Chapter 55

Leveraging the Cloud for Large–Scale Software Testing:
A Case Study – Google Chrome on Amazon

Anjan Pakhira
University of Newcastle upon Tyne, UK

Peter Andras
University of Newcastle upon Tyne, UK

ABSTRACT

Testing is a critical phase in the software life-cycle. While small-scale component-wise testing is done routinely as part of development and maintenance of large-scale software, the system level testing of the whole software is much more problematic due to low level of coverage of potential usage scenarios by test cases and high costs associated with wide-scale testing of large software. Here, the authors investigate the use of cloud computing to facilitate the testing of large-scale software. They discuss the aspects of cloud-based testing and provide an example application of this. They describe the testing of the functional importance of methods of classes in the Google Chrome software. The methods that we test are predicted to be functionally important with respect to a functionality of the software. The authors use network analysis applied to dynamic analysis data generated by the software to make these predictions. They check the validity of these predictions by mutation testing of a large number of mutated variants of the Google Chrome. The chapter provides details of how to set up the testing process on the cloud and discusses relevant technical issues.

INTRODUCTION

Large-scale software (Goth, 2008) usually integrates a very large number of components developed over many years, possibly by different companies or different units of large software companies. This makes understanding testing and maintenance of such software systems naturally very difficult with plenty of opportunities for leaving unnoticed integration bugs (i.e. pieces of software code that induce undesired behaviors of the software) in the software (Dadzie, 2005; Meyer, 2008; NIST & RTI, 2002; Porter, Siy, Toman, & Votta, 1997; Wolverton, 1974).

Such software is likely to contain millions of lines of code, written by many software developers, and tested in various contexts at component level. While the syntactic, semantic and general logical correctness of components can be assumed to large extent, the system-scale testing of such software is usually patchy due to the quick combinatorial explosion of the space of possible test cases and the high costs associated with detailed testing. Time constraints driven by commercial objectives also act against the realization of the ideal wide-scale testing of the software.

Cloud computing (Armbrust et al., 2009; Buyya, Shin, & Venugopal, 2008) has been introduced recently as very large scale remote computing service based on virtualization technology (Barham et al., 2003; Nanda & Chiueh, 2005). The computing cloud offers the possibility of running large number of virtual computers in parallel in a flexible manner, such that computing resources come on-line as they are needed and when they are not needed the overall resource set scales back automatically. Cloud computing provides flexible storage and processing resources that are accessible remotely through a relatively easy to use interface. Commercial cloud services currently provided by Amazon, Rackspace, Microsoft and Google amongst others; offer considerable flexibility in terms operating systems and other software that are available for use on the virtual machines. These services are currently offered at a low unit price for usage, which makes it attractive to use cloud instead of investing in captive infrastructure of similar capability.

The advent of cloud services in the form of platform-as-a-service (PaaS) like Amazon's EC2 and S3, makes it possible for these platforms to be fashioned for a variety of uses (computational, storage or both). Predominant amongst these is relatively cheap commercial shared scalable web-hosting, where it is possible to scale the number of web servers as required. This flexibility and scalability of use has led to some exploratory research on how cloud resources may be leveraged for software testing (Candea, Bucur, & Zamfir, 2010; Joglekar, 2009).

In theory it may be argued that by virtue of operational flexibility, and ability of the user to scale resource allocation and usage on demand, the cloud can benefit some types of software testing. Wide scale software testing requiring substantial compute and storage resources and where the testing process can be managed (automated) through a workflow needing minimal human intervention is representative of type of testing that may benefit from the use of the cloud. Such testing is generally not practicable on typical desktop computers, due to limitation of resource scalability and consequently time and cost limitations.

In this case study we present the use of cloud computing to support wide-scale testing of large software systems. The objective of our experiment (that forms the context of this case study) is to explore the notion that complex network analysis of software dynamic analysis data may be used to detect functionally important elements of object oriented software systems. The reported work is part of a series of experiments based on the premise that software systems can be viewed as complex networks, and software engineering may therefore benefit from analysis based on that viewpoint. The ultimate aim of these experiments is to verify applicability of network analysis methods to analyze software engineering data, so that these methods may be adopted for software analysis to support software testing.

Our approach to verification is based on black box testing, with a form of fault seeding evaluation and evaluation through a mutation scoring (Abran, Moore, Bourque, & Dupuis, 2004). In this experiment network analysis is applied to predict functional importance of methods of classes in Google Chrome web browser. Efficient and cost effective verification of these predictions required the use of cloud for generating Google Chrome mutants.

1176

27 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/leveraging-the-cloud-for-large-scale-software-testing/119903

Related Content

Volunteer Computing on Mobile Devices: State of the Art and Future Research Directions
Cristiano Tapparello, Colin Funai, Shurouq Hijazi, Abner Aquino, Bora Karaoglu, He Ba, Jiye Shiand Wendi Heinzelman (2015). *Enabling Real-Time Mobile Cloud Computing through Emerging Technologies (pp. 153-181).*
www.irma-international.org/chapter/volunteer-computing-on-mobile-devices/134205

Evaluating the Performance of Monolithic and Microservices Architectures in an Edge Computing Environment
Nitin Rathoreand Anand Rajavat (2022). *International Journal of Fog Computing (pp. 1-18).*
www.irma-international.org/article/evaluating-the-performance-of-monolithic-and-microservices-architectures-in-an-edge-computing-environment/309139

Development of Community Based Intelligent Modules Using IoT to Make Cities Smarter
Jagadish S. Kallimani, Chekuri Sailusha, Pankaj Latharand Srinivasa K.G. (2019). *International Journal of Fog Computing (pp. 1-12).*
www.irma-international.org/article/development-of-community-based-intelligent-modules-using-iot-to-make-cities-smarter/228127

Fog Computing Qos Review and Open Challenges
R. Babu, K. Jayashreeand R. Abirami (2018). *International Journal of Fog Computing (pp. 109-118).*
www.irma-international.org/article/fog-computing-qos-review-and-open-challenges/210568

Industry 4.0 in Manufacturing, Communication, Transportation, Healthcare
R. Felista Sugirtha Lizy, Ibrahim M. H.and Chinnadurai Manthiramoorthy (2024). *Emerging Technologies and Security in Cloud Computing (pp. 422-450).*
www.irma-international.org/chapter/industry-40-in-manufacturing-communication-transportation-healthcare/339411