International Journal of Control Theory and Applications

ISSN : 0974–5572

© International Science Press

Volume 9 • Number 40 • 2016

Survey on Utilities of Software Testing in Cloud Computing

M. Maheshwari[a] Deepti Venugopal[a] Pavamana Bhat[a] and Chaman Betrabet[a]

[a]B.Tech Software Department, SRM University, Chennai, Tamil Nadu India

E-mail: deepti_venugopal@srmuniv.ac.in

Abstract: Cloud computing often referred to as "cloud" is defined as delivery of on-demand computing resources. It covers everything from applications to data centres, is on pay for use basis. This type of computing relies on sharing resources rather than employing local servers or personal devices to handle all the requirements. Cloud computing is often compared to grid computing which is type of computing where all the unused processing cycles in a network are utilized as harnesses to solve problems which is too intensive too solve by stand-alone computers. Cloud computing applies traditional supercomputing, or also high-performance computing power, which is normally used by military and research facilities, to perform tens of trillions of computations per second. In order to deploy resources in efficient manner software testing is widely practiced. It uses software testing for evaluation purposes. Organizations generally pursue load testing and regressive testing and also production service monitoring which are tested by problems like hard set deadlines, limited test budget, high maintenance cost, and large amount of test cases and geographical distribution of users around the globe. Moreover ensuring high quality service delivery and avoiding outages requires testing in data centre or outside data centre or both. Cloud testing is the solution. Effective unlimited storage, quick availability of the infrastructure with scalability, flexibility, and availability of distributing testing environment reduce execution time of large application and leads to cost effectiveness.

Keywords: *Cloud computing, Clustering, Unit testing, control flow graph.*

1. INTRODUCTION

Testing is the process of evaluating a system and its components with the sole purpose of finding whether it would satisfy all the criteria of specified and implied requirements. Testing is executing a system in order to find all the gaps, errors and missing definition of requirements in contrary to actual one. It is a process to identify the difference existing between implied needs and the implemented one. It includes defects, errors, bugs and evaluating features of software item. Its dependence is on the processes used as well as the associated stakeholders of the project. In IT industries large companies employ separate teams to perform the work of testing.

2. TESTING TEAM

The team members are assigned with responsibilities to handle separate units commonly known as unit testing. The professionals involved with testing within their expertise are as follows:

1. Software tester
2. Software developer
3. Project lead
4. Manager
5. End user

The designations are assigned on the basis of people's expertise and domain knowledge. A test case is set of conditions or some variable under which tester determines whether the applied tests check the presence of all requirements. It also helps in understanding better the suitable design for our project.

3. CLUSTERING

Clustering is a type of computing which has become a paradigm of choice for executing large scale science, engineering and commercial applications. This is due to their low cost maintenance high performance and availability of off the shelf hardware components and freely accessible software tools that can be used for developing cluster applications.

4. LOW LEVEL TESTING

Unit testing is a type of software testing which involves the preparation of well specified procedural tests of discrete functionality of a program which provides confidence that a software works as intended. Unit tests are referred to as white box tests because they are written with full knowledge of internal structure of functions and modules under test. These tests are typically developed by the testers who actually wrote the code which are commonly automated. Programmers use testing framework as j-unit for java or the test framework in ruby. The objective is not to test each test path within a unit but to focus on tests dealing with areas of high risks, uncertainty and criticality. Each test focuses on single aspect and is grouped under test suites of commonality.

5. COMPARISON AMONG TYPES OF TESTING

1. BLACK BOX
2. WHITE BOX
3. GREY BOX

Table 1
Comparisons among testing

Black-Box	*White Box*	*Grey Box*
The internal working need not be known.	Testers have full internal structure knowledge.	Tester has limited functionality knowledge.
It is also known as closed box, functional testing.	It is also known as clear box, structural code-based testing.	Also known as translucent testing.
Performed by testers, developers and end users.	Performed by testers and developers.	Performed by end users, testers and developers.
Testing is based on external expectations-internal knowledge not needed.	Testing is done on the basis of data flow diagrams and database diagrams.	Full internal knowledge is known and test designs are done accordingly.

6. TRADITIONAL VS. MODERN METHODOLOGY

Traditional Software development phases include:

1. Requirement gathering
2. Design
3. Code and build
4. Testing
5. Maintenance

This is an erroneous **methodology** because the more time you take to find errors in the later stages the more expensive it becomes to maintain. Execution costs increases exponentially.

Modern testing follows W-Model where testing is performed in parallel with traditional steps.

The new steps performed are:

1. Requirement gathering -testing
2. Design- testing
3. Code and build- testing
4. Installation-testing
5. Maintenance-testing
6. Deployment-review

In the requirement phase checking is done to ensure whether the gathered requirements fulfil The business vision intended. Testing of design ensures compatibility of design with original Plan. This phase also covers the possibility of generating rough functional data. Installation testing checks compatibility with different platforms. Conclusively in the maintenance stage when test fixes are made they are sent for re-testing and This is called regression testing. Comparison of automation and manual testing forms the pillar of software testing as these are utilized by developers to develop best quality software.

7. MANUAL VS. AUTOMATED

Table 2
Differences between manual and automated testing

Manual Testing	*Automated Testing*
Process done in manual form.	Process performed using automation tools
It is the first step without which automation can't be performed.	It is continuous form performed after Manual testing.
Testers perform random testing to find bugs.	Testing performed through running scripts.
Error guessing finds more bugs	Functionalities are repeatedly tested.
Regression testing is tougher.	Regression testing relatively easier.
Manual labour increases.	Manual labour decreases.

8. TESTING ALGORITHM

Random testing algorithm

Test Case generated using Random-Testing: Random test case generation is a method where all the test cases being generated are not based on any algorithm but are based on the assumptions made about the application being used. The following classes are going to be tested and numerous test inputs will be supplied to check recurring faults. The framework being utilized for validating is known as Auto-Test and using this prediction can be made about all the issues detected and undetected. The Auto-Test framework classifies all of the test cases in following categories:

1. Pass -no exception
2. Unresolved- precondition violation in method under test
3. Failed

A sample for method generating testing condition is created in a following manner. At each step selection of method takes place then its execution is determined.

```
CreateTest (time_out):
From
Initialisation _of_ pool
Till timeout
Start loop a: = choose (method to test ())
CreateTest for method (a)
end;
```

1. Method Initialisation _of_ pool creates empty pool of objects which needs to be tested.
2. The Method: method to test returns all sets of methods which are being tested.
3. The non-deterministic method selects a random element from sets or lists.
4. The method CreateTest creates a call for method a.

8.1. Data flow Graph

Control dependence is explained in terms of relation existing between different nodes. Here nodes represent statements and the edges between them the control flow existing between them. To ease the analysis entry and exit points are provided in the graph.

EXAMPLE :

```
int a ,b ,c;
read a ,b ,c;
if(a<b)
 if(b<c)
a = c;
c = a;
endif;
endif;
print a ,b ,c;
```

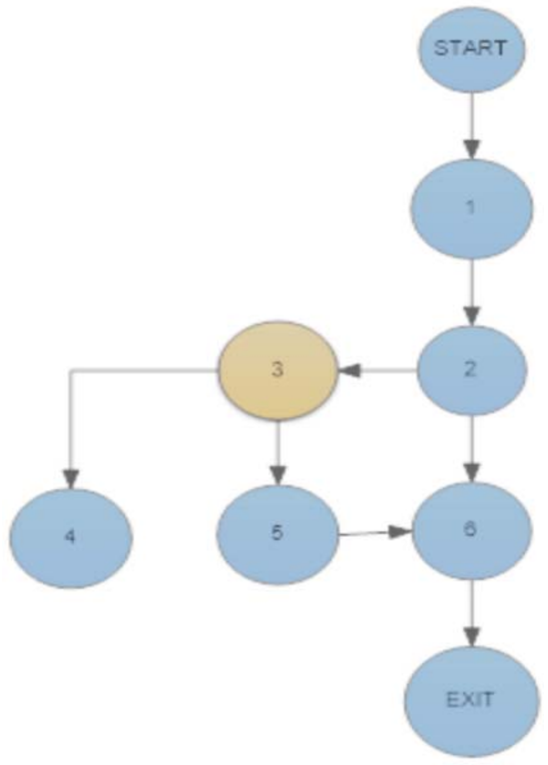

Figure 1: Control flow graph of sample program

8.2. Genetic Algorithm

The algorithm evaluates the data to be tested by execution using test data as input. And the predicates used in the program are recorded. This list of predicates is then compared with the control flow list of predicates generated using decision making process with nodes and edges.

SAMPLE ALGORITHM:

Algorithm: generate data

Input: Instrumented version of program to be tested

1. CDG: control flow graph
2. Initial population of test cases
3. List of test Requirements

Output: final list of test cases and requirements

Declare: cdg paths-> A set of recorded paths

Scoreboards-> Record of test requirements

Current Population, new Population->Se of Test Cases Target-> Test Requirement for which test case is generated. MaxAttempts-> Returns true when all the maximum attempts allowed has been executed. Out of time-> Returns true when time limit is reached.

In the above flow graph for generate data Aristotle Analysis is being used to generate a program map a control flow and instrumented version of the program. Then compilation of instrumented program takes place followed by storing of executable file. There are four major components:

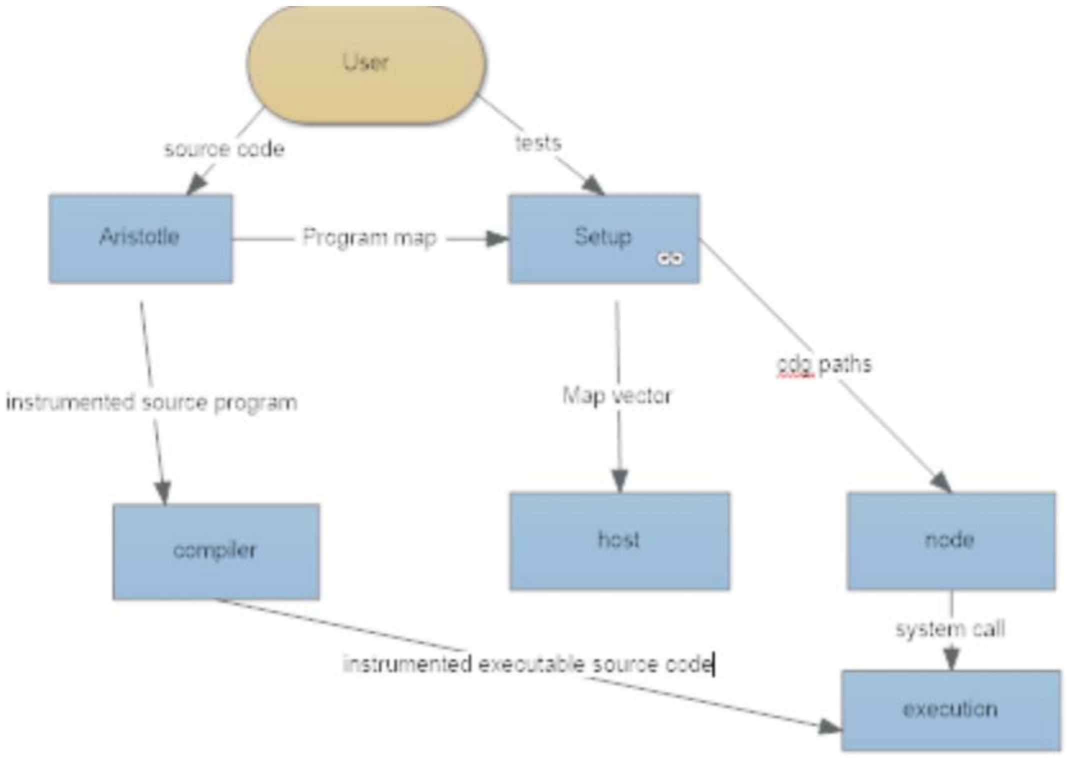

Figure 2: Case Study for the generate Data program mentioned above in a data flow diagram

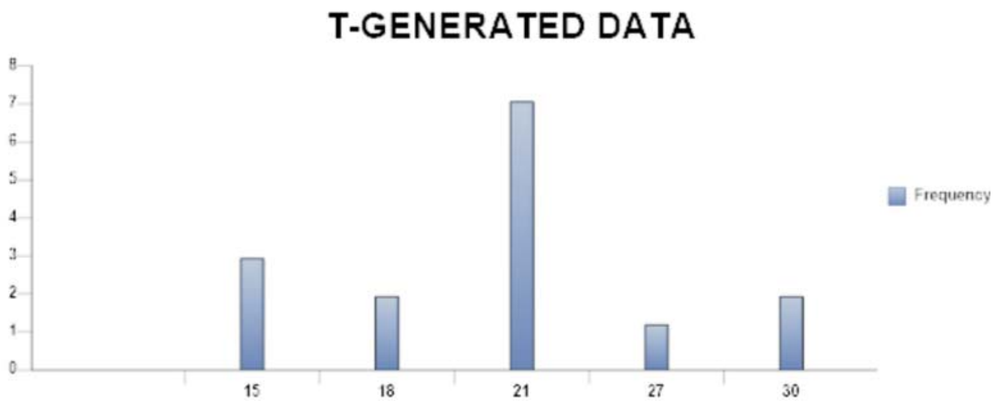

Figure 3: Statistical analysis of t-gen data

Setup: It is responsible for initializing data

Host: It is responsible for dispatching data

Node: It implements algorithm to generate test cases

Execution: Responsible for program execution using nodes' test cases.

In the above flow graph for generate data Aristotle Analysis is being used to generate a program map a control flow and instrumented version of the program. Then compilation of instrumented program takes place followed by storing of executable file. There are four major components:

Setup: It is responsible for initializing data

Host: It is responsible for dispatching data

Node: It implements algorithm to generate test cases

Execution: Responsible for program execution using nodes' test cases

X axis represent no of iterations

Y axis represents frequencies of generated data

9. NEW EMERGING TESTING TECHNIQUES

Penetration Techniques: It is a type of testing performed to check the vulnerabilities in a software product. Tests are conducted to find the weak spots in the software system. The basic intention of using penetration testing is to find the security breaches. If found then certain measures are decided by testers to rectify the issues and find solutions to tackle with the vulnerabilities. These vulnerabilities usually can be found in operating systems, services and applications flaws and also risky end user behaviour. This is done to check the user's adherence to security policies. There are two types of pen tests: Dynamic as well as static. These tools determine whether sufficient encryption has been performed on software pieces and also finds the presence of backdoors installed by hard core coders. Veracode is a tool that employs penetration testing which uses binary scanning approach for best results. As it returns less positive results testers can focus on remediating faults rather than sifting through non threats. These tests alert the companies about the existing flaws in their system.

PERFORMED FOR:

Websites networks servers.

10.1. Penetration tools

10.1.1. Metasploit

It is a framework used for conducting pen test. It is based on concept of exploit which includes code that can surpass some of the security measures and enter a secured system. If entered it runs a payload which is a code that performs operations on target machine. It has a command line and GUI clickable interface. It is compatible with Linux, OS and windows. It is a commercial product.

10.1.2. Wireshark

It is a protocol analyzer which is popular for providing minute details about network protocols, packet information and decryption methods. This is also compatible with Linux, OS and windows.

The information retrieved from this tool can be viewed using GUI or through TTY-mode.

10.1.3. Core impact

Core impact pro can be used for testing mobile application, network device penetration, password identification and cracking etc. It also has command line and GUI interface which works with windows. This is one of the expensive tools in the it industry.

10.2. Software testing tools

10.2.1. Open source tools

TET

Test management tools: The goal behind creating the test environment toolkit was to produce a working test driver that accommodated the required and anticipated future regarding testing needs of developing community. To achieve this goal input was taken from the wide sample of community was utilized for specification and development of TET'S functionality and interfaces.

10.2.2. Functional testing tools

Selenium : Selenium is the testing framework being used to perform testing of web application across numerous browsers and platforms like windows, Mac, Linux. It also assists testers to write tests in different programming languages like Java, PHP, C#, Python, Ruby and Perl. It provides playback and record features to write the tests without the need to learn Selenium IDE. Selenium supports some of the largest well-known browser vendors who make sure that they have Selenium as a native part of their browser. It becomes the base for most of the software testing tools.

Testing Whiz: It is a test automation tool which has code-less scripting by Cygnet InfoTech which is a CMMI level 3 IT solution providers. The enterprise edition's tools offer a complete package of various automated testing solutions:

1. Web Testing
2. Software Testing
3. Database Testing
4. API Testing
5. Mobile app Testing
6. Regression test suite maintenance
7. Optimization
8. Automation
9. Cross browser Testing

It offers various features like:

1. Key word driven, distributed testing
2. Record and playback framework automation
3. Object internal recorder
4. 290+ inbuilt JavaScript and testing commands
5. Integration with tools like Jira, Mantis and Fogs Bugz which tracks bugs.
6. Integration with test management tools like HP Quality Centre
7. Risk based testing
8. Continuous Delivery

10.3. Other pent tools

10.3.1. Watir

It is an open source testing tool made up of ruby libraries for automating web applications. Features provided are:

1. Testing web based language applications.
2. Cross browser testing
3. Compatible with tools like Rspec cucumber
4. Also tests web pages, buttons, forms links and their responses.

10.3.2. WatiN

1. It is an open source C# developed application testing tool that was inspired from Watir.
2. It supports web testing for .Net programming languages and is licensed under apache 2.0
3. It's features are:
4. Supports html and AJAX web testing
5. Integration with unit tools
6. Automation testing with IE and Firefox
7. Screenshot generation
8. Native support for page and control model

10.3.4. Sahi

It is a testing automation tool to automate web app testing. This open source is compiled in java and JavaScript. Features are as follows:

1. Performs multi browser testing
2. Supports extJS, zk, dojo, YUI framework
3. Record and playback testing.

10.3.5. Soap UI

In USA the technology being used is SOAP UI which works in agile framework. It is the world's most widely used open source API testing tool for SOAP and REST API'S. It offers SOA web testing tools with services for functional testing, REST API testing WSDL overage along with message assertion testing and test refactoring with an experience of more than ten years. Soap UI is the de facto method which ensures quality while developing APIs and web services. When it was initially created in 2006 there were no open source tools in the market. The initial idea behind Soap UI was "let's get lot of people for helping." Developers are contributing in code and also providing feedback making it the product which is popular in the market. Because it is an open source it's usage and adoption spread quickly as it was launched. REST API has benefitted from the large number of testers and developers who tried the product and resulted in spreading awareness of its functionality. While the open source version can be seen as Swiss army knife for testing Soap UI pro is the tool with the sharpest edge. It is applied to the functional testing area Soap pro focuses on enhancing efficiency and usability. With Point and Click Testing, we can drag and drop instead of manually writing the code. The Form Editor creates a form from our request and further eliminating the need for us to spend time on repetitive coding. These functions in combination with The Outline Editor who simplifies and exposes the XML structure, makes our testing more fun and less time consuming. If creativity, flow and speed are important to users as testers, Soap UI Pro is too.

10. ANALYSIS AND CONCLUSION

It has been observed that different companies with operational models which are 90% based on cloud services, where the rest of the 10% is constituted of in-house servers. The basic response after enquiring about security issues relating to cloud services was that cloud service provider will be taking care of it and they will not be concerned about it. This isn't necessarily the case with every cloud service provider, since some of the CSPs have a good security model while others clearly won't. Some of the vulnerabilities experienced by the developers are:

Session Riding: It happens when an attacker steals a user's cookie to use the application in the name of the user. An attacker might also use CSRF attacks in order to trick the users into sending authenticated requests to randomized web sites to achieve various goals.

Virtual Machine Escape: In virtual environments, the physical servers usually run multiple virtual machines on top of hypervisors. An attacker can exploit a hypervisor remotely by using a vulnerability present in the hypervisor. Such vulnerabilities are quite rare, but they do exist. Additionally, a virtual machine can escape from the virtualized sandbox environment and resulting in gaining access to the hypervisor and consequentially all the virtual machines which run on it. So in order to decrease the risks generation increasing in the system, due to vulnerabilities

VAPT is preffered.Vulnerability Assessment and Penetration Testing (VAPT) are two different types of vulnerability testing. Both tests have different strengths and are often combined for achieving more complete vulnerability analysis. Penetration Testing and Vulnerability Assessments performs different types of tasks, usually with different results, within the same areas of interest. Vulnerability assessment tools are used to discover which vulnerabilities are generally present, but they don't differentiate between flaws which can be exploited to cause damage and those which cannot. Vulnerability scanners alert companies to the preexisting flaws in their code and their locations.

Penetration tests attempt to exploit the vulnerabilities in a system for determining whether unauthorized access or other malicious activity is possible and try to identify which flaws pose a threat to the application. Penetration tests find exploitable flaws and measure their severity. A penetration test is meant to show how damaging a flaw could be in a real attack rather than finding every flaw in a system. Both penetration testing and vulnerability assessment tools provide a comprehensive picture regarding the flaws that exist in an application and the risks that are associated with those flaws.

Vulnerability Assessment and Penetration Testing (VAPT) provides all enterprises with more comprehensive application evaluation than any stand alone test can provide. Using the Vulnerability Assessment and Penetration Testing (VAPT) approach gives an organization a more detailed view of the threats being faced by it's applications, enabling the business to protect its system's and data from malicious attacks in a better way. Vulnerabilities can be found in applications from third-party vendors and software made internally but most of these flaws are easily fixed once found. Using a VAPT provider enables IT related security teams to focus on mitigating all the critical vulnerabilities while VAPT provider continues to discover and classify vulnerabilities.

REFERENCES

[1] Roy.R.Pargas, Mary Jean Harold and Robert.R.Peck "Test data Generation using genetic Algorithm", Journal of Software Testing and Verification, 1999

[2] Karambir, Kuldeep Kaur "International Journal of advanced Research in Computer Science and Software engineering", ijarcsse, June 2013

[3] Akalanka.Mailewa and Jayantha Herath"A Survey of effective and efficient software testing", Micsymposium

[4] Kung, David and Hong Zhu"Software verification and validation", Wiley Encyclopaedia of computer science and engineering, 2008

[5] Amman Paul and Jeff Offut"Introduction to Software Testing", Cambridge University Press, 2008

[6] Base36"Automated vs. Manual Testing: Pros and Cons of Each", NP.2013, WEB 21 March 2015

[7] Amman Paul and Jeff Offut"Introduction to Software Testing", Cambridge University Press, 2008

[8] M.I.P.Salas and E.Martins "Security testing methodology for vulnerability system", Science direct, 2015

[9] Abdul Razzaq, Faruq Ahmed, Ali Hur "Semantic Security against Web application attack", Elsevier.com, 2014

[10] Jai Narayan Goel and B.M.Mehtre"Vulnerability Assessment and Penetration Testing as a cyber defence Technology", 2015.