A Survey of Testing Methodologies for Cloud-based applications

Sajjan G. Shiva, Sandeep Ramesh, and Ramya Dharam
Department of Computer Science
University of Memphis
Memphis, TN-38152, USA
{sshiva, sramesh, rdharam}@memphis.edu

***Abstract* – In present days, Cloud Computing is gaining widespread popularity among business owners and users for hosting and delivering services such as software applications, data access, storage, and management, etc., over the Internet. This technology offers users an on-demand access to shared resources, services, and applications online by eliminating the need for tedious installation procedures. Testing Cloud-based applications to verify and validate their conformance with existing business models is termed as Cloud Testing. It is gaining increased attention from both industry and research to ensure correctness, functional and non-functional, of the Cloud-based applications. In this paper, we provide a survey of various methodologies, in addition to conventional testing techniques, used to test the Cloud-based applications/services provided by Cloud.**

Keywords – Cloud Computing, Cloud Testing, Infrastructure as a Service (IaaS), Testing as a Service (TaaS), Software as a Service (SaaS), Platform as a Service (PaaS), Cloud Services

I. INTRODUCTION

Cloud Computing is a means of accessing remote software applications, data and storage resources using internet. Business applications and customers can use the resources provided by Cloud services online without installing them on their local machines. This in turn reduces the complexity involved in configuring the applications, and saves a lot of time spent on setting up the computation environment.

In order to efficiently provide resources such as applications, platforms, or infrastructure to the users, and to achieve efficient utilization of the services, various service driven business models are implemented by Cloud service providers. Based on the types of services provided by Cloud Computing environments, Cloud services can be classified into three main categories [2] and [3] as discussed below:

- Software as a Service (SaaS) where users are given access to applications and products online immediately when requested. For example the online services from Google Docs can be accessed by any user by simple creating an account and accessing the service directly from a web browser, without requiring any additional installations.
- Platform as a Service (PaaS) where developers are assigned their requested runtime environment to design and implement their code/ applications. For example, Windows Azure [11] is a cloud services operating system that serves as the development, service hosting and service management for windows Azure platform. This platform helps developers to host and manage web application through Microsoft datacenters with on-demand compute and storage options.
- Infrastructure as a Service (IaaS) where computing resources such as power, storage, networking components are available to users for their computations. Amazon Elastic Computer Cloud (Amazon EC2) [12] offers computing resources such as computing power and memory on demand in the Amazon Web Services (AWS) cloud.

Cloud Testing is a means of testing various Cloud services/applications that use resources (hardware, software, and infrastructure) found in the Cloud. It refers to the verification and validation of the resources that are available on demand, to ensure their conformance to the expectations of the Cloud Computing business models.

One of the distinct features of Cloud Testing, termed as Testing as a Service (TaaS), refers to offering on-demand static/dynamic testing services over the Cloud to its customers. TaaS can be used for validation and verification of various products owned by individuals or organizations. For instance, instead of installing load testing product to test proprietary application, you can avail this offering on demand.

Conventional testing methodologies are suitable for on-premise applications and environments where as in case of Cloud offerings, equal or more emphasis is required on non-functional requirements. Some of the challenges faced by vendors when migrating from in-premise application to Cloud-based platform are listed as follows:

- Data Integrity to ensure that the stored user data in the cloud is not altered by anyone.

- Security and Privacy to assure high level user authentication, and to handle issues such as unauthorized access, etc.

- Availability to ensure that user requests for services such as storage, resources are available on demand.

- Scalability to support multiple users simultaneously.

- Interoperability to ensure that users or applications of different platforms are supported.

- Disaster Recovery to handle any failure issues with respect to cloud services.

- Multi-tenancy to ensure that the cloud services can be used and offered by different classes of service providers and customers.

The main objectives of Cloud Testing [2] are discussed below:

- Verifying the Cloud-based applications to assure that they meet all the functional and non-functional requirements that include services, business process, performance, scalability, etc.

- Validating the software services (SaaS) provided in a Cloud environment, including software performance, scalability, security and measurement based on certain economic scales and pre-defined SLAs.

- Checking the Cloud-based functional services that include provisioning of resources, for example auto-provisioned functions.

- Testing/ensuring the 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.

Various forms of Cloud-based software testing as mentioned in [2] are summarized below:

- Testing of a Cloud - deals with validating a Cloud from an external view based on service capabilities and features. Cloud and SaaS vendors are mainly interested in performing this testing.

- Testing inside a Cloud – deals with testing the quality of a Cloud from internal view based on infrastructure and specified Cloud capabilities. This type of testing is mainly performed by Cloud vendors as they have complete access to the internal Cloud infrastructure and internal Cloud features.

- Testing over Clouds – deals with testing of applications over Clouds to check their conformance with the system level service requirements and specifications, and is usually performed by Cloud-based application system providers.

- Testing a SaaS in a Cloud – deals with testing the quality of services offered in a Cloud (or a SaaS program). The testing involves unit testing, integration, system function validation and regression testing, as well as performance and scalability evaluation. It also includes testing service APIs and connectivity in a Cloud environment, SaaS-based security and functional features, and scalability and elasticity features.

It would be interesting to study cloud testing from a stakeholder's point of view. This requires us to look at the system from a cloud service provider, vendors, and customer's prospective as discussed below[14]:

1. Developers and testers: offer cloud resources as service that serves as a platform for vendors and customers to access various cloud based resources. These services need to be tested to make sure that they provide virtualization, resource pooling, resource-reusability, reallocation, and multi-tenancy.

2. Administrators: handle the development of proprietary or third party applications, and they need to make sure that infrastructure and other offered services will perform efficiently under pressure in a real-world condition. Also, the vendors need to ensure that their developed applications provide an on-demand scaling, self-provisioning, and maintenance of full software lifecycle.

3. End user's: consumers of cloud applications and services offered by either service providers or vendors. Testing has to done to provide reliable and uninterrupted access to those resources, and offer guaranteed Quality of Service (QoS). They can use cloud applications as per their convenience and get billed for the resources that they use. Also, the system must be thoroughly tested to make sure that it accurately monitors their usage.

Many small and medium-sized businesses find cloud-based testing an ideal approach since they no longer need to make upfront investment in infrastructure, tool license, configuration, and maintenance of test environments, and they pay only for what they use [15]. It transforms their Cap-Ex to Op-Ex. The various problems that were faced

by dropbox in the recent years prove the need for testing cloud based applications. [16]

In this paper, we provide a survey of various methodologies, in addition to conventional testing techniques, used to test the Cloud-based applications/ services provided by Cloud i.e. our focus is on testing a SaaS in a Cloud. In Section II, we discuss various methodologies presently used to test SaaS based Cloud applications, and a summary of currently used Cloud Testing Tools is also discussed.

II. CLOUD TESTING METHODOLOGIES

In this section, we provide a survey of different techniques available to test Cloud-based applications [2], [3], and [4].

Conventional testing techniques are used to measure the quality of applications regardless of platform (in-premise/Cloud-based) on which they will be executed and are discussed below:

1. Component Testing
2. Function Testing
3. Integration Testing
4. GUI Interface Testing
5. Concurrence testing: multi-user
6. Risk-Based Testing
7. Acceptance Testing
8. Accessibility Testing
9. Regression Testing
10. Installation and configuration
11. Installation Qualification Testing
12. API Testing and Unit test
13. Navigability Testing
14. Automation
15. Mutation Test
16. Performance
17. DB Testing
18. Smoke Test
19. Web Services
20. Web Compatibility

In addition to conventional testing techniques, Cloud-based applications must be tested for more added features to assure that they meet their business models. Below is a list of most commonly used testing techniques for Cloud-based applications:

1. Performance Testing – Used to test the behavior of an application or a system when subjected to increased load from multiple locations.

2. Latency Testing – deals with testing the latency i.e. the time between requests made and response received after deploying it in the Cloud. Since access to Cloud applications are made by using internet, latency testing plays an important role.

3. Load Testing – test the application when subjected to requests from many users, in terms of multiples of hundreds.

4. Stress Testing – testing the system/ or applications to identify issues when used at 2x, 3x, or increased expected usage.

5. Capacity Testing – validate the Cloud resources used by the applications to determine the threshold usage of the Cloud resources.

6. Fail-over Testing – test the behavior of applications under anticipated load with component failure during the course of a test.

7. Application Security Testing – testing whether it is appropriate to migrate or design an application to run in the Cloud, and to identify the appropriate platform for the application.

8. Governance Risk Compliance Testing – focuses on listing threats, vulnerabilities, and risks that are associated to all three parts of Cloud Computing – IaaS, SaaS, and PaaS.

9. Availability Testing – the goal is to ensure that Cloud services are available all times and there are no unplanned sudden downtimes.

10. Interoperability Testing – the main focus of this testing is to ensure that the application under test is compatible and can be executed in multiple environments/ Cloud platforms. It also involves testing the ease with which the application can be switched between different infrastructures.

11. Disaster Recovery Testing – the main goal is to assure that the Cloud services under test are available all the time. In case of any disaster or failure, the applications must ensure a minimal recovery time to avoid adverse effects on the customers.

12. Multi-Tenancy Testing: the focus of this testing is to ensure that users from different organizations are able to use the Cloud services. It also involves validating the Cloud offering such that the service is customized for every client's requirements.

13. Customization and configuration Testing – Assure the quality of tenant-based customizations and configurations in SaaS databases, workflows, user interfaces, and functional services.

14. Connectivity testing – involves validating the service APIs and connectivity interfaces

provided by the Cloud and SaaS to its customers/users.

15. User Interface Portability and Compatibility – involves testing user interfaces of the Cloud-based application, and validating its compatibility when migrated to a Cloud platform.

16. Continuous Upgrade Testing – involves continuous validation of SaaS for any new upgrades i.e. whenever new tenants are added, and/or existing software is changed.

Few of the most commonly used tools for Cloud Testing are summarized below [4] and [5]:

Performance Monitoring tools such as: [4]

- Perfecto Mobile's
- Keynote
- Monitis
- Cloudsleuth
- BrowserMob
- CloudTools
- GFI
- LoadStorm
- CloudHarmony
- InterMapper
- BlazeMeter

Web Functional/Regression Test Tools [4]

- Windmill
- QEngine
- Soasta
- Selenium

Large Scale Performance & Scalability Testing [5]

- ETHZ [6]
- YCSB [7]
- Benchmark [8]

Fault Tolerance and Recovery Testing [5]

- D-Cloud [9]
- Pre-Fail [10]

III. Conclusion

Cloud Computing technology has recently emerged as a new paradigm for offering services to users over the Internet. There is a need for real world testing to be done on mobile and web based applications. Testing of the Cloud services, termed as Cloud Testing, pays an important role in Cloud Computing to verify and validate the conformance of the applications with their requirements and business models. In this paper, we provide a survey of various Cloud Testing techniques presently used, and their objectives.

IV. References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, September 2011.

[2] J. Gao, X. Bai and W. T. Tsai, "Cloud Testing – Issues Challenges Needs and Practice," Software Engineering: An International Journal, no. 1, vol. 1, pp. 9-23, 2011.

[3] N. Mehrorta, "View Point – Cloud-Testing vs. Testing a Cloud," 10th Annual International Software Testing Conference, 2011.

[4] Belatrix Software Factory, "Cloud Testing Best Practices".

[5] X. Bai, M. Li, B. Chen, W. T. Tsai, and J. Gao, "Cloud Testing Tools," IEEE 6th International Symposium on Service Oriented System Engineering (SOSE), 2011 .

[6] J. Moreno, "A Testing Framework for Cloud Storage Systems", Diss. Master Thesis, Eidgenössische Technische Hochschule, Zürich, 2010.

[7] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," 1st ACM Symposium on Cloud Computing, pp. 143–154, 2010.

[8] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the Weather Tomorrow? Towards a Benchmark for the Cloud," Second International Workshop on Testing Database Systems, pp. 9:1 - 9:6, 2009.

[9] T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, T. Hanawa, and M. Sato, "D-Cloud: Design of a Software Testing Environment for Reliable Distributed Systems using Cloud Computing Technology," 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 631–636, 2010.

[10] P. Joshi, H. S. Gunawi, and K. Sen, "PreFail: a Programmable Tool for Multiple-Failure Injection," ACM International Conference on Object oriented Programming Systems Languages and Applications," pp. 171–188, 2011.

[11]http://www.windowsazure.com/en-us/develop/net/fundamentals/intro-to-windows-azure/

[12] http://aws.amazon.com/ec2/

[13]http://tecires.ecs.soton.ac.uk/cloud_computing.php

[14] http://tecires.ecs.soton.ac.uk/cloud_computing.php

[15]http://www.cognizant.com/Futureofwork/Documents/ITIS-Cloud-Based-Testing.pdf

[16]http://www.cognizant.com/insights/perspectives/taking-testing-to-the-cloud