Chapter 14

Testing of Future Internet Applications Running in the Cloud

Tanja Vos
Universidad Politécnica de Valencia, Spain

Paolo Tonella
Fondazione Bruno Kessler, Italy

Joachim Wegener
Berner & Mattner, Germany

Mark Harman
University College London, UK

Wishnu Prasetya
University of Utrecht, The Netherlands

Shmuel Ur
Bristol University, UK

ABSTRACT

The cloud will be populated by software applications that consist of advanced, dynamic, and largely autonomic interactions among services, end-user applications, content, and media. The complexity of the technologies involved in the cloud makes testing extremely challenging and demands novel approaches and major advancements in the field. This chapter describes the main challenges associated with the testing of applications running in the cloud. The authors present a research agenda that has been defined in order to address the testing challenges. The goal of the agenda is to investigate the technologies for the development of an automated testing environment, which can monitor the applications under test and can react dynamically to the observed changes. Realization of this environment involves substantial research in areas such as search based testing, model inference, oracle learning, and anomaly detection.

INTRODUCTION

The Future Internet (FI) will be a complex inter-connection of services, applications, content and media running in the cloud. It will offer a rich user experience, extending and improving current hyperlink-based navigation. Key technologies contributing to the development of FI services and applications include a rich, complex, dynamic and stateful client. This client interacts asynchronously with the server, where applications are organized as services and run in the cloud, taking advantage
of dynamic service discovery, replacement and composition. Adaptivity and autonomy improve the user experience, by dynamically changing both the client and the server side, through capabilities such as self-configuration and self-healing. As a consequence, FI applications will exhibit emergent behavior which makes them hard to predict.

Our society will become increasingly dependent on services built on top of this complex and emerging Future Internet. Critical activities such as public utilities, social services, government, learning, finance, business, but also entertainment will depend on the underlying software and services. As a consequence, the applications running on top of the Future Internet will have to meet high quality and dependability demands. Not only the functional quality aspect is important, but non-functional aspects like performance, security, and privacy will become increasingly more important. All these make verification and validation for quality assurance of FI applications extremely important. In this chapter, we discuss how to address the FI testing challenges, by describing the features of an integrated environment for continuous evolutionary automated testing, which can monitor the FI application under test and adapt to the dynamic changes observed. FI testing will require continuous post-release testing since the application under test does not remain fixed after its release. Services and components could be dynamically added or even programmed by customers and the intended use could change significantly. Therefore, testing should be done continuously after deployment to the customer, either in vitro or in vivo. The testing environment we describe integrates, adapts and automates techniques for continuous FI testing (e.g. dynamic model inference, log-based diagnosis, oracle learning, classification trees and combinatorial testing, concurrent testing, regression testing). To make it possible for the above mentioned techniques to deal with the huge search space associated with FI testing, evolutionary search based testing will be used. Search-based algorithms will be used to guide the solution of identified problems so as to optimize properly defined objective functions. In this way, we can address the ultimate challenge of FI applications: testing unexpected behavior that may originate from the dynamism, autonomy and self-adaptation involved.

**BACKGROUND**

FI testing demands for major advancements in several areas of software testing. We discuss the state of the art in each of these areas separately, in the following.

**Beyond the State of the Art of Search Based Techniques**

The current state of the art in search based techniques is described in (Harman, 2007; Harman & Afshin, 2010). The area of testing is the most prominent software engineering domain for the application of search techniques. Search based testing techniques have been applied to various real world complex systems (e.g., embedded systems) (Vos et al., 2010; Baars et al., 2010) to deal with automated test case generation for structural (white-box) as well as functional (black-box) testing. Also the testing of various non-functional properties have been investigated (Afzal et al., 2009). While these testing targets remain relevant for FI applications as well, the continuous, autonomous testing framework that we envision introduces new opportunities for search based exploration of the solution space. Correspondingly, novel fitness function definitions and search algorithms will be required.

Innovative approaches to genetic programming applied to testing may also contribute to FI testing. So far, genetic programming has received limited attention in testing. It has been successfully used to conduct unit testing of object oriented code, by providing a simple and effective mechanism to bring the object under test to a proper internal