IJITKM Volume 8 • Number 1 June-Dec 2014 pp. 85-90 (ISSN 0973-4414)

Cloud Testing: Issues, Challenges and Performance Analysis

Anish Soni
HCTM Technical Campus, Kaithal
soni_anish@yahoo.com

Abstract: Cloud computing have dominant role in all over the modern world, It is not only changes the way of obtaining computing resources (such as computers, infrastructures, data storage, and application services), but also changes the way of managing and delivering computing services, technologies, and solutions. Cloud computing leads an opportunity in offering testing as a service (TaaS) for SaaS and clouds. Meanwhile, it causes new issues, challenges and needs in software testing, particular in testing clouds and cloud-based applications. This paper provides a comprehensive tutorial on cloud testing and cloud-based application testing. It answers the common questions raised by engineers and managers, and it provides clear concepts, discusses the special objectives, features, requirements, and needs in cloud testing. It offers a clear comparative view between web-based software testing and cloud-based application testing. In addition, it examines the major issues, challenges, and needs in testing cloud-based software applications. Furthermore, it also summarizes and compares different commercial products and solutions supporting cloud testing as services.

Keywords: cloud computing, cloud-based software testing, testing cloud services, performance testing and evaluation, and scalability testing.

INTRODUCTION

Cloud computing fetches magnificent attention recently as it changes the way computation and services to customers[1], For example, it changes the way of providing and managing computing resources, such as CPUs, databases, and storage systems. Today, leading players, such as Amazon, Google, IBM, Microsoft, and Salesforce.com offer their cloud infrastructure for services. In 2010, Garner estimated that "the cloud service market will reach $150.1 billion in 2013". Similarly, Merrill Lynch also predicted that "cloud computing market will reach $160 billion in 2011". A recent study of Market Research Media forecasts that U.S. government spending on cloud computing is entering an explosive growth phase at about 40% CAGR over the next six years. Expenditure will pass $7 billion by 2015. Merrill Lynch estimates that within the next five years, the annual global market for cloud computing will surge to $95 billion. Cloud computing not only brings new business opportunities, but also causes some major impacts on software testing and maintenance. A major impact is known as Testing as a Service (TaaS) in Clouds[2]. TaaS cloud infrastructures is considered as a new business and service model, in which a provider undertakes software testing activities of a given application system in a cloud infrastructure for customers as a service based on their demands. Although there are many published papers discussing cloud architectures, technologies, and models, design, and management, cloud testing and TaaS are still new subjects in software testing community. Hence, test engineers and quality assurance managers encountered many issues and challenges in testing modern clouds and cloud-based applications. The rest of this paper is structured as follows. The next section discusses cloud testing concepts, including definitions, objectives, motivation and benefits, testing forms, types, and environments as well as activities. Section 3 is devoted to the discussion about what is new in cloud testing. It focuses on the new requirements, features, and differences between cloud testing and conventional software testing. Section 4 examines the primary issues, challenges, and needs in cloud testing. Section 5 reviews the existing emergent technologies, solutions, and tools in software cloud testing. Section 6 reviews the published papers in cloud testing and cloud-based software testing. Finally, the conclusion remarks are given in Section 7.

2. SCENARIO OF CLOUD BASED REMOTE TESTING

According to [3], cloud computing Manages a cost-effective and flexible means through which scalable computing power and diverse services (computer hardware and software resources, networks and computing infrastructures), diverse application services, business processes to personal intelligence and collaboration are delivered as services to large-scale global users whenever and wherever they need. Cloud computing is the next stage of the Internet evolution[4]. A typical cloud must have several distinct properties: elasticity and scalability, multi-tenancy, self-managed function capabilities, service billing and metering functions, connectivity interfaces and technologies. In addition, a cloud supports large scale user accesses at distributed locations over the Internet, offers on-demand application services at anytime, and provides both virtual and/or physical appliances for customers. There are three types of clouds[5]: a) private clouds, which are internal clouds based on a private network behind a firewall; b) public clouds, which are the clouds with public accessible services over the Internet; and c) hybrid clouds, which are made of different types of clouds, including public and private clouds.

2.1 What is Cloud Testing

Cloud testing[6] is the sort out an answer to the less-than-realistic performance test that originates within the infrastructure of one of our clients. Whenever we use cloud testing, we take advantage of hardware and bandwidth that more closely mimics our observed, real world conditions. Essentially, we execute the test in cloud-based infrastructure and bandwidth." (R V Ramanan, President – Global Delivery and Chief Software Architect, Hexaware Technologies) In short, cloud-based software testing refers to testing and measurement activities on a cloud-based environment and infrastructure by leveraging cloud technologies and solutions. It has four major objectives. To assure the quality of cloud-based applications deployed in a cloud, including their functional services, business processes, and system performance as well as scalability based on a set of application-based system requirements in a cloud. To validate software as a service (SaaS)[7] in a cloud environment, including software performance, scalability, security and measurement based on certain economic scales and pre-defined SLAs. To check the provided automatic cloud-based functional services, for example auto-provisioned functions.To test cloud compatibility and inter-operation capability between SaaS and applications in a cloud infrastructure, for example, checking the APIs of SaaS and their cloud connectivity to others.

2.2 Why is Cloud Testing Important

If we comparing with current software testing, cloud-based testing have several different benefits listed below.

- Reduce costs by leveraging with computing resources in clouds – This refers to effectively using virtualized resources and shared cloud infrastructure to eliminate required computer resources and licensed software costs in a test laboratory.
- Take the advantage of on-demand test services (by a third-party) to conduct large-scale and effective real-time online validation for internet-based software in clouds.
- Easily leverage scalable cloud system infrastructure to test and evaluate system (SaaS/ Cloud/Application) performance and scalability.

2.3 Performance of Cloud-Based Software Testing

There are four different forms of cloud-based software testing[8]. Each of them has different focuses and objectives.

- Testing a SaaS in a cloud – It assures the quality of a SaaS in a cloud based on its functional and non-functional service requirements.
- Testing of a cloud – It validates the quality of a cloud from an external view based on the provided cloud specified capabilities and service features. Cloud and SaaS vendors as well as end users are interested in carrying on this type of testing.
- Testing inside a cloud - It checks the quality of a cloud from an internal view based on the internal infrastructures of a cloud and specified cloud capabilities. Only cloud vendors can perform this type of testing since they have accesses to internal infrastructures and connections between its internal SaaS(s) and automatic capabilities, security, management and monitor.
- Testing over clouds – It tests cloud-based service applications over clouds, including private, public, and hybrid clouds based on system-level application service requirements and specifications. This usually is performed by the cloud-based application system providers.

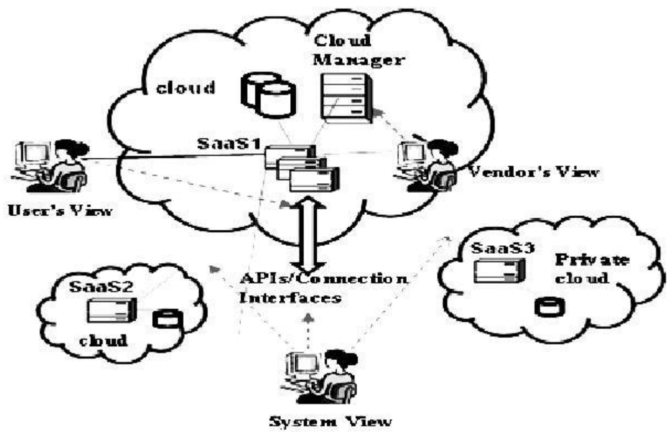

Figure 1 Different View for Cloud-Based Software Testing

3. PROPOSED ISSUES AND CHALLENGES IN CLOUD TESTING

There are a number of issues and challenges in testing clouds and cloud-based software. Here we discuss them from the following four areas.

3.1 On-demand test environment construction

How to set up a testing environment systematically (or automatically) for on-demand testing services in a cloud? Although the current cloud technologies support automatic provision of required computing resources for each SaaS (or application) in a cloud, there are no supporting solutions to assist engineers to set up a required test environment in a cloud using a cost-effective way. It is necessary to provide an on-demand test environment for TaaS customers. To do this, TaaS vendors need to provide a systematic solution to establish a required test environment based on the user's selection. In addition, engineers also found that there is a lack of cost-effective solutions for them to easily leverage their cloud-based applications (or SaaS) in a cloud with the existing test tools because most of them are not cloud-enabled.

3.2 Scalability and performance testing

Although many published papers discuss system performance testing and scalability evaluation in the past two decades, most of them address issues and solutions in conventional distributed software or web-based software systems. According to our recent literature survey on this subject, most existing papers focus on scalability evaluation metrics and frameworks for parallel and distributed systems [9][10][11]

3.3Testing security and measurement in clouds

Security testing has becoming a hot research subject with many open questions in current software testing community. Since security becomes a major concern inside clouds and security services become a necessary part in modern SaaS and cloud technology, engineers must deal the issues and challenges in security validation and quality assurance for SaaS and clouds.

3.4 On-demand testing issues and challenges

In TaaS, software testing services must be controlled and managed based on on-demand testing requests. This kind of new testing service model raised several issues and challenges.

- What is the automatic test process for TaaS to support on-demand automation testing?
- What are the well-defined cost models for testing as a service?
- How can engineers use a systematic approach to model and define QoS requirements?

What are the intelligent approaches to coping with the failures of software testing scripts/ test cases in an on-demand testing process?

3.5 Remote testing issues and challenges

Supporting on-demand software validation in clouds must address the regression testing issues and challenges caused by software changes and bug-fixing. However, most existing research in software regression testing pays most attention to re-test a specific software version in a pre-configured test environment. The multi-tenancy feature of clouds may cause the difficulty to apply the exsiting research work in cloud testing, specially for on-demand software regression testing service whenever software changes. In addition, we also lack of dynamic software validation methods and solutions to address the dynamic features of SaaS and clouds, for example automatic provisioned/de-provisioned features.

4. A Cost Model Based Analysis

Building on the example from the previous section, we propose an alternative cost model drawing from linear optimization. The model uses the concept of opportunity cost to balance automated and manual testing. The opportunity cost incurred in automating a test case is estimated on basis of the lost benefit of not being able to run alternative manual test cases. Hence, in contrast to the simplified model presented in Section 2, which focuses on a single test case, our model takes all potential test cases of a project into consideration. Henceforth, it optimizes the investment in automated testing in a given project context by maximizing the benefit of testing rather than by minimizing the costs of testing.[12]

4.1 Fixed Budget

First of all, the restriction of a fixed budget has to be introduced to our model. This restriction corresponds to the production possibilities frontier described in the previous section. R1: $na * Va + nm * Dm \leq B$ na := number of automated test cases nm := number of manual test executions Va := expenditure for test automation Dm := expenditure for a manual test execution B := fixed budget Note that this restriction does not include any fixed expenditures (e.g., test case design and preparation) manual testing. Furthermore, with the intention of keeping the model simple, we assume that the effort for running an automated test case is zero or negligibly low for the present. This and other influence factors (e.g., the effort for maintaining and adapting automated tests) will be discussed in the next section. This simplification, however, reveals an important difference between automated and manual testing. While in automated testing the costs are mainly influenced by the number of test cases (na), manual testing costs are determined by the number of test executions (nm). Thus, in manual testing, it does not make a difference whether we execute the same test twice or whether we run two different tests. This is consistent with manual testing in practice – each manual test execution usually runs a variation of the same test case [13]

4.2 Maximizing the Benefit

Third, to maximize the overall benefit yielded by testing, the following target function has to be added to the model. T: Ra(na) + Rm(nm) max Maximizing the target function ensures that the combination of automated and manual testing will result in an optimal point on the production possibilities frontier defined by restriction R1. Thus, it makes sure the available budget is entirely and optimally utilized.

4.3 Real Example

To illustrate our approach we extend the example used in Section 3. For this example the restriction R1 is defined as follows. R1: $na * 1 + nm * 0.25 \leq 75$ To estimate benefit of automated testing based on the risk exposure of the tested object, we refer to the findings published by Boehm and Basili [5]: “Studies from different environments over many years have shown, with amazing consistency, that between 60 and 90 percent of the defects arise from 20 percent of the modules, with a median of about 80 percent. With equal consis- tency, nearly all defects cluster in about half the modules produced.” Accordingly we categorize and prioritize the test cases into 20 percent highly beneficial, 30 percent medium beneficial, and 50 percent low beneficial and model following alternative restrictions to be used in alternative scenarios. R2.1: $na \geq 20$ R2.2: $na \geq 50$ To estimate the benefit of manual testing.

we propose, for this example, to maximize the test coverage. Thus, we assume an evenly distributed risk exposure over all test cases, but we calculate the benefit of manual testing based on the number of completely tested releases. Accordingly we categorize and prioritize the test executions into one and two or more completely tested releases. We model following alternative restrictions for alternative scenarios. R3.1: $nm \geq 100$ R3.2: $nm \geq 200$ Based on this example we illustrate three possible scenarios in balancing automated and manual testing. Figures 4a, 4b and 4c depict the example scenarios graphically.

• **Scenario A** – The testing objectives in this scenario are, on the one hand, to test at least one release completely and, on the other hand, to test the most critical 50 percent of the system for all releases. These objectives correspond to the restrictions R3.1 and R2.2 in our example model. As shown in Figure 4a the optimal solution is point S1 (na = 50, nm = 100) on the production possibilities frontier defined by R1. Thus, the 50 test cases referring to the most critical 50 percent of the system should be automated and all test cases should be run manually once.

Scenario B – The testing objectives in this scenario are, on the one hand, to test at least one release completely and, on the other hand, to test the most critical 20 percent of the system for all releases. These objectives correspond to the restrictions R3.1 and R2.1 in our example model. As shown in Figure 4b any point within the shaded area fulfills these restrictions. The target function, however, will make sure that the optimal solution will be a point between S1 (na = 50, nm = 100) and S2 (na = 20, nm = 220) on the production possibilities frontier defined by R1. Note: While all points on R1 between the S1 and S2 satisfy the objectives of this scenario, the point representing the optimal solution depends on the definition of the contribution to risk mitigation of automated and manual testing, Ra(na) and Rm(nm).

• **Scenario C** – The testing objectives in this scenario are, on the one hand, to test at least two releases completely and, on the other hand, to test the most critical 50 percent of the system for all releases. These objectives correspond to the restrictions R3.2 and R2.2 in our example model. As shown in Figure 4c a solution that satisfies both restrictions cannot be found.

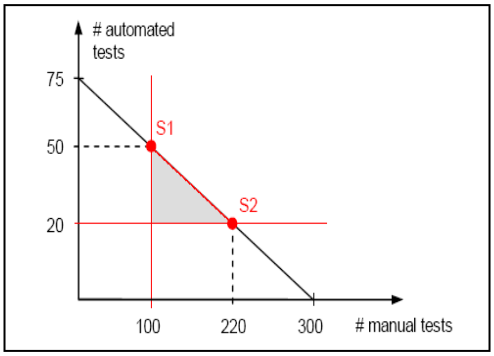

Figure 2: Automated based Cloud testing

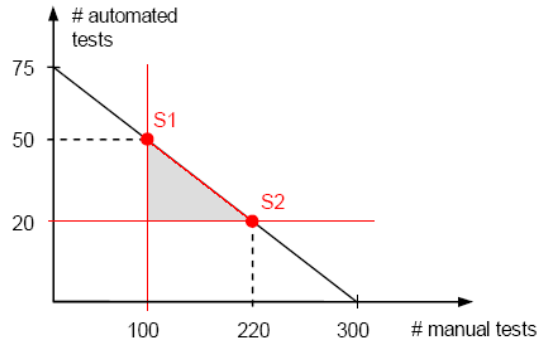

Figure 3: Manual based Cloud testing

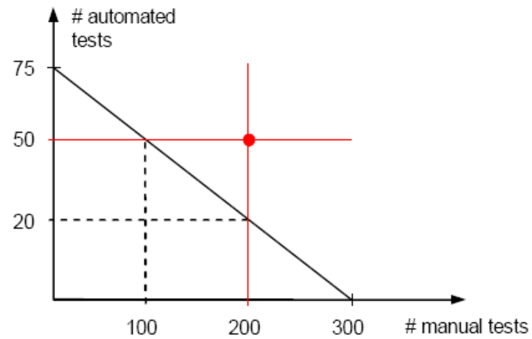

Figure 4: Scenario of Auto vs. Manual Test

5. CONCLUSION AND FUTURE WORK

Cloud testing is becoming a hot research topic in cloud computing and software engineering community. As the advance of cloud technology and testing as services, more research work must be done to address the open issues and challenges in cloud testing and TaaS. More innovative testing techniques and solutions, and QoS standards are needed to support on-demand testing services in a scalable cloud infrastructure. This paper provides a comprehensive review and tutorial on cloud testing by discussing the related concepts, issues, and challenges. The major contributions of this paper include its insightful discussion about cloud testing in terms of its special requirements, benefits, and features as well as the comparison with conventional testing. Moreover, cloud testing opportunities, current major players, and existing research work are reviewed. The Conclusion of this research and review paper is analyze to the manual testing drawback in software testing rather more benefits of automated software testing tools. The enlightened of this modern approaches leads to the new Methodologies of software test automation. The destination of software testing is considered to succeed when an error is detached. Effective Conclusions are given below. Software testing is an art. Most of the testing methods and practices are not very different from 20 years ago. In the current era there are many tools and techniques available to use. Good testing also requires a tester's creativity, experience and intuition, together with proper techniques. Testing is more than just debugging. Testing is not only used to locate defects and correct them. It is also used in validation, verification process, and reliability measurement. Although manual testing is not expensive but is no more effective rather automated testing because automation is a good way to cut down cost and time. Testing efficiency and effectiveness is the criteria for coverage-based testing techniques.

REFERENCES

[1] J. Hurwitz, M. Kaufman, and R. Bloor, "Cloud Computing for Dummies," Wiley Publishing, Inc. 2010.

[2] T. Vengattaraman, P. Dhavachelvan, R. Baskaran, "Model of Cloud Based Application Environment for Software Testing," (IJCSIS)
International Journal of Computer Science and Information Security, Vol. 7, No. 3, 2010.

[3] Leah Muthoni Riungu, Ossi Taipale, Kari Smolander, "Software Testing as an Online Service: Observations from Practice," In Third International Conference on Software Testing, Verification, and Validation Workshops (ICSTW), 418-423, 2010.

[4] Tauhida Parveen, Scott Tilley, "When to Migrate Software Testing to the Cloud?," In Third International Conference on Software
Testing, Verification, and Validation Workshops (ICSTW), 424-427, 2010.

[5] W. K. Chan, Lijun Mei, Zhenyu Zhang, "Modeling and Testing of Cloud Applications," In 2009 IEEE Asia-Pacific Service Computing Conference (APSCC 2009), Singapore; Dec 7-11, 2009.

[7] George Candea, Stefan Bucur, Zamfir Cristian, "Automated Software Testing as a Service (TaaS)," In the 1st ACM Symposium on Cloud Computing, 2010.

[8] H. Lu, W. K. Chan, and T. H. Tse., "Testing pervasive software in the presence of context inconsistency resolution services," In the 30th International Conference on Software Engineering (ICSE 2008), 61–70, 2008.

[9] G. Goth, "Googling test practices? Web giant's culture encourages process improvement," IEEE Software, vol. 25, no. 2, 92-94, 2008.

[10] R. Collard, "Performance innovations, testing implications," Software Test & Performance Magazine, Vol. 6, No. 8, 19-20, August 2009.

[11] Y. Yang, C. Onita, J. Dhaliwal, X. Zhang, "TESTQUAL: conceptualizing software testing as a service," In the 15th Americas conf. on information systems, 6-9.08, San Francisco, California, USA, paper 608, 2009.

[12] Leo.van der Aalst, "Software testing as a service (STaaS)," Internet: http://www.tmap.net/Images/Paper%20STaaS_tcm8-47910.pdf, [May 06, 2009].

[13] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, G. Candea, "Cloud9: A software testing service," In The 3rd SOSP Workshop on Large Distributed Systems and Middleware (LADIS), Big Sky, MT, October 2009.