CLOUD COMPUTING BASED LAYERED PARAMETERIZED TESTING ENVIRONMENT

L.Thenmozhi[1], G. Ilavarasi[2], J. Sujatha[3]

[123] Department of Computer Science,

Sriram College of Arts & Science, Perumalpattu,Tiruvallur.

ABSTRACT

Cloud computing is emerged as new tools in organization and cooperates. While the platform-as-a-service (PaaS) model has efficient application development and deployment, its numerous abstraction layers and dependencies have made testing more difficult.Cloud computing, which is a relatively new approach to distributed computing, uses cloud infrastructures that repeatedly scale to support an application's hardwarerequirements, and therefore, must have high consistency in order to meet user expectations.To support this vision, we suggest layered parameterized tests (LPTs)—generalized integration tests appropriate for cloud applications with multiple processing layers.Software testing has been one of the best practice areas for migrating to cloud atmosphere. Test automation topic is often visited when software testing is considered over the cloud. We present a review of software testing above the cloud environment and body of work in the field.

Keywords

Cloud Computing, Software testing,Cloud applications, Testcases,Distributed computing.

1.INTRODUCTION

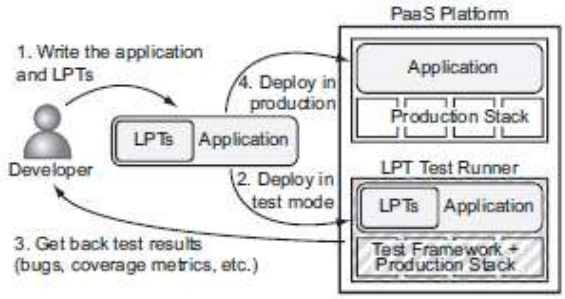
We provide a review of software testing over the Layered testing and categorize the body of work in the field. Cloud computing has become a new computing paradigm where the cloud could provide both virtualized hardware and software resources that hosted remotely and provide a use-on-demand service model.

Platform-as-a-service (PaaS) provides such as Google App Engine or Microsoft Azure, provide easy-to-use interfaces with automated deployment fully integrated into development environments. However, testing tools for apps running on PaaS are still immature, test automation is inadequate, and developers are left with the laborious and error-prone task of manually writing large numbers of individual test cases. Cloud computing has become a new

computing model where the cloud could provide both virtualized hardware and software resources that hosted remotely and provide a use-on-demand service model.

2. CLOUD LAYERED TESTING OVERVIEW

We propose layered parameterized tests (LPTs) for integration testing of PaaS-based cloud applications. LPTs describe families of integration tests (in the spirit of parameterized tests across several application layers. We rely on developer-provided onion objects to describe the layering of the data abstractions in the application; objects encode the multiple interpretations of input data as, e.g., an HTTP request, a JSON object, etc. For the automatic generation of thorough testcases from LPTs, we introduce layered symbolic execution(LSE), an automated program analysis that is tailoredfor the layered structure of cloud applications. Finally, we present a design and early prototype for aPaaS-integrated testing serviceCloud Testing uses cloud infrastructure for software testing.



Development flow for using a PaaS-integrated testing service.

3. CLOUD COMPUTING ANDTESTINGENVIRONMENT

Testing a Cloud includes availability, security, performance, interoperability, disaster recovery and multi-tenancy testing. Cloud testing are challenged by several problems such as limited test budget, meeting deadlines, High costs per test, large number of test cases, little reuse of tests and geographical distribution of users. The aim of cloud testing is to ensures high quality service delivery and avoiding data outages requires testing inside datacenter or outside the datacenter or in both places. Larger datacenters also mean larger installations with high-speed interfaces and the ability to maintain service availability obligations.

4. CLOUD FUNCTIONAL TESTING

Functional cloud testing is the testing of all the features and functions of a system which includes hardware and software. Functional cloud computing testing is performed for both remote and local applications.

4.1 System Testing

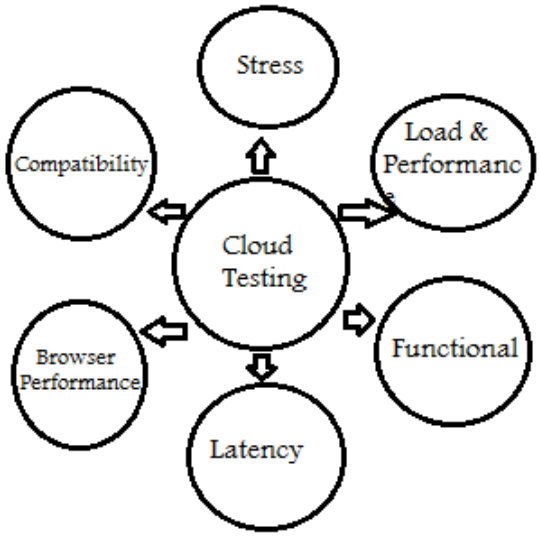
System testing techniques is used to prove the systems behavior within its own boundaries. It is critical to prove that the system functions as it has been designed when the system components work together, inputs and outputs are as expected and the overall resulting system is a high-quality cloud system

4.2 Integration Testing

Integration cloud testing allows the business to verify the cloud solution will work within the current infrastructure and environments which ultimately proving that the implementation of a cloud solution does not detrimentally impact any existing systems.

4.3 User Acceptance Testing

User Acceptance Testing will be done to prove that delivered cloud solution meets business requirements so that the user accepts the developed cloud solution. User acceptance testing is done on both on-premise and off-premise.



Types of cloud testing

5. CLOUD NON-FUNCTIONAL TESTING

Non-functional testing is done to ensure that a web application meets the specified performance requirements. Non-functional testing technique is also known a performance testing technique. In cloud, the applications scalability scope is much wider than in conventional performance testing techniques [Keerti Kulkarni].

5.1. Business Requirement Testing

Before migrating their business to a cloud computing solution, the organizations and cooperates must carefully analyze and document their business requirements clearly, precise and unambiguously. Business requirements are foundations for building a cloud computing solution. These business requirements can be achieved through reviews, periodical customer meets and workshops. Later, this in turn provides a perfect system is constructed which is capable of delivering the business requirements [Prakash & Gopalakrishanan, 2012].

5.2. Cloud Availability Testing

Cloud services must be available for all times. There should be also assurance that the there is no abrupt downtime such that the business of the client must not be adversely affected [Keerti Kulkarni; Prakash & Gopalakrishanan, 2012].

5.3. Cloud Security Testing

Security testing which is an indispensable part of testing applications due to increase in security breaches in business. This can provide assurance that business critical data is stored and transported safely. To identify methods of gaining access to a system by using common tools and techniques used by hackers can very well guarantee the security ofCloud solutions http://www.toolsjournal.com/testing-lists/item/404-10-cloud-based-testing-tools].

5.4. Cloud Scalability and Performance Testing

Cloud Scalability is another major area of concern where adequate amount of testing is needed. Cloud Computing solutions always claim to be scalable on demand. Load or Stress testing can be used to prove that the developed cloud solution can be scale as required with the help of software tools. Hence Cloud solution can be accurately measured and its capacity is verified. Cloud Performance testing techniques allow us to measure the cloud systems performance accurately. Performance testing with the load testing techniques allows getting an accurate image of the solution’s ability on the cloud [Prakash & Gopalakrishanan, 2012; http://www.toolsjournal.com/testing-lists/item/404-10-cloud-based-testing-tools]. Performance is generally tied to an application’s capabilities within the cloud infrastructure. Finding out thresholds, bottlenecks & limitations is a part of performance testing. For this, testing performance under a particular workload and vary the nature of traffic on-demand is necessary

5.5 Cloud Load and Stress Testing

Application stability is a major factor as the user count is expected to be increases. Load testing of an application involves creation of heavy user traffic and measuring its response. There is also a need to tune the performance of any application to meet certain standards. Measure response times and isolate issues related to specific actions while system is subjected to increasing load from different locations and multi user operations. It is imperative to identify issues as system is tested to breaking points maximum expected capacity or often beyond the expected usage. Stress Test is used to determine ability of application to maintain a certain level of effectiveness beyond breaking point or maximum expected capacity or beyond the expected usage [http://www.toolsjournal.com/testing-lists/item/404-10-cloud-based-testing-tools; Eljona Proko & Ilia Ninka, 2012]. It is essential for any application to work even under excessive stress and maintain stability. Stress testing assures this by creating peak loads using simulators. But the cost of creating such scenarios is enormous.

5.6. Latency Testing

Cloud testing is utilized to measure the latency between the action and the corresponding response for any application after deploying it on cloud [Prakash & Gopalakrishanan, 2012; http://www.toolsjournal.com/testing-lists/item/404-10-cloud-based-testing-tools].

5.7. Ability Testing Techniques

Ability testing is done to ensure that the cloud environment is able to gives its service on-demand to users. In this category, the compatibility, interoperability and multi-tenancy ability of cloud computing environment is tested [Spirent Communications, 2010; Eljona Proko & Ilia Ninka, 2012].

5.8. Compatibility and Interoperability Testing

In cloud environment, different software's and operating systems is used and created on demand which makes the compatibility testing must. A cloud application must capable to work and executed across multiple environments and various cloud platforms. Hence, it is very easy to migration of a cloud applications and platforms from one infrastructure to another infrastructure [Spirent Communications, 2010; http://www.toolsjournal.com/testing-lists/item/404-10-cloud-based-testing-tools].

5.9. Disaster Recovery Testing

The cloud service provider has always preferred that his cloud services must be available all the time to end-users but actually it is not achievable. There may be some chance of failure so the disaster recovery time must be low. Cloud verification must be done to ensure the service is back online with minimum adverse effect on business

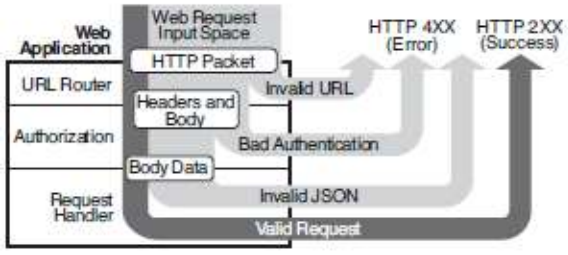
5.10. Multi-Tenancy Testing

Multi-tenancy testing ensures that the multiple clients and organizations using on-demand services activated at a given time. Cloud service should be customizable for each client and provide data and security level to avoid any access related issues

6.A PAAS TEST INTERFACE

We propose an automated testing service integrated in PaaS. Developers write layered parameterized tests (LPTs) and upload them with the cloud application to be executed by the test service. The service uses LPTs to automatically generate application inputs (e.g., web requests and persistent data) that exercise the applicationlayers of interest. The developer writes LPTs by specifying the structure of the application inputs and a target property to be checked.

Layered Parameterized Tests an LPT specifies a family of executions (or, equivalently, classes of inputs) plus a set of properties (expressed as assertions) that are expected to hold for all these executions.



A sample cloud application

7. KEYS TO SUCCESSFUL CLOUDTESTING

Cloud –based test environment gives testing teams greater control to build and execute tests, analyses application performance and look for bottlenecks and stress areas while tests are running. The cloud agrees to testers to scale from thousands to millions of users to access the breaking point and capacity thresholds to combat highlyunpredictable demand levels. This gives testers a clearer picture of possible runtime errors, which reduce production errors. Once a tester logs in and executes a test, the results are available to developers, who can judge performance

and fix anomalies, ever the cloud itself. It reduces communication gap between testers and developers regarding errors.

Other values to raise success of cloud testing are as follows:

- Understanding a platform provider's elasticity modular configuration model.
- Service Provider's or vendors evolving in monitoring services.
- Potentially engaging the service provider as an on-going operations partner if producing commercial off-the

shelf (COTS) software.

- Being willing to be used as a case study by the cloud service provider.Project Managers can monitor the

overall progress of the project and drill down into specific tasks for review. This reduces cycle times andimproves application deployment, which reduce problems and providing users with an enhanced testingexperience.

The ability and cost to simulate web traffic for software testing purposes has been an inhibitor to overall webreliability. The low cost and accessibility of the cloud's extremely large computing resources provides the ability toreplicate real world usage of these systems by geographically distributed users, executing wide varieties of userscenarios, at scales previously unattainable in traditional testing environments. Minimal start-up time along with qualityassurance can be achieved by cloud testing.

8. ISSUES WITH TRADITIONAL TESTING

Physical machines restrict the number of replicable test labs. This contributes to high capital costs when a large number of testing labs are needed. Mobile introduces functionalities, tools, and applications that testing teams are not prepared for with existing tools. Organizations are caught off guard by the growth of multiple platforms, particularly mobile.

Tests are conducted on-premise in a closed environment with limited access. Off-shore locations are required to conduct their own tests locally, even if systems are identical.New Features in Cloud Testing- Cloud based testing has several unique advantages listed below comparing with current software testing. 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. 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.

9. CONCLUSION

The future is going to be Cloud computing solutions for large as well as small businesses. And that will bring a major wave in technology infrastructure. An approach for testing cloud platforms and infrastructures is suggested. An intelligent framework has been created that can significantly accelerate testing and provide for parallel test development.

In this paper, we have presented a classification of current research studies, identified gaps in the literature and investigated the correlation of software Layered testing with different deployment models of cloud computing. We projected a new PaaS-integrated service for automatic and thorough testing of layered cloud applications. Instead of traditional test cases, developers write layered parameterized tests covering a wide range of application behavior, while the service takes care of the concrete test case generation using the novel layered symbolic execution algorithm.

Our future research will be focusing on filling these gaps for achieving 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 formore proliferated use over the cloud.

10.REFERENCES

[1] Vinit B. Mohata, Dhananjay M.Dakhane, Ravindra L.Pardhi ., Cloud Based Testing: Need of Testing in Cloud Platforms.

[2] G. Gowri, M. Amutha., Cloud Computing Applications and their testing methodology.

[3]Ismail, Hasan S¨ozer., A Survey of Software Testing in the Cloud

[4] William Jenkins, Sergiy Vilkomir, Puneet Sharma., Framework for Testing Cloud Platforms and Infrastructures

[5] Dr. Rahul Malhotra, Prince Jain.,Testing techniques and its Challenges in a cloud computing environment

[6] BECK, K. Simple Smalltalk testing: With patterns. http://www.

xprogramming.com/testfram.htm.

[7] ARTZI, S., DOLBY, J., JENSEN, S. H., MOLLER, A., AND TIP, F. A framework for automated testing of JavaScript web applications. In Intl. Conf. on Software Engineering (2011).

[8] TILLMANN, N., AND SCHULTE, W. Parameterized unit tests. In Symp. on the Foundations of Software Eng. (2005).