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
Software Testing in the Cloud
Migration and Execution
Tilley, S.; Parveen, T.
2012, XV, 103 p. 40 illus., Softcover
ISBN: 978-3-642-32121-4