TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

KHUSBOO SHAH

SURVEY ON CLOUD BASED TESTING TOOLS

Master of Science Thesis

Examiner: Professor Hannu-Matti Järvinen
Instructor: Matti Vuori
Examiner and topic approved by the Council of the Faculty of Computing and Electrical Engineer on 9 October 2013

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Information Technology
SHAH, KHUSBOO: Survey on cloud based testing tools
Master of Science Thesis, 59 pages
February 2014
Major: Software Systems
Examiner: Professor Hannu-Matti Järvinen
Keywords: cloud computing, cloud testing, service provider, testing, software testing, testing tools, test automation

Cloud computing has been adopted by many business organizations due to its simplicity and innovative technology. It has changed the way software used to be developed and delivered to customers. Every company aims to develop high quality software, with reduced cost and complexity. Cloud computing provides a platform for software testing where any application can be tested in cloud environment. The main goal of this thesis was to provide an understanding about cloud testing methodologies and explore cloud service providers offering various testing tools and technologies for different types of testing (functional and non-functional testing).

This thesis incorporates description about cloud computing, its benefits, and how it has been helping the organizations with different service models. Taking cloud computing as the base of understanding, cloud testing methodologies and its architecture are explained. In addition, the cloud service providers along with the tools they provide for testing were explored and some of their features were discussed. The pros and cons were analysed in order to find whether cloud-based testing tools were suitable for the organizational environment and its differences were compared with the traditional approach of testing. The research for this thesis was carried out via companies (service providers) websites, blogs, articles, journals and other papers. Communication mediums such as chat, email and phone were also used for the research on tools.

The study indicates that cloud testing is often used as a performance testing and is mostly implemented using software as a service model. In addition, organizations can be benefited by embracing cloud testing methodologies, if they know how to exploit it properly. The results of this study suggest that though cloud testing is attracting more businesses with its promise of minimal maintenance and low costs, it can sometimes be risky. Cloud testing provides more choices for the organizations regarding how to run the infrastructures, save costs and time and delegate liabilities to third-party providers. On the other hand, it can also be challenging in terms of security, performance and other issues. This thesis also analyses things to be considered before moving to the cloud system.

PREFACE

Professor Hannu-Matti Järvinen provided me with an opportunity to write a master's thesis giving me a choice between three different topics. Among them, I chose "survey on cloud based testing tools" as my research topic because cloud testing is rapidly growing in the IT industry and remains a hot topic for research. Another reason is, since I had worked as a software test engineer earlier, I wanted to grow and explore more in this field. This topic has always been my subject of interest and to have done research on this has deepened my understanding about it.

This thesis was examined by Professor Hannu-Matti Järvinen and my instructor Matti Vuori from the department of pervasive computing. I would like to convey my gratitude to both of them for their guidance and continuous feedback regarding my report writing, which further motivated me on the topic and completing my thesis on time.

I would also like to thank my friends and family who have encouraged and supported me throughout this stay in abroad. Without their support, I would not have been able to complete my studies on time.

Tampere, February 2014

Khusboo Shah

TABLE OF CONTENTS

TERMS AND DEFINITIONS

COM	Component Object Model (COM) is a software architecture used to build component based applications.
CSS	Cascading Style Sheets (CSS) is a style sheet language that defines display and formatting of a document.
DOM	Document Object Model (DOM) defines the logical structure of documents and the way a document is accessed and manipulated.
EC2	Elastic Compute Cloud (EC2) is a hosting service provided by Amazon Web Services. Its web service interface allows us to run our computing resources on Amazon's computing resources.
GUI	Graphical User Interface (GUI) is a program interface that that allows a visual way of interacting with the items such as windows, icons and menus.
HTTPS	Hypertext Transfer Protocol Secure (HTTPS) is a protocol for secure communication over internet.
IDE	Integrated Development Environment (IDE) is a software application that provides programming, debugging and testing environment for software developers.
SDLC	Software Development Life Cycle (SDLC) is a term used in software engineering process such as planning, analysis, implementation, design and maintenance.
SOA	Service-Oriented Architecture (SOA) is a software architecture pattern that provides application as services to other applications.
SSH	Secure Shell (SSH) is a protocol used for secure data communications.
SSL	Secure Socket Layer (SSL) is a protocol for providing the secure transmission of message over the internet.
S3	Simple Storage Service (S3) is online data storage offered by Amazon Web Services.

1. INTRODUCTION

The popularity of cloud computing for businesses is undeniable. This growing popularity has led many organizations to move in the cloud for better efficiency. Section 1.1 in this chapter introduces a small background on how cloud computing got popular and its use to testing. Section 1.2 describes about the problem of the thesis and why it has been important for the organizations to migrate in cloud. Sections 1.3 discuss the objectives of the thesis. In addition, Section 1.4 describes the research procedure and Section 1.5 discusses in short how the thesis is organized.

1.1 Background

During the recent years, cloud computing has gained tremendous attention in the industry. The cloud is now delivering the computing tools and resources in the form of many different kinds of services, from individual applications to overall computing infrastructures. The exact date of when cloud computing was coined is still unknown, but it got popular in the year of 2006 when Amazon introduced Elastic Compute Cloud (EC2) which provided the services for renting virtual machines. This service provider became popular in computing environment from mid-size to large companies providing numerous benefits such as cost efficiency, data storage, backups, updates and recovery. This trend is more likely to grow in upcoming years.

Considering this technology along with its benefits, a new concept of testing has been evolved that is, cloud testing. Cloud computing gives a platform for cloud testing but what exactly is cloud testing? Many people still have the confusion what exactly cloud testing is: "Is it testing with the help of something that is in the cloud, or simply testing something that is in the cloud?" (Shajee, 2012). It isn't much complicated to understand after having the clear concept of cloud computing and its service models. Cloud testing is simply making the use of cloud infrastructure to run and manage the test. This technology provides many benefits and mitigates all the problems faced in the traditional approach of testing.

According to Riungu et al. (2010) service-oriented architecture (SOA) and software-as-a-service (SaaS) have immensely affected the software industry. Every company's goal would be to produce high quality software which is flexible and easy to use. This evolving nature of the technology has made organizations think about the new concepts, methods and tools on testing (Riungu et al. 2010). Riungu et al. (2010) points out that software testing as an online service provides on-demand service with a daily operation of maintenance and support through web browsers. With this upgrade and maintenance provided by the cloud vendor, it will definitely improve tester's workload. Gartner (2010) had predicted that, 20 percent of the business would have no ownership in IT asset by 2012, and would acquire the IT assets from the cloud. The exact percentage is

unknown, but it is true that many companies have already started to use the cloud for the purpose of testing.

As the essence, cloud means providing services; it obviously reduces the effort of an organization. Testing with the help of cloud avoids the purchase of expensive hardware and software and allows faster scaling and with a considerable price. It reduces the licensing and capital expense by 50-75% using the virtualized resources. It also shortens development and testing setup times (Chittanai, 2012). It also helps to improve the developers and testers' communication (Riungu-Kalliosaari et al. 2012). Moreover, configuring the test environment becomes simple allowing faster test execution and with increased number of test sessions within the same period (Forno, 2012). Therefore, this approach promotes the faster development of the product improving the quality and productivity.

1.2 Statement of the problem

The rapid growth in the field of cloud computing has motivated the organizations to focus on their core business rather worrying about the set up and maintenance infrastructure. It has opened many new opportunities in the field of software testing. Cloud computing is changing the way of delivering the software (Riungu-Kalliosaari et al. 2012). Traditionally, testing has required a large amount of computing resources (servers, storage data, networks etc.) to test the application. It was also difficult to cover all of the test scenarios from different geographic locations. Nevertheless, these resources (hardware and software cost along with its maintenance) are expensive and consume a lot of time and effort in delivering the product. Therefore, it is necessary to understand how cloud can be properly utilized in the case of software testing as companies still have a very vague understanding how the cloud could be utilized.

Every organization would like to keep pace with the new and revolutionizing trends of IT. There is always a huge competition in every aspect of industry (for example, in sales, new technologies, designs, etc.) and because of this organization should be aware of all the latest technologies and tools that are flexible and scalable to save time, reduce cost and improve the way organizations work. Cloud testing helps in setting up the infrastructure, reduces the cost, and provides on-demand services with broader network access. Moreover, it also reduces the time in the software life cycle which is the most important aspect to be considered in the IT industry. According to Riungu-Kalliosaari et al. (2012), many organizations have already started to provide cloud based testing services such as Soasta, Keynote and BlazeMeter. At the same time, some of the organizations are still not willing to take this challenge, because of the lack of understanding about the general possibilities or the practicalities starting to utilize the cloud.

The focus of this research is to provide a better understanding about the cloud testing tools and its methodologies. It also portrays how the traditional approach of testing is different from the modern approach of testing. There are many papers and articles

discussing about cloud computing, but considering this as a platform to software testing there is very little information available.

1.3 Purpose of the study

Because of the advances in cloud computing and the rising demands for software testing as an online service, many organizations have shifted their resources in the cloud (Riungu et al. 2010). Considering this new environment together with the demand for testing as an online service, this study aims to create a better understanding about cloud testing methods and tools for different types of testing conducted in the cloud. The objectives of this study are:

- Literature review of scientific and industrial documents for the understanding about cloud testing tools and its methodologies.
- Assessment of cloud service vendors testing related service offerings and their related material.
- Analysis of the findings and the overall situation.

1.4 Research methodology

The information for this thesis was basically collected from the research papers such as journals, conference papers and other papers. Since cloud testing has not been so popular yet, the research in this area is scarce. For the other part of the thesis, research was carried out via the companies' websites, blogs, articles etc. Companies' websites here refer to the service providers, who provide the cloud testing tools for various kinds of functional and non-functional testing. Communication mediums such as chat, email and phone were used for the finding of the tools.

1.5 Organization of the thesis

Organization of this thesis work is broadly divided into seven chapters. In chapter one, background of the thesis study is introduced followed by problems, objectives and research methodology. In chapter two, studies of modern testing methods were carried out with a comparison to the traditional testing approach. In chapter three, cloud computing was introduced along with its types and service models. In chapter four, cloud testing was introduced followed by test architecture, test management and test environments in cloud. In chapter five, different types of testing tools with some details were presented for various testing types. In chapter six, pros and cons of the research were discussed with analysis. Finally, in chapter seven the research work was concluded by explaining the implications of the study and some findings.

2. A VIEW INTO MODERN WAY OF TESTING

Software testing is an activity that aims at providing quality related information about the system, such as does it perform and behave as required, does it really meet the customer's real needs, is it as reliable, secure and efficient as expected, what kind of defects does it have and how the system could be used with their effects. It has become one of the challenging activities for many software engineering projects and is one of the five main technical activity areas of the software engineering lifecycle (Parveen & Tilley, 2010). This chapter includes the discussion about the traditional approach of testing and its drawbacks to the organizations. It also discusses the quality needs of the modern testing and common processes of testing that are followed in the organizations.

2.1 Quality needs of modern systems

Today, if the users find the application that performs better for them or their business, they will take no time shifting to a different platform that is better in regard, be it performance, reliability, security or any other quality factor. So, the quality of an application is one important factor that matters while delivering the software. These days with the growing complexity of business applications, it has been difficult for the organizations to build and maintain quality software, covering the real-world testing scenarios. To meet the modern systems' quality needs, the application must meet the requirements like better functionality, security, performance and compatibility. An application should be able to handle the sporadic traffic and heavy load from large amount of user's. Generally, in traditional testing, the applications are only tested inside the organization's firewall in test environments which is limited and might fail to meet the performance issues.

Modern tests system has demands for realistic simulation of the production environment and this test cannot be accomplished from a single location. We need different tools to test multiple scenarios providing analysis, monitoring, scheduling and reporting of the applications. This approach of testing requires being scalable and easy to use so that test engineers can start the test early and deliver results as quickly as possible. It becomes more complex in a large system, when regression testing is carried out consisting of hundreds to several thousands of test cases. In this case, even the high configuration machines might take a long time to execute. This might not be feasible where test needs to be repeated (Parveen & Tilley, 2010). It needs considerable resources that might not be readily available. For example, an application might need to be tested with multiple scenarios like different operating systems, several database clients, multiple

browsers and server interactions (Parveen & Tilley, 2010). Running these test scenarios every time requires management of different infrastructure. So, there should be the proper management of the resource that reduces the time and helps in testing efficiently. Therefore, there are so many things to test for the improvement of its quality that require good environments and many different tools. Nevertheless, all these are ultimately a concern of cost which might be too expensive for an organization.

2.2 Traditional approach of testing

Traditional testing approach has always been very complicated and expensive. It had to be conducted in internal test environments where the infrastructures are inefficient to meet the modern requirement of software. Traditional approach requires a dedicated infrastructure of sufficient hardware storage, adequate bandwidth for realistic simulation, skilled professionals to run, monitor, and analyze the results of the test. It would also require good tools that enable the real-time monitoring of the servers and applications under test. As shown in **Figure 2.1**, this is a system centric model, where the business needs are identified first. The project team members request the infrastructure from the IT infrastructure team and they procure and provision it (Doddavula & Gawande, 2009). This process might take a longer time of about 1-2 months for specifying, procuring, configuring and deploying the hardware. This might also include deploying and configuring the software (Middleware/DB) and the applications settings. The application is then tested and deployed on that infrastructure.

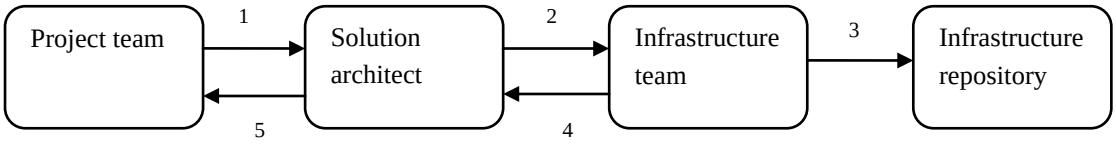


Figure 2.1. *This is a sample how traditional approach of testing is carried out. (1). The project team proposes the requirements to the solution architect (2). Solution architect tries to resolve the plan with infrastructure team (3). Infrastructure team now provides the infrastructure (4). Infrastructure team assigns the request to the solution architect (5). Finally the solution architect fulfils the requirement of the project team.*

Testing in this approach has many challenges to be faced in different aspects such as business, agile business, costs and speed. The testing of a project needs to be started in an early phase. In this testing approach, organizations might have to spend "months" building the test environments and information systems. The main task of the tester should be to focus on the quality of the product rather worrying about the environment setup. Another most important problem is companies' lack of funds. Initially, companies need to be able to start in small and add as needed. They cannot invest in environments and tools that might only be used once. This testing also affects business as many organizations are moving towards agile software delivery. Traditional test tools struggle to work in the agile environment. Tools and technologies change with time so companies

should also follow the latest tools and the new approaches of delivery if it is helpful for their business. In this model, organizations have to invest a lot of time and effort creating the infrastructure. Moreover, the infrastructure and resources might be too expensive for an organization to handle. Therefore, they might have to think more than once before buying and maintaining the resources. Additionally, the infrastructures might not be utilized all the time, it might only be utilized when the project is ongoing or when there is a need for testing. Otherwise, it will stay idle. The continuous effort needed in this approach of testing might also reduce the agility of an organization in creating the new business (Doddavula & Gawande, 2009). Above all, the traditional approach of testing provides in-house testing facilities that can only mimic the real-life environments.

2.3 Common practices for testing processes

According to IEEE definition, software testing is the process of analyzing a software item to detect the differences between existing and required conditions (that is, bugs) and to evaluate the features of the software item (Testing, 2014). It is also defined as the process of detecting the bugs in the software application by verifying that the software behaves as specified. Software testing identifies important defects, flaws, or errors in the application that must be fixed. This identification reduces the errors and maintenance effort that ultimately reduces the overall cost. Software testing also validates the software that is; it checks the system correctness by assuring that the specification meets the criteria of the user (Trivedi, 2012). In common practices, mostly testing are performed at the end of the software life cycle, but the more it is delayed, more it turns out to be expensive, difficult and problematic.

The complexities of the software projects are increasing day by day. Consequently, it is the matter of quality, functionality and reliability of the software. The test methods and tools for this are also changing rapidly. The common testing is a process that normally starts with a test planning, designing the test cases if necessary, preparing for the execution of individual tests (for example, functionality testing, compatibility testing, installation testing etc.) and evaluating the test until it is finalized as shown is **Figure 2.2**. This process also includes building the test environments for different types of testing which consists of servers with high configurations, multiple operating environments, networks, bandwidth, configuring tools etc. Building this environment might take a longer time than planned as discussed earlier in Section 2.1.

The testing processes in general also include different software development life cycle (SDLC) models such as waterfall model, agile model, top-down and bottom-up approach. Waterfall model is the earliest SDLC model used in software engineering process. The whole process is divided into separate phases where one phase acts as the input to the next phase sequentially. It consists of the phases of requirements, design, coding and test, system integration, and operation and maintenance. This model has been with the industry for over 30 years (Waterfall, 2013b). Despite being used for such a long

period, it has some disadvantages like difficultly of going back to other phase in case of change required, deliverables based on documented requirements and no prototype for a customer in the middle of the project (Waterfall, 2013a).

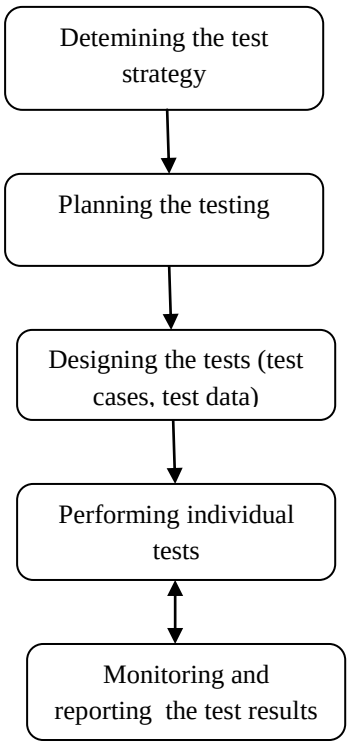


Figure 2.2. *Flow representing common practices for testing a process. An approach to software quality is selected initially. The goals and objectives of the tests are identified with the proper test approach. Test cases, test data and environment set ups are designed for the different testing types if required. Individual tests are executed and logged. If the test gets failed, then it is re-executed and the results are logged again.*

Agile methodology proposes an alternative to the traditional model. It is an iterative and incremental approach of development that provides rapid delivery of an application. It works with sprints, that is, having a defined duration for task deliverables (usually considered in weeks). This methodology provides the access to the project phases, throughout the development of lifecycle (Agile, 2013). These days agile methodology is mostly followed by the organizations but it highly depends upon the organizations which models they follow. According to Trivedi (2012), software testing lifecycle consists of five phases: requirement study, test case design and development, test execution, test closure and test process analysis.

Testers should be clear about the requirements before they start the testing. Test planning involves defining the scope of the test with some test strategy. It defines the milestone, schedules the task and defines the number of resources involved during the test process and employing them if necessary. It also determines the coverage criteria that is, the test scenarios needed to be tested, and the exit criteria. Test analysis and de-

sign refer to reviewing the test basis (understanding the base documents such as requirements, design documents in order to start the test analysis and creating the test cases), identifying the test conditions, designing the test cases and defining its procedure. It also includes setting up the test environment and purchasing the tools needed for testing. This process might take longer time than planned as many things need to be considered as cost, speed, time etc. Once the test design is ready then the individual test execution of different testing types is performed such as functional test, compatibility test, performance test and system test. The test cases are executed by following the test procedures and a proper log is maintained to track pass or fail of the test case. The test cases might have to be re-executed to verify defect corrections. Generally for every test, a test criterion is set which determines the exit criteria specified in the test planning. After the delivery of the software, the testing is closed with some activities such as evaluation of the test, lessons learnt for future testing, storing the test locations in a document for later use and final reporting (Trivedi, 2012).

2.4 Pressure and drivers on development of testing

Developing software with excellent quality is gaining more attention these days and is the most complex task of all. There are numerous factors affecting the quality of testing. That includes the budget of testing, speed, extensiveness and effectiveness of tools, support from management, skills of the tester, motivation of the tester etc. (Perry & Rice, 2013).

Many organizations do not want to spend large amount of budget for the testing tools or hiring new people for testing purpose. The problem for this might be because of very limited budget. However, the organizations should be aware that the skill of the testers has major effects in the testing process. Testers should possess a certain level of professional skill. If not, they should be trained for the better output. Testing is not an art; it should rather be followed with some engineering principles. As the rapid business of technology keeps on changing with each generation, the testing team should be updated with all the new strategies of testing, tools, methodologies and techniques applicable to software testing life cycle. Speed in implementing the system is also one of the important individual issues to be considered. It highly depends upon the organizations, which testing methodologies and tools they follow for the development of testing. If they follow the traditional approach of testing (as discussed in Section 2.2), it will obviously take a long time in testing as well as delivering the product. So, appropriate tools with proper test methodologies should be adopted by the organizations that provide quality results consuming less time. Additionally, testing is often understaffed for the work load required. Therefore, testing teams should be involved in scheduling from the very beginning of the software life cycle and the clear prioritization for validating the test should be made. Moreover, testers should build a good relation with developers and they should be treated equally in the project (Perry & Rice, 2013).

3. CLOUD COMPUTING

Cloud computing is gaining its popularity day by day as it has been improving the business processes continuously. It is a paradigm which eliminates the headaches of managing the computing resources (hardware and software) which was one of the most difficult and time consuming tasks in the traditional approach. This chapter gives an introduction to cloud computing and explains how it can be beneficial to the organizations. Section 3.2 and 3.4 describe the different types of cloud and their service models that can be deployed depending upon the organization's choice.

3.1 What is cloud computing?

Cloud computing is a technology that supports the processing of large volumes of data which relies on sharing computing resources like networks, servers, storage, applications, and services. These resources are basically provided on the basis of the type of application or the platform needed as a service by any third parties. Now, these third parties (service providers) are the ones who manage services over the world, which collectively is referred as the cloud. Therefore, the cloud is an internet-based computing system where IT infrastructure is outsourced to specialized companies via the internet.

U.S. National Institute of Standards and Technology has defined (Mell & Grance, 2011) cloud computing as "a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable resources (for example, networks, servers, storage, applications, services etc.) that can be rapidly provisioned and released with minimum management effort or service provider interaction". This definition in simple word means the dynamic configuration of the pool of resources that can be constantly monitored by the service providers and without worrying about the resource management it can be accessed from anywhere. Mell & Grance (2011) point out some characteristics of cloud computing as on-demand self-service, rapid elasticity, and measured services.

As stated above, cloud computing provides services by outsourcing the computing resources, giving a smart solution for users to handle complex IT infrastructure. It not only provides the resources, but also guarantees a quality of service (QoS) products (CPU speed, I/O bandwidth and memory size) with inexpensive resources on demand of users which can be easily accessed. When we talk about the cloud, we should also have the clear concept of virtualization. Virtualization is the technology used behind the cloud computing where the real hardware is partitioned giving the ideas of virtual ma-

chine and virtual network. This concept is implemented in cloud computing to provide flexible and scalable hardware services.

There are numerous other definition of cloud computing. According to Armbrust et al. (2010), it is the application that is delivered over the internet. Now this application is provided as a service that is hosted in the data center including hardware and system software. Qian et al. (2009) defines cloud computing as a technique where services are provided in a low cost over the internet. Ultimately, both of these definitions have some common meaning. According to Aymerich's definition (Aymerich et al. 2008, p114), cloud computing is a subset of grid computing (every resource of the computer are shared with every other computer in the system) sharing special computing resources. He points out that cloud computing is changing the way of interacting as being more online, rather than the traditional standalone mode.

Therefore, anyone who has internet connection and a standard browser can access the cloud computing applications without worrying about the resource management. Cloud computing affects every life cycle of software development including software testing which will be discussed later in Chapter 4. The cloud model consists of four deployment models and three service models which are discussed below.

3.2 Types of cloud

This section introduces the deployment models in the cloud. Generally, there are four types of cloud models: public cloud, private cloud, hybrid cloud and community cloud.

3.2.1 Public cloud

Public cloud is a type of cloud where computing resources are available over internet for multiple users and are organized and managed by some third party. The third party mentioned here are the cloud service providers, who provide on-demand scalable and flexible services to the client. Mell and Grance (2011) define public cloud as the infrastructure that is provisioned for general public. This model is managed by some business, academic, or government organization or their combination. It exists on the premises of the cloud provider. According to Armbrust et al. (2010), the cloud available as "pay-as-you-go" to general public is referred as public cloud and its services sold is referred as utility computing. Pay-as-you-go is a billing system where one pays for only the resources that are consumed that is, the usage is metered.

This is the most popular model among all the models as it helps the consumers deploy a service in the cloud with a very reasonable cost. Most computer users have been using some of the software and services provided by this model such as Google Drive, Dropbox, and Gmail.

3.2.2 Private cloud

Private cloud is a computing model where the cloud infrastructure is deployed and provisioned for a single organization. It is also defined as the internal data centers of a business which are not available to general public (Armbrust et al. 2010). This model is dedicated to a particular organization which is implemented within a corporate firewall. Because of this, there is no doubt that it ensures security for large data, but might sometimes lack on demand computing power and reduce the cost savings. This operation can be performed in-house or with third-party without the restrictions of network bandwidth, security exposures, and legal requirements.

Mell and Grance (2011) define public cloud as the infrastructure that is provisioned for exclusive use by a single organization comprising multiple customers that can be owned and managed by some organization or third party. Using private cloud would be a better choice when consumers have to deal with large data sets where security and data privacy is a primary concern.

3.2.3 Hybrid cloud

As the name itself suggests, hybrid is a mixture of two models. Hybrid cloud is a model where a cloud infrastructure is formed by the combination of public and private cloud models that can be in-house or provided externally. This cloud model is good for the organization that is more prone to move public cloud to private and vice-versa across a lot of different resources. For example, a company that is using public cloud (Software as a Service provider) is also concerned about the security. In this case organizations can use public cloud to deal with general customers and private cloud for data security within a firewall.

Mell and Grance (2011) define hybrid cloud as a composition of two or more distinct cloud infrastructures that might be private, public or community having bound with some proprietary technology that enables data and application portability. By the use of portability they meant that any organizations whose data and applications are hosted in public cloud can shift in private cloud according to their need.

Generally, private and hybrid cloud infrastructures are only used for specific business purposes, where public cloud is not a suitable choice.

3.2.4 Community cloud

Community cloud is a cloud model where the computing resources are shared among several organizations and is managed by one or more participating organizations. This cloud model is basically designed for working in a joint project. For example, a joint software development environment for an open source project community. Mell and Grance (2011) define community cloud as a community of consumers who share the resources (such as security requirements, mission and policy) that are owned and man-

aged by one or more organizations in the community or a third party. It can either exist on or off premises.

3.3 The benefits of the cloud

Working with cloud computing technology can vastly benefit many business organizations. The primary benefit of cloud computing is cost efficiency, scalability, faster time to market and on-demand elastic infrastructure. Traditionally, organizations had to buy the computing resources and the license fees which were very expensive. However, now the cloud is available in much cheaper rate which can reduce the infrastructure expenses. The cloud providers provide the resources which help the organizations in setting up the initial infrastructure allowing their product to be delivered fast in the market (Doddavula & Gawande, 2009). Additionally, it can also globalize the workforce providing infinite storage capacity on demand that can be accessed from anywhere having the internet connection.

According to Armbrust et al. (2010), the pay-as-you-go model applies to storage and network bandwidth. This depends upon the computation which is different according to the provider. Google AppEngine for example scales with respect to load and customers are charged according to the cycles used. Likewise, AWS charges according to the number of hours used (Armbrust et al. 2010). This helps in scaling quickly, conserving the resources, money as well as helping to focus more on efficiency. Armbrust et al. (2010) also argued that elasticity is one of the important aspects of economic benefit of cloud computing as it transfers the over provisioning resource cost including operation and maintenance cost to the cloud provider.

3.4 Service models

Cloud as a service means delivering what the customer wants as a service over the network. Using these services customers can avoid their overhead cost that come as a part of maintaining the service (Shajee, 2012). The service models include Infrastructure-based, Platform-based and Software-based services as shown in **Figure 3.1**. Generally, IaaS and PaaS are used by technical IT community where as SaaS is used by business and IT community.

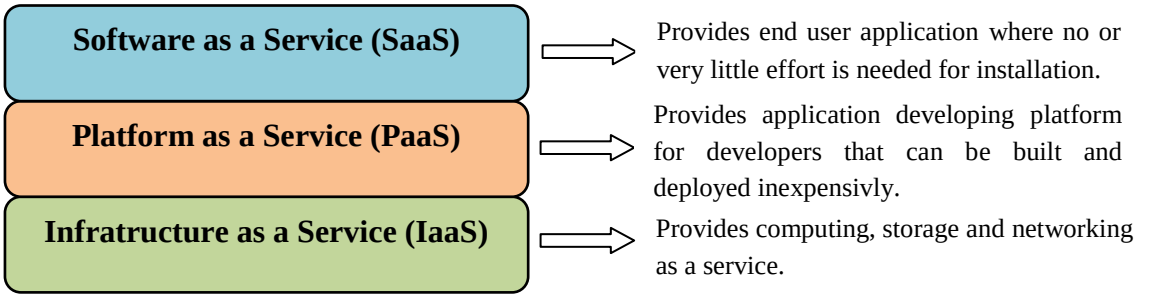


Figure 3.1. *Service models in cloud infrastructure. The visibility to the end-users increases while moving from IaaS to SaaS.*

3.4.1 Infrastructure as a service

Infrastructure as a Service (IaaS) is a service providing model that offers hardware computing resources infrastructure to the clients. These resources might include virtual machines, data storage, firewalls, networks etc. that an organization might use to deliver their solutions. In this service, clients can run their own application on the resource provided and are responsible for managing and security of the application by their own. Mell and Grance (2011) define IaaS model as the capability to provision the computing resources where they can run and deploy the arbitrary software (OS and applications). In the cloud computing model, IaaS is considered as the bottom layer.

According to Doddavula & Gawande (2009), IaaS provides the middleware capabilities such as compute, storage, data stores and messaging on-demand using the infrastructure of public and private cloud. Amazon web services (AWS) is one of the biggest IaaS providers. One of the examples could be EC2 service where they offer VMs with a software stack. The users in this case will have different rights to the server such as customizing and installing different software and configuring the access permissions. Other examples are Amazon S3, Rackspace etc. Therefore, in IaaS model, the consumers will have direct access to manage the operating system, storage, network connectivity and the firewalls, but cannot control the cloud infrastructure.

3.4.2 Platform as a service

In this model, an already built application infrastructure platform service is provided to the clients. This model provides a kind of API that manages the programming platform or other solutions for development. It usually supports different programming languages such as Python, Ruby, .NET languages and Java. Mell and Grance (2011) define PaaS model as the capability of consumers to deploy their applications on the cloud infrastructure using the tools, services and libraries provided. In this model, the clients only have the access to deployed applications, but have no control over the computing resources like network, servers and operating systems.

PaaS are built on the top of infrastructure services and are high level abstraction to the cloud. This platform provides the framework for developing web based applications, analytics and batch framework (Doddavula & Gawande, 2009). Additionally, the clients

using this service will have the access to develop and deploy with complete life cycle management of the application. The popular providers of PaaS services include Google AppEngine, Azure and Amazon Web Services.

3.4.3 Software as a service

Software as a Service is a model that provides application software for the consumers where they need not install any application to their local machine and need not purchase any hardware resources. Using this model, customers can be benefited as they need not manage the computing resources for the application and can use this application on demand. This model can ensure the optimum utilization of the resources (Doddavula & Gawande, 2009). Here, the cloud providers are responsible for the updates, security, OS compatibility, servers, storage etc. Mell and Grance (2011) define this model as the capability of the consumer to use provider's applications which are accessible through the web browsers or program interface.

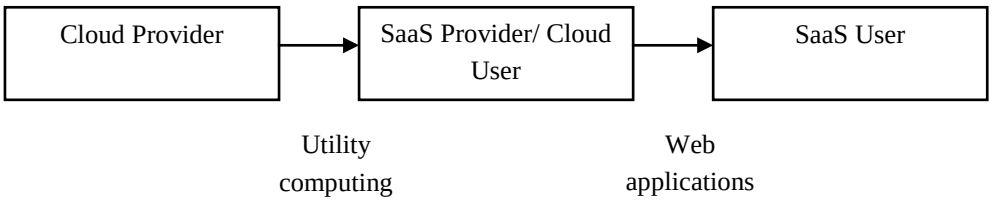


Figure 3.2. *Illustration of cloud computing SaaS provider and user. This illustration is based on Armbrust et al.(2010, p52).*

SaaS encompasses IaaS and PaaS. Therefore, it has become more popular as it can be managed with less effort compared with other cloud models. SaaS applications are usually hosted at provider's network which is served as a web application for SaaS users as shown in **Figure 3.2** (Liu et al. 2010). This model is available as on-demand service which requires very little or no deployment in client's location (Liu et al. 2010). The most common SaaS applications provided often as free services are Gmail, Google Drive, other email services etc. An example application provided free, but with paid enhancements is Dropbox (more shared disk can be bought for Dropbox, which is their business idea). Additionally, the services like Customer Relationship Management (CRM), Document management, different testing tools like Loadstorm and JMeter are available on 'pay-as-you-go' model. In IT industry, SaaS application has become renowned for providing cloud testing tools as services in cloud infrastructure. The clients use these tool services for performance testing, functional testing, regression testing, usability testing etc. Some of the SaaS providers in the field of testing are Salesforce, NetSuite, Soasta and BlazeMeter.

4. CLOUD TESTING

The complexities encountered in the traditional approach to testing make organizations difficult to test the applications and identify the errors early in the development cycle. This chapter introduces a new approach of testing; cloud testing, which eliminates some of the difficulties encountered in the traditional approach such as infrastructure set up, expensive individual costs for hardware and software and a lot of time and effort consumption required for environment set up. The chapter includes an introduction about cloud testing, its architecture, test environment and test management in cloud environment.

4.1 Introduction

Testing with a virtual machine is not a new concept. Even though it has been helping the organizations eliminating the need for huge investments, some of the organizations are still failing to achieve flexibility and scalability (Maruvada, 2012). These virtual machines can sometimes be problematic. It might be the case when applications such as servers, databases and other computing resources are not working properly inside the virtualized environments or the virtual machines might not be available as needed (Parveen & Tilley, 2010). To remove these challenging aspects, cloud computing offers a new platform for software quality assurance and testing that is, cloud testing. Cloud testing at its simplest utilizes the cloud computing infrastructure for software testing. It uses infrastructure based, platform based and software based cloud services to test the application by minimizing the cost and time with improved product quality.

Software testing in cloud refers to testing and measurement activities on a cloud-based environment by leveraging cloud technologies and solutions (Gao et al. 2011). It is mainly used for the testing of cloud based applications using cloud-based tools and technologies. Khamer (2012) defines cloud testing as "a subset of software testing where simulated real world web traffic is used to test cloud based web applications". Cloud testing also utilizes specific cloud characteristics, including redundancy and performance scalability. According to Shajee (2012), the main purpose of cloud testing is to leverage what is out there on the network for the cost effective testing. It can test numerous other types of applications including mobile applications, which is used by a number of customers from different geographic locations (Jadhav, 2012).

Cloud testing can also be described as an online service model. The services provided for software testing uses the combination of cloud computing models that is, IaaS, PaaS and SaaS as described earlier in Section 3.4. Additionally, it also offers testing as

a service (TaaS) for SaaS and cloud-based applications. This service involves the testing activities, which is performed by the service provider for the organizations. This is a new challenge and opportunity for the organizations to meet the quality systems. These services provide on-demand network access to the computing resources such as servers, storage, applications and services that can be easily provisioned and released. Riungu et al. (2010) defines testing as an online service as "a model of software testing used to test an application as a service provided to customers across the internet". This service provides operations on a daily basis with maintenance and testing support through web based browsers.

Cloud testing has gained considerable attention in recent years as a new paradigm for developing and delivering computing application and services. Riungu-Kalliosaari et al. (2012) points out that, as the growth of cloud based service has increased, it is certain that the need for testing them also increase. According to them, this approach helps to test faster than the traditional approach, which consists of a large room of servers running different variations of operating systems. Using the cloud testing approach, different servers can be hired and decommissioned when not in use. The testing based on cloud offers a combination of lower costs, reduction in capital expenditure, on-demand access, reduced maintenance, enhanced collaboration, greater levels of efficiency and most importantly, shorter time to market for key business applications. These characters make cloud testing an attractive alternative to the traditional approach of testing. Also, one of the studies has indicated that organizations might save from 40% - 70% of cost if they adopt this technology as a testing (Fujitsu, 2010).

Therefore, testing in the cloud has potential to revolutionize the way testing is performed. It introduced a new challenge to the software industry in terms of data integrity, security, performance and accessibility. There are various tools that have been offered from different service providers for cloud testing. These tools can be used for numerous testing types such as functional, performance, compatibility, security and usability. This will be discussed more in detail in Chapter 5.

4.2 Test architecture

Testing in the cloud basically aligns the concept of using the cloud as SaaS application. This does not mean other service application models (IaaS and PaaS) are not used during testing. This approach highly depends upon the organizations and their work procedure. Different organizations have their own tools, technologies and work flows. When we talk about cloud testing, we need to understand that virtualization is the basic building block for cloud offerings. Generally, there are three key actors involved in a cloud procedure: service developer, consumer and provider. Service developer develops the application and deploys in the cloud. The services are then consumed by the end users and the cloud providers are responsible for managing and provisioning the cloud infrastructure. All of these actors will access the cloud through the series of access points for

example, an API or a UI. The cloud is made up of three service models and each of the models may or may not be dependent upon each other.

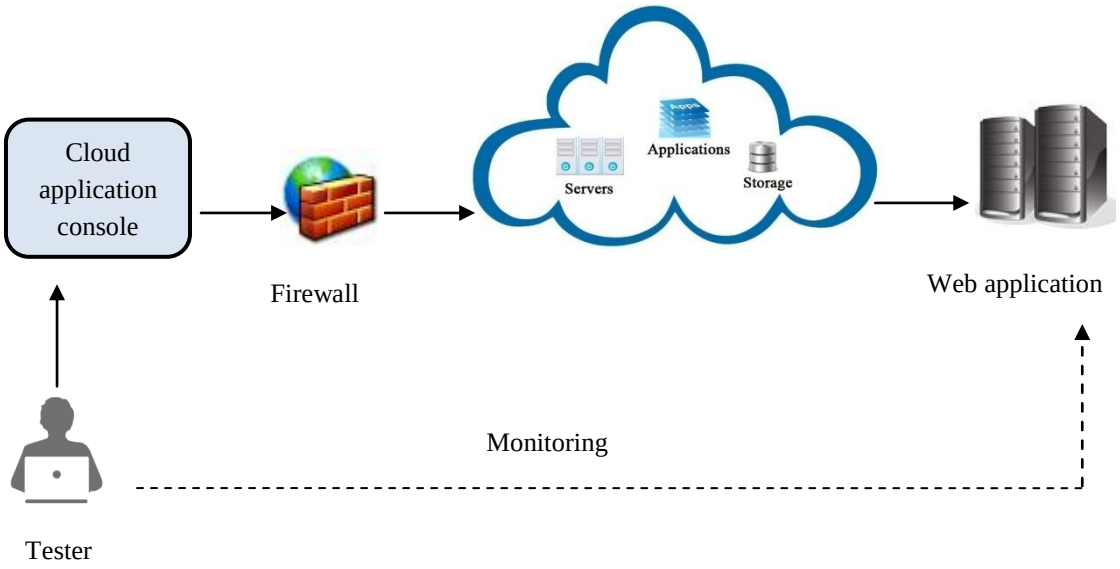


Figure 4.1. *General test architecture for cloud testing that illustrates the flow of testing any application in the cloud. The dashed line for monitoring means test can be monitored simulaneously or after completion.*

The test architecture running in the cloud might differ according to the providers and the organizations. A general form of test architecture is shown in **Figure 4.1**. The organizations first select the testing tool suitable for them from the service provider. The configuration of the tools is different depending upon the service provider. For some tools, no installation is required in the local machine whereas some tools needs to be configured. The testers can start the test through the test console that is available in the cloud. They can create, execute and record the tests through the test console. The tests then pass through the firewall and thus creating the virtual instances to handle the load in the cloud. The results can then be monitored and controlled by the user from the test console if necessary. Similarly, the process continues if any changes are made to the test.

For some of the test tools, cloud provider id (for example, provided by Amazon EC2) is necessary to access the tool. For example, to access the AgileLoad test tool, Amazon EC2 id is necessary. After having this id, testers have to deal with the test console and the injector (machine where the testing is to be performed). The test console needs to be configured with the IP address in the test tool. After the configuration, testers can choose the location where they want to launch the injector from Amazon EC2, for example Helsinki (Cloud, 2013). Therefore, having the account in Amazon EC2 one can easily configure the injector and enter to the machine located in Helsinki where the testing tool is already installed.

The application is spilt into five separate layers in cloud: physical layer, virtualization layer, tool layer, service layer and proposition layer as shown in **Figure 4.2**. The physical layer consists of multiple physical computing servers, storage and networks that provide resources to meet service demands. Relying on the storage capacity of the servers, multiple VMs can be started and stopped according to the demand. In addition, multiple VMs can run concurrently with different operating systems on one machine, hence providing flexibility with the utilization of physical infrastructure. There are several players in the market who are delivering solutions for network, storage and compute virtualization for data centers. Some of the players in the space are VMWare, Citrix, HP and Dell.

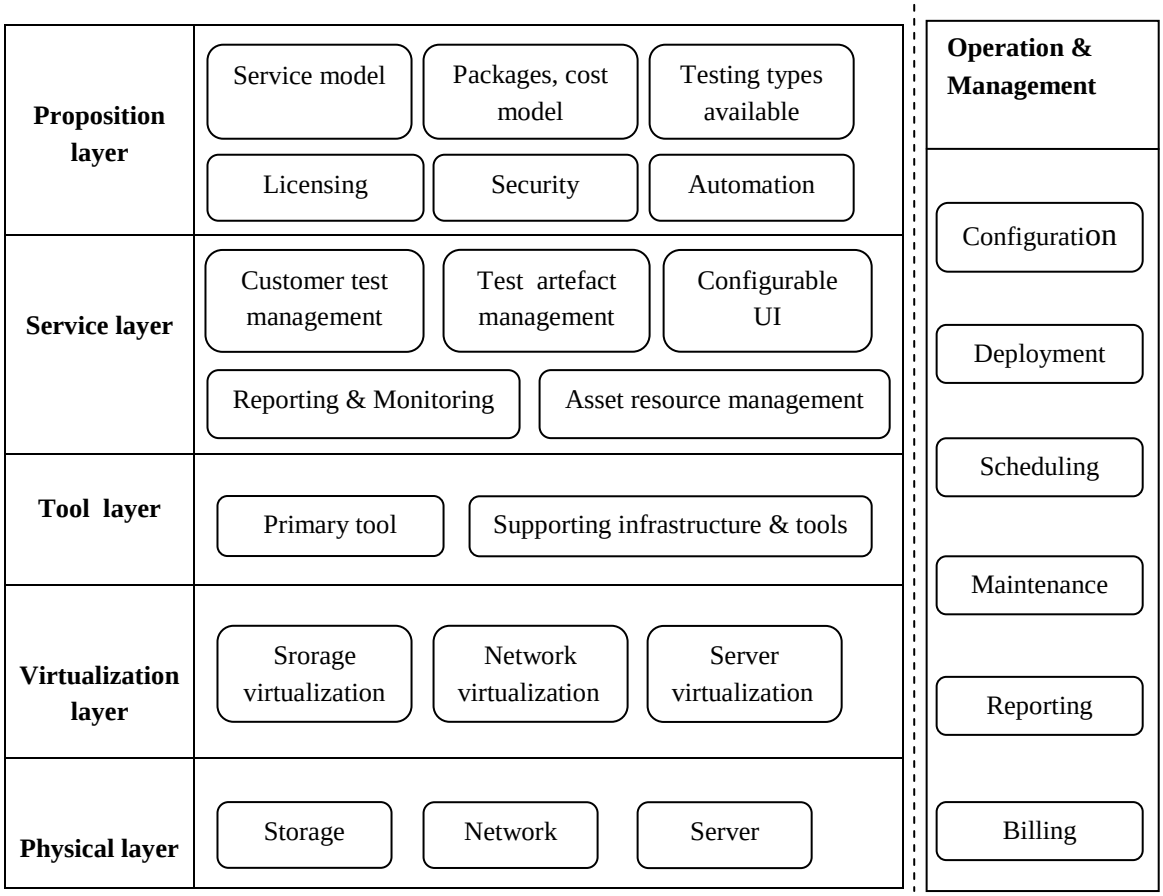


Figure 4.2. *High-level layer based architecture of cloud testing. These layers include physical, virtualization, tool, service and proposition starting from the base. Along with this layer there lies operation and management department for controlling overall procedure.*

The tool layer lies above the virtualization layer which provides primary as well as supporting tools for setting up the infrastructure. Likewise, the service layer consists of customer test management, test artefact management, resource management along with reporting and monitoring service. Lastly, proposition layer manages the service model layer according to the service level agreement. This layer is also responsible for service

packaging, licensing, security mechanisms, different testing types performed and the way testing is to be performed, that is, manual or automated testing. Furthermore, operation and management department is responsible for the process control of overall operation in the business. They ensure that the operations meet a proper schedule and everything is managed and handled efficiently from using the resources to meeting the requirements.

4.3 Test environment

Test environment means planning and setting up the test infrastructures for the purpose of testing before the market launch of an application. It is necessary to test the application after the implementation phase for the high quality of software. High quality software is the one that meets the entire requirements set, is sufficiently defect free, is delivered on time and is maintainable. To meet all of these criteria for the development of high quality software, a good test environment is necessary. A test environment is the setup of infrastructures consisting of hardware and software on which the testing team can perform the testing of a newly built product and follow the test process. The test process can be categorized into four phases: modeling the software environment, selecting the test scenarios, executing and evaluating the test scenarios, and measuring the testing process (Whittaker, 2000).

Setting up the infrastructure includes physical setup and logical setup. The physical setup consists of hardware and network connections whereas the logical setup consists of server operating system, database server, front end running environment, browser (in the case of a web application) or any other software component depending upon the project requirements. This testing environment is built on both server and client ends. According to Eickelmann and Richardson (1996), software test environments provide a means of test process along with integrating testing tools for supporting the testing capabilities across the testing process. They claim that software test environments might support test planning, test management, test measurement, test failure analysis, test development and test execution.

Test environments in cloud generally operate in a virtualized environment with the concepts of virtual machine; a software replica of an underlying real machine. Virtualization is one of the key components in the cloud computing. It behaves as a separate computer, requesting CPU, memory, hard disk, network and other hardware resources from the virtualization layer (Rouse, 2011). These machines are created within a virtualization layer where multiple virtual machines can operate on the same host machine (also called hypervisor). According to VMware (2006), there are two architectures used in virtualization: hosted and bare-metal (hypervisor). A hosted approach as shown in **Figure 4.3** runs on the top of the client or server operating system just like any other application to host a guest OS (runs inside a virtual machine). In bare-metal architecture, a low-level virtualization layer replaces the host OS as shown in **Figure 4.4**. This is also known as virtual machine manager that allows multiple OS to share a single host.

This concept is more efficient than the hosted architecture as it has the direct access to the hardware resources.

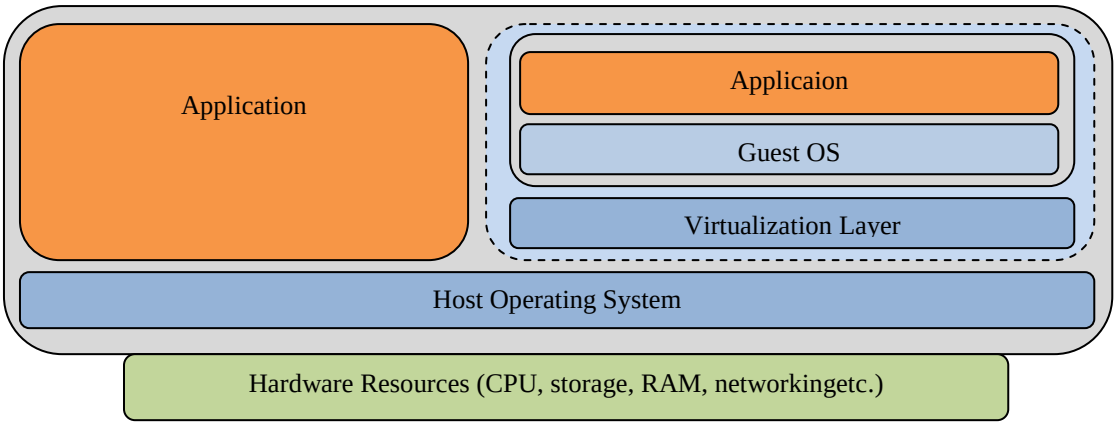


Figure 4.3. *Hosted Architecture. The virtualization layer is installed and runs as an application. It relies on host operating system for physical resource management and support (VMware, 2006).*

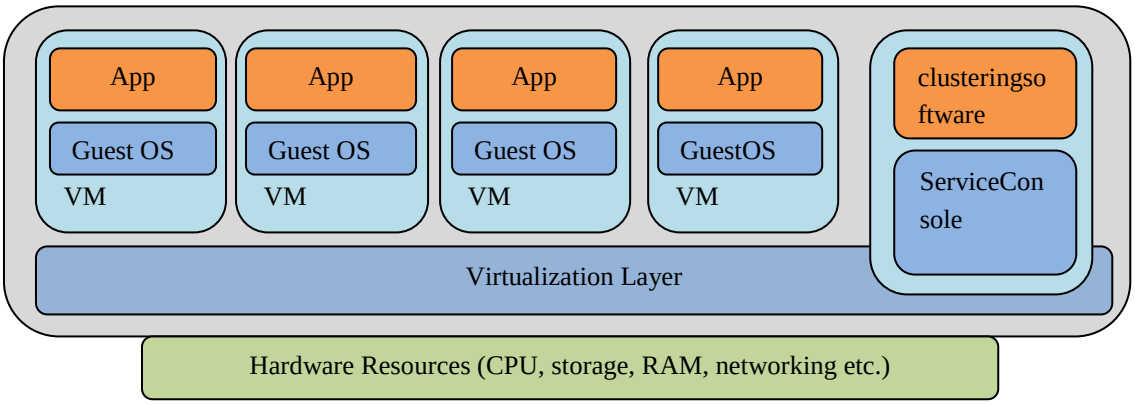


Figure 4.4. *Bare-Metal (Hypervisor) Architecture. Multiple operating systems are shared to a single hardware host (VMware, 2006).*

The cloud environment manages the operation, execution and processes of virtual machines, virtual servers and infrastructure including the back end hardware and software resources. Everything is virtualized in the cloud: physical resources, infrastructure as well as middleware platforms. With this environment, the business applications are consumed as services in the cloud. There are many service providers available for managing the resources in the cloud environment (for example, Amazon EC2, Soasta, Microsoft Azure and Keynote). The cloud test environment can be selected depending upon the client's testing needs. For example, if a customer needs the entire infrastructure along with the testing tools installed, then they can use SaaS service available. Likewise, there are other services available from cloud such as PaaS and IaaS as discussed in Section 3.4. The cloud test environment allows optimizing the computing resources and simplifies the developing and testing infrastructure. They provide a hosting envi-

ronment that is scalable, available on demand and flexible while helping the organizations to plan, install, configure and validate complex test environment within no time (Sahoo, 2013).

The cloud application virtualizes the computing resources and simplifies the development and test infrastructure. This environment shortens the test cycles and eliminates the risk compared to the traditional method (Cloud, 2012). In the traditional approach of testing, organizations focus more on setting up the infrastructure, investing huge amount of money for the computing resources. This lengthens the testing cycle and might be less efficient in the real-time environment. Thus, cloud test environment provides a customized environment for the development and testing team to test more thoroughly throughout the life cycle (Cloud, 2012).

4.4 Test management

Testing activities are controlled with a test management of the system. Test management is the practice of organizing and controlling the software test processes effectively. This process includes requirements management, document management, configuration management, test planning, test execution, test scheduling, defect reporting, test measurement and all the other things that are involved in a standard test process. For the high quality of software products that meet the customer needs, a proven test process framework should be followed by the organization. This will not only increase the customer satisfaction but may also determine the success or failure of a project.

Test planning is the main activity in the generic test process as most of the key decisions are made in this phase (Tian, 2005 p85). For example, defining the goals of testing under a specific environment, determining the test strategy, testing types to be performed, allocating the resources etc. After the planning phase, tests should be run. During the test execution phase, test cases are run with various configured environments and the results are logged in for later use. The failed tests are tracked for monitoring development progress and product quality. Test measurement is also considered as one of the essential parts of the test management process. It keeps track of the test runs with success and failure rates that is used for analysis and demonstrating the product reliability at the end of the project. In addition, the quality metrics calculated can be effective for both the testing and the development team for the quality improvement in a new project (Tian, 2005 p92).

Because of the increased complexity of software products, effective test management is necessary for the key success of a project. It is more troublesome when the quality assurance team grows and the system cannot keep up. So, on the way of saving time and money, organizations sacrifice flexibility, either experimenting with the new tools or using the same old methods. With the advent of cloud computing, test management can be handled easily without having to configure a physical machine. By configuring a set of virtual machines, user can activate the computing resources and utilize them for the purpose of managing the test process.

In the earlier chapters, we have discussed many tools that are available for performance, functional and other testing types. There is also cloud based affordable solutions for the management of overall test processes that have emerged for making life easier for a quality assurance team. This management process includes test life cycle management (requirement management, test case management, task assignment etc.), test execution (automation, team management, test priorities etc.), issue tracking, and monitoring. Additionally, the system can also easily manage multiple test environments and standardize the test processes so that all the team members can use the system to keep track of the progress of testing and measure the quality for the next release (Test, 2013). Moreover, it provides a real-time management solution from the starting of the project through the entire software development life cycle. Most of the tools in cloud systems also have the features to integrate with several bug tracking tools (JIRA, Bugzilla, Redmine, Mantis etc.) and test automation tools (Selenium, Sahi, TestComplete etc.). Some of the tools for test management in cloud are Testuff (SaaS test management), Zephyr and PractiTest. These tools provide the efficiency by leveraging the past information such as documents and test cases that can be reused for new product versions.

Using test management in the cloud can result in a faster release and better quality of products than the traditional approach of testing. Different companies have different techniques of test management. Generally, they use different tools for managing different activities and this can take a long time where one has to deal with the problems of individual tools if encountered. Therefore, the cloud approach of test management is more beneficial as it saves more time for testing without having to play with different tools.

5. TESTING TYPES PERFORMED IN CLOUD

This chapter explains the different types of testing that can be performed in cloud. There are various kinds of service providers available depending upon the testing types. These providers and the tools they provide for the testing will be discussed in detail in this chapter. In addition, the chapter also lists about the pricing models of the tools which are different for every organization.

5.1 Performance testing

Performance testing is a type of testing which measures the system throughput and latency, indicating how well a software system meets its requirements under certain workload (Kumar, 2012). It tests whether or not the software has met speed, scalability and stability requirements under expected workload with a variety of concurrent users. It determines how much traffic a website can handle, and ensures that applications are ready for a market launch. Before launching any application, it is very important to identify the performance accepting criteria of a system to know the number of visitors supported in a site and how the response time and error rates are handled at the arrival of a large number of users. **Figure 5.1** shows the flow of performance testing in a software development life cycle. After the development of any application, test infrastructure is set up and then test is executed for the performance of the application, before it goes to production. Performance testing is responsible for testing the load, stress and scalability of any system.

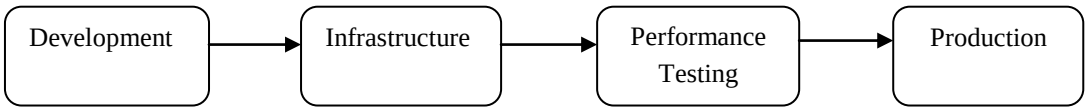


Figure 5.1. *Performance testing flow in a software development life cycle. It is necessary that before the software goes to production, organizations need to monitor the amount of load it can handle. If not, it might get expensive in maintenance phase.*

These days performance testing in cloud has become more beneficial in comparison with the traditional approach of testing where you have to put a lot of effort. The traditional approach of performance testing was carried out in an internal environment in a company which required servers, hardware, production software to host the system under test, that is quite expensive (Kumar, 2012). So, performance testing in cloud ad-

dresses the issues encountered in the traditional model. Cloud computing can carry out performance, scalability, and stress testing in a timely and economical manner (Riungu-Kalliosaari et al. 2012). Organizations performing this testing probably get the bigger benefit from cloud computing, considering both the economic aspect and scalability (Forno, 2012). Testing in the cloud means paying only for what is needed for the required time. This means it is no longer necessary to purchase expensive hardware and software with annual licenses. In addition, configuring the test environment becomes very easy and almost instant, allowing for faster test execution which has the potential to increase the number of test sessions that can be run in the same time period (Forno, 2012). The actual resources used by the application may grow or shrink based on the application load. Performance testing in cloud allows us to test the maximum number of users that are allowed to access the web from different geographic locations in the real environments. The cloud model amplifies elasticity in the application platform. There might be a massive increase in the Internet traffic at some point, in terms of number of users, where an application might get hit from all corners of the world (Kumar, 2012). At this point, application should be able to handle the maximum number of users without any error and degrading its speed. Performance might also change because of the traffic load coming from different geographical locations. Poor application performance directly impacts customer loyalty, reduces revenue and damages your brand. Therefore, performance testing in a cloud infrastructure can help test engineers to take the right step in developing the system, based on evaluating the test results (Kumar, 2012). **Table 5.1** gives a list of performance testing tools along with its service provider and the application where it can be used.

Table 5.1. *List of tools available for performance testing in the cloud*

Tools	**Service Provider**	**Testing types supported**	**Application**
Loadpro	Keynote http://www.keynote.com/	Load testing	Website and mobile
Apache Jmeter	BlazeMeter http://BlazeMeter.com/	Performance testing	Website and mobile
Loadstorm	CustomerCentrix http://loadstorm.com/	Load testing	Website
CloudTest	Soasta http://www.soasta.com/	Performance testing	Website and mobile
CloudTest Lite	Soasta http://www.soasta.com/	Functional and Performance testing	Website and mobile
LoadImpact	Load Impact http://loadimpact.com/	Load testing	Website
Neoload	Neotys http://www.neotys.com/	Load testing	Website and mobile

Tools	Service Provider	Testing types supported	Application
Gomez	Compuware http://www.compuware.com/	Performance testing	Website
SoaTest	Parasoft http://www.parasoft.com/	Unit, integration, regression, system, security and load testing	APIs, web and mobile, Service Oriented Architecture (SOA)
LoadRunner	HP http://www.hp.com/	Performance testing	Website
LoadUIWeb	SmartBear http://SmartBear.com/	Load testing	Website and mobile
Neustar	Neustart http://www.neustar.biz/	Load testing	Website
WebLOAD	RadView http://radview.com/	Load testing	Website and mobile
WAPT Cloud	WAPT http://www.loadtestingtool.com/	Load testing	Website and intranet
Loadster	Loadster http://www.loadsterperformance.com/	Performance testing	Website, web apps and Web services
CLAP	Persistent http://www.persistentsys.com/	Load and performance testing	Website and web services
Monitis	Monitis http://www.monitis.com/	Load testing	Website
AgileLoad	AmazonEC2 (needs to have Amazon id for the access of the tool, see section 4.2) http://www.agileload.com/	Load Testing	Web and mobile
Silk Performer CloudBurst	Borland http://www.borland.com/	Performance testing	Web and mobile
App Loader	NRG Global, AWS http://www.nrgglobal.com/	Smoke, regression, performance, functional and security testing	Web
ProxySniffer	Apica http://www.apicasystem.com/	Load testing	Web and mobile
Blitz	Blitz https://www.blitz.io/	Load testing	Websites, web apps and APIs

5.1.1 Keynote

Keynote (2013) was founded in 1995 with the purpose of providing tools for monitoring the performance and availability of websites, including mobile sites. Keynote claims to be one of the largest cloud monitoring networks in the world. It is comprised of more than 7 000 measurement computers and mobile devices in over 275 locations and 160 metropolitan areas worldwide. With the help of Keynote, customers can improve their web performance and the quality of their mobile communication. Keynote provides on-demand testing and measurement services with realistic web testing, using geographically diverse servers. It provides different representative panels of real online users (over 145 000) from different demographics and language which makes easy for the user to evaluate their product.

Keynote (2012) was ranked in the 54th position out of 100 in Forbes magazine (an American business magazine) titled "Best small companies" in 2012. They provide different products for web performance monitoring, web load testing, web privacy monitoring and real user experience testing. Their three market leading testing and monitoring product platforms are Keynote Perspective, Keynote DeviceAnywhere and Keynote SIGOS. Over 2 000 customers depend on Keynote Perspective as they provide performance monitoring and load testing for web and mobile sites. DeviceAnywhere is used for testing and monitoring the performance, functionality and usability of mobile applications and websites. Similarly for mobile, fixed and VoIP communications, SIGOS provides end-to-end Quality of Service testing and monitoring solutions.

Test Perspective (Keynote, 2013a) is a self-service website load testing tool which runs load tests on demand (pay only for what you use) and receive immediate feedback on modifications you make. This tool provides a simplified process to run as it includes a sample script where just by following the guided workflow leads to launch the result immediately. The tests can be set up on our own and run on demand. It utilizes Keynote's global test and measurement network that includes load generating agents from multiple geographic locations and internet backbones (Keynote, 2013a). This tool allows users to discover the amounts of load that the application can handle in order to determine the point at which the response time becomes unacceptable. Customers can add the fields for test duration, peak virtual users (a virtual representation of user behaving the same way as a real user would) and time to reach a peak. It also provides a load distribution graph where the customers can analyze and confirm the capacity handled. Keynote provides a unified dashboard where all the live load testing indicators can be viewed on a single screen (Keynote, 2013b). This is available to every member in a team. Buying a license or installing servers or networking gear is not needed. Keynote provides this tool for a free trial account of 14 days allowing 3000 per Virtual User Minutes on account (Keynote, 2013a).

Load Pro (Keynote, 2013c) is another tool provided by Keynote which is a fully managed (SaaS) web load testing service for production websites and applications, including mobile web browsers. It uses real world internet traffic to reveal the perform-

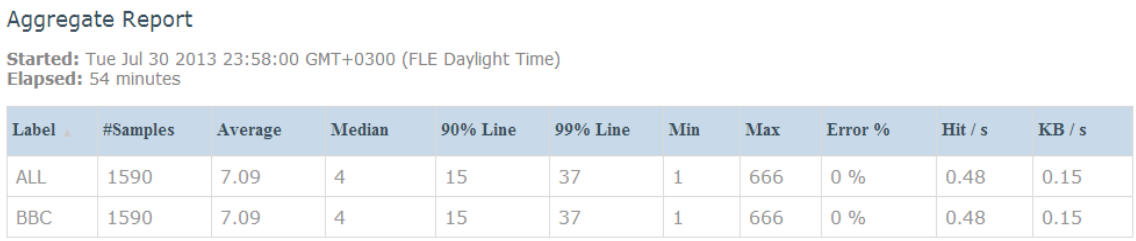
ance of production websites just as the customers will experience it. This tool is basically designed for e-business where customers depend on their site to handle heavy traffic and execute transactions in a high demand conditions (Keynote, 2013d). It provides features such as test planning, script developing, reporting and makes initial test run easier for the platform. It also has the real time dashboard feature where customers can monitor the test with the different status of page views, page errors, overall response time etc. (Keynote, 2013c). With the increase in the site performance change, this tool dynamically reacts according to modeled user behaviour.

5.1.2 BlazeMeter

BlazeMeter (BlazeMeter, 2013a) claims to be a self-service, on-demand and secured load testing tool that is compatible with Apache JMeter for performance testing frameworks. BlazeMeter is a cloud-based platform (PaaS) for load and performance testing as well as cloud monitoring that can create load testing scripts, perform real-time monitoring, and launch new servers (Riungu-Kalliosaari et al. 2012). It was introduced with the aim of providing easier environment, to run complex load and performance tests quickly and affordably. Therefore, it does not matter whether you are a QA expert or an individual testing a simple website; anyone can run this test either by using existing JMeter scripts or by simple entering a domain name (BlazeMeter, 2013a).

BlazeMeter provides a real user simulation for web and mobile application with geographically distributed testing across eight different locations (BlazeMeter, 2013b). Over 6 000 users use BlazeMeter service which includes banks, government agencies, media streaming companies, SaaS companies and many start-ups. It provides extra large testing capacity with single or cluster of servers through which one can test up to 100 000 simulated browser users or more. It can also be used for stress testing of SQL systems (Microsoft SQL, MySQL, Oracle), HTTP/S, web services, mobile and even raw TCP/IP (BlazeMeter, 2013b). There is no need for any setup or installation required to use this service. BlazeMeter also provides a WordPress plugin, a Drupal module, and a Jenkins CI plugin which simplifies load testing. WordPress is an open source blogging tool or content management system that is used to create a website or blog. Likewise, Drupal is also a free and open source content management framework having its own features. On the other hand, Jenkins is an open source continuous integration server that is built in Java. It provides ease for the developers to integrate changes to the project and for the users obtaining the fresh built. BlazeMeter has their own data security policy with the top priority of keeping the customers data private and secure (BlazeMeter, 2013c). They tend to keep the data encrypted at each receive and transmit of the account information. They have dedicated server instances to run each load tests which not only improves the performance but also assures that the resources used by one client won't be accessible by another client. BlazeMeter uses Amazon Web Services (AWS) cloud for their data storage which provides a secured service that contains a number of ways to restrict unauthorized access without hindering the customer's demand (BlazeMeter, 2013c).

BlazeMeter provides plugin for JMeter as a cloud testing tool. JMeter (JMeter, 2013) is an open source desktop Java application for load testing and measuring the performance on the server/network under heavy load. JMeter analyses the load and performance on static and dynamic resources such as static files, servlets, FTP servers, Java Objects, Database, Perl/CGI scripts, queries and more (Emily, 2008). It has the features such as: portability (can run on any JVMs), concurrency, efficient GUI with fast execution and replaying of test results. Although it is better known for a performance testing tool, it can also be used to measure the functional behaviour of client/server application such as web applications or FTP applications. Emily (Emily, 2008) describes JMeter as freely distributed testing application which is highly extensible through a provided API (Application Programming Interface). Since it measures the response time and all other server resources such as CPU loads, memory usage, and resource usage, it can effectively be used in functional test automation (Emily, 2008). BlazeMeter provides a plugin for the open source Apache JMeter that allows the user to integrate BlazeMeter tools (cloud based reporting and test management tools) as a part of JMeter desktop software with no extra cost. Using the BlazeMeter service, JMeter users can compare the reports, metrics and KPIs (Key Performance Indicator) online during the test (BlazeMeter, 2013d). Each test usually lasts for 55 minutes. After running the test, we get a report of the load result consisting of real time performance KPIs for response time, latency, errors and request or second. It also provides the option for downloading all the report data in CSV or JSON format. Users can also access the scripts, compare log files and report from the previous runs (BlazeMeter, 2013e).The following figure **Figure 5.2**, is a sample report that is generated for the website http://www.bbc.co.uk/.



Aggregate Report

Started: Tue Jul 30 2013 23:58:00 GMT+0300 (FLE Daylight Time)
Elapsed: 54 minutes

Label	#Samples	Average	Median	90% Line	99% Line	Min	Max	Error %	Hit / s	KB / s
ALL	1590	7.09	4	15	37	1	666	0 %	0.48	0.15
BBC	1590	7.09	4	15	37	1	666	0 %	0.48	0.15

Figure 5.2. *Sample aggregate report generated for BBC news site.*

5.1.3 CustomerCentrix

This is a service provider company specializing in SaaS products basically for load testing tools. The company is mainly focused on building the customer centered applications focusing on their business needs for absolute customer satisfaction. Initially, they created load testing tools for their own benefit of web application testing purpose, as the traditional tools were too expensive for them, and later they continued to deliver world class software in SaaS model. Their common goal defines as "Put our customers at the center of business decisions" (CustomerCentrix, 2013).

Loadstorm is a cloud load testing tool provided by CustomerCentrix for web applications and websites. It is designed to be a simple and inexpensive tool for running the performance test where the users need not learn a new scripting language. As this is provided as Software as a Service application (SaaS), there is no need to configure server software, no need to download anything in the local machine and no hardware to buy. Loadstorm can generate around 300 000 virtual users from the cloud which helps to stimulate real user behaviours such as registration, login and clicking links. After the completion of the test, Loadstorm provides a report consisting of the real-time graphs of summary test results, request by error code, and request by elapsed time. One of the sample summary test result is shown in **Figure 5.3**.

	Response (average)	Errors	Requests	RPS (average)	RPS (peak)	Throughput (average)	Throughput (peak)	Total Transfer
HTML	5.623	1,797	257,237	71	112	2,912 kB/s	4,677 kB/s	10 GB
Other *	2.859	2,156	619,009	172	274	6,532 kB/s	10,062 kB/s	22 GB
Total	3.671	3,953	876,246	243	385	9,444 kB/s	14,673 kB/s	32 GB

Figure 5.3. *Example of a Loadstorm test results summary report*

Similarly, the request by error code report displays the number of errors occurred and the things that caused the error. Request by elapsed time displays the test results in sequential order. Fields such as elapsed time, user count, requests per second, average response time are displayed for a minute interval. In addition, Loadstorm can also be used for regression testing and agile testing (Cloudbook, 2013).

5.1.4 Soasta

Soasta claims to be the leading cloud based testing solutions provider company, for functional and performance testing of mobile and web applications (Soasta, 2013a). It is a privately held company that provides customers with uniquely innovative, flexible, fast, affordable and exceptionally effective solutions (Soasta, 2013b). Soasta has large experience in real-time analysis and ensures that the web's application and services perform in a higher quality, scalable and predictable manner. They provide the solutions for mobile performance testing and test automation, real user monitoring and web performance testing with the tools like CloudTest, CloudTest Pro, TouchTest and mPulse which are available on-demand from the cloud or as an application installed inside user's environment. Soasta has the platform where customers can test applications hosted on private, hybrid clouds such as those built with technologies from VMware and Eucalyptus Systems and public cloud including Microsoft Azure, IBM Cloud and Amazon Web Services (Soasta, 2011). It also provides planning, implementation, execution, and analysis of the services of which the solutions are used in e-commerce, e-learning, media, entertainment services etc.

Soasta provides CloudTest tool with a single, powerful integrated platform which supports load, performance, functional, and web (UI/Ajax) testing. It is available as on-demand service in the cloud that enables developers to test and monitor their web application. CloudTest is basically designed for testing modern web and mobile application which can even handle complex tests (Soasta, 2013a). It allows designing, executing and analyzing of performance tests on an intuitive platform. It helps to analyze the load by displaying the metrics of average response time, receive rate, division of message sent by the domain, through put and error rates and many other metrics as the number of virtual user increases. It can easily handle millions of geographically dispersed mobile or web users. With CloudTest, customers have a wide view of web performance including application and database servers, load balancers, instrumented code and bandwidth which helps them to analyze fast, the amount of traffic their website can handle. One of its features is a data extraction layer that captures the key performances metrics from both inside and outside the firewalls. Soasta also provides customers with pre-built metrics of reports and charts for documenting test results so that they do not have to build it from the scratch (Soasta, 2013c). Therefore, CloudTest has the features of test building (the creation of tools supporting dynamic content such as AJAX5, HTML5), test management, real-time analytics and access to global test cloud.

5.1.5 Load Impact

Load Impact is a software company founded in 2009. It is specialized in the field of performance measurements and load testing (LoadImpact, 2013a). It is on demand service where no installation is required (SaaS) on the clients' computer and can be run directly over the Internet without requiring the license fees. Load Impact helps to configure and run clients test, provide live traffic simulation and gives some advice on optimizing the test. With the help of this service, customers always have the access to test configurations and test results (LoadImpact, 2013b).

The tool provided by Load Impact service for web load testing can scale up to 100 000 concurrent users in a single test, testing multiple scenarios with different geographic locations (LoadImpact, 2013c). It has a graphical display where customers can see the online users using web with user load time which can be updated dynamically. With this tool, customers can script the user scenarios in a language called Lua (http://www.lua.org), provided with their IDE for code compilation and inline documentation. It helps to generate load scripts and analyze the web page with different server metrics which requires no programming knowledge. This tool is flexible as it supports any HTTP-based application or service with proper scheduling, meaning the load test can be configured at any time with certain durations (LoadImpact, 2013c). There is also the page analyser tool provided by LoadImpact which gives the information about the web site performance by emulating the well-known browsers with connect time, download time, time in queue, time to first byte, etc. (Sergiy, 2013).

5.1.6 Neotys

Neotys is a privately owned company founded in 2005 for testing web and mobile applications. It has been providing services to more than 60 countries for the better performance and quality of web and Rich Internet Applications (Neotys, 2013a). Neotys has also been providing on-site and online training, consulting and cloud testing services (Neotys, 2013b). With Neotys, Customers can use the same load testing solution for both internal testing and testing in the cloud. Neotys has received many awards which can create a good print on new customer's mind. The first award that it received was in the year 2007 for "Young Innovating Enterprise" which was issued by the Ministry of Research in France. Neotys also achieved "Best Quality Tool Award 2010", a Tool Challenge award at Vienna Software Quality days 2010. In addition, it is also included in the report "The Testing Tools Landscape: 2010" where its cloud testing tool was mentioned (Neotys, 2013c). Neotys is integrated with multiple cloud platforms (Amazon EC2, HP Cloud Services, Rackspace, CloudSigma and many more) and supports realistic, large scale test across multiple geographical zones. To ensure that the data is secured while testing, the communication between the controller and load generators should be encrypted. SSL can be used to secure the communication between the browser and the tested server; this helps to secure the data that is sent to the load generators during the test (including account information) as well as the data that is retrieved, including error messages (Neotys, 2011).

The load testing tool provided by Neotys is Neoload which is designed to test web and mobile applications. It can realistically simulate a large number of users and monitor the infrastructure servers to detect potential problems like memory leaks, bottlenecks, poor network connection, software configuration issues etc. The clients using this tool can test more quickly and efficiently as it supports the latest web application technologies such as Web 2.0, ERP/CRM, SOAP, Oracle forms, Ajax, GWT, and Flex (Neotys, 2013d). Neoload can support different Operating Systems (OS) like Windows, Linux, Solaris, and AIX. It can also work with different RDBMS such as Oracle, MySQL, SQL Server and DB2. This tool can run both in-house (inside the firewall) and cloud (outside the firewall) from multiple geographical locations. To start the load testing in cloud, the customers should launch a cloud session and then start the test, where several tests can be launched within the same cloud session. Here, the cloud load generators can be started and managed directly from the Neoload controller GUI, either new session can be started or the existing session can be continued. After the test is over, the cloud session can be stopped and results can be analysed.

Neoload also provides the statistics and tools required for the in-depth analysis so that looking into the errors, the team can figure out the corrective measures like setting of the server or the application code. The main feature of Neoload that is different from other load testing tools is that it provides separate results for each geographical zone, where one can compare the results (Neotys, 2013e). The Neoload architecture consists of two main components: the Controller and the Load Generator as shown in **Figure**

5.4. The Load Generator can be deployed on as many machines as required depending upon the load. It simulates the users accessing the application to be tested. With this architecture, one can run any kind of test with high traffic volumes. From the controller, the user can create and record the test scenarios, execute it by controlling the load generator's output and monitors the web infrastructure's component on the server (Neotys, 2013e).

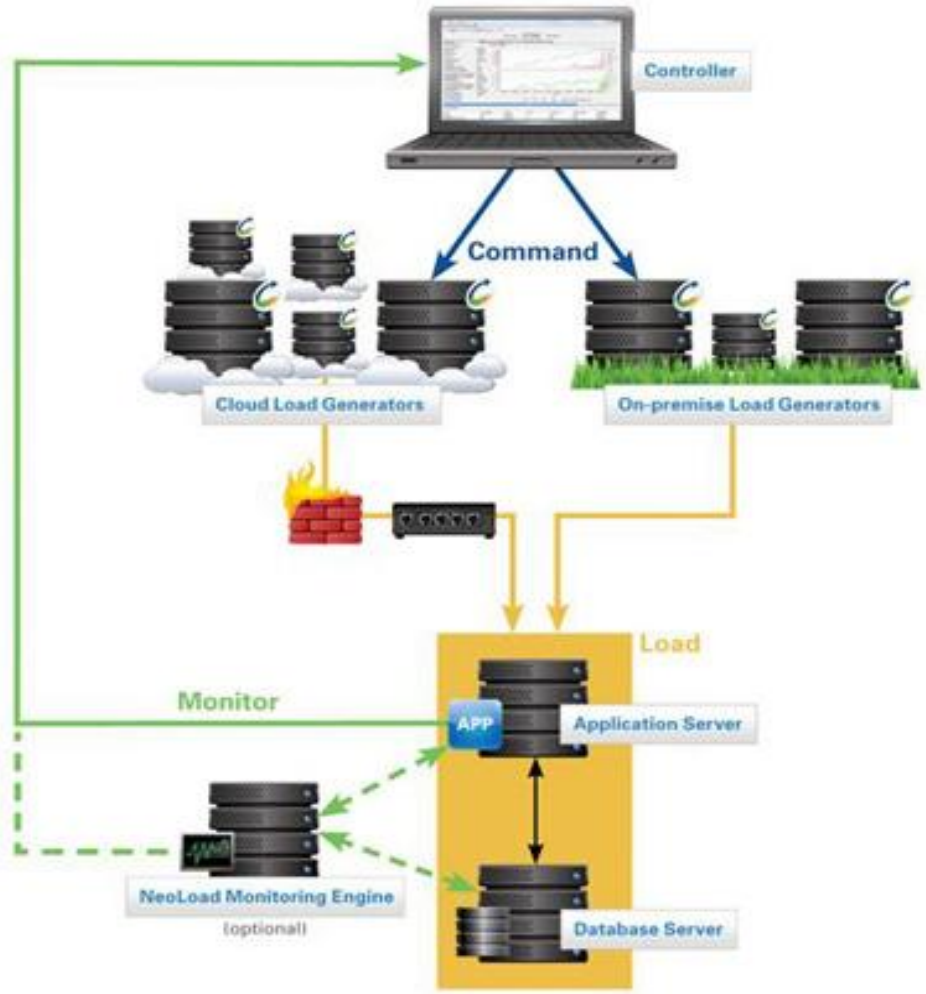


Figure 5.4. *Illustration of Neoload architecture. This diagram displays the work flow of Neoload for load testing in the cloud (Neotys, 2013e).*

5.1.7 Compuware

Compuware is a public company founded in 1973, which provides software, experts and best practices to ensure technology works well and delivers value (Compuware, 2013a). It has achieved many awards each year starting from 2006 till ongoing. It provides the solutions to application performance management, IT project & portfolio management, IT governance, mainframe, professional IT services, and professional service automation. Compuware initially worked with Amazon cloud provider but later clients started demanding more in terms of data centers with different geographic locations. Therefore,

Compuware chose Dimension Data as their hosting platform to spin the servers up and down on demand (Compuware, 2011a).

Compuware provides Gomez as a web load testing tool which is designed to test the performance of web applications. Gomez is on demand performance testing solution, which generates scalable, high-volume, data center based and real world load from thousands of Gomez Last Mile desktops around the globe. It is a part of the application performance management platform which is useful for checking its efficiency on different levels. With most of the company's experience, it is said to cover all kinds of performance issues and is able to run the test from many different locations.

Generally there are two ways of creating test in Gomez: single tests and Gomez recorder. Single test is a simple and common way of performing a test where they have only one step that is, loading a page. In this test, the measurements are taken during the runs and alters are triggered during the error. On the other hand, Gomez recorder is a desktop application which is used to record, playback and edit test scripts. This is basically used for checking whether a page is loaded properly. A variety of performance tests can be performed using this tool such as last mile, mobile, passive actual monitoring, backbone and others. For example, last mile tests check the physical abilities meaning how fast a page can be loaded using high or low broadband or dialed up connections in different countries (Mirońska, 2012).

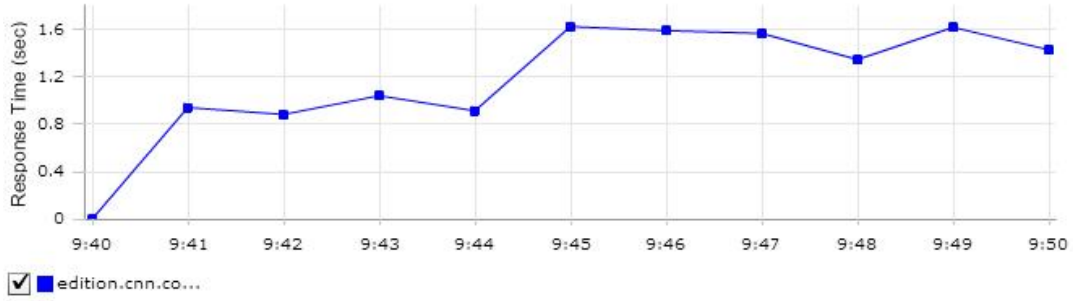


Figure 5.5. *Graph showing the transaction response time of CNN website (Compuware, 2011b).*

A sample website of CNN (http://edition.cnn.com/) was run by the author to test the performance of the website. The results shown were included as four graphs for data from the cloud. They are total virtual users, transaction response time, total transactions per minute and total failures per minute. **Figure 5.5** shows the graph of transaction response time which shows the total length of time it took to load the page along with all its objects. With the increase in load, the transaction response time stays flat and with the increase in transaction response time a bottleneck might be encountered. Similarly, **Figure 5.6** shows the number of times the page or one of its object has encountered an error during the test. This can be studied in more detail by clicking the link available in the test page (Compuware, 2011b).

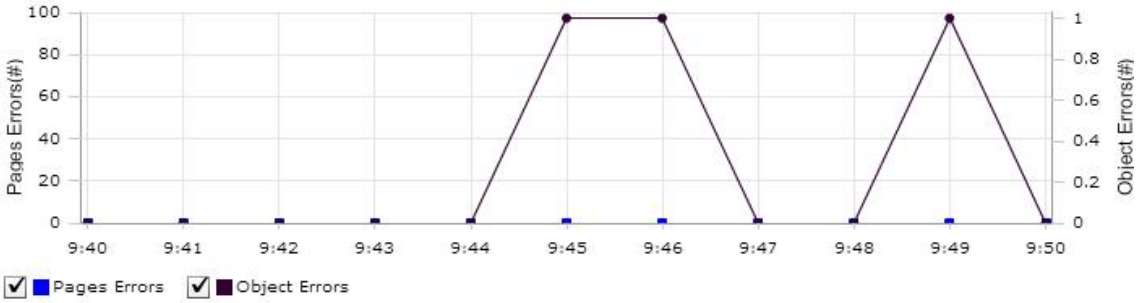


Figure 5.6. *Graph showing the total failure per minute of CNN website (Compuware, 2011b)*

5.2 Functional testing

Functional testing is a type of testing that verifies whether the application meets the functional system requirements. This applies usually to the system and especially the UI level functional testing where the detailed implementation of the system is ignored and the system is only tested focusing on how it was supposed to work according to the specification and design documents (Functional, 2013a). Functional testing is an important step in software development life cycle which ensures that the product works as the users expect it or as specified. The traditional approach of functional testing uses a systematic process including requirement definition, test planning, test design, test execution and analysis (Functional, 2013b). To test the functionality of a system, a clear understanding about it is necessary. Functional testing is dominant in the testing types such as integration testing, interface testing, system testing, regression testing etc. It includes different phases such as test requirements gathering, test planning, test data preparation and test execution (Functional, 2013b). Functional testing can be carried out using manual or automated testing.

The emerging new technologies, the versatile demands of the market, and the dynamic nature of the application has led many organizations to use automated testing. Automated testing is performed with the help of software for the execution of the test and can be repeated for a number of times which are impractical with manual testing. With automated testing long term efficiency can be improved whereas with short term requirements, manual testing can be more effective. While the cloud is often thought only in context of test automation, it obviously supports exploratory testing too, and cloud testing tool vendors can be expected to provide more support for that in coming years. There might be a case where the companies need to test a non-web based application with the direct access. This case can be resolved with the case of remote desktop service to access the test environment, but software vendors are not always willing to grant the access or there might be problems with the machine (Forno, 2012). So in this case, the cloud provides quicker and safer access to test the application. Functional testing on the cloud environments is easier than in the traditional environments as they provide a very easy setup where one can run many test suits in parallel. It also provides a faster and more efficient way of testing in multiple browsers for web applications in

desktop and mobiles. There are other interesting functionalities provided by the tools which will be discussed in later sections. **Table 5.2** below is the list of tools available for functional testing in the cloud.

Table 5.2. *List of functional testing tools that are available in the cloud.*

Tools	Provider	Testing types supported	Application
Janova	Janova http://www.janova.us/	Functional testing	Web
TestComplete	SmartBear http://SmartBear.com/	GUI test, unit tests, regression tests, whitebox tests, data-driven tests, keyword-driven tests, distributed tests, coverage tests and manual tests	Desktop, web and mobile
TestMaker	Push To Test http://www.appvance.com/	Performance and functional testing	Web and mobile
Selenium	Sauce Labs https://saucelabs.com/	Functional testing	Web and mobile
Windmill	-	Functional testing	Web
QEngine	-	Performance testing and functional testing	Web and web services

5.2.1 Janova

Janova was founded in 2002 (Janova, 2011a). It is a web-based automated testing tool that is provided as Software as a Service application. Their main feature is that test can be written in plain English. Using this tool, a web application test can be created in seconds, eliminating the need to have technical knowledge of coding. It claims to be twenty times faster than traditional web testing tools (Janova, 2011b).

To create a test in Janova, a plain English language is used to write the test suite just like a user story. An example of test suite is shown in **Program 5.1**.

Program 5.1. *An example of Janova script to check the sign in button of Google home page*

```
Given I go to the Google homepage
When in the Google homepage I click on "Sign in"
Then in the Login page I should see the "Email Address" element
And in the Login page I should see the "Password" element
And in the Login page I should see the "Sign in" element.
```

The test suite in **Program 5.1** is written as if someone is instructing a user how to manually navigate through the application and what the result should be. This is some-

thing like writing the steps of the test case. Once the feature and its scenarios are created, the variables should be defined which can be done with any development tool for inspecting the element such as fire bug (Janova, 2011c). After the test is ready, it is run in the cloud and reported as pass or fail. In the case of failure of the test, this tool not only shows where it failed but also tells why it failed (Janova, 2011b).

Janova has the benefits such as no download necessary, no infrastructure cost, can run unlimited test on cloud and easy to read the detail reporting. With parallel batch processing ability, Janova can run up to 20 tests simultaneously in the cloud, which gives the power of 20 machines. It provides free 24/7 supports for the customers. Additionally, it also helps customers to know about the product providing the online classes of training, based on organization's needs. Janova can be used for 15 days as a trial version and after that some payment is necessary to be made (Janova, 2011d).

5.2.2 SmartBear

SmartBear is a private software company providing tools for developers, testers and IT professionals for the purpose of development, testing and quality analysis and web monitoring the applications on desktop, mobile and in the cloud (SmartBear, 2010a). It was founded in 1999 with the name of "AutomatedQA Corporation" and changed its name to "SmartBear Software Inc." in July 2010 (SmartBear, 2010b). SmartBear has users in 90 countries, in small business, software companies and government agencies (SmartBear, 2010a). The tool runs equally well inside the company firewall, in a private or public cloud and on smartphone, tablets and other mobile devices (SmartBear, 2010c). It provides testing tools for high accuracy and coverage with automated testing and for test management.

SmartBear provides TestComplete as an automated functional testing application that runs in Microsoft Windows. Creating a test with TestComplete is claimed to be as easy as browsing through web application. Just by pointing and clicking the test, it can capture all the actions inside the browser. For example, web page loading, link navigation, data entry and retrieval, form submission etc. So, it can record the automated test scripts; test can be written from the scratch or the existing automated scripts can be extended with new functionality. TestComplete provides multiple ways to identify and locate objects inside web applications such as searching elements with more than one property value or locating by XPath notation. It also provides access to DOM elements, attributes, methods and CSS attributes, which further allows performing advanced verifications (SmartBear, 2010d). This tool is not limited to testing of only one implementation technology. It can work with any Windows and web applications that are developed with various programming languages. Additionally, it has built-in support for multiple scripting language like VBscript, JScript, Delphiscript, C++script and C#script (SmartBear, 2010e). With TestComplete, testing teams can perform several testing types such as functional GUI test, unit tests, regression tests, white-box tests, data-driven tests, keyword-driven tests, distributed tests, coverage tests and manual tests (SmartBear, 2010f).

Performing a multiple browser test can be problematic and time consuming test for many quality assurance departments. With TestComplete, one can easily perform cross browser testing. It supports the browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, Opera and Apple Safari (SmartBear, 2010g). Since it is based on a COM based architecture, everything is visible in TestComplete (panels, project items, specific scripting objects) as they are implemented in plug-ins. Customers can also create their own plug-ins for new panels, new scripting objects, new project items, new menu and toolbar items etc. (SmartBear, 2010h). It also provides advanced reporting for the deep analysis of the test results and test logs. These all features can be performed without having to know the programming knowledge of the application.

5.2.3 Push To Test

Push To Test is an open source test automation and application performance management solutions provider that was founded in 2001 (TestMaker, 2008a). It provides a service with continuous integration platform that combines grid technology and cloud computing to run tests across single or multiple cloud-based test environments (Riungu-Kalliosaari et al. 2012). They offer customers a testing platform with cloud testing focusing on the latest technologies where the test assets can be reused and managed properly (TestMaker, 2008a). They also offer customers a long term plan for building test infrastructure that leverages the open source technologies. Push To Test provides support for advanced testing, including agile, load and integrating open source test flows into clients systems. They are already in use at 600 000 sites and they claim to be the most popular stress testing platform. Moreover, they can handle the challenging issues of load and user experience testing for social/mobile apps, including the client-side code (TestMaker, 2008b).

Push To Test provides TestMaker as a tool for the testing of the performance and functional issues in web, Rich Internet Applications (RIA) using Ajax, Flex, Flash and Business Process Management (BPM) applications. This is different from other tools as it supports different kinds of testing in one tool and needs to be installed in user's computer, providing specific commands to support automatic deployment to the cloud testing environment. For this, a cloud testing service should be identified (for example say Amazon EC2). Now, TestMaker creates TestNodes in EC2 instance, runs the test and then monitors the result. It can operate for functional testing (smoke test, regression test, and integration test), load and performance testing (scalability test and stress test) and production monitoring in a distributed network of TestNode (TestMaker, 2008c). This tool uses a single test script for functional, performance and production monitors that can be written in Selenium, Sahi, SoapUI etc. These test scripts can be implemented using languages such as Java, Ruby, Python and PHP. It allows the organizations to work in an integrated test environment with agile approach. Push To Test provides security by supporting HTTPs and Secure Socket Layer (SSL) protocols. These protocols use Java cryptography architecture to work with SSL certificate to handle the error, when server side SSL certificates are unknown (TestMaker, 2008d).

5.2.4 Sauce Labs

Sauce Labs was founded in 2008. It provides testing of the web and mobile applications without the need for setting up and maintaining Selenium testing infrastructure (SauceLabs, 2008a). It provides a Selenium based testing service that supports automated testing for mobile and web applications (Riungu-Kalliosaari et al. 2012). Sauce supports functional testing with Selenium as well as JavaScript unit testing, mobile web application testing and manual testing. It can run Selenium tests and JavaScript unit tests across over 200 mobile and desktop browser/OS platforms. There are more than 150 000 users of Sauce Labs (SauceLabs, 2008a). Sauce Labs is a huge supporter of open source community consisting of projects such as Appium, Monocle, and Selenium builder (SauceLabs, 2008a). There are also other providers that support Selenium automated tests in cloud such as Nerrvana, Testbot and Soasta.

Selenium is a software testing framework that supports the test automation for testing the functionality of the web browsers. It consists of tools like Selenium Grid, Selenium IDE, Selenium Webdriver, and Selenium RC. With this availability of tools, testers select one according to their project needs. Selenium supports the execution of tests on multiple browser platforms (SauceLabs, 2008c). A Selenium test on cloud expands the scope of testing that can run hundreds of tests in parallel to faster testing. It can test the key functionalities, features and cross platform compatibility with the automated test suites. While running the test, with the help of breakpoints one can manually stop the test session and investigate the problem (SauceLabs, 2008b). Additionally, a console log for JavaScript errors are provided when the test against Google chrome is carried out. For the easier debugging, a live video and screen-shots are also provided for every test on the cloud. Moreover, local and firewall testing support a secured way of connection through Sauce connect which is a combination of secure tunneling and smart caching (SauceLabs, 2008b). Sauce Labs claims to take security concerns very seriously and thus have built many industry standards security measures, so that customers can test their products with confidence. They have the facility of fiber connections for the networks to the company that is more secured than standard Internet connection. Sauce Labs uses Ubuntu LTS, Secure Shell (SSH) network protocol and every time a fresh VM for the tests (SauceLabs, 2008d).

5.3 Other testing types

Along with the performance and functionality of the system, there are other testing types available as well that can be performed in the cloud. The non-functional testing is the most affected ones because it usually requires a huge investment in hardware and software license and it consists of a variety of testing types (Massimo, Forno). Also, these tests are expensive if the software needs to be fixed after the production phase. This testing consists of performance, security, compatibility, usability, installation test-

ing etc. We have already discussed performance testing in Section 5.1 and have seen that it is one of the main test types for which the cloud can offer great benefits.

Compatibility testing checks whether a system is able to run on different operating systems, hardware, software, browser, network, and database. In this type of testing, the test engineers need to define the environment or platform that the system is expected to work. For this, they need to set up the environment with different configurations of hardware, software, bandwidths etc. Moving this testing to cloud can reduce costs by avoiding the need to buy license for every operating system. It can also increase the speed of creating multiple operating systems with the required configurations with the virtualized techniques (Massimo, Forno).

Security testing is a type of non-functional testing that is performed to check if the system is vulnerable to any attacks. There might be the case where the third party might hack the system and easily access all the information without any authorization. So, there must be some way to protect your data and information. Executing this test on cloud ensures all the essential information to be stored in highly secured manner. Basically, the security mechanism must be tested in three dimensions: effectiveness, accuracy and performance. Testing the effectiveness of a security system checks whether the security mechanism can defend against the attacks, accuracy testing checks whether the system produces any false positives and performance testing checks whether the system can pass the acceptable amount of traffic. The security devices for this testing should be able to cover the things such as inspecting for malware, fending off the service attacks, encrypted traffic, and data leakage testing (Cloud, 2011). With the rapid evolution of virtualization technologies, many organizations are embracing private cloud (internal shared services) along with this technology for better security. In addition, it is even easier to manage security in the cloud as there is a single point of control over multiple systems leading to better business continuity and easy recovery to a system (Security, 2013). Moreover, the cloud approach of testing would be safer and efficient when compared with traditional internally managed test environments.

Usability testing is another important testing that can never be ignored. It is a means of assuring whether intended users can carry the task efficiently, effectively and with satisfaction (Gaffney, 1999). It often happens that the developers become blind to the usability defects or because of being too familiar with their own product they can easily get around the problems. However, customers can find it difficult to use the application. According to Gaffney (1999), usability testing should be carried out in various stages of design processes or at least during the pre-release so that any issues identified can be addressed earlier (Gaffney, 1999). Usability testing is also available in the cloud providing fastest and cheapest approach to testing. For example, UserTesting is a company that offers usability testing in the cloud. They have freelance testers available who tests the application and their test session video is captured along with their comments (Usability, 2012). Now the website owners can watch the recorded test session and improve their usability problems. Thus, usability testing can be better performed in cloud consuming less time and in a cheaper way.

5.4 Pricing models

Today software industry demands high quality software products with reduced cost and timeframe. With the traditional test approach it would be very challenging to maintain this. Cloud testing service offers organizations a possibility to use the tools with reduced cost and time which allows them to deliver a better quality product. Most particularly for the start-ups and small organizations, selecting the services and associated tools could be one of the difficult decisions to be made. There are various pricing mechanisms available according to the demands of the customers. Mostly the pricing in the cloud testing is based on demands and subscription plans. The on demand instances lets you pay only for the resources that have been used and are billed accordingly. This payment procedure has no long term commitments and reduces the complexities of planning, purchasing and maintaining the expensive resources into variable costs. Subscription plan is a payment plan made on the basis of an agreement in order to receive the services. Generally the service providers for the cloud testing provide monthly subscription.

BlazeMeter, for the load testing in cloud provides two pricing strategies that are, monthly plans and pay per test packs. They claim monthly plan to be their most popular plan and are perfect when organizations needs to be changed from month to month. As shown in **Table 5.3**, the monthly subscription plan of BlazeMeter is categorized into basic, medium (pro, pro plus and hi volume) and enterprise levels of subscriptions. In this plan, if some test hour is left over then it gets rollover to the next month subscription (BlazeMeter, 2013f).

Table 5.3. *Sample price list for a monthly subscription of BlazeMeter (BlazeMeter, 2013f)*

Factors	**Basic**	**Medium**	**Enterprise**
Max Concurrent Users	1 000	10 000	100 1000
Max Load Server(IPs)	1	10	100
Test Hours Per Month	20	25	Unlimited
Maximum test duration	1	6	Unlimited
Price	$199	$999	Call (requirement basis)

The other plan is pay per test (on-demand) which is perfect for occasional load testing. It also consists of basic, medium and hi volume levels of plan as shown in **Table 5.4**. This is different from the monthly subscription plan in a sense that no commitment is needed for this plan and the test credits are valid for 30 days (BlazeMeter, 2013f). The plans involved in BlazeMeter (monthly and pay per test) do not require any setup fees and the test should be started within minutes. According to one of the members of BlazeMeter, the price for BlazeMeter gets updated at the end of every year. The pricing provided by other service providers is also similar to the above scheme. It mainly depends upon different factors like maximum user load, the size of the website, test dura-

tion, the number of credits etc. For example, Load Impact offers 5 credits per month for free, after registering to their site (LoadImpact, 2013d).

Table 5.4. *Sample price list for pay per test (on-demand) of BlazeMeter (BlazeMeter, 2013f)*

Factors	**Basic**	**Medium**	**Hi Volume**
Maximum Concurrent Users	1 000	10 000	40 000
Max Load Servers(IPs)	1	10	40
Max Test Duration(Hr)	1	1	1
Price	$19 per test (5 test minimum order)	$149 per test (3 test minimum order)	$299 per test (2 test minimum order)

The pricing list of Neoload is slightly different from other providers. It differs according to End-User Permanent or Rental License. The cost for these license types also depends upon the configuration elements such as the license term, number of virtual users required and the number of optional modules selected. The modules include the protocol module list (Flex, GWT, Java Serialization module, SOAP etc.) or Monitoring, Integration and Collaboration module list (Application Servers, Databases, Network). The rental license depends upon the daily rental or term license (consecutive days). The minimum number of days for the daily rental service is 4 days and it should be used within one year of purchased date. Once the license is activated, it can be used for 10 * d days, where d is the number of days rented. Apart from these conditions, the pricing lists are similar to other service providers depending upon the customers demand (Neotys, 2013f).

Table 5.5. *Sample price list from Sauce Labs services*

Features	**Manual**	**Automated**	**Small team**
Windows/Linux/Android	200 min	1000 min	4000 min
Mac/iOS	80 min	400 min	1600 min
Parallelization	4 VMs	5 VMs	10 VMs
Price	$12/month	$49/ month	$149/ month

The pricing strategy for functional testing is slight different from the performance testing. Sauce Labs provides monthly plans as shown in **Table 5.5**. It depends upon the operating systems used and the number of parallel VMs used for simultaneous testing. If the user uses more than the allocated plan, then the subscription will not be closed. Instead an extra charge will be added per minute. The subscription level can be changed at any time if required. Sauce Labs also provide an enterprise level plan where the user gets features such as unlimited testing minutes, live implementation and training and up to 50 parallel VMs.

Different service providers have different strategy for the pricing, but the most important thing is that services are available according to the client's demand and are very flexible.

6. ANALYSIS OF BENEFITS AND POTENTIAL PROBLEMS

Before moving to the cloud, it is necessary to have a good understanding about testing in a cloud environment. Organizations should know the benefits as well as problems that they might come across while testing different types of applications in the cloud. Sections 6.1 and 6.2 discuss the benefits and potential problems of cloud adoption.

6.1 Benefits

This section introduces the importance of cloud computing to testing and how it can be efficiently utilized for testing in the cloud. Cloud testing can help business grow from different aspects such as setting and maintaining the infrastructure, lowering the cost, providing on-demand service, rapid elasticity, and most importantly, it reduces the time of the overall development cycle. The titles of sub-sections give a brief description explaining how cloud can be beneficial to the organizations.

6.1.1 Reduced cost for the infrastructure

The cost is one of the important aspects to be considered by the organizations. All the organizations would want to deliver a quality product trying to lower the overall IT expenses. According to Shajee (Shajee, 2012), cost is the main factor that needs to be reduced for the customers to maintain the infrastructure and software. So, cloud testing can be a better solution for the organizations to keep up with. It can eliminate the need to invest in stand-alone servers for hardware, software and licensing fees. Organizations can also save some money on computing resources, licensing fees and the maintenance cost as all these things will be managed by the cloud provider.

Riungu et al. (2010) points out that online software testing service would help to cut down cost as it is available as on-demand basis. This would reduce the man hour cost for infrastructure and hardware cost along with the licensing fees. The cloud shifts the task of IT resource management and all the risks that are associated with cloud to the vendors. This means that, vendors will be responsible for any disturbances caused in the cloud systems. According to Riungu-Kalliosaari et al. (2012), it is very easy to use multiple servers (open-source stacks or even licensed servers) with cloud where only the testing software needs to be installed and testing can be started. These responsibilities are handled by the cloud providers and they assure to make a financial benefit to the customers (Shajee, 2012). The set up infrastructure will be handled by the cloud when

testing needs to be performed and later it can be decommissioned, which helps organizations to save money (Riungu-Kalliosaari et al. 2012). This is different from the traditional approach where testing was performed in an internal environment full of many servers with the variety of operating systems in a room. The traditional approach required a huge amount of investment and, above all, these servers had to stay idle when testing were not performed.

Cloud testing also provides the benefits with on-demand resources available. With this feature, organizations can carry out testing without the need to purchase any additional expensive hardware for supporting web environments. The tests generally depend on the different features like the number of virtual users, the number of times the test is performed and many others. So depending on the feature, the cost can be compromised. In addition, there are various ways to perform testing in the cloud. Cloud tools are mostly delivered as SaaS applications, but it is not necessary that organizations have to use it that way. Organizations can use IaaS to conduct their tests and thus will only be charged of utilized test duration. In this way, organizations can save their hardware related costs (setup, installation, required floor space and maintenance) increasing the operational efficiency. Therefore, cloud testing provides a cost effective solution with high return on investments.

6.1.2 Convenience and better availability of tools

Cloud services allow the user to access data from anywhere and anytime. With this feature, user can easily access the information, view it, modify as necessary, and save it back to the original location to be available. Cloud computing helps the organizations by providing various options for test tools and environments that best suit them (Riungu-Kalliosaari et al. 2012). With this choice, most of the providers also have the option to use full featured tools with a limited trial period. This allows the users to test their application and check the compatibility with the tool before purchasing the full version. In addition, these tools can be purchased with "on-demand" meaning that the organizations can use them whenever needed without having to think about the availability of computing resources. They can use the tools in hourly basis, monthly basis or based on the number of tests performed and do not have to purchase a yearly license for the testing tool. Moreover, this kind of readily available testing tools increase the agility in the testing and shortens the overall development cycle with a faster time to market of software products (Riungu-Kalliosaari et al. 2012).

Cloud computing globalizes software testing and provide a chance for providers to serve customers throughout the world. Clients can also compare different providers' tools and use them accordingly (Riungu et al. 2010). In addition, everything is easier to handle and services can be rapidly provisioned through the tools which leads to easier management in cloud. It also provides regular upgrade and maintenance of the software handled by the cloud providers (Shajee, 2012). Another thing that comes up with convenience is easy setup of the testing tools. As most of the cloud applications use SaaS model, very little or no software or hardware is needed. It is mostly accessed through a

browser or thin client. So this alone makes SaaS testing tools more convenient with less testing time and effort compared to the traditional testing approach. So, the feature of instantly accessible solution and on-demand nature of SaaS applications is the epitome of convenience. In addition, there can be several service providers for the same testing tool that are available in cloud. These features allow customers to easily compare the price, service level agreements and quality of different service providers.

6.1.3 Flexibility

In the IT industry, testing has always been viewed as a challenge because of the sporadic infrastructure and the resource it demands. This has made difficult for the organizations to simulate the real world environment and achieve flexibility, even after working with the virtualization technology. To give IT industry, a growth and to achieve flexibility cloud testing has arrived to mitigate the difficulties encountered in traditional infrastructure (Maruvada, 2012). With cloud testing, it is easy and quick to reach the maximum demands of customers and thus provide the whole process with flexibility, on-demand service, and cheaper to be resized, eliminating the overheads.

Cloud-based testing provides flexibility for the testing professionals both inside and outside of the workplace. With the help of cloud, they can easily access the documents and applications from any device such as smartphones, laptops and tablets supporting the internal and external collaboration. Using the cloud services, the customers can start and stop online services as needed. This can lead to a tremendous benefit for the organizations allowing them to save their infrastructure cost. Also, there might be some situations where organizations running their IT resources in private clouds might need extra resources at peak hours. In this case, there will not be a problem of getting the additional capacity at the very moment from the cloud providers. Using traditional solutions this might be problematic. The cloud also provides flexibility in terms of price as it does not require customers to purchase a full package. They can simply use the resources or applications for the time they need themselves and pay only for the used services. Additionally, business can scale up and down in terms of computing resources depending upon the customers need and situation that suits them, allowing flexibility for the change required.

6.1.4 Fast, scalable and efficient testing

According to Riungu-Kalliosaari et al. (2012), using the cloud computing services, one can easily test the developed application within no time using different environments and provision the results available with much less time and effort. This process shortens the testing time and leads to a faster development of the whole application. Also, organizations need not worry about the infrastructure setup and focus more on actual business. In other words, this can increase the mental concentration in the work and bring up improvement in the quality of development and testing.

The cloud services are often based on pay-as-you-go basis which means the computation is based on the virtualization level. For example, Google AppEngine charges on the basis of cycles used which is determined by the load increase or decrease (Amburst et al. 2010). This feature of cloud can scale quickly in response to the traffic and data storage. Ambrust et al. (2010) points out that an idle computer uses two-third of the power of a busy computer which could be conserved with the concept of cloud computing (Amburst et al. 2010). This motivates the consumers to concentrate more on efficiency that is, releasing and acquiring only the resources that are necessary, thus helping in scaling resources as well as money.

Cloud-based testing can also generates a realistic test environment giving realistic test result. It allows working with multiple operating environments allowing the users to test from different perspectives and giving the feedbacks. In the traditional approach, testing is generally done in internal test environment which lacks real test traffic. For example, in the case of performance testing it is important to figure out the number of users a site can handle and its behaviour with increased load before moving it to production. Cloud testing can easily handle this type of testing simulating the real test environment and giving client the worst case results. In addition, it also reduces the maintenance requirements as this is done through APIs without having to install the system (Cloud, 2010).

The results for the testing turn out to be of a better quality as most of the important things are managed by the IT specialist in the cloud (Forno, 2012). Also, there will be faster test executions where many tests can be covered within a short interval of time. In addition, time and effort for deploying new applications and upgrading them can be minimized. The cloud also gives an opportunity for business organizations (either small or large) to have a managed data centers. Also, testing tools and other test materials can be managed properly in some cloud management tools to gain stability and security. This feature will certainly help in the future if testing has to be performed again for different applications. Using the cloud technology, the burden for all the infrastructure management matters is lifted to someone else, thus increasing the efficiency of testing.

6.1.5 Improved communication

Cloud based testing can maintain a proper sink between the developers and the testers throughout the project. As both of them will be in the cloud having equal access to the system, they can develop, modify, test and be updated with the latest status of the project. This can improve the interaction between the developers and the testers and helps the tester team to immediately follow the developers' action (Riungu-Kalliosaari et al. 2012). Because of the agile approach empowered by the cloud testing, organization can easily address the need for customers (Riungu-Kalliosaari et al. 2012). They can build the system according to the requirement and customers can also check and provide their immediate feedback. Using this kind of technology also changes the way software is delivered and thus improves the communication with the end users.

According to Maruvada (2012), the cloud service provides access to required resources whenever needed and can be used from any part of the world. So, no matter from whichever part of the world people are working from, cloud testing can provide efficient and more thorough testing with improved communication. In addition, Riungu-Kalliosaari et al. (2012) points out that cloud computing also improves the interaction between vendors. If one vendor is running out of resources then they can hire resources from another vendor.

Cloud environment can give an efficient collaboration to the business. It helps to communicate and share information easily across different locations. A testing team needs to begin the collaboration at an early stage of test life cycle. This enables the team to share on-demand status and solve problems encountered together. The more efficiently teams collaborate; the quicker organizations can reap financial benefits. In addition, this also reduces the risk during the entire software development and testing lifecycle and improves the overall quality of the end application.

6.2 Potential problems

Cloud computing has become one of the exciting and fastest growing segments of IT industry. However, there are other issues such as security, performance and vendor quality to be faced by the cloud, as it absolutely depends upon the network connection. This section discusses the challenges that the organizations might encounter while moving to the cloud.

6.2.1 Security

Security issues are a major concern for every organization, especially the one related to test data management (Riungu-Kalliosaari et al. 2012). We discussed earlier that the data storage in cloud can be considered "physically" safe and there is no need to take backups but still, security threats remain. Since the applications and data will be stored by third party vendors, it is a great challenge for the organizations to have the confidence and trust them for the client's code to be secured. It is just like running your applications on someone else's hard disk which is a risky job. With this, the organizations might encounter the security issues such as data loss, phishing and data integrity (Dillion, 2010).

In a virtualized environment, it is more complex to handle the security as it needs to be handled in both the physical and the virtual machine. This means if something goes wrong with the physical machine, then all the virtual machines residing in it gets unsecured (Ertal et al. 2014). In the public cloud, both the users and providers might face security risk such as unauthorized access, malware infection and loss of sensitive information (Balduzzi et al. 2011). With these risks, the security provided in public cloud is still considered as a big issue especially due to insufficient encryption techniques. So, before moving to public cloud it is important to check if the private keys of encryption are shared with other users. Also, sometimes there might be a situation where the pri-

vate information is leaked because of sudden disruption of service (Malhotra & Jain, 2013).

Some of the organizations have to perform testing depending on the customers data that might be highly confidential. So, there is a big risk for the customers to provide their private data on cloud because of the insufficient protection. These types of organizations will have to think twice before moving on to cloud and select the cloud provider very carefully (Malhotra & Jain, 2013). However, for smaller organizations, data may be safer with the cloud provider rather than on their own premises. As the violation of data security in the cloud is a paramount concern to an organization, storing and handling of the data safely should never compromise confidentiality (Riungu-Kalliosaari et al. 2012).

6.2.2 Performance

Cloud testing highly depends upon the speed of internet connection and it is certain that slow connectivity will lead to performance degradation. This might affect the test results as the service providers might have varying connection speed (MacVittie, 2011). Researchers from Australia found out that the stress testing offered by three different service providers, Amazon's EC2, Google's AppEngine and Microsoft's Azure have generated variation by a factor of 20 in the response time. This research was conducted with the simulation of 2000 concurrent users to measure how the cloud providers would scale and response to the demand (Winterford, 2009). This shows that there might be variance in performance and availability due to load change.

Organizations might also face latency issue if the test is performed under different cloud provider's environment (cross cloud bottlenecks). This might, however, be expensive as organizations may need to pay double for the throughput (Performance, 2009). Organizations running their tests in private cloud also face latency issue as the cloud is shared by many users. Application performance in cloud can vary significantly based on individual's bandwidth, disk space, memory, and network connections. Sometimes, lack of proper resources also leads to poor application performance. Additionally, if the internet connection is slow it reduces the productivity of the employee and efficiency of the testing also goes down. Nevertheless, it is difficult to predict the linearity of the performance in the cloud.

6.2.3 Quality of cloud vendors

After shifting to cloud, organizations need not worry about computing resources rather they can focus on the quality, availability, reliability and performance of the resources. So, it is very important for the companies to have the guarantee on service delivery. This is generally provided through service level agreement that is negotiated between customers and service providers. It is very challenging for the service providers to provide such kind of agreement as they might have to face many implementation problems (Dillon et al. 2010). However, this would help in an increase in performance as it con-

tains an expected range of availability. Some vendors offer only limited types of configurations often making it difficult to create real-time test environments.

Sometimes there might be a problem of site inspection or auditing as the exact physical location of hardware and software is unknown. There is also the case that service providers' system has some technical problem where the whole server goes down for a while. In this case, the cloud customers might be in a big trouble as the failure would take down a number of important sites on the internet. With these complexities faced by the vendors, it might be difficult for them to handle the growth and give it a continuity focusing on better quality. Furthermore, the assessment of the quality of the vendors is a complicated thing and in this thesis there was only a possibility to discuss just a couple of issues related to that and it would need some proper research and development of guidance for companies (both the customers and the cloud testing service providers).

6.2.4 Lack of standards

Because of continuous attention to cloud computing to the IT business, many companies have already embraced this technology, while some of the companies are still in chaos due to lack of industry standards (Hayner, 2011). According to Riungu (Riungu et al. 2010), cloud providers have a challenge of providing a standard pricing models that meet the customers' expectations. Without having a standard process, it is difficult for the customers if they have to compare a product with another.

Each service providers operate in their own way having their own architecture, pricing charts and other models which might create an interoperability issue (Dillon et al. 2010). So, until and unless proper standards are developed regarding the pricing models, security handling, data handling and an education methodology, there will still be a lot of confusion. In addition, having standards will give the customers better knowledge and help them concentrate on the core competencies (Hayner, 2011). Also, during the research it was found that the information provided by the service providers was not consistent in the website. For example, some of the providers did not completely reveal their price in their websites and most of them did not have proper security methods in cloud. Therefore, standards help customers to make the best choice that suits them and play a huge role in the adoption of any new technology.

7. CONCLUSIONS

Cloud computing has brought a new vista of opportunity for testing, but is it good to migrate in the cloud? Organizations are still in great dilemma moving testing to the cloud system. There are many things that make testing in the cloud environment attractive and better than the traditional testing approach, but at the same time it has many issues and challenges to be faced as discussed in Section 6.2.

Because the IT industry always keeps on changing with the latest technology, organizations should also be on the same wheel to meet the demand of the customers. With the change in technology, customers demand also increases and organizations should be ready to fulfil their needs. They would not want to use any obsolete technology and keep themselves in a dark side. Considering on these facts, many business executives are taking advantage of cloud environments to enable a transformation in their business.

Different people have different views on cloud computing. Some of them take the maximum benefit from the cloud where as some think it is unsafe to use cloud. However, the main thing that needs to be considered is that the organizations have to choose the most reliable service provider who can keep the information totally secured. With the right provider, customers can keep their data safe in the cloud. They can also choose hybrid cloud to leverage the best of both the cloud computing; private cloud for confidential data and public cloud for other data. In addition, migration needs to be done with proper planning and they need to check the right vendor who fits their business needs and those providing a proper service level agreement. For most of the organizations, 100% cloud is not feasible; it highly depends upon the business and their project types.

According to Riungu et al. (2010), many software products are moving from traditional desktop applications to online services, and the same trend is expected for software testing as most of the organizations have already stepped their foot in cloud. Before moving to cloud, organizations should analyse the infrastructure by starting with a pilot project for a specific workload and other test conditions. Cloud offers numerous advantages like virtualized hardware, unlimited storage and applications as a service in different platforms. However, migrating to cloud is not that easy or the best solution to be considered. The testing techniques still need to be improved in terms of innovation, security, quality of service standards etc. So, it can be said that the cloud works only if it is well planned and the users should know how to exploit its benefits.

In this thesis, we have discussed many tools and their service providers for performance and functional testing and it is seen that the cloud testing is often used as perform-

ance or load testing rather than functional testing. There are many automated tools available for functional testing and most of the organizations are already using them, but not all of the tools are implemented in a cloud environment. This is subjected to change with the growing research in cloud testing as the topic is rather new among research scholars. Many organizations have already adopted performance tools for testing web and mobile applications in the cloud. These tools are mostly implemented using software as a service model where no installation is required. Most of the service providers use SaaS model to deliver their application, but organizations can also use PaaS or IaaS models for their testing, if they have their own testing tools. Moreover, for some of the testing tools, it is not provided by the owner of the tool, instead it is preconfigured in another cloud provider's machine (for example Amazon EC2) and is purchased through that provider. In any case, despite its complexities, this approach of testing meets the modern quality needs of the system compared with the traditional approach of testing.

With the cloud, any application can be tested, including mobile applications, but this was out of the scope of this thesis. In addition, there might be other tools available in cloud for performance, functional and other type of testing apart from the ones listed in Chapter 5. All of the tools could not be covered because of the limited scope of thesis. Cloud testing is a huge topic to be covered and can lead new research directions to future work. Considering the large benefit of cloud testing for small and medium sized companies, its issue can be resolved if cloud providers work out with some restrictions to make their products best according to the customer's demand. In many cases, organizations will have to find a balance between security, performance and cost, as not all may be practically achievable at the same time. Hence, in the future, cloud testing will no doubt smoothen the rough edge it currently has and will continue to gain popularity.

REFERENCES

Agile. 2013. Agile Methodology. Accessed on 09.12.2013. Available at: http://agilemethodology.org/

Armbrust, Michael; Fox, Armando; Griffith, Rean; Joshep, Anthony D; Katz, Randy; Konwinski, Andy; Lee, Gunho; Patterson, David; Rabkin, Ariel; Stoica, Ion & Zaharia, Matei. 2010. A view of cloud computing. Publication: COMMUNICATIONS OF THE ACM. Volume 53, Issue 4, pp. 50–58.

Aymerich, Franceso M; Fenu, Gianni & Surcis, Simone. 2008. An Approach to a Cloud Computing Network. Publication: IEEE. pp. 113-118.

Balduzzi, Macro; Zaddach, Jonas; Balzarotti, Davide; Kirda, Engin & Loureiro, Sergio. 2011. A Security Analysis of Amazon's Elastic Compute Cloud Service -Long Version-. Publication: ACM. 10p. Available at: http://www.iseclab.org/people/embyte/papers/securecloud.pdf

BlazeMeter. 2013a. About BlazeMeter. Accessed on 13.06.2013. Available at: http://BlazeMeter.com/about-BlazeMeter

BlazeMeter. 2013b. Features of BlazeMeter. Accessed on 13.06.2013. Available at: http://BlazeMeter.com/features

BlazeMeter. 2013c. Data Security Policy. Accessed on 13.06.2013. Available at: http://BlazeMeter.com/data-security-policy

BlazeMeter. 2013d. BlazeMeter's Plug-in for JMeter. Accessed on 13.06.2013. Available at: http://BlazeMeter.com/BlazeMeters-plug-jmeter

BlazeMeter. 2013e. BlazeMeter's Plug-in for JMeter. Accessed on 13.06.2013. Available at: http://community.BlazeMeter.com/knowledgebase/articles/83191-BlazeMeter-plugin-to-jmeter

BlazeMeter. 2013f. BlazeMeter's Pricing. Accessed on 10.12.2013. Available at: http://BlazeMeter.com/pricing

Chittanai, Khamer. 2012. Moving Testing to the Cloud- An Exploration. Testing Experience. December 2012. p. 27-29.

Cloud. 2010. Introduction to Cloud Computing. Accessed on 08.01.2014. Available at: http://www.dialogic.com/~/media/products/docs/whitepapers/12023-cloud-computing-wp.pdf

Cloud. 2011. Testing the Cloud: Definitions, Requirements, and Solutions. Accessed on 28.09.2013. Available at: http://www.ixiacom.com/pdfs/library/white_papers/cloud_testing_white_paper.pdf

Cloudbook. 2013. Cloud Platforms. Accessed on 17.06.2013. Available at: http://www.cloudbook.net/directories/product-services/cloud-computing-directory?category=Platforms&type=Test%20Environments

Cloud. 2012. Cloud application virtualization accelerator. Accessed on 11.12.2013. Available at: http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=SP&infotype=PM&appname=GBSE_GB_AM_USEN&htmlfid=GBS03134USEN&attachment=GBS03134USEN.PDF

Cloud. 2013. Cloud based performance testing with Agile Load. Accessed on 06.01.2014. Available at: http://www.agileload.com/video-center#prettyPhoto/5/

Compuware. 2013a. About Compuware. Accessed on 23.09.2013. Available at: http://www.linkedin.com/company/compuware?goback=%2Enmp_*1_*1_*1_*1_*1_*1_*1_*1_*1_*1&trk=nmp_rec_act_company_photo

Compuware. 2011a. About Gomez. Accessed on 23.09.2013. Available at: http://apcloud.dimensiondata.com/sites/default/files/content/gomez-case-study.pdf

Compuware. 2011b. Load test report of CNN webpage provided by Gomez. Accessed on 23.09.2013. Available at: http://realityload.gomeznetworks.com/FreeTestReport.aspx?key=w6o9LtYkIDezOWZ8itglcQ2&status=complete

CustomerCentrix. 2013. Load Testing Company. Accessed on 17.06.2013. Available at: http://loadstorm.com/load-testing-company/

Dillon, Tharam; Wu, Chen & Chang, Elizabeth. 2010. Cloud Computing: Issues and Challenges. Publication: 24th IEEE International Conference on Advanced Information Networking and Applications. pp. 27-33.

Doddavula. K. Shyam & Gawande. W. Amit. 2009. Adopting Cloud Computing: Enterprise Private Clouds. Publication: SETLabs Briefings. VOL 7. NO 7. pp. 11-18.

Eickelmann, Nancy S & Richardson, Debra J. 1996. An Evaluation of Software Test Environment Architectures. Publication: IEEE Proceedings of ICSE-18. pp. 353-364.

Ertal, L; Singhal, S & Sldamli, G. 2014. Security Challenges in Cloud Computing. 7p. Available at:
http://www.mcs.csueastbay.edu/~lertaul/Cloud%20Security%20CamREADY.pdf

Forno, Massimo. 2012. Software Testing: Adapt and Growth with the cloud. Testing Experience, December 2012. p. 19-20.

Fujitsu. 2010. Confidence In Cloud Grows, Paving Way For New Levels Of Business Efficiency. Accessed on 20.11.2013. Available at:
http://www.fujitsu.com/uk/news/pr/fs_20101116.html

Functional. 2013a. Functional Testing. Accessed on 02.12.2013. Available at: http://www.st.cs.uni-saarland.de/edu/testingdebugging10/slides/01-Functional%20Testing.pdf

Functional. 2013b. Functional-Testing. Accessed on 02.12.2013. Available at: http://www.infosys.com/engineering-services/white-papers/Documents/functional-testing.pdf

Gaffney, Gerry. 1999. Usability Testing. Accessed on 28.12.2013. Available at: http://infodesign.com.au/wp-content/uploads/UsabilityTesting.pdf

Gao, Jerry; Bai, Xiaoying & Tsai, Wei-Tek. 2011. Cloud Testing- Issues, Challenges, Needs and Practice. Publication: SOFTWARE ENGINEERING: AN INTERNATIONAL JOURNAL (SEIJ). pp. 9-23.

Gartner. 2010. Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond. Accessed on 02.08.2013. Available at:
http://www.gartner.com/newsroom/id/1278413

Halili, Emily. 2008. Apache JMeter: A Practical Beginner's Guide to Automated Testing and performance measurements for your websites. Publication: PACKT. 140p.

Hayner, Louis. 2011. Lack of Standards in the Cloud. Accessed on 14.01.2014. Available at: http://www.telecomreseller.com/2011/06/22/lack-of-standards-in-the-cloud/

Jadhav, Varsha. 2012. Testig Applications in the Cloud - a Tactical Approach. Testing Experience, December 2012. p. 37-41.

Janova. 2011a. About Janova. Accessed on 18.11.2013. Available at: http://www.janova.us/index.php/about-us

Janova. 2011b. About Janova. Accessed on 18.11.2013. Available at: http://www.janova.us/PDFs/JANOVA%20Brochure%20.pdf

Janova. 2011c. How Janova works. Accessed on 18.11.2013. Available at: http://www.janova.us/index.php/products/how-janova-works

Janova. 2011d. Why Janova. Accessed on 18.11.2013. Available at: http://www.janovasoftware.com/why-Janova.php

JMeter. 2013. The Apache Software Foundation. Accessed on 14.06.2013. Available at: http://jmeter.apache.org/index.html

Keynote. 2012. America's Best Small Companies. Accessed on 10.06.2013. Available at: http://www.forbes.com/best-small-companies/list/

Keynote. 2013. Company. Accessed on 05.06.2013. Available at: http://www.keynote.com/company/

Keynote. 2013. Keynote At-a-Glance. 4p. Accessed on 05.06.2013. Available at: http://www.keynote.com/docs/datasheets/Keynote_Overview.pdf.

Keynote. 2013a. Test Perspective. Accessed on 05.08.2013. Available at: http://www.keynote.com/products/web_load_testing/load-testing-tools.html

Keynote. 2013b. Keynote Test Perspective 3.0. 4p. Accessed on 05.06.2013. Available at: http://www.keynote.com/docs/datasheets/DS_Test_Perspective.pdf

Keynote. 2013c. LoadPro. Accessed on 05.08.2013. Available at: http://www.keynote.com/products/web_load_testing/load_pro.html

Keynote. 2013d. Keynote LoadPro: Fully Managed Web Load Testing Service. 4p. Accessed on 05.06.2013. Available at: http://www.keynote.com/docs/datasheets/key_loadpro.pdf

Kumar, Satish. 2012. Cloud Performance Testing - Whitepaper. AgileLoad blog. 10p. Available at: http://www.agileload.com/agileload/blog/2012/11/15/cloud-performance-testing.

Liu, Feng; Guo,Weiping; Zhao,Zhi & Chou, Wu. 2010. SaaS Integration for Software Cloud. Publication: IEEE COMPUTER SOCIETY. 8p.

LoadImpact. 2013a. General Imformation on LoadImpact. Accessed on 20.08.2013. Available at: http://www.crunchbase.com/company/load-impact

LoadImpact. 2013b. About LoadImpact. Accessed on 20.08.2013. Available at: https://loadimpact.com/company/about#company

LoadImpact. 2013c. LoadImpact Features. Accessed on 20.08.2013. Available at: http://loadimpact.com/features

LoadImpact. 2013d. Price of LoadImpact. Accessed on 21.08.2013. Available at: https://loadimpact.com/pricing

MacVittie, Lori. 2011. Cloud testing: The next generation. Accessed on 14.01.2014. Available at: http://www.networkworld.com/news/tech/2011/020911-cloud-testing.html?page=1

Malhotra, Rahul & Jain, Prince. 2013. Testing Techniques and its Challenges in a Cloud Computing Environment. Publication: The Standard International Journals. p. 88-93. Available at: http://www.thesij.com/papers/CSEA/2013/July-August/CSEA-0103550201.pdf

Maruvada, Narayana. 2012. Testing in the Cloud. Testing Experience, December 2012. p. 31-33.

Mell, Peter & Grance, Timothy. 2011. The NIST Definition of Cloud Computing. USA. National Institute of Standards and Technology. Accessed 25.9.2013. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Mirońska, Aneta. 2012. Gobal performance testing with Gomez. Accessed on 18.11.2013. Available at: http://www.cognifide.com/blogs/mobile/global-performance-testing-with-gomez/

Neotys. 2013a. Company Summary. Accessed on 19.09.2013. Available at: http://www.knowledgestorm.com/prov_summary_2693536.asp

Neotys. 2013b. Company Overview of Neotys SAS. Accessed on 19.09.2013. Available at:http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=22127125

Neotys. 2013c. Company History of Neotys. Accessed on 19.09.2013. Available at: http://en.wikipedia.org/wiki/Neotys

Neotys. 2013d. Overview of Neoload. Accessed on 19.09.2013. Available at: http://www.neotys.com/product/overview-neoload.html

Neotys. 2013e. Overview of Neoload. Accessed on 20.09.2013. Available at: http://www.neotys.com/product/neoload-cloud-testing.html#!prettyPhoto

Neotys. 2011. Load Testing with the Cloud . A Neotys White Paper. 11p.

Neotys. 2013f. Pricing List for Neoload. Accessed on 23.09.2013. Available at: http://www.neotys.com/store/pricespermanent.html

Parveen, Tauhida & Tilley, Scott. 2010. When to Migrate Software Testing to the Cloud?. Publication: IEEE COMPUTERSOCIETY. p. 424-427.

Performance. 2009. Performance testing in the cloud. Accessed on: 14.01.2014. Available at: http://h20195.www2.hp.com/v2/GetPDF.aspx%2F4AA3-0545EEW.pdf

Performance Testing. 2013. Accessed on 02.09.2013. Available at: http://www.guru99.com/performance-testing.html

Perry, William & Rice, Randall. 2013. Surviving the Top Ten Challenges of Software Testing: A People-Oriented Approach. 202p.

Riungu-Kalliosaari, Leah; Taipale, Ossi & Smolander, Kari. 2012. Testing in the Cloud: Exploring the Practice. Publication: IEEE COMPUTER SOCIETY. 6p.

Riungu, Muthoni; Taipale, Ossi & Smolander, Kari. 2010. Software Testing as an Online Service: Observations from Practice. Publication: IEEE COMPUTER SOCIETY. 6p.

Rouse, Margaret. 2011. Virtual Machine (VM). Accessed on 11.12.2013. Available at: http://searchservervirtualization.techtarget.com/definition/virtual-machine

Sahoo, Kulamani. 2013. An Overview of Testing In Cloud. Accessed on 11.12.2013. Available at: http://www.slideshare.net/imkulu/white-paper-on-testing-in-cloud

SauceLabs. 2008a. About Sauce Labs. Accessed on 01.12.2013. Available at: http://www.crunchbase.com/company/sauce-labs

SauceLabs. 2008b. Features of Sauce Labs. Accessed on 01.12.2013. Available at: https://saucelabs.com/features

SauceLabs. 2008c. Introduction to Selenium. Accessed on 01.12.2013. Available at: http://www.seleniumhq.org/docs/01_introducing_selenium.jsp

SauceLabs. 2008d. Security in the Sauce Labs Cloud. Accessed on 01.12.2013. Available at:
http://info.saucelabs.com/rs/saucelabsinc/images/sauce_security_white_paper.pdf

Security. 2013. Virtualization Cloud Security. Accessed 28.12.2013. Available at: http://www.sans.org/course/virtualization-private-cloud-security

Shajee, Sujith. 2012. Simplifying Testing Cloud and Cloud Testing. Testing Experience, December 2012. p. 16-17.

SmartBear. 2010a. About SmartBear. Accessed on 18.11.2013. Available at: http://www.crunchbase.com/company/smart-bear-software

SmartBear. 2010b. Company Overview of SmartBear Software, Inc. Accessed on 18.11.2013. Available at:
http://investing.businessweek.com/research/stocks/private/snapshot.asp?privcapId=11214573

SmartBear. 2010c. About SmartBear. Accessed on 18.11.2013. Available at: http://SmartBear.com/about-us

SmartBear. 2010d. Web Testing with TestComplete. Accessed on 18.11.2013. Available at: http://SmartBear.com/products/qa-tools/automated-testing-tools/web-ria-testing/web-testing-with-testcomplete/

SmartBear. 2010e. How to Create Automated Tests - TestComplete. Accessed on 18.11.2013. Available at: http://SmartBear.com/products/qa-tools/automated-testing-tools/automated-software-tests/create-automated-tests-with-ease/

SmartBear. 2010f. Software Testing Types- TestComplete. Accessed on 18.11.2013. Available at: http://SmartBear.com/products/qa-tools/automated-testing-tools/software-testing-types/

SmartBear. 2010g. Cross-Browser Testing with TestComplete. Accessed on 18.11.2013. Available at: http://SmartBear.com/products/qa-tools/automated-testing-tools/web-ria-testing/cross-browser-testing-with-testcomplete/

SmartBear. 2010h. Extensions and SDK - TestComplete. Accessed on 18.11.2013. Available at: http://SmartBear.com/products/qa-tools/automated-testing-tools/testcomplete-extensions-and-sdk/

Soasta. 2011. Press Releases. Accessed on 18.06.2013. Available at: http://www.soasta.com/press-releases/new-soasta-cloudtest-pro-delivers-the-worlds-largest-performance-testing-platform/

Soasta. 2013a. General Information on Soasta from CrunchBase Company. Accessed on 18.06.2013. Available at: http://www.crunchbase.com/company/soasta

Soasta. 2013b. General Information on Soasta from CrunchBase Company. Accessed on 18.06.2013. Available at: http://www.soasta.com/company/about/

Soasta. 2013c. Real- Time Analytics. Accessed on 18.06.2013. Available at: http://www.soasta.com/product-features/real-time-analytics/

Qian, Ling; Luo, Zhiguo; Du, Yujian & Guo, Leitao. 2009. Cloud Computing: An Overview. Springer-Verlag Berlin Heidelberg. Volume 5931. pp. 626-631.

Test. 2013. Zephyr Test Management Overview. Accessed on 20.12.2013. Available at: http://www.getzephyr.com/zephyr/zephyr_test_management_overview.php

Testing. 2014. Accessed on 04.08.2013. Available at: http://agile.csc.ncsu.edu/SEMaterials/IntrotoTesting.pdf

TestMaker. 2008a. About Push To Test. Accessed on 28.11.2013. Available at: http://www.linkedin.com/company/pushtotest

TestMaker. 2008b. About Push To Test. Accessed on 28.11.2013. Available at: http://www.crunchbase.com/company/pushtotest

TestMaker. 2008c. TestMaker Test Deployment- Cloud and Grid Testing. Accessed on 29.11.2013. Available at: http://www.pushtotest.com/cloud-and-grid

TestMaker. 2008d. Installation. Accessed on 29.11.2013. Available at: http://www.pushtotest.com/installation.html

Tian, Jeff. 2005. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement. 440p.

Trivedi, Shivkumar H. 2012. Software Testing Techniques. Publication: International Journal of Advanced Research in Computer Science and Software Engineering. October-2012, pp. 433-438.

Usability. 2012. Usability Testing For Cloud Applications. Accessed 28.12.2013. Available at: http://blog.cloud-apps-experts.com/2011/02/usability-testing-for-cloud.html

VMware. 2006. Virtualization Overview. Accessed 11.12.2013. Available at: http://www.vmware.com/pdf/virtualization.pdf

Waterfall. 2013a. SDLC WATERFALL MODEL. Accessed on 09.12.2013. Available at: http://www.tutorialspoint.com/sdlc/pdf/sdlc_waterfall_model.pdf

Waterfall. 2013b. WHY THE WATERFALL MODEL DOESN'T WORK. Accessed on 09.12.2013. Available at: http://www.infoq.com/resource/articles/scaling-software-agility/en/resources/ch02.pdf

Whittaker, James A. 2000. What Is Softwarae Testing? And Why Is It So Hard?. Publication: IEEE software 17(1). Jan-Feb 2000. pp. 70-79.

Winterford, Brett. 2009. Stress tests rain on Amazon's cloud. Accessed on: 14.01.2014. Available at: http://www.itnews.com.au/News/153451,stress-tests-rain-on-amazons-cloud.aspx