A mutation-based model to rank testing as a service (TaaS) Providers in cloud computing.

Abdel-Rahman Al-Ghuwairi
Hashemite University
Software Engineering Department
Zarqa, 13133, Jordan
P.O.Box 33127,00962-796921277
ghuwairi@hu.edu.jo

Hazem Eid
Hashemite University
Software Engineering Department
Zarqa, 13133, Jordan
P.O.Box 33127,00962-786041112
hazemeid@itc.hu.edu.jo

Mohammad Aloran
Hashemite University
Software Engineering Department
Zarqa, 13133, Jordan
P.O.Box 33127,00962-785050797
Mohammedaloran@itc.hu.edu.jo

Zaher Salah
Hashemite University
Department of Computer Information Systems
Zarqa, 13115, Jordan
P.O.Box 330136,00962-53903333
zaher@hu.edu.jo

Aladdin Hussein Baarah
Hashemite University
Department of Computer Information Systems
Zarqa, 13115, Jordan
P.O.Box 330136,00962-53903333
aladdin.baarah@hu.edu.jo

Ahmad A. Al-oqaily
Hashemite University
Department of Computer Information Systems
Zarqa, 13115, Jordan
P.O.Box 330136,00962-53903333
aloqaily@hu.edu.jo

ABSTRACT

With the increase of cloud computing service models, the need to measure and evaluate them are increased as well. In this paper, we proposed a novel measurement approach for the purpose of evaluating the quality of Testing as a Service (TaaS), which is considered as one of the most recent outstanding model within cloud computing environment. (TaaS) as outstanding model include the provision of multi-sub services, such as enabling cloud customer to verify his own code through the use of cloud provider resources. Its goes without questioning that testing over web environment requires high level of resources, time, and effort. Therefore, it should take high attention toward the quality of the used testing technique. Where, the quality of testing technique associated with set of attributes that has the ability to determine testing effectiveness. Thus, in this paper we propose a measurement approach to evaluate the effectiveness of TaaS, over cloud computing environment which relies on the use of mutation score. The main contribution of the proposed model represent in the use of mutation score to evaluate cloud providers ability to perform TaaS, and rank them according to the percentage of TaaS effectiveness.

Categories and Subject Descriptors

[**A**]: Cloud computing and applications

ICC '16, March 22-23, 2016, Cambridge, United Kingdom

DOI: http://dx.doi.org/10.1145/2896387.2896403

Keywords

Cloud computing, Cloud services, Testing as a services, Measurement, Effectiveness, Mutation.

1. INTRODUCTION

Recently, cloud computing service models appears clearly in the field of software system manufacturing and represent as a supporter in industry and individual work-flow. In such fields, the provided service has to satisfy cloud customer requirements and specifications. According to National Institute of Standards and Technology (U.S. NIST), cloud computing defined as "a model for enabling ubiquitous, 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 is composed of five essential characteristics, three service models, and four deployment models" [7]. The previous definition includes three service models that categories the cloud services as: Software as a Service (SaaS) which is cloud provider ability to provide applications running on a cloud infrastructure to cloud customer. These applications are accessible from various customer devices through either a thin customer interface, such as a web browser (e.g., web-based email), or a program interface, where cloud customer doesn't manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings [7]. The second service model is Platform as a Service (PaaS) which is defined as cloud provider ability to deploy cloud customer requirements onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools

supported by the provider, where cloud customer doesn't manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment [7]. The third service model is Infrastructure as a Service (IaaS) which defined as cloud provider ability to provision processing, storage, networks, and other fundamental computing resources where the customer is able to deploy and run arbitrary software, which can include operating systems and applications, where cloud customer doesn't manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls) [7]. In software engineering, software testing takes place when software system developer complete the coding of what has been required as functional and non-functional requirements. Testing becomes an important subject in software life cycle for the purpose of verifying code implementation, where testing tasks and activities can be implemented through the use of tester experiences or through the use of automated testing techniques . The main reason of testing as a phase in software life cycle is to improve the implemented code through set of modification process. Therefore, code testing is a reason behind modifying software code for: improving code performance, fixing a bug, adding features and functionalities. It should be noted that, there are modifications should be implemented immediately and should be delivered back to the developer on time with high effectiveness . In cloud computing, the provision of Testing as a Service (TaaS) helps in code verifying process(es), through the use of cloud provider infrastructure to perform an idealistic testing techniques. Therefore, TaaS should be defined in the context of cloud computing doamin. In the few recent years, many definition has been emerged to determine the conceptual meaning of TaaS in the context of cloud computing environment. George Candea et al. in [1] identified three types of testing service below: - "A public certification service, akin to Underwriters Labs, that independently assesses the reliability, safety, and security of software - A "home edition" on-demand testing service for consumers to verify the programs they are about to install on their PC or mobile device; and - A "programmer's sidekick" enabling developers to thoroughly and promptly test their code with minimal upfront resource investment". After that,we need an individual part to control the process of providing service and negotiations between the provider and the customer. Thus, we used a third-party broker, where cloud broker is a third-party that works as a mediator between the customer and the provider of cloud computing services. In general, a broker is someone (person or company) works as an intermediary between two or more parties in cloud computing during negotiations. To ensure that cloud provider has the ability to provide TaaS service with high level of quality , service provision should be evaluated for the purpose of assess there effectiveness. In this paper, mutation testing technique used for the purpose of measuring TaaS service effectiveness. In more details, mutation testing is a fault-based testing technique that has the ability to measures the effectiveness of test cases that used to find out the level of testing quality. The proposed model relies on the use of mutation operators that substitute sections of the program to produce slight syntactic modifications to the original source code [3]. Therefore, the implementing of one specific operator, generate a new version of the program; the resulting version is called "mutant" . assess mutant is tested against test suites to assess the effectiveness of test cases in detecting faults [8], If the tested mutant produces different results than the original program the tester will induce that the program contains a syntactic error that needs to be corrected. Otherwise, if the tested mutant produces the same expected result as the original program then test cases have to be improved.This type of testing can be considered a defect testing rather than Validation testing . As a complement of mutation technique, mutation score measure used also where it perform the ratio of the number of Dead Mutants over the number of Non Equivalent Mutants. The goal is to have a sore of one (1), which means that all faults in all mutants have been detected; the more dead mutants the higher the score will be.

Currently, making sure that the testing process has achieved high effectiveness is difficult, because the nature of the TaaS service differs from cloud provider to another. To insure a high effectiveness of software testing, we insert a mutations to the original code, after analysing TaaS output, after that mutation score measuring applied on the outputs of TaaS to measure testing effectiveness.

This paper is organized as following section 1 introduction which contains general understanding of our paper, Section 2 related work that shows other studies about this topic, section 3 discuss the proposed procedural model, section 4 conclusion and future work, section 5 acknowledgements and section 6 references.

2. RELATED WORK

Recently, many research has been emerged to discus and handle the issue of using testing technique over cloud computing environment, Therefore, we represent in this section list of related research that take about TaaS.

Zhang et al. in [6] represent an approach to analyse the relationship between testing tasks and establishment of task relationship model. Based on this analyses, [6] propose a dynamic task scheduling strategy that use genetic algorithm, which not only ensures to get the least execution time but also guarantee load balance. The dynamic strategy relies on the use of genetic algorithm that compared with traditional static genetic algorithm on cloudsim. The experimental result shows high level of effectiveness of the proposed strategy that used also to test resources. In our research we suggest a measurement approach to detect TaaS effectiveness based on execution time of cloud provider to complete testing service successfully.

Shou-Yu Lee et al. in [5] produced an approach to discuss the constraints in a traditional service-oriented architecture, such as: resource monitoring and allocation, [5] proposed also a middle-ware for enterprises that shared cloud Computing services, through a custom description language that complete with cross-layer monitoring module through: the development platform interface, service developers can easily design new service components, or reuse the old components to construct a new system. In our approach we suggest a measurement approach to detect TaaS effectiveness based on execution time of cloud provider to complete testing service successfully.

Firdhous et al. in [2] suggest an approach that produced a trust management system along with a trust evolution

mechanism that can be used to measure the performance of services based on the response time of systems. The proposed mechanism formulates trust scores at different service level requirements based on assurance (confidence) levels. Hence this, the proposed mechanism [2] is suitable for managing multiple service levels against single trust score for all the service levels. Also the proposed mechanism [2] is adaptive as it continuously modifies the scores based on the performance of the system. The proposed mechanism has been tested in a simulated environment and the results are found to be satisfactory. In our approach, we suggest a measurement approach to detect TaaS effectiveness based on execution time of cloud provider to complete testing service successfully and represent in evaluating testing as provided service through the using of mutation approach.

Xiaodong Zhang et al. in [10] represent an approach to introduces service equivalent as the basic metric to measure the capabilities of service resources and proposes an optimal service selection model based on capability and quality of service resources and algorithm in order to solve the issues of matching capability of service resource and the optimal selection of service resource based on quality. Then [10] proves that the model can effectively reduce the waste of resources by the test, which achieves the expected goal.

Hosseini et al. in [4] represent an approach to produce a new method for user behavioural trust evaluation that employ user's identification code and mac address to prevent the registration of members with multiple user names and makes available the information of user behavioural trust in other cloud computing environments. Also, calculating the amount of user behavioural trust assigns a negative score for repeated disruptive behaviours. This method rely on the user behaviour from the perspective of cloud provider in six measurement steps and considers the new scores of user, behaviour scores rated in the past, the scores of the other entities and scores of other cloud computing provider's to the user. Principles are presented for calculating scores and total confidence to the end user is obtained as a score. The method helps identify malicious users and negative behaviours. then compare proposed model with other models of user behaviour evaluation that use principles and show the effectiveness of model in detecting malicious behaviour and users. in our approach we suggest a measurement approach to detect TaaS effectiveness based on execution time of cloud provider to complete testing service successfully , we use cloud broker to send the mutated code to cloud provider and then Waite until execution of TaaS process terminate. The response of cloud provider will be measured, and convert into a criteria that has the ability to rank cloud provider ability to perform TaaS in efficient manner.

Wei-Tek Tsai et al. in [9] present an approach to proposes a new approach to manage services on the cloud so that it can facilitate service composition and testing .The authors uses service implementation selection to facilitate service composition similar to Googly Guice and Spring tools, and apply the group testing technique to identify the oracle, and use the established oracle to perform continuous testing for new services or compositions. The authors extends the existing concept of template based service composition and focus on testing the same workflow of service composition. in our approach we suggest a measurement approach to detect TaaS effectiveness based on execution time of cloud provider to complete testing service successfully and represent in evaluating testing as provided service through the using of mutation approach.

3. PROPOSED PROCEDURAL MODEL

In software engineering, testing technique considered as an important phase to construct faultless software system, and to produce software system free from errors. As cloud computing service provision relies in the used of distributed systems, it also relies on the use of testing technique such TaaS, which perform as the responsible of providing code testing as a service. However, cloud providers testing techniques are not idealistic enough to discover and test the requested code. Therefore, we proposed in this paper an evaluating model that takes the control of interrupting TaaS provision process for the purpose of assess the used software tests over cloud computing environment, through the employment of third party broker.Figure 1 represents the operational design of proposed model.

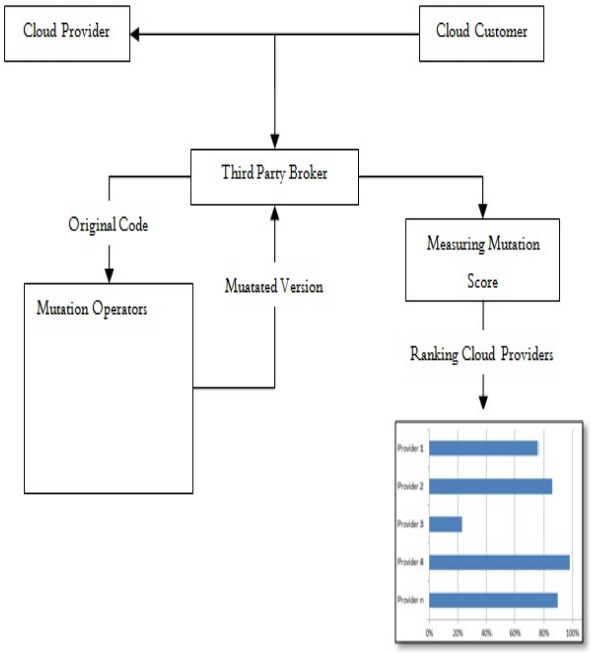

Figure 1: The procedural model

As shown in Figure 1, cloud customer as initiator of service, request TaaS to handle the issue of testing his own code.That invoke the activation of third party broker. Third party broker responsible for receiving the incoming original code (Version one of code) form cloud customer, for modifying it to generate a mutated version (Version two of code). Version two of code is the original code with a set of mutants aims to detriment whether cloud provider testing techniques is effective to use or not. In his turn, cloud broker send Version two of code to cloud provider and then Wait until execution of TaaS process terminate. The tested mutated version as a result of TaaS from cloud provider will be measured by applying mutation score role, and convert the

output of mutation score into a criteria that has the ability to rank cloud provider ability to perform TaaS in effectiveness manner. Cloud computing environment include cloud broker as permanent entity, which has the ability to interrupt service provision, for the purpose of managing what is required. Therefore, we proposed cloud broker module insertion within the process of providing TaaS over cloud computing environment, which associated directly with cloud customer and provider, the acknowledgment of initiating evaluating process of Taas, take place to reduce the expected response time between request and response of service. In more details, TaaS provision broker, interrupt the uploading process of original version of code to generate mutated version. The following process will be done in accordance to generate mutated version, measuring cloud provider testing affiance through calculating mutation score, and rank cloud providers based on measurement result.

A. Mutation Operators

Mutation testing, as fault-based code testing technique relies mainly on alter the syntactic program code for the purpose of insuring whether the used testing technique has the ability to discover (kill) what has been changed, through the use of mutation operator. TaaS typically designed as a service capable to test the uploaded code through the use of test cases, which stated to experiment the implemented code. Mutation operator as a modular of cloud broker takes the responsibility to evaluate cloud provider testing technique through generating a copy of the original code that has set of mutant. While, the original version of code kept to be compared with the result of TaaS. It should be noted that, cloud mutation operator doesn't support new test cases generation, since it contribute to improve quality of the used test suit, and that limit the ability to measure the quality of used testing technique. In the case of equivalent mutants, where is the generated mutant is the same as the original syntactic.

B. Measuring Mutation Score

Measuring of the quality relies in the used of cloud provider response, whether cloud provider discover all mutant in Version two or not. Measuring mutation score, used in the context of our research as determiner of TaaS quality level. The following equation used to calculate the amount of discovered mutant:

$$TaaSquality = (D/T) - E \quad (1)$$

-where:
D=Discovered mutant.
T=Total mutant.
E=Equivalent mutant.
Cloud broker interrupt provider responding to calculate by cloud broker, for the purpose of measuring the quality of the provided TaaS. Third party broker take the control of measuring TaaS mutation score and record it as percentage coupled with detailed description about cloud provider.

C. Mutation-based Model to Rank Cloud TaaS providers

Our main contribution is the use of mutation score as a quality indictor for cloud providers how support TaaS only. Mutation score as a percentage represent the rank of cloud providers, where the result of measuring Taas effectives specific cloud provider recoded in cloud TaaS providers table, to rearrange the rank of cloud provider according to the higher percentage of mutation score. In more details, Taas with higher ability to discover mutant ranked as the first rank, and so on. This phase aims to provide a recommendation report that detailed which cloud customer should select among cloud TaaS providers.

4. CONCLUSIONS AND FUTURE WORK

Testing as a Service (TaaS) considered as an outstanding model to provide testing facilities over cloud computing environment. Cloud customer ability to verify his own code through the use of cloud provider resources is one of the most promised techniques, that needs to monitor and measure for the quality of testing. on the other hand, making sure that the testing technique which has been achieved has high effectiveness is difficult process, because the nature of the TaaS service differs from one provider to another. In this paper, we proposed a procedural model that include a measurement approach to evaluate TaaS effectiveness based on mutation score.Within the proposed model we use mutation score measures to implement it on the outputs of the TaaS process , then convert the output of mutation score into a criteria that has the ability to rank cloud provider ability to perform TaaS in effectiveness manner.
The implementation of measuring TaaS effectiveness required cloud broker insertion as an intermediary between cloud provider and customer. Therefore, the design of TaaS evaluating model takes into consideration to insert third party responsible for interrupting cloud provider and customer service provision flow. With the end of measurement process, cloud broker will be able to rank a providers based on the mutation score result and introduced a report to cloud customers. As a future work, we intend to enhance the ranking list of cloud providers to help cloud customer to insure that they will get high level of quality services, which will be implemented through adding new measures such as: performance, efficiency, reliability, and other quality of service parameters.

5. ACKNOWLEDGMENTS

Many thanks to software engineering department professors at the Hashemite University for their valuable discussions and suggestions. Thanks also to ICC 2016 conference team for their support and help.

6. REFERENCES

[1] G. Candea, S. Bucur, and C. Zamfir. Automated software testing as a service. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 155–160. ACM, 2010.

[2] M. Firdhous, O. Ghazali, and S. Hassan. A trust computing mechanism for cloud computing with multilevel thresholding. In *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pages 457–461. IEEE, 2011.

[3] M. HAFIZ. Mutation testing tool for java.

[4] S. B. Hosseini, A. Shojaee, and N. Agheli. A new method for evaluating cloud computing user behavior trust. In *Information and Knowledge Technology*

(IKT), 2015 7th Conference on, pages 1–6. IEEE, 2015.

[5] S.-Y. Lee, D. Tang, T. Chen, and W. C.-C. Chu. A qos assurance middleware model for enterprise cloud computing. In *Computer Software and Applications Conference Workshops (COMPSACW), 2012 IEEE 36th Annual*, pages 322–327. IEEE, 2012.

[6] Y. Lou, T. Zhang, J. Yan, K. Li, Y. Jiang, H. Wang, and J. Cheng. Dynamic scheduling strategy for testing task in cloud computing. In *Computational Intelligence and Communication Networks (CICN), 2014 International Conference on*, pages 633–636. IEEE, 2014.

[7] P. Mell and T. Grance. The nist definition of cloud computing. 2011.

[8] A. J. Offutt, J. Pan, K. Tewary, and T. Zhang. An experimental evaluation of data flow and mutation testing. *Softw., Pract. Exper.*, 26(2):165–176, 1996.

[9] W.-T. Tsai, P. Zhong, J. Balasooriya, Y. Chen, X. Bai, and J. Elston. An approach for service composition and testing for cloud computing. In *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*, pages 631–636. IEEE, 2011.

[10] X. Zhang, Z. Dechen, L. Nie, T. Zhao, and X. Xiong. An optimal service-selection model based on capability and quality of resource service. In *Service Sciences (ICSS), 2014 International Conference on*, pages 47–52. IEEE, 2014.