International Journal of Engineering Research and General Science Volume 3, Issue 1, January-February, 2015
ISSN 2091-2730

Software Testing in Cloud

J.Jayashree, Dr.B.Persis Urbana Ivy, J.Vijayashreee

School of Information Technology and Engineering,

VIT University, India

jayashree.j@vit.ac.in

Abstract— Cloud computing has developed as another figuring standard that effects a few diverse exploration fields, including programming testing. Testing cloud applications has its own characteristics that interest for novel testing routines and devices. Advancement of more compelling and adaptable programming testing strategies are being provided. This paper gives an account of an efficient review of distributed papers. We give a diagram with respect to fundamental commitments, patterns, crevices, opportunities, difficulties and conceivable Research headings. We give an audit of programming testing over the cloud writing and sort the assortment of work in the field.

Keywords - exploration fields, routines, adaptable, patterms, assortment.

Cloud computing with Software Testing

Cloud computing is Internet computing. Cloud computing can be defined as utilizing the internet to provide technology enabled services to the people and organizations. Cloud computing empowers purchasers to get to assets online through the web, from anyplace at any time without stressing over specialized/physical administration also support issues of the first assets. Moreover, Assets of distributed computing are alert and adaptable. Cloud computing is free processing it is completely not quite the same as framework and utility figuring. Google Apps is the central illustration of Cloud figuring, it empowers to get to administrations by means of the program and conveyed on a great many machines over the Internet. Assets are available from the cloud at any time and from wherever over the globe utilizing the web.

Cloud computing is less expensive than other figuring models; zero support expense is included since the administration supplier is in charge of the accessibility of administrations and customers are free from support and administration issues of the asset machines. Because of this gimmick, distributed computing is otherwise called utility registering, or _IT on interest'. Adaptability is key quality of distributed computing and is accomplished through server virtualization. This new, electronic era of registering uses remote servers set in greatly sheltered and secure server farms for capacity of information and administration, so associations don't have to pay for and look after their inner IT arrangements. After production of a cloud, Sending of distributed computing contrasts with reference to the prerequisites and for the reason it will be utilize The principal service models being deployed are:

Software as a Service (SaaS): Software‘s are provided as a service to the consumers according to their requirement, enables consumers to use the services that are hosted on the cloud server.

Platform as a Service (PaaS): Clients are provided platforms access, which enables them to put their own customized software‘s and other applications on the clouds.

Infrastructure as a Service (IaaS): Rent processing, storage, network capacity, and other basic computing resources are granted, enables consumers to manage the operating systems, applications, storage, and network connectivity.

Cloud Testing uses cloud framework for programming testing. Organizations seeking after testing all in all and burden, execution testing and generation administration observing specifically are tested by a few issues like constrained test plan, meeting due dates, high expenses every test, substantial number of experiments, and practically no reuse of tests and topographical appropriation of clients add to the difficulties. Besides guaranteeing superb administration conveyance and evading blackouts obliges testing in one's datacenter, outside the server farm, or both. Cloud Testing is the answer for all these issues. Compelling boundless stockpiling, speedy accessibility of the foundation with adaptability, adaptability and accessibility of conveyed testing environment lessen the execution time of testing of substantial applications and lead to savvy arrangements.
This paper is a survey on collection of fifteen papers on software testing on cloud computing and has explained in the way the authors has proposed.

SURVEY

"Cloud Chamber: A Self-Organizing Facility to Create, Exercise, and Examine Software as a Service Tenants"
This work [1] describes the Cloud Chamber testbed to investigate autonomic resource anagement of web services in a cloud environment. Cloud Chamber is a virtualized environment which provides web servers as services,facilities to apply loads to the tenant services,algorithms for autonomic organization and reconfiguration of service assignments as demand changes, and sensors to capture resource consumption and performance metrics. The testbed inserts sensors into web servers to collect the resource utilization of CPU cycles, memory consumption, and bandwidth consumption of the individual web services, the web server, and the operating system. This high resolution performance data generates profiles of the resource usage of each web service and the resource availability of each server. The testbed, as described in this work, utilizes these profiles to efficiently place services on servers, thus balancing resource consumption, service performance, and service availability. Once services have been placed, the testbed monitors changes such as traffic levels, server churn, and the introduction of new services. The information gathered is used to calculate configurations of service placement which better meet the changing requirements of the environment. In conclusion the Cloud Chamber meets all the requirements described in the introduction. The Cloud Chamber creates a facility to create, exercise, and examine the behavior of tenants in a Software as a Service environment. Services of various shapes and sizes can be deployed onto a heterogeneous set of nodes providing different amounts of resources. These services can be executed with any size of prescribed load for any length of time. The nodes self-organize autonomously finding and implementing tenant assignments in response to changes in the environment. The authors are unaware of any such facility.

"A Parallel Genetic Algorithm Based on Hadoop MapReduce for the Automatic Generation of JUnit Test Suites"
In this paper [2] we proposed the use of a Parallel GeneticAlgorihtm (PGA) for test suite generation exploiting Hadoop MapReduce and showed a preliminary evaluation of its use on a small cluster. The obtained results highlighted that using PGA allowed us to save over the 50% of time. Since the use of parallel SBST approaches is still in its early phases, several directions can be prospected as future work. First of all a deeper empirical evaluation of the proposed approach is needed to asses on other subjects its strength or weakness, as well as to assess its actual scalability employing different GA settings, numbers of maps, and larger clusters. Also the use of Hadoop MapReduce should be assessed running it not only on standard clusters, but also exploiting cloud computing and graphic cards. Moreover, it can be interesting to realize and compare higher levels of parallelization, such as by parallelizing the genetic operations other than the fitness evaluation. It would be also interesting to verify how other approaches for test suite generation could be take advantages of parallel/distributed computation as the one described in this paper. Finally, as a long-term research goal, it will be desirable to integrate these SBST approaches within a whole Validation-as-a-Service platform, available in the Cloud, to support the entire software testing process.

"A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud"
In this paper [3] we provide an overview of a multi-faceted project targeted at automatically testing the security and robustness of Android apps in a scalable manner. We describe an Android-specific program analysis technique capable of generating a large number of test cases for fuzzing an app, as well as a test bed that given the generated test cases, executes them in parallel on numerous emulated Androids running on the cloud. We have presented a novel framework for automated security testing of Android applications on the cloud. The key contributions of our work are (1) a fully automated test case generation, (2) iterative feedback loop to generate and guide our input in an intelligent manner that ensures code coverage and uncovers potential security defects, and (3) highly scalable *fuzzing* by leveraging the cloud. In our on going work, we are exploring two approaches for improving the test case generation facet of our framework. First, we are developing an evolutionary algorithm for generating tests, as part of which we are modeling the problem of testing an Android app as a genetic problem and developing an appropriate fitness function to evaluate the quality of test cases. Second, we are developing an Android-specific symbolic execution engine for automatically generating test cases. We are extending *Java Pathfinder*, which is capable of symbolically executing pure Java code, to work on Android. In addition, we are creating a graphical reporting environment that would allow the security analyst to visually explore the results of the testing, and in particular obtain metrics (e.g., achieved code coverage, bugs per KSLOC) that could then be used for making decisions as to the overall security and robustness.

"Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey"
This paper [4] makes three contributions. First, we performed a systematic review of software test automation benefits and limitations in academic literature. We collected 24.706 papers, which were reduced to 25 research works (see Table II). Thus, the amount of evidence on these matters is quite shallow as many benefits and limitations are backed up by only one or two sources. Furthermore, we found that while benefits often came from stronger sources of evidence (experiments and case studies), limitations were more frequently reported on experience reports. We think that this is caused by publication bias regarding the benefits. We believe that important further work on this area is to assess the limitations of test automation with rigorous empirical studies, i.e. case studies and

experiments. Second, we conducted a survey of the practitioners' view of software test automation benefits and limitations. The results showed that the main benefits of test automation are reusability, repeatability and effort saved in test executions.These results support the superiority of test automation when several regressions testing rounds are needed. Furthermore, the practitioners indicate that automation improves test coverage, which means that automation has benefits even when excessive regression testing is not needed. Regarding the limitations, we found that automation bares a high initial cost in designing the test cases, buying a test automation tool,and training the staff. Non-surprisingly, the maintenance of automated test cases was also perceived as problematic. Also45% of the practitioners think that current test automation tools offer a poor fit for their needs. The limitations of test automation perceived by the practitioners should outline important future research directions.

"Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds"

We address some of these challenges in [5] through Expertus—a flexible code generation framework for automated performance testing of distributed applications in Infrastructure as a Service (IaaS) clouds. Expertus uses a multi-pass compiler approach and leverages template-driven code generation to modularly incorporate different software applications on IaaS clouds. Expertus automatically handles complex configuration dependencies of software applications and significantly reduces human errors associated with manual approaches for software configuration and testing. To date, Expertus has been used to study three distributed applications on five IaaS clouds with over 10,000 different hardware, software, and virtualization configurations. The flexibility and extensibility of Expertus and our own experience on using it shows that new clouds, applications, and software packages can easily be incorporated. More generally, our evaluation results show the feasibility, extensibility and usefulness of our approach in cloud testing. We have also identified several limitations of our tool and continue our research work to address them. Our final goal is to make Expertus available for public use. Our tool can significantly reduce the deployment and configuration cost of running distributed test scenarios in today's production cloud environments, which indicates great promise for the future.

"A Survey of Software Testing in the Cloud"

This paper [6] reports on a systematic survey of published results attained by the synergy of these two research fields. We provide an overview regarding main contributions, trends, gaps, opportunities, challenges and possible research directions. We provide a review of software testing over the cloud literature and categorize the body of work in the field. Cloud computing and software testing are likely to be active and popular research fields in the near future. Traditional software testing techniques are being adapted for the cloud. 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 a classification of current research studies, identified gaps in the literature and investigated the correlation of software testing with different deployment models of cloud computing. 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 acceptance testing is an open research area for testing over the cloud. Test task management is also among the potential areas for further research. Finally, we believe that interoperability testing needs more emphasis as a research area to ensure reliable service composition by means of integrating services from different service delivery models. 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.

"Path Coverage Testing in the Cloud"

The aim of this paper [7] is to present a new method for automated software esting as a cloud computing service. Unlike actual testing services, our goal is to provide a fully automated testing without human involvement from the service user's or provider's side. We use a program modeling allowing an easy symbolic execution and a scalable parallelization of the testing. Programs are divided into several parts assigned to different nodes (Workers) of the cloud. A particular node (Coordinator) allocates tasks to Workers and collects the final results. In this paper we have presented a new approach for path testing in the cloud. We have first defined the backward symbolic execution method performed to compute path formulas. Then we have proposed a distributed version of our solution allowing to take advantage of cloud infrastructures and to overcome the problem of path explosion. Our solution presents several advantages: It is a backward method: So instead of executing the entire program as in the other methods, it just captures the impact of each statement on the considered predicates. Our solution in the cloud does not require a great preparation, nor dynamic partitioning. No communication is required between workers and the amount of communication between the workers and the coordinator is minimal. Each worker is required to have just the information concerning the paths it computes.

"Emulation of Cloud-Scale Environments for Scalability Testing"

We use an emulation approach in [8], whereby endpoints are modelled and then executed in an emulation environment, which we call "Kaluta". The key aspect is to balance the modelling of the endpoint systems such that it is rich enough to "fool" an unmodified application-undertest into thinking that it is talking to real systems, but lightweight enough such that tens of thousands of instances of

model systems can be executed simultaneously in the emulation engine. We present an industry case study – CA IdentityMinderTM-as-a-
Service – to demonstrate the effectiveness of using emulation to validate the scalability of a cloud hosted application.We have shown that emulation is a feasible approach for large scale testing which fills a gap which is not easy to achieve with other tools. Software systems are becoming increasingly connected with other systems in their environment, and in a large enterprise this can involve connections to tens of thousands of other systems. It is important that software components are tested for the limits of their scalability before being deployed in production. With the increasing trend of cloud computing, software will be exposed to new scales. Giving developers the tools to measure performance at these extremes will be essential to delivering high quality cloudenabled software.

"Testing as a Service (TaaS) on Clouds"

In [9] Cloud computing leads an opportunity in offering testing as a service (TaaS) for SaaS, clouds, and cloud-based applications. This brings new business opportunities, challenges, and demands in innovative service models, testing techniques, QoS standards, and requirements. This paper provides a comprehensive tutorial on testing as a service in a cloud environment. It answers the common questions raised by engineers and managers, and provides clear conceptual discussions about testing as a service (TaaS), including its scope, objectives, motivations and values, distinct features, required techniques, as well as testing environments. It not only presents a classification of different types of testing services in TaaS, but also offers a clear comparative view and perspectives between conventional software testing service and cloud-based testing as a service. In addition, it examines underlying issues, challenges, and emergent needs.Testing as a service (TaaS) is becoming a hot research topic in both cloud computing and software engineering research communities. As the advance of cloud technology and testing as services, more research results are needed to address the open issues and challenges on TaaS infrastructures, techniques, and automation solutions. More innovative testing techniques and solutions, and QoS standards are needed to support on-demand testing services in a scalable cloud infrastructure, for example, SaaS testing adequacy and standards for multi-tenancy. This paper provides a comprehensive review and in-depth tutorial on cloud-based TaaS for SaaS applications. It offers essential tutorial concepts on TaaS definitions, scope, motivations and benefits, as well as classified testing services and test environments. In addition, it discusses in details about TaaS requirements, distinct features, issues, challenges, and needs. Moreover, it highlights the major differences between conventional software testing and cloud-based TaaS.

"Exploiting Cloud Computing for enabling distributed testing of complex systems: the SELEX-SI roadmap"

In [10] Cloud computing represents the most promising way for allowing the seamless access to distributed testbed from any site and for allowing remote testing activities, either at system and integration level. A cloud based infrastructure in charge of connecting all the company premises would allow to run testing experiments from anywhere and, more important, the possibility of reproducing distributed systems deployment scenarios to run integration testing in a pre-installation phase thus dramatically reducing company costs. This paper aims to
describe the cloud research roadmap that SELEX-SI has been designing, the architectural design of the cloud infrastructure and the real ROI that the company expect from introducing such an innovation into the traditional software production process.To conclude, the results gained and illustrated in this work show that KVM is a valuable Open Source alternative to build up a CC environment against both the performance and application requirements exposed by SELEX-SI scenarios. This paves the way to the actual development of a CC platform aimed at supporting the company V&V processes and optimizing costs.

"The Application and Development of Software Testing in Cloud Computing Environment"

Paper [11] Software testing is an important part of software engineering and it's very important to enhance software quality. In the cloud computing environment, it is an unchangeable trend that the traditional software testing turns to "cloud testing". The article discusses the advantages processes and procedures of software testing and cloud computing . Based on those procedures discussed, the intersection between software testing and cloud computing is pointed out. At last, the advantages and development trend of cloud testing is analyzed.

Cloud computing is the next key point of information technology in the cloud environment and the applications of information technology will be changed greatly. The effective combination of the software testing and cloud computing technology is not only the innovation of software testing model, but also extension of the cloud computing in the application field. This paper introduces some knowledge of the cloud computing and software testing , expounds the advantages and developing trend of software testing in the cloud computing environment, and provides certain reference for diversifying the software testing method and developing the cloud computing applications.

" Mechanism for On Demand Tag-Based Software Testing in Virtualized Environments"

In this paper [12] we provide a generic way to manage the tests and provide an efficient mechanism to run the tests selectively. In our approach we make use of well-known build management tool called Jenkins for running the tests on-demand. One unique benefit of our approach is that the input is a single comma-separated-value (CSV) file and it is very easy to add/modify existing tests. The tests

are tagged using well known keywords (viz. database-layer, configuration, regression). When some particular tests need to be run, the user enters 'Tag' in the job parameter and testsuite will be generated dynamically. In this paper we also describe an end-to-end test management system that supports running selective tests with help of jobs created in Jenkins environment. The proposed model of Tests execution is highly useful in high demanding environments like agile software development model, Test driven development model where feature development is many times faster than traditional water fall model. The approach suggested in this paper makes optimum use of cloud resources by distributing the jobs, so this can be utilized specially for testing under virtualized environment. The proposed model can be leveraged for model-based testing (MBT) also called as Mind-mapping tools which naturally extend Tag-based testing. One more interesting enhancement which can be done in cloud environment is to use identity based encryption techniques (IDE) for user authentication. The test management system when deployed on cloud need to be secured from unauthenticated access and traditional encryption techniques like PKI doesn't prevent from unauthorized access. So user authentication and authorization part can be implemented using IDE techniques such as key-policy attribute based encryption (KP-ABE) and cipher-text-policy attribute based encryption (CP-ABE).

"Migrating Load Testing to the Cloud: A Case Study"

In this paper [13] , we focus on migrating conventional load testing tools to the cloud, for which the two significant issues are about multitenancy and load simulating resource management.We propose a four layer model for cloud-based load testing, along with the approach of test request admission control and scheduling to solve these issues. We carried out a concrete case study on our proposed approach and made the efficiency of cloud-based load testing shown successfully by two contrast experiments. In this paper, we propose some techniques of migrating load testing software to the cloud. We present the challenges for load testing to be a cloud service, including requirements for multi-tenancy and the load simulating resource management issue. In our case study of migrating Bench4Q tool to the cloud, a four layer cloud-based load testing model (portal Layer, business Layer, test infrastructure Layer and IaaS Layer) is introduced to meet these requirements. Also, an innovation admission control and scheduling algorithm taking use of the distributed Agent component and the technology of virtualization is proposed to solve the load simulating resource management issue. In the future, we are going to add a multi-protocol support, such as DNS, TCP and FTP. Also, testing for Web service is supposed to be available. We are thinking of reimplementing the Script Recorder to be an automatic one
by analyzing the workflow of Web applications. Moreover, we are going to put the cloud-based Bench4Q platform to the public cloud after improving its robustness.

"A Cloud-Based TaaS Infrastructure with Tools for SaaS Validation, Performance and Scalability Evaluation"

This paper [14] proposes a testingas- service (TaaS) infrastructure and reports a cloud-based Taas environment with tools (known as CTaaS) developed to meet theneeds in SaaS testing, performance and scalability evaluation. The paper presents TaaS concepts and CTaaS, including their infrastructure, design and implementation. In addition, the paper demonstrates the application results of our previously proposed graphic models and metrics for SaaS performance and scalability evaluation. Moreover, the paper reports one case study for a selected SaaS (OrangeHRM) using the developed TaaS environment. Although there are numerous papers discussing testing services for software applications, only a few of them focus on special features, issues, and needs on cloud-based TaaS. This paper first presents our vision and distinct features of TaaS in cloud computing, and then proposes an infrastructure of TaaS in cloud computing. This paper presents its initial prototype and application experiments. Currently, we are extending CTaaS by adding a SaaS tracking server (known as SaaS-Watcher) into the infrastructure to support different types of SaaS program tracking.The future extension of this research includes two folds. The first is to work on multitenancy testing service for SaaS in cloud-based TaaS. The next is to provide continuous testing techniques and tools for SaaS vendors to support testing services.

"SaaS Testing on Clouds – Issues, Challenges, and Needs"

This paper [15] provides a tutorial to discuss SaaS testing, including its concepts, focuses and objective, test process, test environments, and requirements. Moreover, the paper discusses the special SaaS features, and examines the related issues and challenges and needs in SaaS testing. With the advance of cloud computing and SaaS engineering, people begin to realize the importance of SaaS testing, and have encounter many open issues and challenges. This paper provides a tutorial to cover the different perspectives of SaaS testing at the conceptual level. It highlights special testing features, challenges, and needs in SaaS testing and its future research and practices. We believe there is an urgent need in the near future to develop new
test standards, techniques, platforms, solutions for SaaS testing and automation.

REFERENCES:

[1] M. Brent Reynolds, Don R. Hulce and Kenneth M. Hopkinson "Cloud Chamber: A Self-Organizing Facility to Create, Exercise, and Examine Software as a Service Tenants", In 45th Hawaii International Conference on System Sciences , 2012,pp.5546-5555.

[2] Linda Di Geronimo, Filomena Ferrucci, Alfonso Murolo, Federica Sarro "A Parallel Genetic Algorithm Based on Hadoop MapReduce for the Automatic Generation of JUnit Test Suites", In IEEE Fifth International Conference on Software Testing, Verification and Validation, 2012,pp.785-792.
[3] Riyadh Mahmood, Naeem Esfahani, Thabet Kacem, Nariman Mirzaei, Sam Malek, Angelos Stavrou "A Whitebox Approach for Automated Security Testing of Android Applications on the Cloud", 2012.
[4] Dudekula Mohammad Rafi, Katam Reddy Kiran Moses, Kai Petersen "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey", 2012 ,pp.22-28.
[5] Deepal Jayasinghe, Galen Swint, Simon Malkowski, Jack Li, Qingyang Wang, Junhee Park and Calton Pu. "Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds",In IEEE Fifth International Conference on Cloud Computing ,2012,pp.115-122.
[6] Koray ˙Inc, ki, ˙Ismail Arı , Hasan S¨ozer "A Survey of Software Testing in the Cloud", In IEEE Sixth International Conference on Software Security and Reliability Companion,2012,pp.18-22.
[7] Nassima Aleb Samir Kechid "Path Coverage Testing in the Cloud", The 2nd International Conference on Communications and Information Technology,2012,pp.118-122.
[8] Steve Versteeg,_ Cameron Hine,y Jean-Guy Schneidery and Jun Hany "Emulation of Cloud-Scale Environments for Scalability Testing", In 12th International Conference on Quality Software,2012,pp.201-209.
[9] Jerry Gao1,2, Xiaoying Bai2, and Wei-Tek Tsai2,3 "Testing as a Service (TaaS) on Clouds", In IEEE Seventh International Symposium on Service-Oriented System Engineering,2013,pp. 212-222.
[10] Gabriella Carrozza, Massimo Loffreda "Exploiting Cloud Computing for enabling distributed testing of complex systems: the SELEX-SI roadmap",2012.
[11] Peng Zhenlong Ou Yang Zhonghui "The Application and Development of Software Testing in Cloud Computing Environment", In International Conference on Computer Science and Service System,2012,pp.450-454.
[12] "Mechanism for On Demand Tag-Based Software Testing in Virtualized Environments", In Fourth International Conference on Advanced Computing,2012.
[13] Qiang Gao1,2, Wei Wang1, Guoquan Wu1, Xuan Li1,2, Jun Wei1, Hua Zhong1 "Migrating Load Testing to the Cloud: A Case Study", In IEEE Seventh International Symposium on Service-Oriented System Engineering,2013,pp. 429-434.
[14] Jerry Gao, K. Manjula, P. Roopa, and E. Sumalatha Xiaoying Bai W. T. Tsai Tadahiro Uehara "A Cloud-Based TaaS Infrastructure with Tools for SaaS Validation, Performance and Scalability Evaluation", In IEEE 4th International Conference on Cloud Computing Technology and Science,2012,pp. 464-471.
[15] Jerry Ga Xiaoying Bai W. T. Tsai Tadahiro Uehara "SaaS Testing on Clouds – Issues, Challenges, and Needs", In IEEE Seventh International Symposium on Service-Oriented System Engineering,2013,pp.409-415.