

# Framework for Testing Cloud Base Applications

Sehrish Asif, Mariam Rehman, Maria Anjum, Farhat Saleemi

**Abstract** – With the increase in demand of SAAS usage, there is a need of testing SAAS applications properly so that it should be available on cloud in good condition without any kind of functional and non-functional bugs. There is a lack of performance testing of applications which includes availability, scalability, response rate, and throughput. Therefore, there is a need to test performance matrix of an application so that users can use application without any discrepancy. The research work provides a framework for performance testing of cloud base applications which uses Performance Testing Life Cycle (PTLC) as a baseline for testing SAAS applications over the cloud environment. The research work suggests that there is a need of general purpose framework which follows PTLC as a baseline. The research is performed for discovering the whole process starting from documentation to the practical implementation for performance testing of cloud base applications. The research theoretically contributes by providing framework for testing cloud based applications. However, evaluation of the proposed framework is performed with the experts in this domain.

**Index Terms** – Cloud Computing, Software-as-a-Service (SAAS), Testing, Performance Testing, Framework

## I. INTRODUCTION

Cloud computing receives great attention by the whole world researchers, innovators, IT organizations and general users. With the increase in the popularity of cloud computing, cloud testing becomes an important part of this new innovation. Cloud computing refers to the providing software and hardware as a services on a centralized location where everyone can access and take benefit of it. [1]

With the advancement in cloud technology, there is a need to move testing towards cloud computing so that cloud can give quality services to the users. There are several factors which point's migration of testing towards cloud computing. The factors like requirement of whole infrastructure for testing which requires resources such as human, hardware, and software. While keeping in mind that testing is just a periodic activity of the whole process. Now-a-days, applications have become very complex. For testing of these applications, test labs should be created to provide a same deployment environment which requires a lots of capital usage [2].

According to NIST: "Cloud computing is a model for enabling 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. This cloud model promotes availability and is composed of five essential

characteristics, three service models, and four deployment models" [3].

Cloud computing consider two bases. The first one is Service Oriented Architecture (SOA) and the other one is Virtualization. In SOA, loosely and tightly coupled functions are considered for the delivery of integrated services to the end users. In addition, SOA facilitates user to use business valued applications with the required desire which they have as. In virtualization, hardware and functional abstractions facilitates portability and clump of physical possessions [4]. It has some essential key elements which have been addressed below:

- Service measurement
- On demand instant self-service
- Network accessibility for different environments
- Resource pooling
- Rapid elasticity (Increase / decrease) in the required service

Cloud computing offers three types of cloud services models which are Software-as-a-Service (SAAS), Platform-as-a-Service (PAAS), and Infrastructure-as-a-Service (IAAS) [3] [5].

SAAS consists of pre-made applications accessed by web browsers, mobile applications and thin clients with required virtualize software, virtual hardware and virtual network for delivering its services [1]. These are the executable applications usable by the users where each user connects to their respective application according to their need [6] [7] [1] [8].

PAAS platform provided as a service which provide software, hardware and network as a platform. In this platform, users can install and develop their own software application using platforms through internet [1]. Moreover, platform has operating system, data bases / webservers, and developing environment which are used by the developers to develop their applications without considering any upgrading, and maintenance cost [8] [7].

IAAS provides whole infrastructure as a service which includes hardware and network but user can install or develop their own software applications and operating systems [1]. IAAS focuses on physical environment, virtual machine, network infrastructure load balancer and firewall which depends on the model of "pay as you go" where user just need to install the operating software or patches [8] [7][4].

CC is defined as the cloud shared by many organizations and is handled by organizations or the cloud service providers [1].

HC is defined as the combination of different deployment models including PC, prc and CC [1]. In HC, some services handled and managed by the internal organization and some by the external providers [10]

Software testing is a very important integral part of any software development life cycle (SDLC) which spans over all

development phases. It includes the verification and validation of software. In addition, conventional software testing can be done using two ways: i.e., manual testing and automated testing. Software testing is not a single activity. It includes a whole individual process named as software testing life cycle (STLC). STLC takes weeks to months or longer time depending on the complexity of the project [11].

Cloud services procure on line services on deployment models such as public cloud (PC), private cloud (PrC), community cloud (CC) and hybrid cloud (HC).

PC provides general public free or pay as you go model services to the users over the internet. These services include applications, photo saving, online storage, and email services [1]. In addition, medium and small organizations prefer to make use of PC because PrC (Private Cloud) is not affordable for such organizations [9]. There are many benefits of using such services like scalability, easy and inexpensive setup, no resources wastage [10].

PrC is also known as internal cloud or corporate cloud. PrC offered the cloud services to some organization behind a firewall. These hosted services are handled by the internal management of the organization or the cloud owner [1]. Large organizations prefer PrC for settling its own internal environment / infrastructure for providing data with more security which facilitate customization for different products with reusability of hardware and software [9].

Testing in Cloud Computing Paradigm can be classified into two broad categories i.e., testing a cloud and cloud base testing. These are two independent concepts in the testing and have supplementary benefits to IT industry as well as to the users as more people want to use cloud infrastructure or applications testing for providing users with appropriate quality services.

Testing the cloud includes the V & V (Validation and Verification) of the on demand services provided to the users which includes availability, security and interoperability of resources, infrastructure, applications, and environment whereas cloud testing or testing as a service (TAAS) is the on demand service providers which provide testing tools, products and services to the users to test. The providers of such services provide functional and non-functional testing with on demand and on premises. These services adopted by many providers commercially with reduction in cost of hiring resources, test execution and arranging hardware [12].

Cloud testing is aligned with the concept of cloud and SAAS which provides the ability to test by leveraging the cloud, thereby bringing the same benefits that the cloud brings to customers" [13]. Cloud testing methodology includes methods, tools and services to fulfill the required outcome. For this purpose, there are many conventional software testing techniques moved to the cloud services and additional techniques have been developed specially for cloud testing services. [8]

Cloud can be treated as a test item because of containing abandoned of applications for users. It is not sure that all applications running over the cloud are free from any kind of bugs or errors. In addition, testing cloud is not different from testing traditional applications which have same software testing life cycle phases. The difference occurs when we

move down in granularity of cloud where specific parameters comes. Therefore, attention is required for performance and security testing attributes because they are cloud specific attributes where services or product are shared by multi-tenant across the globe. Cloud testing is very complex because of the entry of new tenants and services but cloud as a testing item can cover all aspects of performance testing [8].

Cloud can be treated as a test tool or test-ware because software or product requires a tool for testing which have capabilities of testing, test reporting, and generating testing tools. Cloud as a test tool generally use for security testing and performance testing upgrading [8].

Cloud can be used as a test environment for providing testing. The environment includes hardware, software tools, simulators, and instrumentation for support testing. Cloud as a test environment can be trust worthy but is not able to generate the same outcome as the original environment generates [8].

Cloud can be treated as a test logistics which is more likely to be a testing as a service (TAAS) with the activities like regression automates testing, appropriate scalability, elasticity and modules delivery. Clients and providers can take benefit from test logistics in terms of cheap services, less costs and less sales expense [8].

Above all layers of cloud computing, SAAS is the most talked layer. SAAS is defined as the software services using internet regardless of the location and time. It is a use of licensed purchased software provided by 3<sup>rd</sup> party on the cloud. Hardware location is owned or purchased by the customer but the software licensing may be perpetual or subscription basis. This service can lead the customers with the less burden of maintenance, backup and disaster recovery [6].

Occasionally, SAAS is defined as on demand service in which software and data is put centrally on cloud and can be accessed by the users. It provides services and products according to the customer requirements because all customers have different needs regarding the usage of services and the duty of SAAS providers to provide services to the users which are errorless. SAAS demand increases from 2011-2015 with 79.2 billion USD and SAAS vendors increases which are IBM, Oracle, Microsoft, SAP [14].

SAAS provides two modes of applications to the users which are free applications like Google docs, Gmail, Drop box, Calendar, Picasa and paid applications like sales force customer relationship management (CRM), content management (CM), and human resource management (HRM), enterprise resource management (ERP), Office 365 etc. It has SAAS major players which are Google and Salesforce.com

SAAS applications have some key features like pay as you go- Flexibility, on demand service, reusability, quality of services, scalability, data security, data managed by provider, multi-tenant, availability, configurability, service customizability [15], high system accessibility, and ease in configuration [14].

There are many SAAS based applications which exist on cloud e.g.; online Collaboration (Web conferencing), HR applications (ERP, Salesforce automation, e-recruiting),

CRM, Healthcare [16]. Some of these applications have been discussed below:

**SAAS Content Management System (SAAS CMS)** is the content management system in the cloud environment. It is usually known as cloud content management system. The major examples of SAAS CMS are Google Docs, Acrobat.com and Box.net. SAAS CMS provides a powerful tool for the efficient management of any type of content over the cloud under the SAAS model.

**SAAS Customer Relationship Management System (CRMS)** is the most talkative technology of SAAS model which delivers services to the enterprise business level. It is just not a technology rather it is a strategy for retaining, maintaining, developing, and to dig out maximum customer satisfaction. With the increase in technology, businesses need to run smoothly with customer satisfaction. For this purpose, software tools can be very helpful which provide emergence of technology for customer and customer relationship management. CRMS is an application which focuses on customer satisfaction, decision making, reporting dashboard (GUI, round the clock access to work, data driven results, sales enhancements, better customer service experiences with great satisfaction) [6].

Cloud computing overall growth graph increases the adoption of SAAS testing. With the rapid change in technology and day to day upgrades of application, the availability of SAAS application to the market within very short time evolves cloud computing and increases the SAAS application demand on cloud. The demand leads to the concern towards SAAS application testing as well. SAAS Application testing is a big challenge itself for vendors because of the requirement of repeated validations of applications with hard timelines to deliver products. Furthermore, testing includes the category of application for testing with the usage of cloud resources.

SAAS based applications are becoming popular and provide a source of great benefits to the enterprises. SAAS application testing is far different from the local on premise application testing. SAAS application testing involves performance testing, security testing, data integrity testing, business logic testing, and many others. SAAS application testing has some challenges to take off with which are as follows:

**Security and privacy testing:** In the multi-tenant environment, SAAS applications security and privacy is the major encounter. It guarantees testing for secure SAAS applications so that user's data is not shared among tenants. Some security testing includes testing cookies and SQL injection.

**Short notice period and frequent release:** SAAS applications require frequent releases, and upgrades within short notice period. Assuring the security for the frequent releases should be noticeably by the customer's. For this purpose, the provider provides 1-2 weeks for quality assurance team to dig out the bugs. SAAS applications run smoothly and successfully with having the hard stress of frequent releases and upgrades which need to be guaranteed.

**Performance testing:** SAAS application should perform well in every case like the abandoned number of users, heavy

data in and out from the application. For this purpose, simulating real life scenarios for performance testing is a big dare.

**Integration and migration:** It may create intense circumstances for the users who want to integrate applications with other applications. This may require the exchange of data which will have privacy of data, validation and verification of data and security concerns of data from both side applications. In contrast, migration of data from one SAAS application to the other requires lot of time.

**Business knowledge:** It is very important to understand the business knowledge of SAAS applications so that companies would be able to understand the environment, configurations, impact of changes and the functionality workflow.

**Licensing:** SAAS application licensing should be made according to the different parameters like no of simultaneous users, type of data, functionality work flow, and integration with other applications and migration of data from one application to other [17].

Cloud base application testing involves various techniques that include testing SAAS in a cloud, testing of a cloud, testing inside a cloud, test case independence, known operational environment, and programmatic interface.

In testing a SAAS in a cloud, SAAS applications tests for functional and non-functional requirements in a cloud environment. Testing involves unit testing, regression testing, performance testing, scalability testing, API service connectivity testing and elasticity testing.

In testing of a cloud, cloud tests for external view and validating the quality with respect to the end users, and vendors.

In testing inside a cloud, mainly vendors perform cloud internal view testing for validating the services they provide. Internal view consists of infrastructure and cloud capabilities.

**Applications Characteristics for Testing Applicability on Cloud**

Test case independence should be a part of the application because concurrent execution required for testing and migrating test cases to cloud.

Operational environment should be well suited to the test cases. Applications have different programs, libraries and components which have dependency on environment. These dependencies should be clear and analyzed before application testing.

Application under test should have programmatic graphical user interface (GUI) for the efficient test automation because lack in GUI leads to the degradation in migrating test case, and test automation to the cloud [18].

There are the components which lie in the category of SAAS testing that include application testing, network testing, and infrastructure testing.

SAAS application testing is very essential because they have critical effects on the business organizations. Outputs and the decisions mainly depend on the business software. SAAS application testing covers different aspects of the application like testing component function, testing end to end business workflows, testing browser compatibility,

testing for security of data, testing for the integrity of data, and testing the access privileges.

Network testing is performed for data sensitivity and security. Testing network bandwidth is another perspective i.e. each tenant can send and receive data from application to client network.

Infrastructure testing requires testing of live upgrades, disaster recovery and reliable assurance for SAAS application [14].

## II. SAAS APPLICATION PERFORMANCE AND CALABILITY EVALUATION

In this research work, focus will be on application component of testing with the perspective of performance testing of SAAS applications. Performance is the most important quality attribute of SAAS applications with efficiency and user acceptance. Performance is evaluated in terms of application response time, user acceptance of application, concurrent access of users for the same resource, and resource utilization rate from user point of view. SAAS performance testing makes sure advantages like scalability of the applications.

Definition: "Performance testing is a type of software testing that intends to determine how a system performs in terms of responsiveness and stability under a certain load." [19]

### REASONS FOR PERFORMANCE TESTING:

- How SAAS application performs when demand and throughput increases which leads to the bottlenecks identification and fixing [20]?
- How SAAS meets the need of tenant for shared resources?
- How the entries of a new tenant effect the activity of other tenants in the pool [21]?

### SUB-TYPES OF PERFORMANCE TESTING

- Response Performance Testing: Verifies response time for certain amount of load.
- Batch Performance Testing: Verifies the processing of batch by monitoring resources provided by systems.
- Stability Performance Testing: Verifies stability of a system considering leakage of memory and recovery of system.
- Scalability Performance Testing: Verifies the scalability of the system with the increase in load.
- Failover Performance Testing: Verifies the Failover of the system or component by emphasizing planned fail over.
- Stress Performance Testing: Verifies the stress of system by inserting heavy load which evaluates the boundaries for system or the product [8].

### SAAS APPLICATION EVALUATION LEVELS

There are following SAAS application testing levels:

- SAAS level: SAAS application instances validated and evaluated for private and public cloud.
- On Cloud Level: SAAS application multiple instances validated and evaluated for private and public cloud.

- Over Cloud Level: End to end SAAS application or SAAS application instances validated and evaluated for hybrid cloud.

### SAAS TESTING METHODOLOGIES / FRAMEWORKS

SAAS testing methodologies refers to the methods used which are used to ensure the validation and verification of the SAAS application. SAAS testing methods use agile methods to speed up application testing and to make sure the demands of the current market for achieving goals like timely delivery of software while keeping in mind the quality metrics. SAAS uses agile method and application functions in the cloud computing environment which make stress on automated testing. SAAS testing is different from traditional testing and it eliminates many perspective of testing for SAAS scope [22].

### III. FRAMEWORK FOR PERFORMANCE TESTING OF CLOUD BASE APPLICATIONS

In this section, architecture is proposed for SAAS cloud base applications performance testing. The proposed framework named as "SAAS Performance Testing framework "SAAS PTF". SAAS PTF is a general framework which is designed to test all types of SAAS cloud base applications and follow PTLC as a baseline so that all steps should be performed. Before moving to the framework proposition, general performance testing life cycle is explained in which every kind of application should pass through every kind of testing. So far, no framework in the current studied literature worked according to the Performance Testing Life Cycle (PTLC). PTLC is an important part to be followed by tester for performance testing of any application.

All studied frameworks are scenario based and for some special purpose only. The proposed framework will be general purpose framework which will be used for performance testing of applications which follows PTLC in the base line. SAAS PTF has multiple selections according to the different metrics of application like architecture and implementation, load description, and number of virtual users required for the application. It consists of following steps:

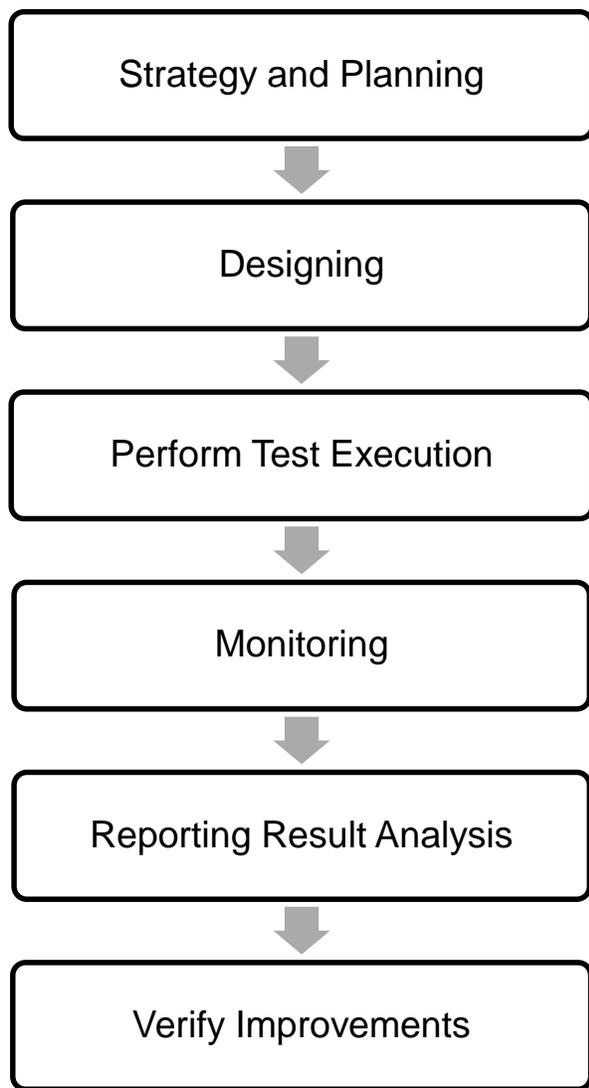


Fig. 1. Main Components for Performance Testing Life Cycle (PTLC) [20], [38] and [39] [40].

Now, all steps will be elaborated with respect to the above mentioned main component with in depth detail.

#### IV. FRAMEWORK OVERVIEW

**Upload Application Over Cloud:** Vendor / user upload application over the cloud for which they want to perform testing. After the successful uploading of the application, strategy and planning is performed for the concerned application with respect to every aspect.

**Strategy and Planing:** In strategy and planning, multiple steps are performed for the performance testing of application under testing. All pre-requisites are well defined and documented which is required for the applications testing. So that no problem occur with respect to the strategy and planning of application.

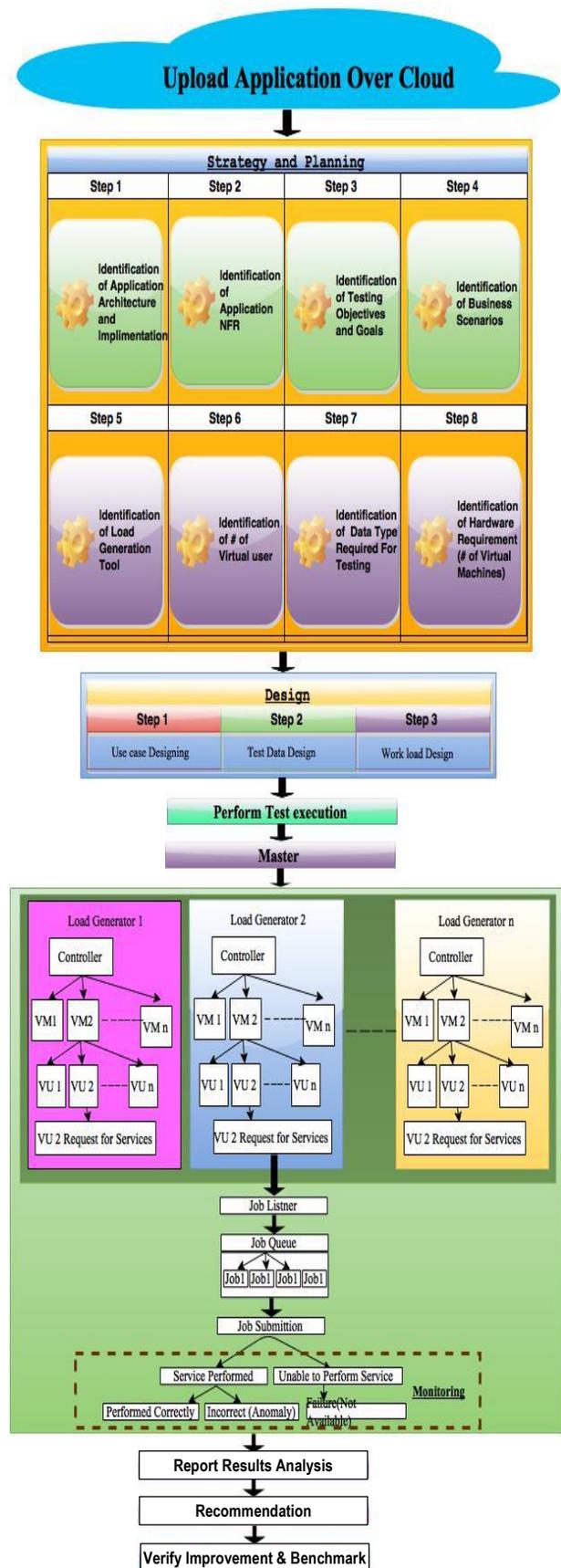


Fig. 2. Detailed framework

Module	Description
<b>Identification of application architecture and implementation</b>	In this step, application architecture and implementation details are identified so that the cloud creates physical and logical environment for testing replica like the application have physically and logically. E.g.; application architecture: centralized, de-centralized, MVC (Model View Controller) Implementation details: Language, IDE use for development
<b>Identification of application non-functional requirements</b>	In this step, application non functional requirements are identified so that performance testing performs on the non-functional requirements. E.g.; response time, throughput etc
<b>Identification of application testing objectives and goals</b>	In this step, application testing objective and goals are identified which are supposed to be achieved in testing results. Goals are the before release certain criteria achieved under certain circumstances. Objectives are the user satisfaction direct metric vendor should achieved before release.
<b>Identification of business scenario</b>	In this step, business scenarios are identified for the application under test. So that performance matrix is identified according to the business flow.
<b>Identification of load generation tools</b>	In this step, load generation tool identify according to the application business scenario.
<b>Identification of # of virtual user</b>	In this step, total number of min and max virtual user are identified for application performance testing matrix like stress testing and load testing.
<b>Identification of data type required for testing</b>	In this step, data type identified which required for giving input to the application for checking response time and throughput.
<b>Identification of # of virtual machine</b>	In this step, total number of min and max virtual machines (hardware and software) identified for application performance testing matrix. E.g.; Hardware: CPU / Memory / Hard disk / Printer / Network card Software: Operation System: Windows, Unix Web Server: Tomcat, IIS

Database: SQL Server, mysql, Oracle Compiling Tool: gcc, NET, jdk Testing Tool: Test management, Static Analysis
--

### Designing

Module	Description																
<b>Use case designing</b>	In this step, use cases are designed for each scenario for acquiring the non-functional matrix report. E.g.; Use case template <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 50%;">Tester</td> <td style="width: 50%;"></td> </tr> <tr> <td>Purpose:</td> <td></td> </tr> <tr> <td>Pre-requisite:</td> <td></td> </tr> <tr> <td>Test Data:</td> <td></td> </tr> <tr> <td>Steps:</td> <td></td> </tr> <tr> <td>Expected Result:</td> <td></td> </tr> <tr> <td>Actual Result:</td> <td></td> </tr> <tr> <td>Status:</td> <td></td> </tr> </tbody> </table>	Tester		Purpose:		Pre-requisite:		Test Data:		Steps:		Expected Result:		Actual Result:		Status:	
Tester																	
Purpose:																	
Pre-requisite:																	
Test Data:																	
Steps:																	
Expected Result:																	
Actual Result:																	
Status:																	
<b>Test data design</b>	In this step, data design for testing which will required in the use case for input in the system under test.																
<b>Test script design</b>	In this step, scripting languages are define and designed for automated testing of a system. E.g.; javascript / Perl / Python / Ruby/ Tcl / Unix Shell Script / vbscript [41]																

### Performance Test Execution:

Module	Description
<b>Master</b>	Master use for the creation, management and termination of (load generator)LG's. It takes test script performance matrices as an input.
<b>Controller</b>	Controller controls the accurate delivery of total number of virtual machines required for processing and the total number of virtual user required for the performance testing of the job.
<b>Virtual Machines(VM)</b>	VM is the mimic / abstract environment on cloud which is approximately same to the original / physical environment required for application.
<b>Virtual User (VU)</b>	VU are the maximum number of users mimic the original requirement of the user for application usage. These users automatically generate requests. They

	all have identical hardware and software configurations
<b>Job Listner (JL)</b>	JL listens all jobs coming from the LG's and user want that job to be fulfilled.
<b>Job Queue (JQ)</b>	All job's queued in JQ and fulfilled on first come first serve (FCFS) basis algorithm or priority basis.
<b>Job Submission (JS)</b>	JS submit the job to be fulfilled and came across two type of scenarios. In first scenario, job fulfilled correctly or may be fulfilled correctly but with the anomaly. In second scenario, job not fulfilled or might not be available for performing(Failure).

Monitoring: It monitors the whole performance matrices and make a log of it.

Reporting Result Analysis: Result analysis would be made on the matrices occurs after whole process performs performance testing.

Appli cation Name	# o f V U	# of M s	# of req uest s	Serv ice Perf orm	Serv ice Not Perf orm	Resp onse Time	Availa bility

### Output Report

Recommendations and Verify Improvements:  
Recommendations made in application according to the results and suggestions make for the further improvements.

## V. FRAMEWORK EVALUATION

This area covers the framework evaluation part which is taken with the help of a questionnaire. For framework evaluation, expert review method is used to take expert review on the proposed work. For this purpose, a questionnaire is prepared to take comments from the experts of related field. The questionnaire consists of some questions related to the framework and experts will respond on it according to their thoughts. The data collected from the questionnaire is 100% so far. According to expert, there is a need of such kind of framework in order to test cloud base applications. The URL for the use questionnaire is as follows: <http://goo.gl/forms/1n9veyqjng>

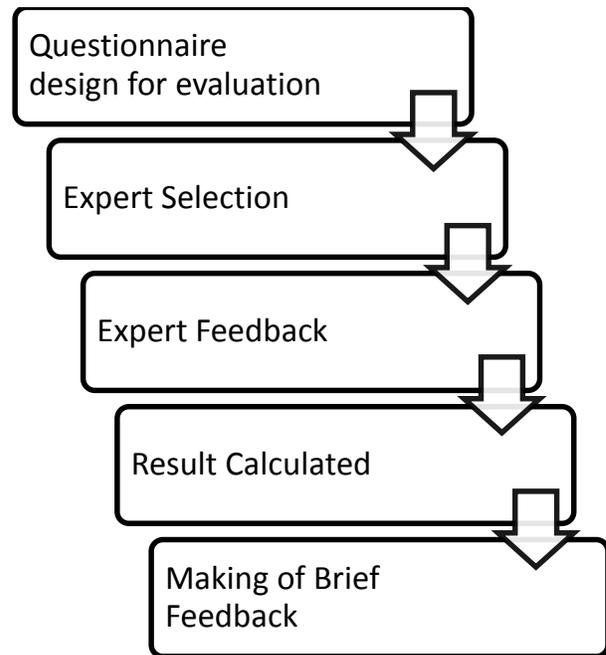


Fig. 3. Framework Evaluation Flow Chart

## VI. CONCLUSION

In this research work, conceptual framework is proposed for performance testing of cloud base SAAS applications. The framework is a structure for testing small as well as large complex SAAS applications.

This is a general framework which can be used for any kind of SAAS application uploaded over the cloud. By using the cloud environment of hardware and software, SAAS applications testing being performed and evaluated.

This framework made concentration of three performance matrices which are response time, availability and throughput by using different types of performance testing which are load testing, stress testing, scalability testing.

By using this framework, testing process is well planned, well documented and provide benchmark for the future testing purpose.

## REFERENCES

- [1] T. Dillon, C. Wu and E. Chang, "Cloud Computing: Issues and Challenges," in *24th IEEE International Conference on Advanced Information Networking and Applications*, 2010.
- [2] Priyanka, I. Chana and A. Rana, "Empirical Evaluation of Cloud-based Testing Techniques: A Systemetic Review," *ACM SIGSOFT Software Engineering*, India, 2012.
- [3] Grance, Tim and P. Mell, "The NIST Definition of Cloud Computing," 2009.
- [4] K. Incki, I. Ari and H. Sozer, "A Survey of Software Testing in the Cloud," in *IEEE Sixth International Conference on Software Security and Reliability Companion*, 2012.
- [5] L. M. Riungu, O. Taipale and K. Smoland, "Research Issues for Software Testing in the Cloud," in *2nd IEEE*

- International Conference on Cloud Computing Technology and Science*, 2010.
- [6] S. Kostojohn, M. Johnson and B. Paulen, CRM Fundamentals, Apress, 2011.
  - [7] J. vemulapati, N. Mehrotra and N. Dangwal, "Saas security testing: Guidelines and Evaluation Framework," in *11th Annual International Software Testing Conference*, 2011.
  - [8] V. Czenter, "about-sqs/whitepaper-book-2013.php," August 2013. [Online]. Available: <http://www.sqs.com/en-group/about-sqs/whitepaper-book-2013.php>. [Accessed 20 January 2015].
  - [9] K. Vinaya and S. Kumar, "Taking Testing to the Cloud," Cognizant Reports, 2013.
  - [10] Vineetha, *Performance Monitoring in Cloud, Infosys View-Point*, Bangalore, 2012.
  - [11] V. Katherine and D. K. Alagarsamy, "Conventional Software Testing Vs. Cloud Testing," *International Journal Of Scientific & Engineering Research*, 2012.
  - [12] Neha, "Cloud Testing Vs. Testing a Cloud," in *10th Annual International Software Testing Conference*, 2011.
  - [13] R. Batra and N. Sharma, "Cloud Testing: A Review Article," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 6, p. 314 – 319, June 2014.
  - [14] V. Naganathan and S. Bellave, "The challenges associated with SAAS testing," Infosys, India, 2011.
  - [15] S. Banerjee and S. Jain, "A survey on SaaS using quality model in cloud computing," *International Journal of Engineering And Computer Science*, vol. 3, no. 1, pp. 3598-3602, January 2014.
  - [16] R. Garg, "SAAS Testing Overview," 28 August 2010. [Online]. Available: <http://www.slideshare.net/ramgarg/saas-testingoverviewppt-5072671>. [Accessed 20 January 2015].
  - [17] J. Rosenblum, "6 challenges of testing saas applications," 11 Sep 2012. [Online]. Available: <http://cloudtweaks.com>. [Accessed 14 Feb 2015].
  - [18] T. Narula and E. G. Sharma, "Framework for Analyzing and Testing Cloud based Applications," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 6, pp. 592-596, 2014.
  - [19] STF, "performance-testing," 15 July 2013. [Online]. Available: <http://softwaretestingfundamentals.com/>. [Accessed 20 12 2014].
  - [20] "PerformanceTesting," September 2011. [Online]. Available: <http://saastesting.wikispaces.asu.edu>. [Accessed 20 April 2015].
  - [21] B. Sengupta and A. Roychoudhury, "Engineering Multi-Tenant Software-as-a-Service Systems," in *PESOS '11 ACM*, 2011.
  - [22] Inc, "saas-testing-methodology-development-process," 2015. [Online]. Available: <http://appenda.com/library/software-on-demand/saas-testing-methodology-development-process/>. [Accessed 14 2 2015].
  - [23] S. Gaisbauer, N. Edwards, J. Kirschnick and J. Ro, "Vats: Virtualized-Aware Automated Test Service," in *Fifth International Conference on Quantitative Evaluation of Systems*, 2008.
  - [24] R. Mathew and R. Spraezt, "Test Automation on a SaaS Platform," in *International Conference on Software Testing Verification and Validation*, 2009.
  - [25] L. Yu, W.-T. Tsai, X. Chen, L. Liu, Y. Zhao, L. Tang and W. Zhao, "Testing as a Service over Cloud," *Fifth IEEE International Symposium on Service Oriented System Engineering*, 2010.
  - [26] S. Bucur, V. Ureche, C. Zamr and G. Candea, "Parallel symbolic execution for automated real-world software," in *In Proceedings of the sixth conference on Computer Systems*, 2011.
  - [27] T. M. King, A. S. Ganti and D. Froslic, "Enabling Automated Integration Testing of Cloud Application Services in Virtualized Environments," 2011.
  - [28] M. A. S. Netto, S. Menon and H. V. Vieira, "Evaluating load Generation Environments for Software Performance Testing," in *IEEE IPDPSW*, Washington DC, USA, 2011.
  - [29] J. Gao, Tsai, P. Pattabhiraman and Xiaoyi, "SaaS Performance and Scalability Evaluation in Clouds," in *Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering*, 2011.
  - [30] Niclas Snellman, "Thesis Niclas Snellman Two side," 2011. [Online]. Available: <http://www.cloudsoftwareprogram.org>. [Accessed 20 April 2015].
  - [31] H. S. Gunawi, T. Do, P. Joshi and Peter, "FATE and DESTINI: A Framework for Cloud Recovery Testing," in *NSDI*, 2011.
  - [32] H. Le, "Testing As A Service for Component-based Developments," in *The Third International Conference on Advances in System Testing and Validation Lifecycle*, 2011.
  - [33] L. Bautista, A. Abran and A. April, "Design of a Performance Measurement Framework for Cloud Computing," *Journal of Software Engineering and Applications(JOSEA)*, pp. 69-75, 2012.
  - [34] T. C. C. T. Michael Lynch, "Testing a Cloud Application: IBM Smart Cloud iNotes," in *TTC'13*, Switzerland, 2013.
  - [35] F. D. Mukoko, Abhaya, K. Kumar and A. Jain, "Cloud Based Performance Testing," *Journal of Software Engineering and Simulation*, vol. 1, no. 2, pp. 01-08, 2013.
  - [36] X. Bai, M. Li, X. Huang, . W.-T. Tsai and J. Gao, "Vee@Cloud: The Virtual Test Lab on the Cloud," in *AST*, San Francisco, CA, USA, 2013.
  - [37] N. Yigitbasi, A. Iosup and D. Epema, "C-Meter: A Framework for Performance Analysis" Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid. IEEE Computer Society, 2009.
  - [38] Keytorc, "Keytorc: Keytorc Performance Testing," 10 January 2014. [Online]. Available: <http://www.slideshare.net>. [Accessed 8 February 2015].
  - [39] Molyneaux, *The Art of Application Performance Testing From Strategy to Tools*, vol. 2, O'REILLY, 2015.
  - [40] T. Warren, "Primer on application performance testing," 17 November 2014. [Online]. Available: <http://www.slideshare.net>. [Accessed 7 Feb 2015].
  - [41] STF, "test-script," 20 Feb 2012. [Online]. Available: <http://softwaretestingfundamentals.com>. [Accessed 20 12 2014].
  - [42] Wikipedia, "Cloudtesting," April 2010. [Online]. Available: [http://en.wikipedia.org/wiki/Cloud\\_testing](http://en.wikipedia.org/wiki/Cloud_testing). [Accessed Thursday January 2015].
  - [43] "comprehensive guide to help you perform saas app testing effectively," testing-whiz, 2015. [Online]. Available: <http://www.testing-whiz.com>. [Accessed 3 January 2015].
  - [44] "comprehensi guide to help you perform saas app testing effectively," [Online]. Available: <http://www.testing-whiz.com>. [Accessed 1 January 2015].
  - [45] V. Beniwal and B. Narula, "Cloud Testing- Types, Service Platforms and Advantages," *International Journal of Computer Applications*, vol. 72, no. 20, pp. 1-6, June 2013.
  - [46] R. Y. C. a. V. S. Buyya, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2008.

- [47] P. Mell and T. Grance, "Draft nist working definition of cloud computing," 2009.
- [48] L. Ciortea, C. Zamfir, S. Bucur and Vita, "Cloud9: A Software Testing Service," *SIGOPS Operating System Review*, vol. 43, pp. 5-10, January 2010.
- [49] V. Priyadharshin and D. A. Malathi, "Survey on software testing techniques in cloud computing," *International Journal of Computer Trends and Technology*, 2011.
- [50] N. Snellman, A. Ashraf and I. Porres, "Towards automatic performance and scalability testing of rich internet applications in the cloud," in *37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, 2011.
- [51] N. Kumar, "Performance testing Qa Insights," 30 October 2012. [Online]. Available: <http://www.slideshare.com>. [Accessed 12 Dec 2014].
- [52] Gao, X. Ba and W.-T. Tsai, "Testing as a Service (TaaS) on Clouds," in *IEEE Seventh International Symposium on Service-Oriented System Engineering*, 2013.
- [53] D. R. Malhotra and P. Jain, "Testing Techniques and its Challenges in a Cloud Computing Environment," *The Standard International Journals (The SIJ)*, vol. 1, July-August 2013.
- [54] M. Wang and D. Krishnamurthy, "Performance Testing Web Applications on the Cloud," in *IEEE International Conference on Software Testing, Verification, and Validation Workshops*, 2014.
- [55] Gao, X. Bai and W.-T. Tsai, " Cloud Testing- Issues, Challenges, Needs and Practices," *Software Engineering : An International Journal (SEIJ)*, 2011.