ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 2, Issue 2, Ver. 2 (April - June 2014)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)

Testing as a Service (TaaS) on Cloud: Needs and Challenges

[I]Shivam Jain, [II]Dr. Devesh Kumar Srivastava
[I,II]Dept. of Computer Science and Engineering, Manipal University, Jaipur, India

Abstract
Cloud computing leads an opportunity in offering testing as a service (TaaS) for SaaS and cloud-based applications. This brings new business opportunities, challenges, and demands in innovative service models, testing techniques, QoS standards, and requirements. Testing as a Service (TaaS) provides testing capability to cloud user. In this paper a rigorous survey introduced in-depth study about Testing as a service (TaaS) over clouds which provide opportunities, challenges and possible research directions. Cloud testing tool and the key technologies for cloud testing are carefully discussed which lead to some useful conclusions.

Keywords
Cloud computing, cloud testing, software testing, Cloud Testing Tool, Testing as a Service (TaaS).

I. Introduction

A. Cloud Computing

Cloud Computing is shared resource as hardware, software, and network. Cloud computing is research oriented topic in IT industry as a model that provides computing resource on demand (when it require) with reduced cost. There are main services Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and Storage as a Service. Cloud computing has emerged as a new computing paradigm that facilitates the development and utilization of highly flexible, elastic services on-demand, and over broadband network access. Cloud Computing supports an everything as a service (XaaS) delivery Model. Cloud computing is a subscription-based service where you can obtain networked storage space and computer resources [1]. A good example of cloud computing is email. Most of all people using cloud computing technology. If you have access Gmail, Yahoo, Hotmail that is cloud computing. Today major players of cloud computing Skype, Basecamp. Let us take the analogy of automobiles to understand the use case of cloud computing. Take the case of car versus taxi cab. Both are automobile with basic functionality of transferring people one from to another. Cloud computing has been enabled by the developments in virtualization, distributed computing, utility computing, web and software services technologies. It is especially based on two key concepts. The first one is service-oriented architecture (SOA), which is the delivery of an integrated and orchestrated suite of function to an end-user. SOA enables end-users to easily search, use and release services on-demand and at a desired quality level. Software testing has been one of the best practice areas for migrating to cloud environment. Virtualization, which is an enabling technology of cloud computing, was first used for quickly creating virtual computing resources with different operating systems (OS) to test software applications on various platforms [1]. Testing new software often requires costly server, storage and network devices only for a limited time [2]. These computing resources are either not used or underutilized after testing, thus incurring extra cost on budget. For example, to test the performance and scalability of a banking application, the system must be stressed with requests from millions of users in a short time interval. This is a realistic scenario that should be tested because people rush to their bank accounts regularly on every payday. Reproducing such a scenario would require the provider to set up a test harness (including the user databases) to emulate the actions of millions of users. Similarly, mobile application providers frequently have to deal with maintaining the quality of their services over a plethora of various combinations of platforms [3]. The computing platforms may encompass various browser technologies with different backend support running on various mobile OS.

B. Cloud Testing

Cloud Testing is a method for testing cloud-based provisions that utilization assets found in the cloud. By resources, we mean any component (infrastructure, hardware and software) important to do the tests. Cloud testing gives an end-to-end result that changes the way testing is carried out and can help an association support its intensity by reducing the cost of testing without contrarily impacting mission critical production applications [9]. This helps ensure unused servers are not sitting idle. Cloud Testing, is a new business and service model where testing activities are performed using cloud infrastructure by leveraging cloud technologies. Advantages of cloud testing include reduced costs from a shared resources and large-scale test environments. While the focus of this discussion is on "testing as a service in the cloud", there are other forms of cloud-based testing such as testing of a cloud, testing inside a cloud and testing over clouds. Similarly, there are different types of cloud test environments. Among the many differences from conventional "dedicated" testing, in cloud testing there are requirements around SLA's and service capabilities, pricing models, and data & traffic simulations[9]. Essentially, TaaS is an outsourced model of conventional testing.

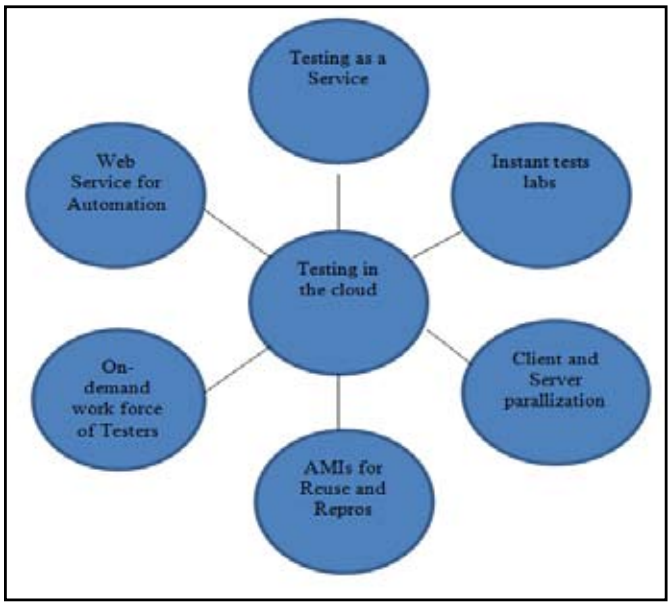

Fig.1 : Cloud Testing

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

In above figure-1 we define few service offered by cloud test.

C. Testing as a Service (TaaS)

There are a few unique characteristics in cloud testing. One of them is testing as a service (TaaS). This is an inventive idea, and it alludes to giving static/alert on-interest testing services in/on/over clouds for the third-parties at any time and all time (365/7/24). One of the essential goals is to lessen the IT budget of organizations to center their core businesses by outsource As indicated by Wikipedia, Taas includes the on-interest test execution of well-defined suites of test material, for the most part on an outsourced premise. The execution could be performed either on customer site or remotely from the outsourced suppliers test lab.

The remainder of the paper is organized as follows; in Section II, presents Research Methodology; Section III presents Challenges and needs; in Section IV; testing tool; in Section V we present conclude the paper and future work.

II. Research Methodology

The main purpose of this paper is to classify research activities performed in cloud-based testing area, clarify the terminology used, identify cloud testing tool, challenges and needs. There are currently two different perspectives on "cloud testing" and both cases can be considered as valid forms of "Testing as a Service":

1) Testing the cloud-resident applications,
2) Providing testing software as services in the cloud, and
3) Both of the above, i.e., testing cloud-resident applications

by means of cloud-resident testing services. The former deals with how applications perform in terms of functional correctness and speed when they are migrated to cloud. The latter deals with migration of the testing process itself into the cloud. This motivation enabled us to distinguish the problem domains of the literature. After thorough review of the selected papers we identified 11 major problem domains depending on the problem/solution domain of the paper. The problem domains that we have identified enable to make a distinction between whether the test service is provisioned for cloud-resident applications or for other platforms (e.g., desktop applications, mobile applications, etc.). During our search for the related literature, we found the following keywords and phrases to be useful:

- cloud application validation
- cloud application verification
- cloud computing testing
- Software testing cloud
- testing cloud applications
- Verification cloud

Cloud computing partially relates to and even depends on prior technologies such as virtualization, web services, utility computing, multi-core and parallel programming and several others. One can go back and analyze how testing processes were affected by these enabling technologies over a long period of time.

III. Challenges, Needs and Cloud Testing Tool

To effective perform cloud testing, it is extremely vital to understand the essential issues, potential challenges and needs. This section first talks about the essential issues and challenges, and after that outlines the key needs in cloud testing. large internet-based solution vendors have recognized the necessity and cost-reduction opportunities in cloud testing for cloud based applications in the near future and large-scale internet-based applications [4].

A. New Requirements and Features in Cloud Testing

There are four new requirements a features in cloud testing.

1. Cloud-based testing environment

This refers to use a selected cloud infrastructure (or platform) as a base to form a test bed equipped with diverse and scalable computing resources, system infrastructures, and licensed tools, which are allocated using auto-provision based on static/dynamic requests. Both virtual and physical computing resources can be included and deployed inside.

2. Service-level-agreements (SLAs)

In cloud computing, all clouds, SaaS, and applications usually provide diverse services to their end users and customers with well-defined service-level-agreement. Naturally, these agreements will become a part of testing and quality assurance requirements, such as system reliability, availability, security, and performance agreements.

3. Price models and service billing

Since utility computing is one of basic concepts and features in cloud computing, so price models and utility billing becomes basic parts and service for testing as a service. In other words, required computing resources and infrastructures (including tools), and testing task services will be charged based on pre-defined cost models and

4. Large-scale cloud-based data and traffic simulation

Applying and simulating large scale online user accesses and traffic data (or messages) in connectivity interfaces is necessary in cloud testing, particularly in system-level function validation and performance testing.

B. Challenges in Cloud Testing

There are various issues and challenges in cloud testing and cloud-based software. Here we examine them from the following three areas.

1. On-demand test environment construction:

Step by step instructions to set up a testing environment deliberately (or naturally) for on-interest testing services in a cloud? In spite of the fact that the current cloud technologies help automatic provision of required computing resources for every SaaS in a cloud, there are no supporting answers for support designers to set up an obliged test environment in a cloud utilizing a cost-effective way. It is important to give an on interest test environment for TaaS clients. To do this, Testing as a Service (TaaS) vendors need to give a methodical solution for secure an obliged test environment focused around the client's selection. Engineers additionally discovered that there is an absence of cost-effective solutions for them to effectively influence their cloud based applications (or SaaS) in a cloud with the current test tool [4].

2. Scalability and performance testing:

Although many published papers talk about system performance testing and scalability evaluation in the previous decades, the vast majority of address issues and results in web-based software systems or conventional distributed software. As indicated by our literature survey on this subject, most existing papers

concentrate on frameworks for parallel and distributed systems and scalability evaluation metrics. Since these system are situated up with preconfigured system assets infrastructures and resources, execution testing and are normally directed in a static and prefixed system environment (for example, a test lab.), so the current evaluation measurements, framework and results did not consider special characteristics in cloud testing, for example, cost-models Testing security, SLA-based requirements, scalable testing environments, dynamic scalability and measurement in clouds.

3. Testing security and measurement in clouds:

Security testing has becoming a hot exploration subject with numerous open questions in software programming testing group. Since security turns into a major concern inside clouds and security services become an essential part in cloud technology and modern Software as a Service (SaaS), engineers must arrangement the issues and challenges in quality assurance and security validation for Software as a Service (SaaS) and clouds.

Here are some related issues and challenges:

- How can we assure the security of cloud based application processes and business data inside a third-party cloud infrastructure [6]?
- What are the Qos models for security oriented quality certification for end-to-end provision prepare and related business data in/on/over clouds?
- What are the test models, test sufficiency, test strategies and tools for security testing for end-end requisitions in/on/over clouds?
- How can we assure and assess user privacy in a cloud infrastructure [7]?

C. Major Needs in Cloud Testing

There are a number of major needs in cloud testing. They are discussed below.

1. Adequate test models and criteria:

To effectively support cloud testing, engineers need new adequate test models and criteria in the following areas.

Scalability models for SaaS/Cloud-based Applications

Engineers need well-defined adaptive test models and evaluation metrics for scalability and performance testing in clouds to validate and measure dynamic system scalability (such as scale-up and scale-down) and system performance (such as improvement and degradation). These models and metrics must address scalable computing resources, dynamic system loads, and pre-defined economic sales (cost/price models) [4].

Adequate integration models and criteria

To address software integration issues in cloud testing, engineers need adequate test models and criteria addressing three types of integration in cloud testing:

a) SaaS (or application) APIs and interactions to legacy systems outside clouds

b) End to end application integration crossing clouds

2. TaaS test processes and QoS standards

Today's commercial SaaS and clouds provide different types of Service Level Agreement (SLA) for users. However, both vendors and users need well-defined testing and quality assurance standards as a theoretic base to establish fair and sound service level agreements between them. Some examples are listed below.

- Although application system and data security should listed as a part of requirements for SaaS and clouds, today's cloud technologies and major players only list the security requirements for cloud infrastructure. One major reason is the lack of testing and QoS standards that address cloud and SaaS security in different aspects including, application data security, end-to-end transaction security, business process security, and user privacy.
- Since (Testing as a Service) TaaS is a service model and new concept for software testing, there is a lack of well-defined TaaS processes and QoS standards for on-demand testing services, for example, price models [4].

3. Innovative test methods and solutions

The new features and new requirements of cloud-based applications and SaaS bring some demands on new test methods and solutions.

Continuous validation and regression testing solutions

Since high system availability is very important to SaaS and cloud-based applications, engineers need automatic re-testing techniques to address the multi-tenancy feature of clouds whenever software are changed for bug-fixing or feature improvement.

Automatic test solutions for cloud interoperability

Because both clouds and SaaS provides their connectivity protocols and APIs, this requires engineers to assure the quality of interoperability of cloud-based applications crossing different clouds in connectivity protocols, firewalls, interactions between SaaS and legacy systems [4].

Innovative test technologies for cloud testing

Cloud-based applications and SaaS must support global online users to access the provided services using diverse client platforms, browsers, and technologies, this suggests that vendors need new effective test technologies to support the validation of application compatibility on different platforms, client technologies, and browsers.

18

CLOUD TESTING TOOL

A) SOASTA

SOASTA [8] is motivated by the necessity to test in production, rather than in a laboratory environment. Today's web applications usually follow agile practices with frequent builds and high change rates. Load testing with legacy tools in the laboratory can be significantly different from testing in the production environment in terms of scale, configuration, user profiles and network environment. Running tests against production websites thus can achieve higher degree of accuracy and confidence, compared with lab practices. Fig. 9. SOASTA Cloud Test architecture [8]

SOASTA Cloud Test 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. The worker nodes can be distributed across public and private clouds to cooperate in a large load testing. Test results from distributed test agents are integrated

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

for analysis. Memory-based analytic techniques are implemented to handle, in real-time, the huge data produced by large-scale testing. Provisioning data are displayed via analytic
Dashboard on a synchronized time-line. Through an Ajax based web UI, testers can operate and supervise the whole process including launching hundreds of load generation servers, creating and running test agents geographically distributed, and analysing test results. Fig. 9 shows SOASTA architecture. Cloud Test is composed of three distributed services to support test creation, test execution and test results analytics, as listed below:
SOASTA repository is responsible for delivering SOASTA Cloud Test objects such as test scenario recordings test composition, and performance data. Analytics Dashboard is a memory-based analytic service to handle large sets of results and analytics from distributed load tests. It is able to correlates many data streams from distributed environments into a single one with synchronized timeline. Meastro is a test engine as a massively multi-thread service, used for test execution (including sending and validating responses). Multiple Meastros can execute different parts of a test composition collaboratively, with the ability to be geographically distributed.

B) iTKO LISA

iTKO LISA [10] 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 virtualization technology. For unavailable or inaccessible resources, LISA provides virtualized services by simulating the target system's dynamic behaviour so that they can respond as live systems. In this way, it breaks dependence constraints of system integration and supports continuous testing LISA quality services are provided from three perspectives: LISA Test, LISA Validate, and LISA Pathfinder. LISA Test: The capabilities it offers to enhance testing include coverage-based testing for heterogeneous distributed architecture, codeless testing, UI testing, load and performance testing. It brings testability to all components deployed in the cloud by LISA development toolkit. It establishes collaborative testing platform with portable and executable test cases that are shared across different teams and environments. It provides a codeless testing environment that allows QA, development and others to rapidly design and execute automated tests. LISA Validate: LISA provides continuous regression testing, triggered by change events, for each software build and in the production environment. Policies are enforced via governance infrastructure to ensure that systems conform to quality requirements at design time, change time and runtime.
LISA Pathfinder: It traces interactions step-by-step, reviewing the dataflow and control flow among the constituents in a composite application. It facilitates testers to see through system execution process to localize defects and identify performance bottlenecks [9].

C) Cloud Testing

Cloud Testing [11] 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. As websites need to be compatible to various browsers and operating systems, Cloud Testing offers a shared test environment so that users need not set up and maintain various testing platform to ensure website portability. Test scripts are recorded by users from local browsers using Selenium IDE. The scripts are then submitted to Cloud Testing to be executed automatically in the cloud with various browsers on operating systems.

D) Cloud Sim

Cloud Sim [12] is built by CLOUD (Cloud Computing and Distributed Systems) Laboratory at the University of Melbourne in Australia. It aims to provide a toolkit for modelling and simulating the behaviour of various
Cloud components including data canters, virtual machines and resource provisioning services. It can be used for analysing and evaluating cloud strategies in a controlled simulated environment. Particularly, Cloud Sim facilitates initial performance testing with less time and effort to set up test environment.

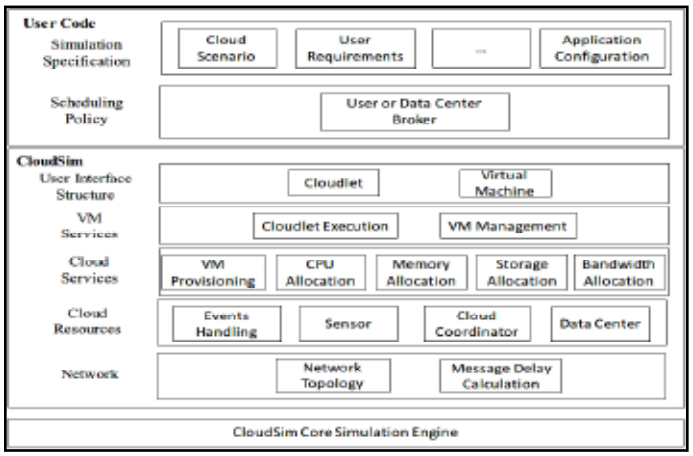

Fig. 2 : Cloud Sim Architecture [12]

As shown in Fig. 5, Cloud Sim is built with two layers: the Simulation layer and the User Code layer. It is designed to support modelling and simulating typical cloud features such as network behaviour, VM allocation, Cloud federations, dynamic workloads and power consumptions. It claims to have been used for research in companies like HP and universities.

E) D-Cloud

D-Cloud [13] is developed by University of Tsukuba in Japan. 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. Fig. 6 shows D-Cloud architecture. D-Cloud can simulate
Various typical faults and inject them into host/guest OS. It supports the definition of fault types, fault injection time and fault duration. A XML-based language is designed for describing test environment configuration and test scenarios. A tester first submit test plan in the specific language to set up infrastructure parameters, workload and fault injection scenarios. D-Cloud then initiates virtual machines to simulate the execution process following the test plan. In this way, D-Cloud enables flexible and observable platform for testing
Distributed system on the cloud.

F) PreFail

PreFail [14], [15] is developed by ParLab (Parallel Computing

Laboratory) at University of California at Berkley. For the cloud built out of tens of thousands of unreliable computers, it usually experiences frequent and diverse failures. Failure testing is thus important to validate the correctness and efficiency of cloud's recovery protocols.

However, it is expensive to enumerate all of the possible failure scenarios. PreFail thus introduces a framework for systematically and efficiently explore failures. PreFail introduces a failure abstraction called failure ID (FID), composed of an I/O ID (abstract information of an I/O call) and the injected failure, to identify every failure. Unlike D-Cloud that provides simulated "actual" faults, PreFail inserts a "failure surface" into the target system between the target system (e.g., HDFS [16]) and the OS library (e.g., Java SDK). In this way, testers can flexible program failure testing policies. PreFail provides optimization techniques to reduce redundant failure scenarios before fault injections and simulation, and parallelization method to divide failure sequences into independent parts that can be exercised in parallel on different machines. PreFail has been adopted by Cloud era Inc. for Hadoop Software testing.

G) Cloud9

Cloud9 [17], an academic research project from EPFL in Switzerland, migrates symbolic execution to the cloud platform. Symbolic execution is an important testing technique introduced in 1970s. It reasons all the possible executions by exploring program path-by-path. In spite of research over 30 years on path exploration, it still faces the challenging problem of scalability. With increasing program size and complexity, its memory and CPU consumptions go exponentially. It is difficult to be applied to industry software in general where software often contains millions of lines of code. Basically, Cloud9 parallelizes symbolic execution on large shared-nothing clusters of computers. To do this, it divides the path exploration work into independent jobs to be allocated to different worker nodes. The conventional search strategy is distributed to the workers, with reduced coupling and coordination among workers. The global load balancer is used to balance the workload of each worker, in order to get high efficiency of testing service.

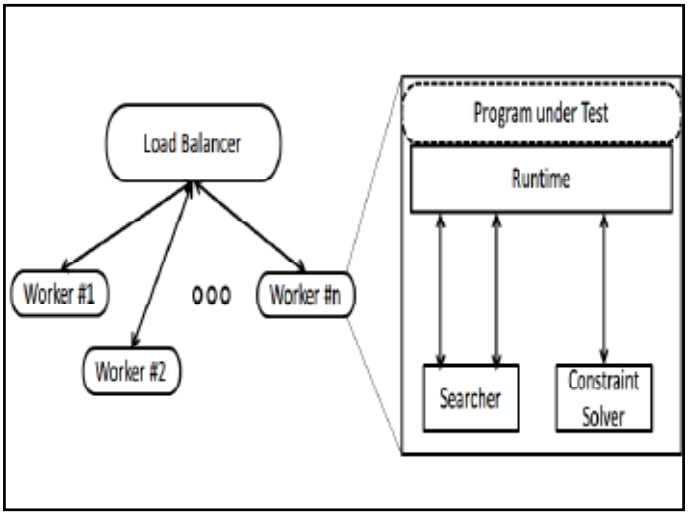

Fig. 3 : Cloud9 Architecture [17]

Fig. 7 shows the architecture of Cloud9. Each worker consists of a runtime, a searcher and a constraint solver. It independently explores a sub-tree of program's execution tree. An initial prototype is implemented with the single-node Klee symbolic execution engine running Fig. 7. Cloud9 Architecture [17] on the infrastructure service provided by Amazon EC2. It reduced testing time of 32 real UNIX utilities on an average by a factor of 47, with a maximum of 250-fold speedup.

H): Hadoop Unit

Hadoop Unit [18] migrates JUnit test framework to Hadoop platform. JUnit test cases are created as independent Hadoop Map Reduce jobs. The map () function receives test jobs as < test name; test command > pair. At each node, the command is executed as a process. The reducer gets < testname; test result > from each map and combines all the results. Experiments shows that a 150-node cluster can produce 30 x improvements compared with sequential test executions on a local computer.

I) YETI

YETI (York Extensible Testing Infrastructure) [19] also provides a cloud version random testing tools. It uses Map Reduce to parallelize the processing of test inputs and results. A preliminary evaluation was carried using Amazon EC2. It showed clearly performance improvements by employing more computers and distributing the jobs.

IV. Conclusions

Cloud computing and Cloud testing are likely to be active and popular research fields in the near future. On the other hand, cloud computing itself is under constant evolution, continuously bringing in new opportunities and challenges for software testing research. In this paper we have presented depth study about Testing as a service (TaaS) on cloud. Opportunities, challenges and possible research directions researchers in this field can benefit from the results in selecting their research direction and identifying new research opportunities for future work. We have observed that Test task management is also among the potential areas for further research. Our future research will be focusing on filling these gaps for achieving a 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 for more proliferated use over the cloud.

References

[1] *(2012) Ibm cp-40 project. [Online]. Available: http://en.wikipedia.org/wiki/IBM CP-40*

[2] *Won Kim, Soo Dong Kim, Eunseok Lee, and Sungyoung Lee "Adoption issues for cloud computing" In Proc. of the 7th International Conference on Advances in Mobile Computing and Multimedia," pages 2–5, New York, NY, USA, 2009.*

[3] *Youssef Ridene and Franck Barbier. "A model-driven approach for automating mobile applications testing" In Proc. of the 5th European Conference on Software Architecture: Companion Volume, pages 9:1–9:7, New York, NY, USA, 2011.*

[4] *Jerry Gao, Xiaoying Bai and Wei-Tek Tsai "Cloud Testing- Issues, Challenges, Needs and Practice" Software Engineering : An International Journal (SEIJ), Vol. 1, No. 1, SEPTEMBER 2011*

[5] *P. Williams, "Value versus cost: governing IT on a reduced budget", ComputerWeekly.com, Friday 08, February 2002.*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

[6] B. Wrenn, CISSP, ISSEP, "Unisys Secure Cloud Addressing the Top Threats of Cloud Computing," (white paper)..
[7] AppLabs, "Testing the Cloud," white paper, Internet: http://www.applabs.com/html/TestingtheCloud_786.html
[8] SOASTA. [Online]. Available: http://www.SOASTA.com/ "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[9] Xiaoying Bai, Muyang Li, Bin Chen, Wei-Tek Tsai, Jerry Gao "Cloud Testing Tools" Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering (SOSE 2011)
[10] ITKO. [Online]. Available: http://www.itko.com/
[11] Cloud Testing. [Online]. Available: http://www.CloudTesting.com/
[12] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23–50, 2011.
[13] T. Banzai, H. Koizumi, R. Kanbayashi, T. Imada, T. Hanawa, andM. 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.
[14] P. Joshi, H. S. Gunawi, and K. Sen, "PreFail: a Programmable Tool for Multiple-Failure Injection," in Proceedings of the 2011 ACM international conference on Object oriented programming systems languages and applications, 2011, pp. 171–188.
[15] H. S. Gunawi, T. Do, P. Joshi, P. Alvaro, J. Yun, J.-s. Oh, J. M. Hellerstein, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, K. Sen, and D. Borthakur, "FATE and DESTINI: A Framework for Cloud Recovery Testing," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-127, Sep 2010. [Online]. Available: http: //www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-127.h
[16] D. Borthakur. The Hadoop Distributed File System: Architecture and Design. [Online]. Available: http://hadoop.apache.org/common/docs/r0. 18.0/hdfs design.pdf
[17] L. Ciortea, C. Zamfir, S. Bucur, V. Chipounov, and G. Candea, "Cloud9: A Software Testing Service," ACM SIGOPS Operating Systems Review, vol. 43, no. 4, pp. 5–10, 2010
[18] 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.
[19] M. Oriol and F. Ullah, "YETI on the cloud," in Third International Conference on Software Testing, Verification, and Validation Workshops, 2010, pp. 434–437.

Author's Profile

Shivam Jain
A research scholar of M.Tech - Software Engineering at Manipal University Jaipur. He did his B.Tech in Information Technology from Kailash Chandra Bansal College of Technology in 2012.

Devesh Kumar Srivastava
Currently he is associated with Manipal Univrsity at Jaipur, Rajsthan India.