Software Testing in Cloud Platform: A Survey

A.Vanitha Katherine,
Associate Professor,
Department of MCA,
PSNA College Of Engineering & Technology,
Dindigul.

K. Alagarsamy,
Computer Centre,
Madurai Kamaraj University ,
Palkalai Nagar, Madurai 625 021.

ABSTRACT
Software Testing is a challenging activity for many software engineering projects and it is one of the five main technical activity areas of the software engineering lifecycle that still poses substantial challenges. Testing software requires enough resources and budget to complete it successfully. But most of the organizations face the challenges to provide enough resources to test their software in distributed environment, with different loading level. This leads to severe problem when the software deployed into different client environment and varying user load. Cloud computing is a one of the emerging technology which opens new door for software testing. This paper investigates the software testing in cloud platform which includes cloud testing models, recent research work, commercial tools and research issues.

Keywords
Software Testing, Cloud Testing, Testing Tools

1. INTRODUCTION
Software testing is an integral and important phase of the software development process. Testing requires expensive dedicated infrastructure and resources that were only used sporadically which scrutinizes the application's performance, reliability, speed, security and functionality. Since, business applications are growing in complexity, it is somewhat difficult for organizations to build and maintain in-house testing facilities that imitate real-time environments. This is where cloud testing has emerged as a fresh approach to testing where cloud computing environments are leveraged to simulate real-world user traffic by significantly decreasing costs [1]. This can also be extended to classical functional, regression and other testing of regular products in a product development cycle from a perspective of cost. In essence, cloud testing is a form of software testing wherein testing is done using resources, machines or servers from the cloud infrastructure. Besides, the entire testing environments can be obtained from the cloud on-demand at a cost that is practical and reasonable due to the pay-for-use nature of cloud computing and with a lead-time that is near impossible within a company's own data center.
Initially, this concept took shape when companies started using numerous machines booted up in the cloud in order to simulate web traffic and carry out performance tests on Web sites. Now remote machines in the cloud are used to provide a common ground for testers to test and developers to isolate and resolve the observed software defects. Apparently, cloud testing has traditionally been used to refer to load and performance testing of Web sites. However, with increasing maturity of technology, all kinds of enterprise software can be tested for functional and performance issues before going in for full fledged enterprise deployment. The below figure 1 indicates that testing and application development rank second (57%) as the most likely workload to be put into the cloud after Web sites (61%)[2]. Hence the cloud platform provides cost effective solution to the software testing, still it poses the some challenges.

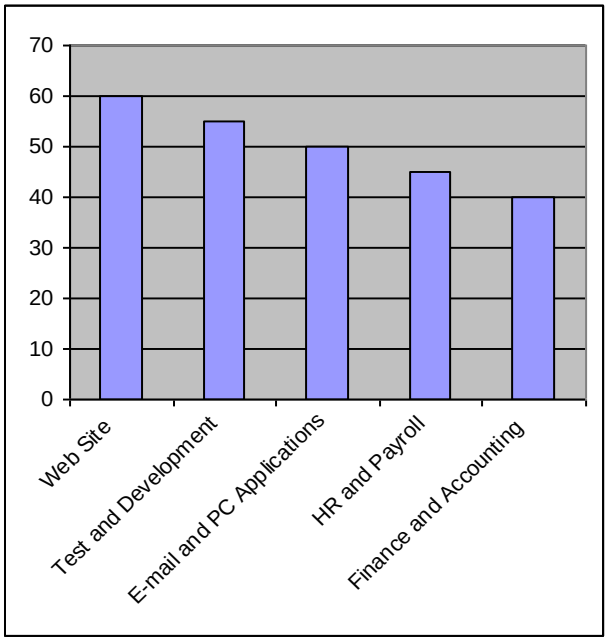

Figure1: TOP Applications in Cloud

The rest of paper is organized as follows; Section 2 presents the cloud testing model. Section 3 reviews the related work. Section 4 highlights the potential risks in the cloud testing. Section 5 presents the commercial tools. Section 6 highlights the research issues and finally Section 7 concludes the paper.

2. CLOUD TESTING MODEL

Software Testing in Cloud has three models.

2.1 Testing as a Service (TaaS)

TaaS concept was initially introduced by Tieto in Denmark in 2009. Software Testing as a Service (TaaS) is a model of software testing used to test an application as a service provided to customers across the Internet. By eliminating the need to test the application on the customer's own computer with testers on site, TaaS alleviates the customer's burden of installing and maintaining test environments, sourcing and (test) support. Using TaaS can also reduce the costs of testing, through less costly, on-demand pricing.

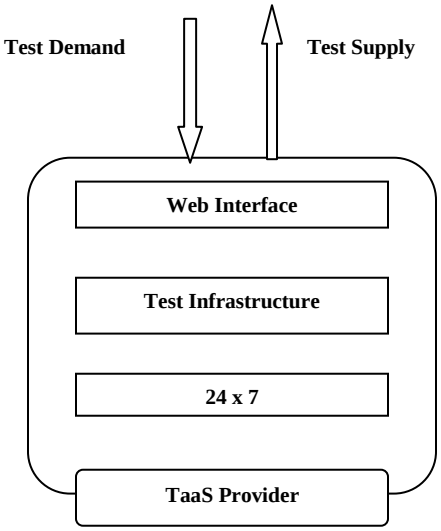

Figure 2: TaaS Process

2.2 Testing Support as a Service (TSaaS)

To enhance testability of autonomic services, TSaaS was proposed so that each service will expose both production and test environment to external users. Test functions (such as specification, execution, configuration and reporting) are exposed as API services. King, et al.,[5], [6] applied autonomic computing concepts to testing of adaptive systems, called autonomic self-testing (ATC). The technique was then migrated to the cloud platform [7], called TSaaS, so that services that are hosted on remote cloud platform can expose their test support APIs for partner providers.A self-test harness is developed to manage testing workflow and activities. It monitors changes or updates on hosted services, utilizes necessary infrastructure services, and invokes TSaaS supporting services to validate the changes. Test operations exposed as supporting services include test setup, input, assertions, and teardown operations. These services are provided for cloud partners during the development, testing and maintenance of tailor-made cloud applications and services. They can also be used for design, build, and deployment of automated tests across administrative domains.

2.3 Testing inside Cloud

Testing applications that are hosted and deployed in a cloud environment, or testing the cloud infrastructure itself.

3. RELATED WORK

This section briefly reviews the published research work relating to software testing using Cloud Platform.

Vengattaraman et al. in [8] propose their initial work on modeling of cloud based application environment for software testing by focusing On-Premises Applications over clouds. Its major objective is to present the relationships between different application services over clouds and external consumer services. But it does not address how to use this model in cloud testing.

Yang Yang at al. [9] discussed that software testing can be conceptualized as a service rather than being viewed as a sequential line of responsibility in software development. In their view, TaaS has two key aspects: (1) a service to developers, and (2) a service to end users. Their paper discusses software testing as a service from software quality assurance perspectives.

Yu, et al., [12] defined a 5-layer TaaS framework based on cloud infrastructure services, including:1) test tenant and test service contributor layer, 2) test task management layer, 3) test resource management layer, 4) test layer (and it has three components: testing service composition, testing service pool, and testing task reduce), and 5) test database layer. This paper proposed an automated testing platform TaaS on a cloud. This platform adopts cloud computing technique to build the elastic resource infrastructures, and provide various kinds of testing services to testing users.

Candea [10] identified three categories of testing services: $TaaS_D$ for developers, $TaaS_H$ for end users and $TaaS_C$ as certification service. They argue that with a pricing model, TaaS can be operated as a public service and as a business, targeting at the ”long-tail” small business companies.

Liviu Ciortea et al. in [11] introduce Cloud9, a cloud based testing service that promises to make high quality testing fast, cheap, and practical. Cloud9 is the first parallel symbolic execution engine to run on large shared- clusters of computers, and its test harness uses the aggregate memory and CPU resources based on compute utilities like Amazon EC2. The paper reports their initial prototype results. In addition, some initial cloud-based test experiments are reported.

Banzai et al [12] from University of Tsukuba in Japan developed “D-Cloud”, is a dedicated simulated test environment built upon Eucalyptus, an open-source cloud infrastructure providing similar functionalities as Amazon EC2. It uses QEMU, an open-source virtual machine software, to build virtual machine for simulating faults in hardware including disk, network and memory.

Parveen et al [53] migrates JUnit test framework to Hadoop platform. JUnit test cases are created as independent Hadoop MapReduce jobs. The map() function receives test jobs as <

testname; testcommand> pair. At each node, the command is executed as a process. The reducer gets < testname; testresult> from each map and combines all the results. Experiments shows that a 150-node cluster can produce 30x improvement compared with sequential test executions on a local computer.

Moreno et al [14] proposed a new distributed testing architecture for simulating parallel jobs. This framework contains two types of nodes - Master and Slave. Master is unique identified, which is responsible for distribution, synchronization and management of all slave nodes. Master is started with a given test. It waits for enough slaves to connect to it, then sends every slave corresponding tasks. During the execution, the master controls the execution sequence of slaves to guarantee that all tasks in a step start at the same time. The slaves run testing tasks and store test results locally, including nodes' states collected by daemon thread at each node. At the end of the test, master recollects every slave's results, analyzes and then generates statistics and graphs for analyzing test execution.

Baride, et al [15] proposed a cloud-based approach for mobile application testing where infrastructure services are used to simulate diversified mobile devices, hardware configurations, heterogeneous application platforms, and complex dependencies.

4. RISKS ASOOCIATED WITH CLOUD TESTING

In this section, presents some of the potential risks associated with cloud testing [16].

4.1 Security

Security in the public cloud is still a major concern. Cloud test is based on the internet, so it may come up to the situation about the Leakage of private information, internet suspending, and service provider may suddenly announce disruption of service due to a maintenance window, slow internet speed, virus attack and so on. Procedures are being developed to improve security and performance in the public cloud.

4.2 Lack of standards

Presently, there are no universal/standard solutions to integrate public cloud resources with user companies' internal data center resources. Public cloud providers have their own architecture, operating models and pricing mechanisms and offer very little interoperability. This poses a big challenge for companies when they need to switch vendors.

4.3 Infrastructure

Some cloud providers offer only limited types of configurations, technology, servers and storage, networking and bandwidth, making it difficult to create real-time test environments.

4.4 Usage

Improper usage of cloud-based test environments can increase costs.

4.5 Planning

Testing teams should rigorously plan their test environments, from utilization periods through disassembly. They should also be aware of the associated expenses, such as cost of encrypting data, before putting testing in a cloud environment, since these requirements will consume additional CPU and memory. It's important to monitor utilization of cloud resources to avoid over-usage and over-payment.

5. COMMERCIAL CLOUD TESTING TOOLS

This section introduces the Commercial Cloud Testing Tools.

5.1 SOASTA CloudTest

SOASTA CloudTest [16] is a production performance testing tool for Web applications. It can simulate thousands of virtual users visiting website simultaneously, using either private or public cloud infrastructure service.

5.2 iTKO LISA

iTKO LISA [17] aims to provide a cloud-based environment and virtual services for composite application development, verification and validation. It claims to reduce software delivery timeline by 30% or more using its innovative approach to support continuous integration for development and testing.

Central to LISA architecture is its virtualizaion technology. For unavailable or inaccessible resources, LISA provides virtualized services by simulating the target system's dynamic behavior so that they can respond as live systems. In this way, it breaks dependence constraints of system integration and supports continuous testing.

5.3. Cloud Testing

Cloud Testing [18] is initiated by a group of architects and performance experts from UK's largest Website Performance Monitoring & Load Testing Company. It aims to support cross browser and functional testing of Web applications.

Table 1: Types of cloud Testing

S.No.	Type Of Testing	Target
1	Performance Testing Load Testing	To ensure meeting the business requirements specific to cloud computing
2	Interoperability Testing Compatibility Testing	To ensure meeting the business requirements specific to cloud computing
3	Stress Testing Recovery Testing	To ensure data recovery from crashes, hardware failures in a cloud environment.
4	System Integration Testing(SIT) User Acceptance Testing(UAT)	To ensure the developed cloud solution meets the functional requirement.
5	Security Testing	To ensure meeting the Application/ Data security requirement.

6. RESEARCH ISSUES

This section highlights the some of the key research issues [21].

6.1 Define the characteristics of an application under test and the types of testing done on the application.

Parveen and Tilley [20] highlights the characteristics of an application under cloud testing are test case dependency, the operating environment within which to carry out testing and the ability of an application's interface to be programmed. The types of testing identified to be appropriate for testing in the cloud are unit testing, high volume automated testing and performance testing.

6.2 Evaluate whether certain testing infrastructure in the cloud really helps to meet a specific performance attribute.

SOASTA, a cloud testing provider, has for almost two years been providing performance testing of web applications in the cloud [16]. However, many other applications as well as the cloud itself need to be tested for various performance attributes e.g. response time, speed and throughput. So a suitable approach to address this issue is by running a test bed which would encourage researchers to explore different aspects of performance testing, reflect on the experiences encountered in the process and thereafter give recommendations based on the results.

6.3. Validate the quality of cloud tested applications at all levels.

Quality is sometimes a highly subjective attribute, varying due to different end-user expectations. So the researchers have to develop a testing methodology to verify and validate the quality of overall testing in the cloud.

6.4 Monitoring and managing the software testing processes.

With the growing number of cloud providers, customers have a wide range of choice from where to get cloud services. Assuming a customer chooses to acquire the services from more than one cloud testing provider, it would be efficient to have some methods, tools and facilities for monitoring and managing the software testing processes from the different providers in an "all-in-one" fashion.

6.5 Management of test data

The management of test data is a delicate issue. In order for effective testing to take place, some testing tasks depend highly on the actual customer or production data. In some cases, due to rules and regulations, customers are prohibited from supplying confidential or production data to third parties. A solution to this problem may be the development of new models or algorithms that would generate almost "identical" test data to facilitate productive testing results.

7. CONCLUSION

Compared with conventional testing methods, cloud testing emphasizes more on system testing and online testing. This is due to the novel design and development methods imposed by cloud computing. This is still an emerging research area with many open problems. This paper highlights the recent cloud testing architecture, tools and research issues. This will really serve the foundation for the new researchers and students those who really interested in software testing using cloud.

8. REFERENCES

[1] Nivedan Prakash, "Cloud testing: attracting demand", http://www.expresscomputeronline.com/20100201/trend01.shtml.

[2] Fujitsu, "Confidence In Cloud Grows, Paving Way For New Levels Of Business Efficiency" 2010.

[3] P.Jogalekar, M.Woodside, "Evaluating the scalability of distributed systems" IEEE Trans. Parallel and Distributed Systems vol.11, no 6, 589-603,2000.

[4] L. Yu, W. Tsai, X. Chen, L. Liu, Y. Zhao, L. Tang, and W. Zhao, "Testing as a Service over Cloud," in 2010 Fifth IEEE International Symposium on Service Oriented System Engineering, 2010, pp. 181–188.

[5] T. King, A. Ramirez, R. Cruz, and P. Clarke, "An Integrated Self-Testing Framework for Autonomic Computing Systems," Journal of Computers, vol. 2, no. 9, pp. 237–249, 2007.

[6] T. King, "A Self-Testing Approach for Autonomic Software," Ph.D.dissertation, Florida International University, 2010.

[7] T. King and A. Ganti, "Migrating Autonomic Self-Testing to the Cloud," in Third International Conference on Software Testing, Verification, and Validation Workshops, 2010, pp. 438–443.

[8] 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.

[9] 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.

[10] G. Candea, S. Bucur, and C. Zamfir, "Automated Software Testing as a Service," in Proceedings of the 1st ACM symposium on Cloud computing, 2010, pp. 155–160.

[11] 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.

[12] 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," in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010, pp. 631–636.

[13] T. Parveen, S. Tilley, N. Daley, and P. Morales, "Towards a Distributed Execution Framework for JUnit Test Cases," in IEEE International Conference on Software Maintenance, 2009., sept. 2009, pp. 425 –428.

[14] J. Moreno, D. Kossmann, T. Kraska, and S. Loesing, "A Testing Framework for Cloud Storage Systems," Master's thesis, Swiss Federal Institute of Technology Zurich, 2010.

[15] S. Baride and K. Dutta, "A Cloud Based Software Testing Paradigm for Mobile Applications," ACM SIGSOFT Software Engineering Notes, vol. 36, no. 3, pp. 1–4, 2011.

[16] Cognizant reports, "Taking Testing to the Cloud". March 2011.

[17]SOASTA. [Online]. Available: http://www.SOA STA.com/

[18] ITKO. [Online]. Available: http://www.itko.com/

[19]CloudTesting.[Online].Available:http://www.CloudTesting.com/

[20] T. Parveen, and S. Tilley, "When To Migrate Testing to the Cloud," In proc. 2nd International Workshop on Software Testing in the cloud (STITC), 3rd IEEE International Conference on Software Testing, Verification and Validation (ICST), April 2010, pp. 424-427, doi:10.1109/ICSTW.2010.77.

[21] Leah Muthoni Riungu, Ossi Taipale, Kari Smolander, "Research Issues for Software Testing in the Cloud" IEEE International Conference on Cloud Computing Technology and Science, 2010