

A Systematic Mapping Study of Empirical Studies on Software Cloud Testing Methods

Amro Al-Said Ahmad, Pearl Brereton, Peter Andras

School of Computing and Mathematics, Keele University, Staffordshire, United Kingdom
{a.m.k.al-said.ahmad, o.p.brereton, p.andras} @keele.ac.uk

Abstract—Context: Software has become more complicated, dynamic, and asynchronous than ever, making testing more challenging. With the increasing interest in the development of cloud computing, and increasing demand for cloud-based services, it has become essential to systematically review the research in the area of software testing in the context of cloud environments. **Objective:** The purpose of this systematic mapping study is to provide an overview of the empirical research in the area of software cloud-based testing, in order to build a classification scheme. We investigate functional and non-functional testing methods, the application of these methods, and the purpose of testing using these methods. **Method:** We searched for electronically available papers in order to find relevant literature and to extract and analyze data about the methods used. **Result:** We identified 69 primary studies reported in 75 research papers published in academic journals, conferences, and edited books. **Conclusion:** We found that only a minority of the studies combine rigorous statistical analysis with quantitative results. The majority of the considered studies present early results, using a single experiment to evaluate their proposed solution.

Keywords-systematic mapping study; cloud software testing methods; software testing; empirical studies

I. INTRODUCTION

Systematic review (SR) is a methodology that aims to be reliable, exhaustive, and auditable to allow researchers to collect evidence on a particular research question, topic area, or subject of interest [1][2]. The SR plays a major role in supporting academic research as well as enriching practices in software engineering [3]. The SR process starts with the development and validation of a review protocol [1]. The review protocol provides a plan for the process of conducting a review, including study selection and data extraction, with the aim to answer the research questions [4]. The protocol preparation is followed by locating potentially relevant studies in an automatic or manual way, selecting primary studies based on inclusion and exclusion criteria, extracting data, and reporting the SR, including its limitations [1]. A mapping study is a form of SR which provides a classification of the relevant research for a particular subject without necessarily assessing the quality of each study [1].

Software testing is one of the main technical activities in the software development cycle, which consumes more than 30% of a project's budget, effort, and time [5]. When the budget and time are not sufficient to cover all test cases, suites, and scenarios, an efficient strategy that involves tools and technical solutions will be key to enhancing and speeding up the testing process.

Cloud computing provides integrated services that help to create an environment for speeding up the development process by allowing organizations to transfer some of the development processes -such as testing, deployment, installations, and tracking failures- into the cloud. In the context of testing, cloud computing has been described as a resource that offers virtualization, storage, and software services that can reduce the time and cost of managing and applying large test suites [8]. Virtualization can be used in large-scale testing [9], and the cloud can support on-demand test laboratories [10]. Furthermore, it can be used for auto-run and management of test suites [11]. On the other hand, the cloud has changed the way services are delivered. As cloud-based services have grown in popularity, so has the need for testing those services.

This work presents a mapping study which addresses the functional and non-functional testing methods on/using cloud-based services. The study provides an overview of primary studies, published in the period of 2010-2015, that evaluate cloud testing methods. Our methodology is based on a well-defined protocol to build a structure and classification scheme to analyze the research area of cloud-based testing. Our mapping study collected 247 research papers from which a total of 69 primary studies reported in 75 research papers were selected. We look at how methods are applied, and what is being tested using those methods. Several papers that report the same study are included as a group. We identify each study using the notation [S+ID] where ID is the numeric identifier of the study- the studyID is included at the end of the bibliographic data of each appropriate paper in the reference list.

The rest of the paper is organized as follows: Section II presents related work. A description of the research methodology is presented in Section III, while Section IV presents and analyses the mapping study results. Finally, we present our discussion and conclusions in Section V.

II. RELATED WORK

There have been a number of literature surveys and reviews and one mapping study within the software cloud testing area. A systematic mapping study is reported in two research papers, [6][12], using the 5W+1H model for reporting systematic reviews. Studies are categorized based on research questions, authors and countries, research objectives, research ideas, patterns of papers on different types of cloud service and publication type, immediacy of article citation, and article inter-relevance. The mapping study does not include clear inclusion/exclusion criteria, however. Further, the study covers published papers dated during the period 2010-2012.

Some literature surveys have been published in conferences and journals. In particular, one study focuses on publications dated during the period 2009-2012 and classifies relevant literature according to the type of testing activities for cloud services and the type of application domains [7]. An overview of research related to cloud testing tools, types, and challenges, and a comparison of testing tools are presented in [13]. A survey that identifies the need for cloud testing tools and presents the current testing methods and tools has also been published [14]. Studies [15][16] provide an overview of software testing as a service (TaaS), while literature survey [17] highlighting the current situation of security measurement and testing on the cloud. Study [18] discusses SaaS testing on the cloud, including tools, issues, challenges, and needs.

We checked that all of the primary studies that are reported in these reviews were located by our search process and either complied with our inclusion criteria or were excluded based on our selection criteria. We found no comprehensive mapping studies about cloud software testing.

III. RESEARCH METHOD

Here we describe our systematic mapping methodology, as recommended by [1], to provide an overview of empirical studies about cloud software testing methods, to answer the research questions, and reveal the current situation regarding the research topic. A protocol was developed and updated throughout the study.

A. Research questions

The major focus of this study is to determine and classify the available information regarding functional and non-functional cloud testing methods, and the subject and attributes of the testing methods. The research questions addressed by this mapping study are:

Question 1: What types of functional and non-functional testing methods have been evaluated on/using cloud-based services?

Question 2: How were these testing methods applied, and what was being tested?

B. Search and Selection Process

An initial search was performed using ScienceDirect, ACM Library, IEEE Xplore, Springer, and Wiley, identifying publications' sources and dates for the topic of the study. This led to the selection of relevant journals and conference proceedings, as well as the targeted publication period (2010-2015).

The search strategy included manual and automatic searches. Relevant high-ranking (based on ISI impact factor and reputation) journals and conference proceedings were selected in the domains of software testing or cloud computing. Some high-ranking magazines such as IEEE Software were excluded, because no empirical studies related to cloud software testing methods were found during the initial search, and the search and selection process stages.

A manual search is more time-consuming than an automated search. However, it gives better completeness in

terms of the number of relevant studies found [1]. Two manual searches were conducted: one in the journals, and the other in the conference proceedings. An automated search of the International Conference on Software Engineering (ICSE) proceedings and the IEEE Cloud community conferences proceedings was conducted. The search strings were: "(Cloud OR Cloud services) AND (Testing)" and "(Testing Cloud services)". The snowballing method was used at the end of the second stage of the selection process in order to find more primary studies. The selected journals are: Automated Software Engineering; Journal of Systems and Software; Information and Software Technology; ACM Trans. on Soft. Eng. and Methodology; Software Testing, Verification, and Reliability; Software Quality Journal; Empirical Software Engineering; Software: Practice and Experience; Journal of Software: Evolution and Process; Journal of Cloud Computing; IEEE Trans. Softw. Eng.; and IEEE Trans. on Services Computing. Conference proceedings are: ISSTA; IEEE/ACM ASE; IEEE ICST; IEEE/ACM ISBCC; IEEE/ACM ICSE; IEEE SERE; ACM SoCC; ACM FSE; IEEE SOSE; and IEEE Cloud Computing Community Conference List Proceedings.

C. Study Selection

Our inclusion criteria were: peer-reviewed research papers (more than 5 pages) based on empirical research; experimental reports, case studies, or feasibility studies, with evidence; published between 2010-2015; describe testing methods used for cloud-based testing and provide an evaluation of the method used. The selection involved a three-stage process: screening based on the paper title, abstract, and keywords; reading the whole paper by the lead researcher to check exclusion criteria; and applying snowballing to included primary studies' list of references, and repeating stages 2 and 3 on the added studies.

D. Data Extraction

Data for all studies was extracted by Al-Said Ahmad, while the second and third authors performed extraction for a random sample of studies. Data was compared and reconciled as necessary. In addition to publication data, the extracted data are: type of study- e.g. experiment, case study, feasibility study; study aims and objectives; security testing options- vulnerability scan and assessment, security review, security audit, penetration test, or INP (if not provided); scalability testing options- scalability testing, scaling-up, scaling-down, or INP; performance testing options- load testing, stress testing, endurance testing, or INP; reliability testing options - regression testing, load testing, or INP; model-based testing options- model-based security testing, model-based assessment, model-based performance testing, or INP; injection-based testing options- mutation testing, fault injection, or INP.; functional testing options - functional testing or INP.; test coverage options- percentage of coverage or INP.; number of experiments (examples) and case studies, with a brief description; validation method options- simulation and modeling, cross-validation, qualitative data analysis, quantitative data analysis, or by a single example; and prototype study or not.

TABLE I. STUDIES UNDER TESTING METHODS

Category		Studies	#
Functional testing		S1, S2, S3, S4, S5, S7, S8, S11, S12, S14, S15, S17, S19, S20, S24, S27, S28, S30, S32, S34, S35, S36, S38, S42, S44, S48, S50, S51, S52, S54, S55, S56, S58, S64, S67, S68	36
Security testing	Vulnerability scan and assessment	S5, S7, S12, S21, S23, S25, S26, S33, S49, S46, S47, S53, S57	13
	Security review	S5, S7, S24, S26, S41, S47	6
	Security audit	S5, S7, S46	3
	Penetration test	S12, S16, S33	3
Scalability testing	Scalability testing	S3, S28, S37, S39, S48, S65, S66, S69	8
	Scaling-up	S13, S31, S42, S45, S53, S60, S62, S67	8
	Scaling-down	S9, S31, S42, S45, S13, S60, S62, S67	8
Performance testing	Load testing	S1, S3, S4, S6, S8, S9, S10, S11, S13, S14, S17, S18, S19, S20, S24, S28, S29, S31, S36, S37, S39, S43, S45, S40, S48, S50, S51, S59, S60, S61, S63, S65, S66, S67	34
	Stress testing	S9, S15, S18, S31	4
	Endurance testing	S9, S18, S31, S37, S45	5
Reliability testing	Regression testing	S4, S9, S30, S34, S42, S50	6
	Load testing	S1, S3, S11, S42, S48, S56, S59	7
Model-based testing	Assessment	S1, S8, S15, S27	4
	performance	S29	1
	security testing	S5, S25	2
injection-based testing	Mutation testing	S25, S52	2
	Injection-based testing	S1, S3, S5, S6, S11, S19, S35, S36, S37, S40, S47, S53, S54, S56	14

IV. RESULT AND ANALYSIS

A. Overview of result

We identified 69 primary studies reported in 75 research papers related to evaluated testing methods using cloud-based services and resources. The search was conducted using our methodology, and obtained a total of 247 papers. 92 papers entered phase 1 of the search and selection process. 17 papers failed to meet our inclusion criteria during the data extraction. 70 papers were found via snowballing and 48 papers were eliminated due to the exclusion criteria. Of the 75 papers, 13 papers (19%) came from journals, 57 papers (75%) came from conference proceedings, and 5 papers (6%) were book chapters.

30 (43%) primary studies used quantitative data analysis; however, 20 of these did not report specific statistical tests. 6 (9%) studies used simulation and modeling techniques, 3 studies used cross-validation, one study used qualitative data analysis, and another study used both qualitative and quantitative data analysis. About 41% (28) of the primary studies evaluated their method using an example, and six of those studies provided some numerical data.

40 (58%) of the studies are feasibility studies, providing results about a limited scope and often partial implementation of the proposed approach or methodology, without considering a real-world scenario or complex software under test (SUT). There are 32 (47%) studies which describe a complete prototype implementation and there are

11 (16%) studies which present a single complete case study. There are only 18 (26%) studies which describe more extensive experiments (e.g. multiple case studies).

B. Results for Research Question 1

The primary studies were classified according to the testing methods, i.e., functional and non-functional. Table I shows the classification scheme that we developed after applying the methodology described in section III, which was based on the used testing methods. We classified the studies into seven main categories. Of the 69 studies, 36 (52%) studies involved functional testing methods, 55 (80%) studies involved non-functional testing methods, 14 studies focused only on functional testing, and 33 studies focused only on non-functional testing. Table I presents the studies included for this classification.

In the context of non-functional testing, 16 (23%) studies covered security testing, while 35 (51%) used one or more types of performance testing, 17 (25%) studies applied scalability testing methods, 12 (17%) studies used reliability testing, 13 studies applied mutation testing and injection-based testing to test non-functional features, and three studies applied a model-based technique to test a non-functional feature.

C. Results for Research Question 2

The studies were classified into eight groups (see Table II). Where a study could be included in more than one group, we based our decision on the main purpose of the study. Studies that had a purpose that was not related to any of the other seven groups were labelled under ‘other ways of testing’.

TABLE II. MAIN PURPOSE OF STUDIES

Category	Studies (S)	#
Web services/app testing	S18, S21, S31, S43, S58, S59, S63, S65, S66, S67	10
Mobile testing	S23, S24, S44, S48, S53, S61, S68	7
Vulnerability and security configuration testing	S7, S12, S16, S25, S26, S33, S41, S46, S47, S49, S57	11
Benchmarking	S6, S9, S10, S13, S20, S36, S37, S40	8
Testing SaaS	S5, S11, S19, S28, S30, S38, S39, S42, S45, S64	10
Testing cloud services	S3, S8, S14, S15, S22, S27, S29, S32, S60, S62	10
Large-scale testing	S1, S2, S17, S35, S51, S52, S54	7
Other ways of testing	S4, S34, S50, S55, S56	6

1) Web services and web application testing

The feasibility study [19] examined the performance of web applications running on the three types of Amazon EC2 instances. Based on httpref (performance testing tool) PHP script workload and in-cloud load generator, the system stability was checked by generating load requests on the web server for a whole week. The study [20] presents a framework integrated with benchmarking and monitoring tools. A number of smaller-scale experiments are carried out to test performance and scalability of a web application using different instance types to measure the response time, compute units, and throughput. A framework for web security in the cloud [21], which examines vulnerability scanning for web applications, proposes a prototype TaaS

framework for security testing, and is evaluated through experiments using 456 web applications, with 21,141 critical vulnerabilities detected. A prototype hybrid cloud testing platform called AGARIC is presented in [22] that uses both cloud resources and human resources to test web applications in a scalable way. Two experiments were conducted: one with 10 computers and a local server to test a simulated application and another one using resources deployed in LAN and dokuwiki.org as the SUT.

ASTORIA [23], a prototype for automatic testing of performance and scalability on rich Internet applications, was tested with 1,000 virtual users in Amazon EC2. The study [24] presents an experiment for static testing the performance of web applications to measure their reliability. They use two VMs using VMware, and generated and executed test cases automatically by JMeter tool. Four studies present testing for SOA applications and web services using cloud-based resources. The first [25], is a feasibility study presenting a prototype to capture web service change at runtime by using functional regression testing to verify the selected services. The second [26] provides a cloud-based scalable PaaS for a dynamically chosen node in the IaaS layer. They use load testing for scaling-up and down in a case study of their previous work [27], WS-TaaS, testing the load capacity of three real web services. They simulate the service environment, applying 959 slices for deploying WS-TaaS on PlanetLab and using 50 nodes as the test node. The study [28] concerns cloud-based performance testing for web services. It reports prototype experiments in Amazon EC2 with 100 test tasks for 3 performance test methods, with each task assembled with 2 web services.

2) *Mobile testing*

Five studies present a TaaS framework for mobile testing, and two studies report testing of mobile applications. One study presents a TaaS framework for mobile development [29], evaluating the framework with one example and implementing a web user interface using a VAADIN framework, Google App Engine application, and Jersey RESTful web services. A simulation-based mobile testing environment [30], emulating mobile devices using VMs and IaaS is evaluated using analytical techniques. A prototype mobile TaaS framework [31] is tested using a functional approach, comparing the result with two other test script generations. An automated TaaS is presented in [32], with a feasibility case study to evaluate it using private cloud services, with 9 mobile devices, 5 mobile applications and 84% test case coverage. The study [33] uses a prototype framework for load balancing implemented with OpenStack with 63 hosts and 400 requests, comparing the proposed method with other algorithms. A white-box automated security testing approach [34] for cloud-based Android apps is evaluated by an example run over 1,000 test cases using 100 parallel instances. The study [35] presents a testing approach with experiments evaluated on a combination of 1,000+ emulated instances and 10 actual devices.

3) *Vulnerability and security configuration testing*

A real-life case study [36] with six design stages is evaluated using a sequence of interviews. [37] presents a

penetration TaaS, with two case studies that let POTASSIUM capture the exact SUT into a mirror and save it as a live snapshot. They ran a penetration test against the snapshot using a cluster of three different memory size Ubuntu VMs. An automated risk assessment framework (Nemesis) is presented in [38], involving vulnerability assessment by using their previous work [39]. To evaluate their approach, a cloud environment and its services are designed using OpenStack, applying the framework on 10 IT products. A security testing approach is presented in [40] targeting two situations. First they aim to determine the vulnerabilities of Ubuntu Server with the OpenStack node; second they aim to determine the vulnerabilities of cloud instances with different operating systems. A prototype framework for vulnerability assessment in cloud systems is presented in [39] and [41], with one example about developing an automated process for their proposed approach. Security validation as a service is presented in [42], with two hosts providing the proposed service to two midsize business processes, repeating the requests every 15 minutes for security validation. A vulnerability scan and assessment approach is presented in [43] with four test cases: two cases for security assessment from inside the cloud, and two from outside the cloud. The study [44] presents a prototype model-based security testing approach. The authors employed risk analysis to test the cloud environment, which is evaluated by one example using VMware's vCloud.

The study [45] presents an approach for detecting security vulnerabilities by checking for software updates and scanning virtual machines, with one experiment using Debian penetration suite, repeated 20 times. Another vulnerability assessment approach [46], applies three different scenarios to explain how the cloud affects the security vulnerability. A model-driven approach is shown in [47] to facilitate the creation of security configurations. The approach is assessed by applying it to a model developed using the Oryx tool.

4) *Benchmarking*

The study [48] presents a benchmarking-as-a-service framework that automatically scales the injection load platform. Three experiment scenarios were performed, with two SUTs selected to test in these scenarios. The study presented in two research papers [49], [50] introduces performance and scalability testing of SaaS using IaaS. The experiments measure the performance of two SaaS applications using three public clouds, and three private clouds, evaluating both the scaling up and out in Amazon EC2, and scaling out in Emulab and Open Cirrus. The study [51] presents a modeling framework (ROAR) for automated cloud resource allocation, optimization, and benchmarking. In two experiments using Amazon and Google clouds, they use the ROAR to deploy multi-tier applications to cloud providers and an auto-scaling engine. The study [52] presents C-MART, a benchmark application emulating a web application running in the cloud. C-MART is run against datacenter benchmarks comparing the results. The study [53] proposes a cloud-based load testing model for cloud infrastructure. The validation involved two experiments for benchmarking as a service using two e-

commerce systems (TPC-W), one with MySQL and the other with NoSQL. The study [54] presents a toolset called DS-Bench, which operates through benchmarks and fault injectors that simulate the overload in system resources, aiming to measure dependability.

A framework is presented in [55] to facilitate performance comparisons of cloud data serving systems, using 6 server machines to verify the scalability of YCSB. They run one experiment with PNUTS on a 47 server clusters with a database that contains 120 million records. A benchmark for virtualized and cloud environments is presented in the study [56]. They run several experiments using Libvirt, oVirt, Sar, Faban, KVM, and Collectd.

5) *Testing SaaS*

In the context of testing SaaS, [57] introduces a novel model-driven security engineering approach for multi-tenant SaaS applications. To evaluate the proposed approach, they applied it to seven open-source web-based applications developed using ASP.Net. The study [58] presents Trio, an open-source Java prototype topology robustness indicator that simulates failure sequences. By using a domain-specific language (CloudML), Trio is used to evaluate the robustness of various topologies through a number of experiments. The study [59] presents an approach to automate performance testing of cloud applications and a prototype based on load-testing tools and using IBM's WAIT expert system. Two experiments were conducted: one to evaluate the overhead using JPetStore and IBM WebSphere Portal applications, the other to evaluate the productivity of the approach by injecting three common performance issues in JPetStore.

TaaS with tools are presented in [60], which describes a single case study with 100% test coverage. Using the OrangeHRM (SaaS) application with two functional features, and two black-box test methods, system-level test cases have been designed for each feature. The prototype study [61] aims to improve the test effectiveness and efficiency of SaaS using a regression testing approach with 61%-72% test coverage. The study reports one case study using two versions of an industrial application. They generate test cases from the requirements scenarios and execute each test case manually. A prototype testing approach to detect scalability bottlenecks in NoSQL schemas is presented in [63]. Concurrent writes are generated by running a servlet on Google App Engine. A case study uses an article-oriented scenario, creating one single article, and a series of 20, 100, 500, and 1,000 write requests runs against the single article. The study [63] presents a code generation tool for automated performance testing of distributed applications in IaaS called Expertus. Experiments were performed using three SaaS solutions deployed on five IaaS solutions.

A prototype approach to support SaaS continuous testing and policy enforcement is presented in [64]. The study describes one case study using test cases generated from Metadata. The test cases are ranked based on their importance, WebStra's framework ranking, and their history. They establish a test oracle by voting and automatically analyse the oracle using statistical techniques. The study [65] presents a testing model that evaluates SaaS performance and analyses scalability in the cloud. A case study is reported

using Amazon EC2 with 4 load configuration scenarios. An automated integration testing approach of SaaS is introduced in [66]. A prototype of unit testing framework is described using Windows Azure and Visual Studio 2010.

6) *Testing Cloud Services*

The study [67] presents a tool for automated quality of service and scalability analysis for system reliability testing using load variation and fault injection. Experiments were performed to evaluate the proposed tool using seven user loads to measure the scalability and the quality of the SUT. A study presented in two research papers [68], [69] uses integration testing of data-centric and event-based dynamic service compositions. Four distributed performance test experiments were run on a single virtual machine using Ubuntu Linux. A testing framework for test scripts and test case generation that measures service performance, called CLTF, is presented in [70]. The authors applied the framework to over 1,300 realistic cloud services from 50 projects collected from the enterprise private PaaS cloud.

A prototype model-based assessment approach is presented in [71]. They evaluated the proposed approach with a case study simulating system prototypes in the face of hostile environment conditions. Another study [72] presents a cloud service selection model through a set of experiments. They used 59 real cloud services to do real-time performance evaluation. The study [73] presents a prototype testing framework for cloud platforms and infrastructures. To evaluate their framework a case study was conducted with 18 Google App Engine test cases. A prototype platform is presented in [74] for testing services and users. The platform enables the setting up of unit testing by selecting the most suitable unit testing method and cloud service, test case generation, execution, and reporting testing result in an automatic way. Another prototype framework for cloud services test cases generation is introduced in [75], with one experiment. The system is separated into a web service semantics side that generates test cases from source code and transmits these to the UDDI side that allows the users to discover cloud services. Paper [76] presents a simulated cloud service based testing approach. The study proposes a solution for testing and quality estimation for both bottleneck detection and fault diagnosis using an offline testing technique. The study [77] presents a scalability testing approach to model the performance for cloud-based services at different abstraction levels. The paper constructs preliminary models for IaaS, and the benchmark program (SaaS) on the cloud using Amazon resources and services.

7) *Large-scale Testing*

The study [78] presents a model-based testing approach using a local cloud to test the global properties of a large-scale system. An experiment was conducted on two clusters of 32 nodes to validate the functionality of two popular clouds' open-source distributed hash tables, data insertion, and retrieval. An analysis of crowdsourcing testing methods for a large-scale system by using INP is presented in [79]. Three experiments are presented: to determine the min-time test case combinations, to compare the proposed approach with the performance of CPLEX ILP formulation, and to evaluate the performance of the proposed testing approach.

The study [80] presents peer-to-peer load testing approaches to isolate bottleneck problems in a large-scale system. The experiments used load testing validate performance having point-to-point connection between the test driver and the SUT, or using tools that provide a test driver to allow submission of operations based on load type, with one machine to simulate the SUT and five others to simulate the clients. The study [81] presents an investigation of cloud computing to facilitate the testing of large-scale software. They evaluate the proposed mutation functional testing using a case study on Google Chrome and Amazon EC2 with 820 implemented mutations.

The study [82] presents a case study of resource management infrastructure to enable integration testing of distributed real-time and embedded system applications. They used a modelling tool (CUTS) to evaluate an infrastructure-level system (RACE) scenario in the Emulab test cloud. A study is reported in three research papers about D-Cloud [9], [83], [84], a simulated Eucalyptus-based testing environment for large-scale distributed systems. The authors apply D-Cloud to two real systems: a highly available server system and RI2N. The study [85] is a feasibility study that introduces a framework for testing the IaaS-based delivery model, which is evaluated by using FaultVM and D-Cloud.

8) Other ways of testing

The study [86] presents an automated verification approach for virtual machine patches with 3 stages of experiments. An approach to manage, compose and test services on the cloud is presented in [87]. The study provides limited data on the results. Test case generation using JUnit is presented in [88], with three series of experiments. They determined the performance of the JUnit test execution using one machine, then they used HadoopUnit to coordinate testing on four nodes in a cluster, finally they tested the reduction of map tasks by increasing the workload of each map task. A simulation test case generation using parallel symbolic execution is presented in [89] based on MC/DC test cases and suite generation with six case examples.

Cloud9 [90], [91] is a prototype platform for automated testing of real-world applications that run on Amazon EC2, private clusters, and multi-core machines. 5 case studies are reported, using different operating systems and simulated services. Scalability Explorer, an automated framework for scalability testing is presented in [92], introducing scalability testing as TaaS through one experiment to evaluate a web service-based distributed matrix multiplication system hosted on Amazon EC2.

V. DISCUSSION AND CONCLUSION

This systematic mapping study has discussed 69 software cloud testing unique primary studies reported in 75 research papers. The paper presents a state-of-the-art analysis of existing cloud-based testing methods that were experimentally evaluated during the period 2010-2015. This was done methodically by following a well-defined protocol.

It is possible that not all relevant studies were located. However we used the related review papers to validate our search for primary studies. We used multiple reviewers to check the quality of the extracted data.

The majority of the studies present only preliminary results, often describing an example of the software cloud-based testing methods or a simple application experiment to evaluate the proposed approach. Many of the considered studies rely on limited scope or relatively simple implementations and case studies. Only a minority of the studies used quantitative analysis combined with rigorous statistical tests. The considered studies spread relatively evenly across the testing topic categories that we used in this paper. Many of the studies present early work and results that their authors expect to lead to further more extensive studies. Often the assessment of the proposed solutions is based on a single experiment. These indicate growing interest across the field of cloud related testing and also the potential for much more research to follow the early results.

REFERENCES

- [1] B. Kitchenham, D. Budgen, and P. Brereton, "Evidence-based software engineering and systematic reviews," Abingdon: CRC Press, 2015.
- [2] B. Kitchenham, "Procedures for performing systematic reviews," Keele University, Keele, 2004, 1-26.
- [3] F. Silva, et al., "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Infom. and Soft. Tech.*, vol. 53, no. 9, pp. 899-913, 2011.
- [4] P. Brereton, B. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. of Systems and Software*, vol. 80, no. 4, pp. 571-583, 2007.
- [5] G. Tasse, "The economic impacts of inadequate infrastructure for software testing," *National Inst. of Standards and Technology, RTI Project 7007.011*, 2002.
- [6] C. Jia, Y. Cai, Y. Yu, and T. Tse, "5W 1H pattern: A perspective of systematic mapping studies and a case study on cloud software testing," *J. of Systems and Software*, vol. 116, pp. 206-219, 2016.
- [7] K. Inçki, I. Ari and H. Sözer, "A Survey of Software Testing in the Cloud," *Software Security and Rel. Companion (SERE-C)*, 2012 IEEE Sixth Int. Conf. on, Gaithersburg, MD, 2012, pp. 18-23.
- [8] T. Parveen and S. Tilley, "When to Migrate Software Testing to the Cloud?," *Software Testing, Verification, and Validation Workshops (ICSTW)*, 2010 Third Int. Conf. on, Paris, 2010, pp. 424-427.
- [9] T. Hanawa, et al., "Large-Scale Software Testing Environment Using Cloud Computing Technology for Dependable Parallel and Distributed Systems," *Software Testing, Verification, and Validation Workshops (ICSTW)*, Int. Conf. on, Paris, 2010, pp. 428-433. [S35]
- [10] H. Liu, and D. Orban, "Remote network labs: An on-demand network cloud for configuration testing," *ACM SIGCOMM Computer Communication Review*, Vol. 40, no. 1, pp. 83-91, 2010.
- [11] S. Gaisbauer, J. Kirschnick, N. Edwards and J. Rolia, "VATS: Virtualized-Aware Automated Test Service," *Quantitative Evaluation of Systems (QEST)*, Fifth Int. Conf. on, St. Malo, 2008, pp. 93-102.
- [12] C. Jia and Y. Yu, "Using the 5W+1H Model in Reporting Systematic Literature Review: A Case Study on Software Testing for Cloud Computing," *13th Int. Conf. on Quality Software*, Najing, 2013, pp. 222-229.
- [13] S. Nachiyappan, and S. Justus, "Cloud Testing Tools and its Challenges: A Comparative Study," *Procedia Computer Sci.*, Vol. 50, pp. 482-489 2015.
- [14] X. Bai, M. Li, B. Chen, W. Tsai and J. Gao, "Cloud testing tools," *Service Oriented System Engineering (SOSE)*, IEEE 6th Int. Symp. on, Irvine, CA, 2011, pp. 1-12.
- [15] J. Gao, X. Bai, W. Tsai and T. Uehara, "Testing as a Service (TaaS) on Clouds," *Service Oriented System Engineering (SOSE)*, IEEE 7th Int. Symp. on, Redwood City, 2013, pp. 212-223.
- [16] B. Floss and S. Tilley, "Software Testing as a Service: An Academic Res. Perspective," *Service Oriented System Engineering (SOSE)*, IEEE 7th Int. Symp. on, Redwood City, 2013, pp. 421-424.

- [17] R. Denz and S. Taylor, "A survey on securing the virtual cloud," *J. of Cloud Computing*, vol. 2, no. 1, p. 17, 2013.
- [18] J. Gao, X. Bai, W. T. Tsai and T. Uehara, "SaaS Testing on Clouds - Issues, Challenges and Needs," *Service Oriented System Engineering (SOSE)*, IEEE 7th Int. Symp. on, Redwood City, 2013, pp. 409-415.
- [19] J. Mukherjee, M. Wang and D. Krishnamurthy, "Performance Testing Web Applicat. on the Cloud," *Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE Seventh Int. Conf. on, Cleveland, OH, 2014, pp. 363-369. [S18]
- [20] M. Vasar, S. Srirama, and M. Dumas, "Framework for monitoring and testing web application scalability on the cloud," *Companion Volume (WICSA/ECSSA)*, The ACM WICSA/ECSSA 2012, New York, USA, 2012, pp. 53-60. [S31]
- [21] Y. Tung, C. Lin and H. Shan, "Test as a Service: A Framework for Web Security TaaS Service in Cloud Environment," *Service Oriented System Engineering (SOSE)*, IEEE 8th Int. Symp. on, Oxford, 2014, pp. 212-217. [S21]
- [22] J. Wu, C. Wang, Y. Liu and L. Zhang, "Agaric: A hybrid cloud based testing platform," *Cloud and Service Computing (CSC)*, Int. Conf. on, Hong Kong, 2011, pp. 87-94. [S63]
- [23] N. Snellman, A. Ashraf and I. Porres, "Towards Automatic Performance and Scalability Testing of Rich Internet Applicat. in the Cloud," *37th EURO MICRO Conf. on Software Engineering and Advanced Applicat.*, Oulu, 2011, pp. 161-169. [S65]
- [24] A. Ali and N. Badr, "Performance testing as a service for web Applicat.," *IEEE Seventh Int. Conf. on Intelligent Computing and Infom. Systems (ICICIS)*, Cairo, 2015, pp. 356-361. [S59]
- [25] M. Cooray, J. Hamlyn-Haris and R. Merkel, "Test Reconfiguration for Service Oriented Applicat.," *Utility and Cloud Computing (UCC)*, Fourth IEEE Int. Conf. on, Victoria, NSW, 2011, pp. 300-305. [S58]
- [26] H. Sun, X. Wang, M. Yan, Y. Tang and X. Liu, "Towards a Scalable PaaS for Service Oriented Software," *Parallel and Distributed Systems (ICPADS)*, Int. Conf. on, Seoul, 2013, pp. 522-527. [S66]
- [27] M. Yan, H. Sun, X. Wang and X. Liu, "WS-TaaS: A Testing as a Service Platform for Web Service Load Testing," *Parallel and Distributed Systems (ICPADS)*, IEEE 18th Int. Conf. on, Singapore, 2012, pp. 456-463. [S67]
- [28] L. Zhang, Y. Chen, F. Tang and X. Ao, "Design and implementation of cloud-based performance testing system for web services," *Communications and Networking in China (CHINACOM)*, 6th Int. ICST Conf. on, Harbin, 2011, pp. 875-880. [S43]
- [29] O. Starov and S. Vilkomir, "Integrated TaaS platform for mobile develop.: Architecture solutions," *Automation of Software Test*, 8th Int. Workshop on, San Francisco, CA, 2013, pp. 1-7. [S24]
- [30] J. Gao, W. Tsai, R. Paul, X. Bai and T. Uehara, "Mobile Testing-as-a-Service (MTaaS) -- Infrastructures, Issues, Solutions and Needs," *2014 IEEE 15th Int. Symp. on High-Assurance Systems Engineering*, Miami Beach, FL, 2014, pp. 158-167. [S48]
- [31] S. Zhang and B. Pi, "Mobile Functional Test on TaaS Environment," *Service-Oriented System Engineering (SOSE)*, 2015 IEEE Symp. on, San Francisco Bay, CA, 2015, pp. 315-320. [S44]
- [32] I. Villanes, E. Costa and A. Dias-Neto, "Automated Mobile Testing as a Service (AM-TaaS)," *2015 IEEE World Congr. on Services*, New York City, NY, 2015, pp. 79-86. [S68]
- [33] C. Tao, J. Gao and B. Li, "Cloud-Based Infrastructure for Mobile Testing as a Service," *2015 Third Int. Conf. on Advanced Cloud and Big Data, Yangzhou*, 2015, pp. 133-140. [S61]
- [34] R. Mahmood, et al., "A whitebox approach for automated security testing of Android Applicat. on the cloud," *Automation of Software Test (AST)*, 7th Int. Workshop on, Zurich, 2012, pp. 22-28. [S23]
- [35] H. Turner, et al., "Building A Cloud-Based Mobile Application Testbed," in *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, S. Tilley and T. Parveen, Ed. IGI Global, 2013, pp. 382-403. [S53]
- [36] O. Rebollo, D. Mellado, E. Fernández-Medina, and H. Mouratidis, "Empirical evaluation of a cloud computing Infom security governance framework," *Infom. and Soft. Tech.*, vol. 58, pp. 44-57, 2015. [S7]
- [37] R. Li, et al., "Potassium: penetration testing as a service," *Cloud Computing (SoCC '15)*, The Sixth ACM Symp. on, New York, USA, 2015, pp. 30-42. [S16]
- [38] P. Kamongi, M. Gomathisankaran, and K. Kavi, "Nemesis: Automated Architecture for Threat Modeling and Risk Assessment for Cloud Computing," *ASE Big Data/ Social Informatics/ PASSAT/ BioMedCom Conf.*, Harvard University, USA, 2014, pp 1-10. [S57]
- [39] P. Kamongi, S. Kotikela, K. Kavi, M. Gomathisankaran and A. Singhal, "VULCAN: Vulnerability Assessment Framework for Cloud Computing," *Software Security and Rel. (SERE)*, IEEE 7th Int. Conf. on, Gaithersburg, MD, 2013, pp. 218-226. [S33]
- [40] S. Ristov, M. Gusev, and A. Donevski, "Openstack cloud security vulnerabilities from inside and outside," *CLOUD COMPUTING*, 2013, pp. 101-107. [S49]
- [41] S. Kotikela, K. Kavi and M. Gomathisankaran, "Vulnerability assessment in cloud computing", *The Int. Conf. on Security & Management (SAM)*, pp. 67-73, 2012. [S33]
- [42] L. Compagna, P. Guillemot and A. D. Brucker, "Business Process Compliance via Security Validation as a Service," *IEEE Sixth Int. Conf. on Software Testing, Verification and Validation*, Luembourg, 2013, pp. 455-462. [S26]
- [43] A. Donevski, S. Ristov and M. Gusev, "Security assessment of virtual machines in open source clouds," *Infom. & Communication Technology Electronics & Microelectronics (MIPRO)*, 36th Int. Convention on, Opatija, 2013, pp. 1094-1099. [S47]
- [44] P. Zech, M. Felderer and R. Breu, "Towards a Model Based Security Testing Approach of Cloud Computing Environments," *Software Security and Rel. Companion (SERC-C)*, IEEE Sixth Int. Conf. on, Gaithersburg, MD, 2012, pp. 47-56. [S25]
- [45] R. Schwarzkopf, M. Schmidt, C. Strack, S. Martin, and B. Freisleben, "Increasing virtual machine security in cloud environments," *J. of Cloud Computing*, vol. 1, no. 1, p. 12, 2012. [S12]
- [46] H. Li, P. Liang, J. Yang and S. Chen, "Analysis on Cloud-Based Security Vulnerability Assessment," *e-Business Engineering (ICEBE)*, IEEE 7th Int. Conf. on, Shanghai, 2010, pp. 490-494. [S46]
- [47] M. Menzel, R. Warschofsky, I. Thomas, C. Willems and C. Meinel, "The Service Security Lab: A Model-Driven Platform to Compose and Explore Service Security in the Cloud," *6th World Congr. on Services*, Miami, FL, 2010, pp. 115-122. [S41]
- [48] A. Tchana, N. D. Palma, B. Dillenseger, and X. Etchevers, "A self-scalable load injection service," *Software: Practice and Experience*, vol. 45, no. 5, pp. 613-632, 2013. [S6]
- [49] D. Jayasinghe, et al., "Variations in Performance and Scalability: An Experimental Study in IaaS Clouds Using Multi-Tier Workloads," *IEEE Trans. Serv. Comput.*, vol. 7, no. 2, pp. 293-306, 2014. [S9]
- [50] D. Jayasinghe, et al., "Variations in Performance and Scalability When Migrating n-Tier Applicat. to Different Clouds," *Cloud Computing (CLOUD)*, IEEE Int. Conf. on, Washington, DC, 2011, pp. 73-80. [S9]
- [51] Y. Sun, J. White, S. Eade, and D. C. Schmidt, "ROAR: A QoS-oriented modeling framework for automated cloud resource allocation and optimization," *J. of Systems and Software*, vol. 116, pp. 146-161, 2015. [S10]
- [52] A. Turner, A. Fox, J. Payne, and H. Kim, "C-MART: Benchmarking the Cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1256-1266, 2013. [S13]
- [53] Q. Gao, et al., "Migrating Load Testing to the Cloud: A Case Study," *Service Oriented System Engineering (SOSE)*, IEEE 7th Int. Symp. on, Redwood City, 2013, pp. 429-434. [S20]
- [54] H. Fujita, et al., "DS-Bench Toolset: Tools for dependability benchmarking with simulation and assurance," *IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, Boston, MA, 2012, pp. 1-8. [S36]
- [55] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," *Cloud computing (SoCC)*, The 1st ACM Symp. on, New York, USA, 2010, pp. 143-154. [S37]
- [56] M. A. El-Refaei and M. A. Rizkaa, "CloudGauge: A Dynamic Cloud and Virtualization Benchmarking Suite," *Enabling Technologies:*

- Infrastructures for Collaborative Enterprises (WETICE), 19th IEEE Int. Workshop on, Larissa, 2010, pp. 66-75. [S40]
- [57] M. Almorsy, J. Grundy, and A. S. Ibrahim, "Adaptable, model-driven security engineering for SaaS cloud-based Applicat.," *Automated Software Engineering*, vol. 21, no. 2, pp. 187-224, 2013. [S5]
- [58] F. Chauvel, H. Song, N. Ferry, and F. Fleurey, "Evaluating robustness of cloud-based systems," *J. of Cloud Computing*, vol.4, no.1, 2015. [S11]
- [59] A. Portillo-Dominguez, M. Wang, J. Murphy and D. Magoni, "Automated WAIT for Cloud-Based Application Testing," *Software Testing, Verification and Validation Workshops (ICSTW)*, IEEE Seventh Int. Conf. on, Cleveland, OH, 2014, pp. 370-375. [S19]
- [60] J. Gao et al., "A cloud-based TaaS infrastructure with tools for SaaS validation, performance and scalability evaluation," *Cloud Computing Technology and Sci. (CloudCom)*, IEEE 4th Int. Conf. on, Taipei, 2012, pp. 464-471. [S28]
- [61] H. Srikanth and M. Cohen, "Regression testing in Software as a Service: An industrial case study," *Software Maintenance (ICSM)*, 27th IEEE Int. Conf. on, Williamsburg, VI, 2011, pp. 372-381. [S30]
- [62] S. Scherzinger, E. Almeida, F. Ickert, and M. Fabro, "On the necessity of model checking NoSQL database schemas when building SaaS Applicat.," *Testing the Cloud (TTC)*, Int. Workshop on, Lugano, 2013, pp. 1-6. [S38]
- [63] D. Jayasinghe et al., "Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds," *Cloud Computing (CLOUD)*, IEEE 5th Int. Conf. on, Honolulu, HI, 2012, pp. 115-122. [S39]
- [64] W.-T. Tsai, Q. Shao, Y. Huang, and X. Bai, "Towards a scalable and robust multi-tenancy SaaS," *Proc. of the Second Asia-Pacific Symp. on Internetware - Internetware '10*, 2010. [S42]
- [65] J. Gao, P. Pattabhiraman, X. Bai and W. Tsai, "SaaS performance and scalability evaluation in clouds," *Service Oriented System Engineering (SOSE)*, IEEE 6th Int. Symp. on, Irvine, CA, 2011, pp. 61-71. [S45]
- [66] T. King, A. Ganti, and D. Froslic, "Enabling automated integration testing of cloud application services in virtualized environments," *Conf. of the Center for Advanced Studies on Collaborative Res. (CASCON)*, Marin Litoiu, Eleni Stroulia, and Stephen MacKay (Eds.), IBM Corp., Riverton, NJ, USA, 2011, pp. 120-132. [S64]
- [67] D. Preuveneers, T. Heyman, Y. Berbers, and W. Joosen, "Systematic scalability assessment for feature oriented multi-tenant services," *J. of Systems and Software*, vol. 116, pp. 162-176, 2015. [S3]
- [68] W. Hummer, O. Raz, O. Shehory, P. Leitner, and S. Dustdar, "Testing of datacentric and event-based dynamic service compositions," *Software Testing, Verification and Rel.*, vol. 23, no. 6, pp. 465-497, 2013. [S14]
- [69] W. Hummer, O. Raz, O. Shehory, P. Leitner and S. Dustdar, "Test Coverage of Data-Centric Dynamic Compositions in Service-Based Systems," *Fourth IEEE Int. Conf. on Software Testing, Verification and Validation*, Berlin, 2011, pp. 40-49. [S14]
- [70] J. Zhou, B. Zhou and S. Li, "Automated Model-Based Performance Testing for PaaS Cloud Services," *Computer Software and Applicat. Conf. Workshops (COMPSACW)*, IEEE 38th Int., Vasteras, 2014, pp. 644-649. [S29]
- [71] K. Ravindran and A. Adithan, "Verification of non-functional properties of cloud-based distributed system services". *Automation of Software Test (AST 201 4)*, 9th Int. Workshop on. ACM, New York, USA, 2014, pp. 43-49. [S15]
- [72] L. Qu, et al., "CCCloud: Context-Aware and Credible Cloud Service Selection Based on Subjective Assessment and Objective Assessment," *IEEE Trans. Serv. Comput.*, vol.8, no.3, pp. 369-383, 2015. [S8]
- [73] W. Jenkins, S. Vilkomir, P. Sharma and G. Pirocanac, "Framework for testing cloud platforms and infrastructures," *Cloud and Service Computing*, Int. Conf. on, Hong Kong, 2011, pp. 134-140. [S32]
- [74] L. Yu et al., "Testing as a Service over Cloud," *Service Oriented System Engineering (SOSE)*, Fifth IEEE Int. Symp. on, Nanjing, 2010, pp. 181-188. [S22]
- [75] C. Wu and Y. Lee, "Automatic SaaS test cases generation based on SOA in the cloud service," *Cloud Computing Technology and Sci. (CloudCom)*, IEEE 4th Int. Conf. on, Taipei, 2012, pp. 349-354. [S27]
- [76] W. Lo, X. Liu, R. Sheu, S. Yuan and C. Chang, "An Architecture for Cloud Service Testing and Real Time Management," *Computer Software and Applicat. Conf. (COMPSAC)*, IEEE 39th Annual, Taichung, 2015, pp. 598-603. [S60]
- [77] K. Hwang, Y. Shi and X. Bai, "Scale-Out vs. Scale-Up Techniques for Cloud Performance and Productivity," *Cloud Computing Technology and Sci. (CloudCom)*, IEEE 6th Int. Conf. on, Singapore, 2014, pp. 763-768. [S62]
- [78] G. Sunyé, E. Almeida, Y. Traon, B. Baudry, and J.-M. Jézéquel, "Model-based testing of global properties on large-scale distributed systems," *Infom. and Software Technology*, vol. 56, no. 7, pp. 749-762, 2014. [S1]
- [79] Y.-H. Tung and S.-S. Tseng, "A novel approach to collaborative testing in a crowdsourcing environment," *J. of Systems and Software*, vol. 86, no. 8, pp. 2143-2153, 2013. [S2]
- [80] J. Meira, E. Almeida, Y. Le Traon and G. Sunye, "Peer-to-Peer Load Testing," *IEEE Fifth Int. Conf. on Software Testing, Verification and Validation*, Montreal, QC, 2012, pp. 642-647. [S17]
- [81] A. Pakhira, and P. Andras, "Leveraging The Cloud for Large-Scale Software Testing - A Case Study: Google Chrome on Amazon," in *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, S. Tilley and T. Parveen, Ed. IGI Global, 2013, pp. 252-279. [S52]
- [82] J. Hill, and D. Schmidt, "Using Test Clouds To Enable Continuous Integration Testing Of Distributed Real-Time And Embedded System Applicat.," in *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, S. Tilley and T. Parveen, Ed. IGI Global, 2013, pp. 174-195. [S51]
- [83] T. Banzai, et al., "D-Cloud: Design of a Software Testing Environment for Reliable Distributed Systems Using Cloud Computing Technology," *Cluster, Cloud and Grid Computing (CCGrid)*, 10th IEEE/ACM Int. Conf. on, Melbourne, Australia, 2010, pp. 631-636. [S35]
- [84] T. Hanawa and M. Sato, "D-Cloud: Software Testing Environment for Dependable Distributed Systems Using Cloud Computing Technology", in *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, S. Tilley and T. Parveen, Ed. IGI Global, 2013, pp. 340-355. [S35]
- [85] T. Hanawa et al., "Customizing Virtual Machine with Fault Injector by Integrating with SpecC Device Model for a Software Testing Environment D-Cloud," *Dependable Computing (PRDC)*, IEEE 16th Pacific Rim Int. Symp. on, Tokyo, 2010, pp. 47-54. [S54]
- [86] Y. Yamato, "Automatic verification technology of software patches for user virtual environments on IaaS cloud," *J. of Cloud Computing*, vol. 4, no. 1, 2015. [S4]
- [87] W. Tsai, et al., "An Approach for Service Composition and Testing for Cloud Computing," *10th Int. Symp. on Autonomous Decentralized Systems*, Tokyo & Hiroshima, 2011, pp. 631-636. [S34]
- [88] E. Bower, T. Parveen, and S. Tilley, "Performance Analysis Of A Distributed Execution Environment For Junit Test Cases On A Small Cluster," in *Software Testing in the Cloud: Perspectives on an Emerging Discipline*, S. Tilley and T. Parveen, Ed. IGI Global, 2013, pp. 96-112. [S50]
- [89] Matt Staats and Corina Păsăreanu, "Parallel symbolic execution for structural test generation," *Software testing and analysis (ISSTA)*, 19th Int. Symp. on. ACM, New York, USA, 2010, pp. 183-194. [S55]
- [90] S. Bucur, V. Ureche, C. Zamfir, and G. Candea, "Parallel symbolic execution for automated real-world software testing," *th Computer systems (EuroSys)*. ACM sixth Conf. on, New York, NY, USA, 2011, pp. 183-198. [S56]
- [91] G. Candea, S. Bucur, and C. Zamfir, "Automated software testing as a service," *Cloud computing (SoCC)*. 1st ACM Symp. on, New York, USA, 2010, pp. 155-160. [S56]
- [92] P. Moura and F. Kon, "Automated scalability testing of software as a service," *Automation of Software Test (AST)*, 8th Int. Workshop on, San Francisco, CA, 2013, pp. 8-14. [S69]