EVALUATING THE EFFICIENCY AND SCALABILITY OF CLOUD-BASED TESTING ENVIRONMENTS FOR SOFTWARE QUALITY ASSURANCE

V. Sumathi[1] & Dr. Prasadu Peddi[2]
[1] *Ph.D. Research Scholar, Department of Computer Science,*
Shri J.J.T. University, Rajasthan, India
[2] *Professor & Research Guide, Department of Computer Science,*
Shri J.J.T. University, Rajasthan, India
***Corresponding Author :** V. Sumathi*

DOI - 10.5281/zenodo.10039341

ABSTRACT:

Testing software involves running different kinds of programmes and apps in order to locate and fix any bugs that may be present. The testing of a significant number of software products at the same time might take a very long period and be very expensive. In this work, a cloud-based distributed system for testing software is described. This system relies on cloud computing concepts. Virtualization is utilised to give a wider range of options for the design of networks and their components. The data gathered from the capture of the network traffic reveals that the amount of time needed to test a file grows in proportion to its size. It is necessary to expand the number of testers since the testing procedure required more time when just one tester and two users were utilised.

Keywords: Cloud-based testing, software quality assurance, scalability, efficiency, virtualization, test automation.

INTRODUCTION:

It is not difficult to develop Java code; nevertheless, the difficulty is in determining whether or not the code is correct and whether or not it produces the result that is desired; hence, all software scripts need to be tested.

Testing is still the most difficult obstacle in the process of developing software. The act of looking for and locating faults that occur during the execution of a programme is known as software testing. The end goal of this process is to get a programme that is free of errors and to assess the capability or usefulness of the software. Testing software is an essential component in producing high-quality software [1].

The testing of software has developed into an essential component of a variety of IT services [2].

Due to the fact that software testing is an important part of the software development process, and that significance is growing in today's world.

The term "cloud computing" refers to the sharing of resources including networks, devices, and software applications. See figure 1 for an explanation of the four primary types of cloud computing services: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), and storage as a service [3].

The concept of computing in the cloud has evolved into a computer model. Where can a virtual devices and hardware be provided, both of which have been remotely hosted, and where can a usability service model be provided when required? The term "cloud computing" refers to the capability of accessing shared resources and common search structures in order to deliver services when they are required. across the network to other activities that may accommodate varying requirements. It gives customers the ability to build, deploy, and administer their applications "on the cloud," which requires the virtualization of resources in order to maintain and realise itself [4].

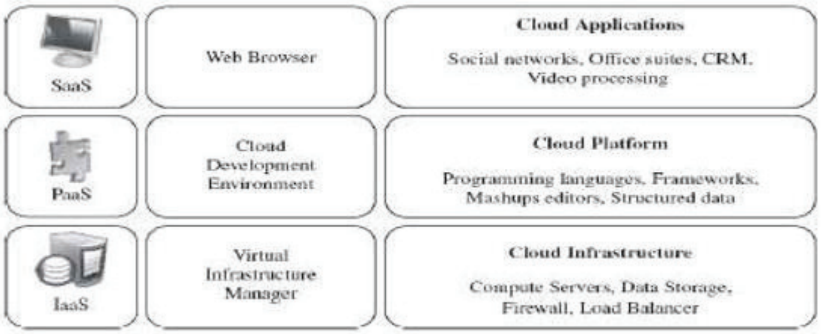
SaaS
Web Browser
Cloud Applications
Social networks, Office suites, CRM, Video processing
PaaS
Cloud Development Environment
Cloud Platform
Programming languages, Frameworks, Mashups editors, Structured data
IaaS
Virtual Infrastructure Manager
Cloud Infrastructure
Compute Servers, Data Storage, Firewall, Load Balancer

Fig. 1: Cloud Computing Service Model

Testing software before moving it to the cloud has shown to be one of the most effective methods in this transition. During the testing phase of new software, expensive server, storage, and network components are needed, but only for a brief period of time. After the testing phase is over, these computer resources are not put to use, which results in further costs being incurred. It is necessary for service providers to test their offerings on every available platform in order to assure the reliability of their offerings. When software testing is examined in relation to the cloud, the subject of test automation is brought up often. The realisation of economies of scale is one of the primary motivating factors for the widespread use of cloud computing. It is a form of service that is paid for on a per-use basis, which eliminates the need for an initial investment in many circumstances. It's possible that cloud services may be permitted to dynamically adapt and update, which calls for some modifications to be evaluated as they're being executed.

The conventional method of software testing may be replaced with an approach that is both flexible and effective if it is done in the cloud, which can minimise the amount of resources, both hardware and software, that are required.

In this article, the notion of virtualization is utilised to give a variety of options for the design of networks and their components.

Virtualization is a method that hides the physical features of computer resources from the manner in which other systems, applications, or end users interact with those resources. This may be accomplished by simulating the environment in which the resources are located.

"Virtualization" is defined as "a technique that may split or combine the resources of a computer system amongst many operating systems or applications, to create the appearance that each one accesses the actual resources" [6].

Windows Server 2008 Hyper-V was made available for purchase and download in June of 2008. The Windows Hypervisor is a 64-bit microkernel that the operating system is based on. The Windows Hypervisor operates directly on top of the underlying hardware, enabling several operating systems to operate simultaneously inside partitions, and guarantees strong isolation between the partitions by enforcing access controls for important system resources such as memory and CPUs [7].

OBJECTIVES:

The purpose of this work is to construct a cloud-based system that can evaluate computer programmes that are written in the Java programming language. The following is a list of this system's specifications:

- The platform is of the cloud-based Software as a Service kind (SaaS).
- The service that this system introduces is the testing of programmes and applications that are written using the Java programming language with regard to the correctness of the unit's writing, which is (package, class, object, function, etc.), and

the result will be as a message that includes the success or failure of the individual units.

- One or more testers may be used by the system in addition to the single server.
- The user is unaware that the server has split the task up across other testers.

LITERATURE REVIEW:

T. Vengattarman and colleagues published a paper in 2010 in which they presented a model of a cloud computing environment [8].

In the work that they published in 2010, Takayuki Banza and others posited a system that could test software in a D-cloud environment [9].

In the year 2011, Philip and Carmelo Ragusa provide in their article a software testing subject relating to cloud computing that is more inclusive and the computing environment for that [10].

Hung Youlan and Pen Zhenlong published a study in 2012 that focused on discussing the operations and methods used to evaluate software and cloud computing[11].

Nassima Aleb Samir presents a novel method to test software automatically as a service using cloud computing in the year (2012) [12].

Neha Joha and Amit Srivastava outline how to leverage a cloud computing strategy and its services to lower testing cost and install testing effort in their article that was published in 2012. Neha Joha and Amit Srivastava also authored the article.

In the year 2014, Devesh Kumar and Shivam Jain published a paper in which they presented a comprehensive analysis on testing as a service in the cloud [14].

A study was presented in 2014 by V.Priyadharshini and A. Malathi that gives an overview of trends, opportunities, problems, concerns, and requirements in cloud testing and cloud-based application[15.]

The authors Sarah Hosseini et al. offer a new framework for cloud testing in the year (2015). This new framework is based on the ISTQB standard framework and is structured according to the requirements and cloud testing steps[16].

In (2015), J.Jayashree, and others have written a paper that provides an account of an effective review of articles that have been circulated. They provide a graphic that illustrates essential commitments, patterns, fissures, possibilities, and obstacles, as well as possible research directions. they organise the many pieces of work that are being done in the field and do a review of programming testing that is done over the cloud [17].

Khushwant Virdi and others wrote a paper in 2015 that deals with the provisioning of testing for software as a service via the use of clouds [18]. In the year 2021, Akanksha Singh and others will present a paper that focuses mostly on cloud computing and the ways in which it is being utilised to test software. In addition to that, it provides a concise introduction to the several risks that are posed by cloud testing [19].

CLOUD SYSTEM FOR SOFTWARE TESTING (CSST):

Figure 2 depicts the CSST system, which is a networked-distributed system made up of three programmes that interact with each other through a network. These applications are the server app, the user app, and the tester app.

The user app runs on the user computer, which may be thought of as the user's point of entry into the system. This is also the point at which the user interacts with the application interface. The server application runs on the primary computer of the system and is responsible for managing the whole system. The tester application runs on the tester's machine, where it checks the code; the server, however, may interact with more than one user at a time and may house an endless number of testers who are testing at the same time.

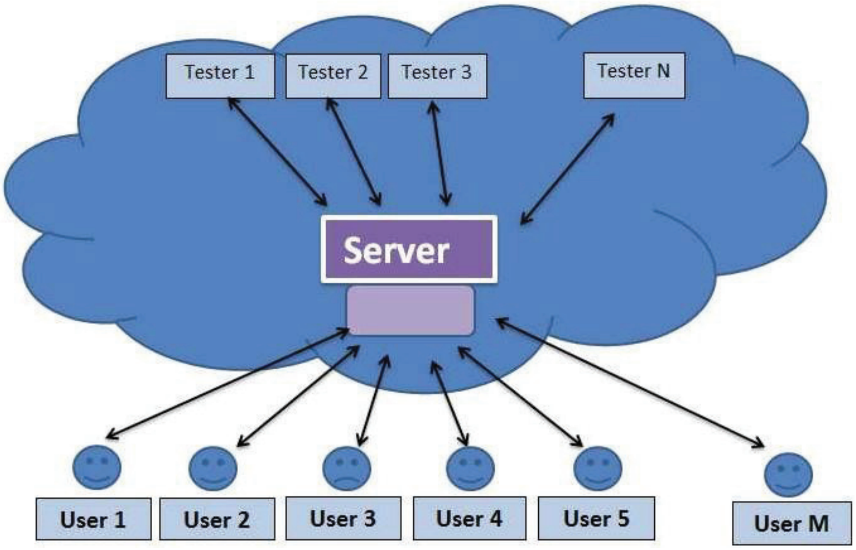

Figure 2: CSST System

Components of the system applications interfaces that have been proposed:

- User app interface: The user app interface is activated by performing a double click on the app's icon.

- Interface for the server application: the server application has been installed on the primary computer of the system.
- Interface for the tester app: the tester app runs on the tester computer and directly connects with the server; the system may have an endless number of tester computers, each of which has its own tester app.

As was mentioned earlier, the CSST system is a networked-distributed system; consequently, there are many factors that affect its operation. Some of these factors are related to the network, such as network congestions; others are related to system hardware (Server specifications, tester computer specifications); and still others are related to the user himself. All of these factors affect both the speed at which files are tested and the results that are returned to the user.

The CSST system is used in virtualized networks that are configured by Hyper-V manager. It consists of three virtual computers that work on the mother computer (depending on the mother computer specifications available), and it connects them by means of a virtual network. The server app works on the mother computer, and it turns the three other virtual computers into testing computers and/or user computers. It adopted three different situations, each of which was dependent on the opportunities provided by the network:

- One user- one tester- server.
- One user- two tester- server.
- Two user- one tester- server.

The traffic on the network is recorded with the help of the wire shark application [20], and after analysing the network, the following points were discovered:

There is a correlation between the size of the file and the amount of time required for testing. In the first scenario (one user, one tester, one server), the necessary time was 0.025 seconds for a file of size 2KB to be transferred. This time was needed when the file size was 1KB. When the file size was 3KB, the needed amount of time to submit, test, and get the result was 3.89 seconds, whereas the required amount of time was 0.89 seconds. When we submitted a file that was 1 megabyte in size, the connection was established after 4.289 seconds. This finding reveals a positive correlation in the testing time for the whole connection, and it is also quite evident to us that this was the case.

This time is impacted by the state of the connection, the network requirements, and the use of the user's machine.

The testing time was found to be longer when there was just one user and two testing machines active at the same time. This is due to the fact that the server needs some time to choose one of the testing computers from the pool. The testing took 0.93 seconds while transmitting a file that was 1 KB in size; this time is longer than the time that was taken in a specific example of sending a file of the same size but with just one user and one tester.

The time necessary to transmit a file with a size of 2 KB was 2.442 seconds, which is more than the time needed to test a file with the same size. However, in the scenario of (one user and one tester), the time it took to send a file with a size of 3 KB was 4.539 seconds.

The amount of time needed to transfer a file that is 1 megabyte in size is 5.294 seconds. This amount of time is much longer than the amount of time we need in the event that the file's size is 1 megabyte (one user and one tester). There is a correlation between the passing of time and the accumulation of more data in the file.

In this particular scenario, in addition to the various variables that effect time that we discussed before.

Table1: Comparison of the Time required to test between (one user + one tester, one user+ two testers)

File Size	One user+ one tester	One user+ two testers
1KB	0.025 second	0.938 second
2KB	0.89 second	2.442 second
3KB	3.089 second	4.539 second
1MB	4.289 second	5.294 second

In the scenario when there were two users logged in at the same time but only one tester, it seemed that network congestions and server scheduling had an impact on the amount of time needed for testing. The total time required for the simultaneous transmission of a file with a size of one kilobyte by two users was 3.235 seconds. This duration is longer than the two examples that came before it because the tester computer tests the first user file and then performs the second, based on the priority to the waiting line and resending the result to users, while it only took 4.649 seconds when a file of size 2 KB was received. Additionally, the time necessary to deliver a file that was 3 KB in size was 5.058 seconds. The instance of sending a file that was 1 megabyte in size by two users at the same time took 9.254 seconds to test, which is longer than the cases of sending the file by (one user and one tester) and sending the file by (one user and one tester) (one user and two testers). In addition, the circumstances that were discussed before had an impact on this case. The results of this comparison are shown in Table 2, which compares the amount of time needed to test with one user and one tester, one user and two testers, and two users and one tester.

Table2: Comparison of the Time required to test among (one user + one tester, one user+ two testers , two users + one tester)

File Size	One user+ one tester	One user + two testers	Two users + one tester
1KB	0.025second	0.938second	3.235 second
2KB	0.89 second	2.442 second	4.649 second
3KB	3.089 second	4.539 second	5.058 second
1MB	4.289 second	5.294 second	7.456 second

CONCLUSION:

There is a positive link between the increase in file size and the needed amount of time to test these files, as shown in table 2. However, the amount of time required to test these files is less than the amount of time required to test the file of each individual user on his own computer. This indicates that utilising the CSST system will cut down on the amount of time needed to test the files of multiple users at the same time. In the works that are yet to come, we have recommended that the CSST system be deployed over the internet and that users take advantage of the Java programming language, which has been developed to enable the CSST system to be programmed by android to play on smart phones.

REFERENCE:

[1]. ChauhanȦ Rasneet Kaur, Iqbal SinghḂ, (2014), “Latest Research and Development on Software Testing Techniques and Tools", International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161.

[2]. Jayakumar Kamala Ramasubramani , Alain Abran , (2013)," A Survey of Software

Test Estimation Techniques", Journal of Software Engineering and Applications,2013, 6, 47-52 .

[3]. Jain IShivam, Srivastava Devesh Kumar,(2014), " Testing as a Service (TaaS) on Cloud: Needs and Challenges" , International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014), Vol. 2, Issue 2, Ver. 2.

[4]. Gowri G., Amutha M.,(2014), " Cloud Computing Applications and their Testing Methodology", International Journal of Innovative Research in Computer and Communication Engineering,Vol. 2, Issue 2.

[5]. Chandane Swapnil H., Bartere Mahip M., (2013), " New Computing Paradigm: Software Testing in Cloud, Issues, Challenges and Need of Cloud Testing in today‟s World", International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 .

[6]. Sahoo J., Mohapatra S., Lath R., 2010 ,"Virtualization: A Survey On Concepts ,Taxonomy And Associated Security Issues", Second International Conference on Computer and Network Technology, pp. 222 - 226

[7]. IBM, 2005," IBM Systems Virtualization Version 2 Release 1”International Business Machines Corporation.

[8]. Vengattaraman,T.; Dhavachelvan ,P.;(2010)," A Model of Cloud Based Application Environment for Software Testing";(IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3.

[9]. Banzai ,Takayuki ; Koizumi, Hitoshi ; Kanbayashi, Ryo ;Imada ,Takayuki ; Hanawa ,Toshihiro ; Sato, Mitsuhisa ;(2010);" D-Cloud: Design of a Software Testing Environment for Reliable Distributed Systems Using Cloud Computing Technology"; ACM International Conference on Cluster, Cloud and Grid Computing;.

[10]. Robinson, Philip; Ragusa, Carmelo;(2011);" Taxonomy and Requirements Rationalization for Infrastructure in Cloud- based Software Testing" Third IEEE International Conference on Cloud Computing Technology and Science.

[11]. Yang Zhonghui ,Peng Zhenlong Ou ; Youlan, Huang;(2012);"The Application and Development of Software Testing in Cloud Computing Environment"; International Conference on Computer Science and Service System.

[12]. Kechid, Nassima Aleb Samir ;(2012) "Path Coverage Testing in the Cloud", The 2nd International Conference on

Communications and Information Technology,pp.118-122.

[13]. Johari, Neha; Srivastava, Amit;(2012); "Software Testing: An Approach using Cloud Computing"; National Conference in Intelligent Computing & Communication Quality Improvement Dept. Center of development and advance computing.

[14]. Jain , Shivam; Srivastava , Devesh Kumar ;(2014);"Testing as a Service (TaaS) on Cloud: Needs and Challenges"; International Journal of Advanced Research in Computer Science & Technology Vol. 2, Issue 2, Ver. 2.

[15]. Priyadharshini, V.; Malathi, A. ,(2013);"Survey on software testing techniques in cloud computing", International Journal of Computer Trends and Technology (IJCTT) – vol 4 Issue 8.

[16]. Hosseini Sarah, Nasiri Ramin, Shabgahi Gholamreza Latif, (2015), " A New Framework for Cloud based Application Testing", International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-3.

[17]. Jayashree J., Ivy Urbana, Vijayashreee J., (2015), " Software Testing in Cloud", International Journal of Engineering Research and General Science Volume 3, Issue 1.

[18]. Virdi Khushwant , Kalyan Radhika , Virdi Navjot Kaur, (2015), " Software Testing as a Service (STaaS) using Cloud Computing", International Journal Of Engineering And Computer Science, Volume 4, Issue 1, Page No. 10129-10235.

[19]. Singh Akanksha, Kardam Kusum, Suaib Mohammad,(2016), " A Review on Shifting Software Testing to Cloud",ACEIT Conference Proceeding 2016.

[20]. Pavithirakini S., Bandara M., Gunawardhana N., Perera S., Abeyrathne M., Dhammearatchi Dhishan, (2016), " Improve the Capabilities of Wireshark as a tool for Intrusion Detection in DOS Attacks ", International Journal of Scientific and Research Publications, Volume 6, Issue 4,www.ijsrp.org.