hadoop Works	43
3.2	Design of HadoopUnit	45
3.2.1	Test Case Extraction	46
3.2.2	The Map Function	46
3.2.3	The Reduce Function	47
3.3	Implementation of HadoopUnit	47
3.4	Using HadoopUnit	49
3.4.1	Install HadoopUnit on Master	49
3.4.2	Configure HadoopUnit for Test Execution	50
3.4.3	Execute Test Cases Using HadoopUnit	51
3.5	Summary	52
4	Case Studies	55
4.1	JUnit	56
4.1.1	Generated Test Cases	56
4.1.2	Hadoop	59
4.1.3	Spring	63
4.2	GCC	64
4.2.1	Business Drivers	64
4.2.2	Technical Factors	65
4.2.3	Trial Migration	65
4.2.4	Actual Migration	68
4.2.5	Results	68
4.3	XYZ	70
4.3.1	Business Drivers	70
4.3.2	Technical Factors	70
4.3.3	Discussion	73
4.4	Threats to Validity	74
4.4.1	The Application Under Test	74
4.4.2	The Type of Tests Executed	76
4.4.3	Changes Made During Migration	77
4.5	Summary	78
5	Conclusions	79
5.1	Summary of Results	79
5.1.1	Migrating to the Cloud	80
5.1.2	Test Execution in the Cloud	80
5.2	Lessons Learned	82
5.2.1	Migrating Testing to the Cloud	82
5.2.2	Test Execution in the Cloud	83
5.3	Closing Remarks	85

Contents	xv
Appendix: SMART-T Questionnaire	87
References	95
About the Authors.	103