Chapter 14
Concolic Test Generation and the Cloud:
Deployment and Verification Perspectives

Nikolai Kosmatov
CEA LIST, France

ABSTRACT

Software testing in the cloud can reduce the need for hardware and software resources and offer a flexible and efficient alternative to the traditional software testing process. A major obstacle to the wider use of testing in the cloud is related to security issues. This chapter focuses on test generation techniques that combine concrete and symbolic execution of the program under test. Their deployment in the cloud leads to complex technical and security issues that do not occur for other testing methods. This chapter describes recent online deployment of such a technique implemented by the PathCrawler test generation tool for C programs, where the author faced, studied, and solved many of these issues. Mixed concrete/ symbolic testing techniques not only constitute a challenging target for deployment in the cloud, but they also provide a promising way to improve the reliability of cloud environments. The author argues that these techniques can be efficiently used to help to create trustworthy cloud environments.

INTRODUCTION

Testing is nowadays the primary way to improve the reliability of software. Software testing accounts up to 50% of the total cost of software development. Automatic testing tools provide an efficient alternative to manual testing and reduce the cost of software testing. However, automatic testing requires considerable investments: purchase and installation of testing tools, additional computing resources to run these tools, employing or training competent validation engineers to

maintain and operate them, etc. These resources are necessary only during the testing steps of software development, and their cost for the company outside this period can be avoided by sharing them between several projects and with other companies.

The paradigm of cloud computing brings obvious benefits for the software testing process. The deployment of software testing services in the cloud makes them easily available for different companies and projects and allows their on-demand usage. The companies do not have

to purchase and maintain powerful servers and testing tools all the time, but use them just when it is required.

On the other hand, for the providers of testing tools, this approach makes it easier to update and to support the tools and to provide flexible on-demand solutions to the clients. Various testing tasks, taking from several seconds up to several weeks, can be optimally scheduled in the cloud. Thus a testing service can be offered to a larger number of companies and becomes appropriate for testing software of almost any size.

Before testing in the cloud becomes widely accepted and used in industry, various technical, security and privacy protection issues must be resolved. In this chapter, we focus on test generation techniques combining concrete and symbolic execution, also known as concolic testing. Concolic testing is an advanced technique of structural unit testing, that is one of the most suitable kinds of testing for the cloud (Parveen & Tilley, 2010). We address two facets of concolic testing in the cloud: migrating concolic testing to the cloud and usefulness of concolic testing for the cloud. The deployment of concolic testing in the cloud raises particularly challenging technical and security problems that do not necessarily appear in other testing methods. Relevant to any version of concolic testing, the security and efficiency concerns become even more critical for a publicly available testing service in the cloud. While for a local deployment used by a restricted number of people, an intentionally malicious software is very unlikely to be submitted to the tool, a publicly available testing service runs a much greater risk. We have recently implemented and deployed an online version for such a technique where we faced, studied and solved many of these problems. We show how a concolic testing tool can be decomposed into safe and unsafe parts in order to preserve the efficiency of the method, and how the unsafe part can be secured. On the other hand, concolic testing provides an excellent means for improving the reliability of the cloud itself. We

present the most recent results on verification of operating systems and cloud hypervisors and show the role of various concolic testing approaches for creating more reliable cloud environments.

We start by providing some background on testing in the cloud and migrating software testing to the cloud. Next, we give an overview of concolic testing tools and outline their main features. We describe the method of PathCrawler, a concolic testing tool for C programs. Some implementation issues, that are usually omitted, will be described here in detail in order to illustrate the particular problems that this technique raises for an implementation in the cloud. Next, we describe the problems we faced, and provide the solutions we found during the implementation of PathCrawler-online, a prototype of testing service in the cloud whose limited evaluation version is available at (Kosmatov, 2010b). This web service implements all basic features of an online test generation service: uploading a C program to be tested, customizing test parameters and an oracle, generating and executing test cases on the program under test, and providing to the user the generated test cases and test coverage statistics. Next, we present recent results on verification of operating systems and cloud hypervisors. We underline several successful applications of concolic testing illustrating how this testing technique can improve the reliability of cloud environments. We finish by pointing out future work directions and a conclusion.

TESTING IN THE CLOUD

Cloud computing is an emerging paradigm (Zhang, Cheng, & Boutaba, 2010). The term *cloud* became popular after Google's CEO Eric Schmidt used the word to describe the business model of providing services across the Internet in 2006. In mid-2009, the U.S. National Institute of Standards and Technology (NIST) gave the following definition of the concept (Mell & Grance, 2011): Cloud computing is a model for enabling convenient, on-demand

Related Content

An IoT-Based Framework for Health Monitoring Systems: A Case Study Approach
N. Sudhakar Yadav, K. G. Srinivasaand B. Eswara Reddy (2019). *International Journal of Fog Computing (pp. 43-60).*
www.irma-international.org/article/an-iot-based-framework-for-health-monitoring-systems/219360

A Tutorial on CloudStack: CloudStack
Srinivasa K. G., Nishal Ancelette Pereira, Akshay K. Kallianpurand Subramanya E. Naligay (2018). *Design and Use of Virtualization Technology in Cloud Computing (pp. 233-242).*
www.irma-international.org/chapter/a-tutorial-on-cloudstack/188131

IoT-Fog-Blockchain Framework: Opportunities and Challenges
Tanweer Alam (2020). *International Journal of Fog Computing (pp. 1-20).*
www.irma-international.org/article/iot-fog-blockchain-framework/266473

A Survey on Privacy Preserving and Trust Building Techniques of Blockchain-Based Systems
G. Balamurugan, Amit Kumar Tyagiand Richa (2023). *Privacy Preservation and Secured Data Storage in Cloud Computing (pp. 430-457).*
www.irma-international.org/chapter/a-survey-on-privacy-preserving-and-trust-building-techniques-of-blockchain-based-systems/333149

Resource Allocation With Multiagent Trading Over the Edge Services
Yee-Ming Chenand Chung-Hung Hsieh (2022). *International Journal of Fog Computing (pp. 1-11).*
www.irma-international.org/article/resource-allocation-with-multiagent-trading-over-the-edge-services/309138