2012 IEEE Sixth International Conference on Software Security and Reliability Companion

A Survey of Software Testing in the Cloud

Koray İnçki*†, İsmail Arı †, Hasan Sözer†
*TÜBİTAK BİLGEM, Information Technologies Institute, Kocaeli, Turkey
koray.incki@bte.tubitak.gov.tr
†Computer Engineering Department, Özyeğin University, İstanbul, Turkey
{koray.incki, ismail.ari, hasan.sozer}@ozyegin.edu.tr

***Abstract*—Cloud computing has emerged as a new computing paradigm that impacts several different research fields, including software testing. Testing cloud applications has its own peculiarities that demand for novel testing methods and tools. On the other hand, cloud computing also facilitates and provides opportunities for the development of more effective and scalable software testing techniques. This paper reports on a systematic survey of published results attained by the synergy of these two research fields. We provide an overview regarding main contributions, trends, gaps, opportunities, challenges and possible research directions. We provide a review of software testing over the cloud literature and categorize the body of work in the field.**

***Index Terms*—cloud computing; software testing; cloud testing; cloud-based software testing; testing cloud services; testing as a service**

I. Introduction

Cloud computing has emerged as a new computing paradigm that facilitates the development and utilization of highly flexible, elastic services on-demand, and over broadband network access. Those attributes are driving many organizations to move their businesses to a cloud platform.

Software testing has been one of the best practice areas for migrating to cloud environment. Virtualization, which is an enabling technology of cloud computing, was first used for quickly creating virtual computing resources with different operating systems (OS) to test software applications on various platforms [1]. Testing new software often requires costly server, storage and network devices only for a limited time [2]. These computing resources are either not used or underutilized after testing, thus incurring extra cost on budget.

Especially in some application domains, software testing requires extensive resources. For example, to test the performance and scalability of a banking application, the system must be stressed with requests from millions of users in a short time interval. This is a realistic scenario that should be tested because people rush to their bank accounts regularly on every payday. Reproducing such a scenario would require the provider to set up a test harness (including the user databases) to emulate the actions of millions of users. Similarly, mobile application providers frequently have to deal with maintaining the quality of their services over a plethora of various combinations of platforms [3]. The computing platforms may encompass various browser technologies with different back-end support running on various mobile OS. To ensure a reliable service, providers have to test their services on all these platforms.

Test automation topic is frequently visited when software testing is considered over the cloud. There are many test automation tools in the market, which address different requirements in a testing life-cycle (e.g., automated test data generation, test case generation, test execution and test evaluation). We believe our evaluation will also motivate the migration of those tools to the cloud.

One of the major drivers of cloud computing adoption is economies of scale. It provides a pay-per-use type of service, thus eliminating the upfront investment in many cases. Testing tools and services are no exception. Development teams can benefit from this paradigm for utilizing test tools when they need it and as much as they need it, thus saving license fees.

We will enrich the discussion with current state-of-the-art software testing as a service over the cloud; the survey will classify related literature according to what type of testing activities these services support for what type of application domains.

The remainder of the paper is organized as follows; in Section II, we give a short background on cloud computing and software testing in the cloud; in Section III we describe our research methodology; in Section IV we present an evaluation of review results and identify gaps as well as opportunities; Section V presents related work; and we conclude the paper in Section VI.

II. Cloud Computing

Cloud computing is a relatively recent term, which basically defines a new paradigm for service delivery in every aspect of computing. It enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [4].

Cloud computing has been enabled by the developments in virtualization, distributed computing, utility computing, web and software services technologies [5]. It is especially based on two key concepts. The first one is *Service-Oriented Architecture (SOA)*, which is the delivery of an integrated and orchestrated suite of functions to an end-user. The functions can be both loosely or tightly coupled. SOA enables end-users to easily search, use and release services on-demand and at a desired quality level. Workflows allow integration of

978-0-7695-4743-5/12 $26.00 © 2012 IEEE
DOI 10.1109/SERE-C.2012.32

services to deliver a business-valued application. The second key concept is *virtualization*. Virtualization allows abstraction and isolation of lower level functionalities and hardware, which enables portability of higher level functions and sharing and/or aggregation of the physical resources.

A. Essential characteristics

Cloud computing exhibits the following essential characteristics. *Rapid elasticity* allows end users to easily and rapidly provision new services and release them, enabling them to pay for what they utilize and how much they use it. *On-demand self-service* is an appealing characteristic for consumers because it provides them the flexibility of provisioning a service exactly when they need it. The services provided over the cloud are *measured services*, which means that consumers only pay for how much service they consume; thus eliminating the need for investing in redundant computing resources. Cloud computing has benefits at the providers' end as well. A cloud computing provider pools its computing resources in order to serve multiple consumers by means of a *multi-tenant* provisioning model.

B. Service delivery models

Even though there are several definitions for delivery models of cloud computing services, three are widely adopted in the literature. *Software as a Service (SaaS)* delivery model is described as providing software applications/services over cloud infrastructure for consumers. These applications are accessible from various platforms through an easy-to-use client interface such as a web browser. *Platform as a Service (PaaS)* delivery model enables consumers to deploy their solutions to the cloud by means of platforms such as application servers and database services provided by the Cloud Platform Provider. *Infrastructure as a Service (IaaS)* is the lowest level of service model in cloud delivery models. In IaaS consumers acquire computing services and can deploy their own custom-configured systems in these resources potentially replicating their own existing infrastructures. Therefore, IaaS can also enable legacy system and software compliance.

C. Deployment models

The deployment model of a cloud platform is also important to consider when delivering or procuring on-line services [4]. Public cloud infrastructures are provisioned for use by any consumer; infrastructure exists in the premise of the provider. Private cloud infrastructure is provisioned for exclusive use of a single organization and can be owned by a single organization, a third party, or some combination of them. Community cloud infrastructure is provisioned for exclusive use by a particular community of users from organizations that constitute the specific community. Hybrid cloud is a composition of two or more of the models above.

D. Software Testing and Virtualization

Software testing is an integral part of the software development life cycle that span over all the development phases. One of the main challenges in software testing is deploying and maintaining a real-world test platform at the outset of a project. Virtualization technology has been utilized in testing various software since its inception in 1960's. IBM's CP-40 project might be considered as the pioneer of virtualization technology [1]. Among other goals of the project, CP-40 was mainly used by researchers as a tool to evaluate and test the performance of operating systems.

Developments in network infrastructure triggered a spur in Web-based service delivery. Riungu et.al. discuss the conditions that influence software testing as an on-line service and elicit important research issues [6]. They define on-line software testing as a model of software testing used to test an application provided as a service to customers across the Internet. This model promotes a demand-driven software testing market by enabling organizations and individuals to provide and acquire testing services on-demand. The concepts that affect software testing as an on-line service are domain knowledge, infrastructure, security, pricing, communication and skills [6]. On-line testing reduces costs related to installing and maintaining testing environment. It also introduces a new market where the providers and consumers can reach skilled test engineers on-demand.

III. Research Methodology

The main purpose of this paper is to classify research activities performed in cloud-based testing area, clarify the terminology used, identify any gaps or open issues that remain, and address those issues at a high level. There are currently two different perspectives on "cloud testing" and both cases can be considered as valid forms of "Testing as a Service" [7]:

1) Testing the cloud-resident applications,
2) Providing testing software as services in the cloud, and
3) Both of the above, i.e., testing cloud-resident applications by means of cloud-resident testing services.

The former deals with how applications perform in terms of functional correctness and speed when they are migrated to cloud. The latter deals with migration of the testing process itself into the cloud. This motivation enabled us to distinguish the problem domains of the literature. After thorough review of the selected papers we identified 11 major problem domains (Table I - II) depending on the problem/solution domain of the paper. The problem domains that we identified enables us to make a distinction between whether the test service is provisioned for cloud-resident applications or for other platforms (e.g., desktop applications, mobile applications, etc.).

During our search for the related literature, we found the following keywords and phrases to be useful:

- cloud application validation
- cloud application verification
- cloud computing testing
- software testing cloud
- testing cloud applications
- verification cloud

Cloud computing partially relates to and even depends on prior technologies such as virtualization, web services, utility

computing, multi-core and parallel programming and several others. One can go back and analyze how testing processes were affected by these enabling technologies over a long period of time. IEEE, ACM, Science Direct and Scirus are the main search engines we utilized for research. Due to proliferation of recent publications directly on Cloud Computing we decided to focus on publications dating 2009–2012, without limiting the search to any specific venue. Almost all conferences today in the fields of computer systems, data mining, software engineering and even consumer electronics hold special sessions on Cloud computing. Therefore, we did not lack any resources. Specifically, in these sessions we looked for papers that mentioned the keywords and phrases listed above. We eliminated any duplicate and incremental report. Our categorization approach resulted four distinct categories: *test level, test type, contribution* and *delivery model*. Based on this analysis we were able to identify trends and gaps.

IV. Evaluation

Our survey of software testing literature resulted eleven problem domains, each of which is analyzed according to Test Level, Test Type, Contribution and Delivery Model.

A. Categorization

Test level categorizes the papers against the levels of traditional V-Model of software testing: unit, integration, system and acceptance testing. *Test type* addresses the type of test that the investigated paper studies: functional, performance, security and interoperability. *Contribution* of papers classified whether they contribute to the field as test execution automation, test case generation, a framework that either defines a tool or a methodology or an evaluation paper. We further investigated and extracted what type of *delivery model* that a specific research builds on.

B. Gaps

In our review, we could not identify any research that deals with effects of cloud deployment model on providing software testing as a service over the cloud. We believe that the deployment model has a critical role in procuring software testing service as it is for other on-line services. For instance, community cloud model might be further investigated for promoting community testing or crowd-testing (e.g., UTest [57]).

Our categorization emphasizes correlation between testing level, testing type and delivery model. It can be seen from Table-I that interoperability testing presents opportunities for further research, which deals with cloud services interoperability. It's also shown in Table-I that acceptance testing has not been studied thoroughly.

We noticed that contribution of a literature and its delivery model are mostly interrelated. For instance, it can be seen in Table-II that [15] introduces a framework for testing and it is classified as an IaaS delivery model; so is [21].

Workload distribution and management over the cloud has being studied by the cloud community. Automated tests might be investigated in terms of their correlation with available task management frameworks or infrastructures. This subject is not studied thoroughly ([53], [7], [52]). We believe that task management issue in distributed and parallel applications has long been studied, and studying those solutions might facilitate task management for testing over the cloud.

As more and more services are migrated to the cloud verification of legacy applications over cloud will gain more attention by the research community. Especially acceptance testing of those applications needs to be well-structured in order to reap the benefits of cloud. Ding et al. describe why post-migration testing is necessary when migrating a complex application to cloud in [39]. They introduce a black-box approach for post-migration testing of Web applications without manually creating test cases. They propose to automate those tests and present a software module called Splitter. The tool executes automated functional test by using actual user workload in real-time to compare responses from the migrated application against those from the original production application before cut-over to the new platform. Migration of legacy systems should not only be investigated in terms of system and functional specifications but also studied in terms of performance, security, unit-level verification and integration of composing services (Table-II).

Many cloud services are provisioned through composition of several services.In the near future, several cloud infrastructure service providers may be utilized in providing value-added cloud services. Thus, interoperability testing of cloud infrastructures has potential for research.

C. Testing for the Cloud

Testing for the cloud defines the testing of applications that are specifically developed to run on a cloud platform. This fact entails that the application might be utilizing parallel computing features of cloud computing or it might be a multi-threaded application. Parallel program testing becomes more critical with the proliferation of cloud computing services.

Cloud service development and deployment, test task management, cloud infrastructure and storage, cloud applications domains are good examples of testing for the cloud. For example, Chan et.al. propose a graph-based modeling approach to cloud applications and attempt to support the approach with a testing paradigm for cloud applications [14]. The testing relates to the notion of model-based testing.

D. Testing on the Cloud

We distinguish the testing activities for on-premise applications as "testing on the cloud". In this type of service, the system under test resides either on-premise or on the cloud for testing purposes, but it's deployed on a platform other than cloud.

Testing for certification is a good example for testing on the cloud. On-demand service delivery attribute of software testing over the cloud paradigm might attract end-users to test the applications which they will install on their PC or

TABLE I
CATEGORIZATION OF LITERATURE BASED ON TEST LEVEL & TYPE

Problem Domain	Test Level				Test Type			
.	Acceptance Testing	System Testing	Unit Testing	Integration Testing	Functional Testing	Performance Testing	Security Testing	Interoperability Testing
Mobile App.s	X	[8], [3]	[9]	X	[3]	[8]	[8]	X
Cloud App.s	X	[10], [11], [12], [13], [14], [15], [16]	[11], [12], [9], [13]	[15], [17]	[10], [11], [12], [13], [14], [15]	[10], [16]	X	X
Desktop App.s	X	[18], [11], [16]	[9]	X	[18], [11]	[16]	X	X
Web Services & App.s	X	[19], [20]	[9]	[5]	[19], [20]	[5], [20]	X	X
Distributed & Parallel App.s	X	[21], [22], [23], [24]	[9]	X	[21], [22], [23], [24]	X	X	X
Cloud Service Dev. & Deployment	X	[25], [26], [27]	[25], [28], [9], [13]	[26], [25]	[27], [28], [13], [29], [30], [31], [32]	[26], [29], [33], [34]	[29]	[29], [30], [35], [36], [37]
Migration to Cloud	X	[38], [39], [40], [41], [6]	[6]	X	[38], [41], [6], [39], [40]	[41]	[41]	X
Cloud Infrastructure & Storage	X	[42], [43], [44], [45]	X	X	[44], [45]	[42], [43]	X	X
Real-Time Systems	X	[46], [47], [48]	[47], [49], [9]	X	[46], [47], [48]	[50]	X	X
Network Config.	X	[51]	X	X	[51]	X	X	X
Test Task Mang.	[52]	[53], [7], [52]	[7], [52]	X	[53], [7], [52]	X	X	X

mobile devices or check the applications' conformance to certain standards [11].

Unit testing activities are another area where on-demand software testing service can be utilized. Symbolic execution concept has been migrated to cloud environment, which facilitates automatic test case generation for unit tests ([47], [48], [9]). Symbolic execution presents opportunities for automatic test generation and test execution automation; but it's not widely studied according to the problem domains we presented (Table-II). Thus it presents further research opportunities.

Testing activities usually mean verification activities. Verification and validation should be considered as a complete service for the quality purposes. Verification and Validation as a Service (VVaaS) should answer both questions: whether the software does the right thing and whether the software is built to do the right thing. Thus acceptance testing should be considered as a new test service to be provided over cloud. VVaaS over the Cloud should be studied and promoted because one of the goals of software testing research is to automate the testing activities as much as possible, thereby

- significantly reducing the cost,
- minimizing human error and
- making regression testing easier.

V. RELATED WORK

There is a vast amount of literature regarding software testing in the cloud and testing cloud services. However, to the best of our knowledge, there is no comprehensive literature review that categorizes exiting body of work according to problem and solution domains. There have been previous works for identifying research issues for software testing in the cloud [31]. These works are based on a survey conducted with industry practitioners, in which issues are categorized from the application, management, legal and financial perspectives. The analysis of this survey reveals the requirements of a cloud-based testing solution from the viewpoint of industry practitioners [6].

VI. CONCLUSION

Cloud computing and software testing are likely to be active and popular research fields in the near future. Traditional software testing techniques are being adapted for the cloud. On the other hand, cloud computing itself is under constant evolution, continuously bringing in new opportunities and challenges for software testing research. In this paper, we have presented a classification of current research studies, identified gaps in the literature and investigated the correlation of software testing with different deployment models of cloud computing. Researchers in this field can benefit from the results in selecting their research direction and identifying new research opportunities for future work. We have observed that acceptance testing is an open research area for testing over the cloud. Test task management is also among the potential areas for further research. Finally, we believe that interoperability testing needs more emphasis as a research area to ensure reliable service composition by means of integrating services from different service delivery models. Our future research will be focusing on filling these gaps for achieving a comprehensive verification and validation model in cloud computing. We will specifically work on issues that facilitate cloud as a platform for acceptance and unit testing, and we will also focus on optimizing existing automated test tools for more proliferated use over the cloud.

TABLE II
CATEGORIZATION OF LITERATURE BASED ON CONTRIBUTION & DELIVERY MODEL

Problem Domain	Contribution				Delivery Model		
	Test Execution Automation	Test Case Generation	Framework	Evaluation	SaaS	PaaS	IaaS
Mobile App.s	[8], [3]	[3], [9]	X	X	X	[8], [3]	X
Cloud App.s	[10], [14]	[10]	[10], [12], [13], [15]	[11], [16]	[10]	[15]	[15]
Desktop App.s	X	[9]	[18]	[11], [16]	X	X	X
Web Services & App.s	[20]	[20], [9]	[20]	[5], [19]	X	[19], [20]	X
Distributed & Parallel App.s	[22], [23]	[9]	[21]	[24]	X	X	[21], [22], [23]
Cloud Service Dev. & Deployment	[25]	[25], [9], [26], [28]	[54], [25], [13], [17]	[27]	[25], [26], [29], [55], [32], [33], [34], [56]	[17]	X
Migration to Cloud	[39]	[39]	X	[38], [41], [6], [40]	[39]	X	X
Cloud Infrastructure & Storage	[44], [43], [45]	[44], [45]	[44], [42]	X	[44], [43], [45]	X	[42]
Real-Time Systems	[47], [48]	[47], [48], [49]	X	[46], [50]	[47], [49]	[47], [48]	X
Network Config.	[51]	[51]	[51]	X	[51]	X	[51]
Test Task Mang.	[53], [7]	X	[53], [52]	X	[7], [52]	[7]	X

ACKNOWLEDGMENT

This research was sponsored by EU-FP7 Marie Curie BI4MASSES project and by TÜBİTAK Center for Software Testing and Quality Assessment (C4STAQ) funded by Turkish Ministry of Development State Planning Organization (DPT).

REFERENCES

[1] Ibm cp-40 project. http://en.wikipedia.org/wiki/IBM_CP-40, 2012. [Online; accessed 12-Feb-2012].

[2] Won Kim, Soo Dong Kim, Eunseok Lee, and Sungyoung Lee. Adoption issues for cloud computing. In *Proc. of the 7th International Conference on Advances in Mobile Computing and Multimedia*, pages 2–5, New York, NY, USA, 2009.

[3] Youssef Ridene and Franck Barbier. A model-driven approach for automating mobile applications testing. In *Proc. of the 5th European Conference on Software Architecture: Companion Volume*, pages 9:1–9:7, New York, NY, USA, 2011.

[4] Peter Mell and Timothy Grance. The nist definition of cloud computing (draft) recommendations of the national institute of standards and technology. *Nist Special Publication*, 145(6):7, 2011.

[5] Mladen A Vouk. Cloud computing: Issues, research and implementations. *ITI 2008 30th International Conference on Information Technology Interfaces*, 16(4):31–40, 2008.

[6] Leah Muthoni Riungu, Ossi Taipale, and Kari Smolander. Software testing as an online service: Observations from practice. *IEEE International Conference on Software Testing Verification and Validation Workshop*, 0:418–423, 2010.

[7] Lian Yu, Wei-Tek Tsai, Xiangji Chen, Linqing Liu, Yan Zhao, Liangjie Tang, and Wei Zhao. Testing as a service over cloud. In *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*, pages 181 –188, June 2010.

[8] Srikanth Baride and Kamlesh Dutta. A cloud based software testing paradigm for mobile applications. *ACM SIGSOFT Software Engineering Notes*, 36(3):1–4, 2011.

[9] Matt Staats and Corina Păsăreanu. Parallel symbolic execution for structural test generation. In *Proc. of the 19th International Symposium on Software Testing and Analysis*, pages 183–194, New York, NY, USA, 2010.

[10] Zhen Ming Jiang. Automated analysis of load testing results. In *Proc. of the 19th International Symposium on Software Testing and Analysis*, pages 143–146, New York, NY, USA, 2010.

[11] George Candea, Stefan Bucur, and Cristian Zamfir. Automated software testing as a service. In *Proc. of the 1st ACM Symposium on Cloud Computing*, pages 155–160, New York, NY, USA, 2010.

[12] Jan S. Rellermeyer, Michael Duller, and Gustavo Alonso. Engineering the cloud from software modules. In *Proc. of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 32–37, Washington, DC, USA, 2009.

[13] Liang Zhao, Anna Liu, and Jacky Keung. Evaluating cloud platform architecture with the care framework. In *Proc. of the 2010 Asia Pacific Software Engineering Conference*, pages 60–69, Washington, DC, USA, 2010.

[14] W.K. Chan, Lijun Mei, and Zhenyu Zhang. Modeling and testing of cloud applications. In *Services Computing Conference, 2009. IEEE Asia-Pacific*, pages 111 –118, Dec. 2009.

[15] A.I. Avetisyan, R. Campbell, I. Gupta, M.T. Heath, S.Y. Ko, G.R. Ganger, M.A. Kozuch, D. O'Hallaron, M. Kunze, T.T. Kwan, K. Lai, M. Lyons, D.S. Milojicic, Hing Yan Lee, Yeng Chai Soh, Ng Kwang Ming, J-Y. Luke, and Han Namgoong. Open cirrus: A global cloud computing testbed. *Computer*, 43(4):35 –43, April 2010.

[16] Leah Riungu-kalliosaari, Ossi Taipale, and Kari Smolander. Testing in the cloud : Exploring the practice. *IEEE Software*, PP:1, 2011.

[17] Nianjun Zhou, Da Peng An, Liang-Jie Zhang, and Chih-Hong Wong. Leveraging cloud platform for custom application development. In *Proc. of the 2011 IEEE International Conference on Services Computing*, pages 584–591, Washington, DC, USA, 2011.

[18] T. Vengattaraman, P. Dhavachelvan, and R. Baskaran. A model of cloud based application environment for software testing. *CoRR*, abs/1004.1773:XX, 2010.

[19] Wang Jun and Fanpeng Meng. Software testing based on cloud computing. In *Proc. of the 2011 International Conference on Internet Computing and Information Services*, pages 176–178, Washington, DC, USA, 2011.

[20] Niclas Snellman, Adnan Ashraf, and Ivan Porres. Towards automatic performance and scalability testing of rich internet applications in the cloud. In *Proc. of the 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications*, pages 161–169, Washington, DC, USA, 2011.

[21] Toshihiro Hanawa, Hitoshi Koizumi, Takayuki Banzai, Mitsuhisa Sato, Shin'ichi Miura, Tadatoshi Ishii, and Hidehisa Takamizawa. Customiz-

ing virtual machine with fault injector by integrating with specc device model for a software testing environment d-cloud. In *Proc. of the 2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing*, pages 47–54, Washington, DC, USA, 2010.

[22] Takayuki Banzai, Hitoshi Koizumi, Ryo Kanbayashi, Takayuki Imada, Toshihiro Hanawa, and Mitsuhisa Sato. D-cloud: Design of a software testing environment for reliable distributed systems using cloud computing technology. In *Proc. of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pages 631–636, Washington, DC, USA, 2010.

[23] Toshihiro Hanawa, Takayuki Banzai, Hitoshi Koizumi, Ryo Kanbayashi, Takayuki Imada, and Mitsuhisa Sato. Large-scale software testing environment using cloud computing technology for dependable parallel and distributed systems. In *Proc. of the 2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, pages 428–433, Washington, DC, USA, 2010.

[24] Simone R. S. Souza, Maria A. S. Brito, Rodolfo A. Silva, Paulo S. L. Souza, and Ed Zaluska. Research in concurrent software testing: A systematic review. In *Proc. of the Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging*, pages 1–5, New York, NY, USA, 2011.

[25] Wei-Tek Tsai, Peide Zhong, J. Balasooriya, Yinong Chen, Xiaoying Bai, and J. Elston. An approach for service composition and testing for cloud computing. In *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, pages 631 –636, March 2011.

[26] C.R. Senna, L.F. Bittencourt, and E.R.M. Madeira. An environment for evaluation and testing of service workflow schedulers in clouds. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pages 301 –307, July 2011.

[27] Atif Farid Mohammad and Hamid Mcheick. Cloud services testing: An understanding. *Procedia CS*, 5:513–520, 2011.

[28] Linghao Zhang, Tao Xie, Nikolai Tillmann, Jonathan de Halleux, Xiaoxing Ma, and Jian Lu. Environment modeling for automated testing of cloud applications. *IEEE Software, Special Issue on Software Engineering for Cloud Computing*, 1:xx, March/April 2012.

[29] Tariq M. King and Annaji Sharma Ganti. Migrating autonomic self-testing to the cloud. In *Proc. of the 2010 Third International Conference on Software Testing, Verification, and Validation Workshops*, pages 438–443, Washington, DC, USA, 2010.

[30] Thomas Rings, Jens Grabowski, and Stephan Schulz. On the standardization of a testing framework for application deployment on grid and cloud infrastructures. In *Proc. of the 2010 Second International Conference on Advances in System Testing and Validation Lifecycle*, pages 99–107, Washington, DC, USA, 2010.

[31] L.M. Riungu, O. Taipale, and K. Smolander. Research issues for software testing in the cloud. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 557 –564, 30 2010-Dec. 3 2010.

[32] Wei-Tek Tsai, Qihong Shao, Yu Huang, and Xiaoying Bai. Towards a scalable and robust multi-tenancy saas. In *Proc. of the Second Asia-Pacific Symposium on Internetware*, pages 8:1–8:15, New York, NY, USA, 2010.

[33] Guanying Wang, Ali R. Butt, Prashant Pandey, and Karan Gupta. Using realistic simulation for performance analysis of map-reduce setups. In *Proc. of the 1st ACM Workshop on Large-Scale System and Application Performance*, pages 19–26, New York, NY, USA, 2009.

[34] Sebastian Gaisbauer, Johannes Kirschnick, Nigel Edwards, and Jerry Rolia. Vats: Virtualized-aware automated test service. In *Proc. of the 2008 Fifth International Conference on Quantitative Evaluation of Systems*, pages 93–102, Washington, DC, USA, 2008.

[35] Yongzhi Wang and Jinpeng Wei. Viaf: Verification-based integrity assurance framework for map-reduce. *2011 IEEE 4th International Conference on Cloud Computing*, pages 300–307, 2011.

[36] Philipp Zech. Risk-based security testing in cloud computing environments. *2011 Fourth IEEE International Conference on Software Testing Verification and Validation*, pages 411–414, 2011.

[37] Xiaofei Zhang, Hui Liu, Bin Li, Xing Wang, Haiqiang Chen, and Shizhong Wu. Application-oriented remote verification trust model in cloud computing. *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pages 405–408, 2010.

[38] Van Tran, Jacky Keung, Anna Liu, and Alan Fekete. Application migration to cloud: A taxonomy of critical factors. In *Proc. of the 2nd International Workshop on Software Engineering for Cloud Computing*, pages 22–28, New York, NY, USA, 2011.

[39] Xiaoning Ding, Hai Huang, Yaoping Ruan, Anees Shaikh, Brian Peterson, and Xiaodong Zhang. Splitter: A proxy-based approach for post-migration testing of web applications. In *Proc. of the 5th European conference on Computer systems*, pages 97–110, New York, NY, USA, 2010.

[40] Tauhida Parveen and Scott R. Tilley. When to migrate software testing to the cloud? In *ICST Workshops*, pages 424–427, 2010.

[41] Parastoo Mohagheghi and Thor Saether. Software engineering challenges for migration to the service cloud paradigm: Ongoing work in the remics project. In *Proc. of the 2011 IEEE World Congress on Services*, pages 507–514, Washington, DC, USA, 2011.

[42] Ji Ho Kim, Sang Min Lee, Dong Seong Kim, and Jong Sou Park. Performability analysis of iaas cloud. In *Proc. of the 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 36–43, Washington, DC, USA, 2011.

[43] Pallavi Joshi, Haryadi S. Gunawi, and Koushik Sen. Prefail: A programmable tool for multiple-failure injection. *SIGPLAN Not.*, 46:171–188, Oct. 2011.

[44] Cuong Pham, Daniel Chen, Zbigniew Kalbarczyk, and Ravishankar K. Iyer. Cloudval: A framework for validation of virtualization environment in cloud infrastructure. In *Proc. of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems&Networks*, pages 189–196, Washington, DC, USA, 2011.

[45] Lorenzo Martignoni, Roberto Paleari, Giampaolo Fresi Roglia, and Danilo Bruschi. Testing system virtual machines. In *Proc. of the 19th International Symposium on Software Testing and Analysis*, pages 171–182, New York, NY, USA, 2010.

[46] Jason S. Bolin, James Bret Michael, and Man tak Shing. Cloud computing support for collaboration and communication in enterprise-wide workflow processes. In *SERVICES*, pages 589–593, 2011.

[47] Liviu Ciortea, Cristian Zamfir, Stefan Bucur, Vitaly Chipounov, and George Candea. Cloud9: A software testing service. *SIGOPS Oper. Syst. Rev.*, 43:5–10, Jan. 2010.

[48] Stefan Bucur, Vlad Ureche, Cristian Zamfir, and George Candea. Parallel symbolic execution for automated real-world software testing. In *Proc. of The Sixth Conference on Computer Systems*, pages 183–198, New York, NY, USA, 2011.

[49] Yunho Kim and Moonzoo Kim. Score: A scalable concolic testing tool for reliable embedded software. In *Proc. of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering*, pages 420–423, New York, NY, USA, 2011.

[50] George Chang, Emily Law, and Shan Malhotra. Demonstration of lmmp automated performance testing using cloud computing architecture. In *Proc. of the 2nd International Workshop on Software Engineering for Cloud Computing*, pages 71–71, New York, NY, USA, 2011.

[51] Huan Liu and Dan Orban. Remote network labs: An on-demand network cloud for configuration testing. *SIGCOMM Comput. Commun. Rev.*, 40:83–91, Jan. 2010.

[52] Lian Yu, Xiaohu Li, and Zhongjie Li. Testing tasks management in testing cloud environment. In *Proc. of the 2011 IEEE 35th Annual Computer Software and Applications Conference*, pages 76–85, Washington, DC, USA, 2011.

[53] James Michael Ferris. Systems and methods for software test management in cloud-based network, 2009.

[54] Shigeru Hosono, He Huang, Tatsunori Hara, Yoshiki Shimomura, and Tamio Arai. A lifetime supporting framework for cloud applications. In *Proc. of the 2010 IEEE 3rd International Conference on Cloud Computing*, pages 362–369, Washington, DC, USA, 2010.

[55] Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Speculative analysis: Exploring future development dtates of software. In *Proc. of the FSE/SDP workshop on Future of Software Engineering Research*, pages 59–64, New York, NY, USA, 2010.

[56] M. Oriol and F. Ullah. Yeti on the cloud. In *Proc. of the Third International Conference on Software Testing, Verification, and Validation*, pages 434 –437, April 2010.

[57] Online software testing services community. http://www.utest.com/, 2012. [Online; accessed 12-Feb-2012].