Development of Software Testing Framework on Cloud

By

Sidrah Inayat

2008-NUST-MS-PhD-IT-41

Supervisor

Dr. Raihan Ur Rasool

A thesis submitted in partial fulfilment of the requirements for the degree of Masters in Information Technology (MSIT)

In

School of Electrical Engineering and Computer Science (SEECS)

National University of Sciences and Technology (NUST), Islamabad, Pakistan

APPROVAL

It is certified that the contents and form of thesis entitled “Development of software Testing Framework on Cloud” submitted by Sidrah Inayat has been found satisfactory for the requirement of the degree.

Advisor: Dr. Raihan Ur Rasool

Signature: _____

Date: _____

Committee Member 1: Dr. Hafiz Farooq Ahmad

Signature: _____

Date: _____

Committee Member 2: Mr. Muhammad Bilal

Signature _____

Date: _____

Committee Member 3: Ms. Sana Khalique

Signature _____

Date: _____

IN THE NAME OF ALMIGHTY ALLAH

THE MOST BENEFICENT AND THE MOST MERCIFUL

TO MY PARENTS & SISTER

CERTIFICATE OF ORIGINALITY

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Sidrah Inayat

Signature: _____

ACKNOWLEDGEMENTS

First of all I am thankful to Almighty Allah for giving me courage and strength to complete this challenging task and to compete with international research community. I am also grateful to my family, especially my parents who have supported and encouraged me through their prayers that have always been with me.

I am extremely thankful to Dr. Raihan Ur Rasool, Dr. Hafiz Farooq, Mr. Muhammad Bilal and Ms. Sana Khalique for their guidance and help in throughout my research work to accomplish the goal. I am highly thankful to all of my teachers who have been guiding me throughout my course work and have contributed to my knowledge. Their knowledge, guidance and training helped me a lot to carry out this research work.

I am especially thankful to Dr. Raihan Ur Rasool for sparing his precious time for my guidance and warmed welcomed me whenever I visited him. I am thankful to him for his motivation, valuable suggestions and continuous guidance throughout my research work. His foresightedness and critical analysis of things taught me a lot about research which will be more helpful to me in my practical life.

Sidrah Inayat

Table of Contents

List of Figures

List of Tables

Abstract

Software testing is any activity used for evaluating an attribute or capability of a program or system and determining that it meets its required results. The difficulty in software testing stems from the complexity of software. There are several major issues that people are facing while testing the applications in their local environment, firstly they have to install multiple software for performing different type of tests like Jmeter is used for load testing LoadRunner for Performance testing, JUnit for Unit testing and many more, actually there is no common platform from where a user can get all these software and can use them according to the need by scaling up and down resources accordingly. Secondly on one hand while testing an application on personal computers results are not credible and accurate all the time because of the involvement of some hardware and software factors such as processing and computing time and storage differences etc. at the same time if an application needs to be tested under certain load for performance testing, in such cases not only the personal computers but even the private clusters become unsuitable. They usually prohibit deficiency of lack of resources, memory and storage for generating the load to test the application under stress. To tackle all these issues the software testing framework on cloud (STFC) is developed that provides multiple open source software testing tools on the cloud , with ease of accessibility, range of resources available on demand, provides credibility of results, without worrying about the management and maintenance.

Chapter 1: Introduction

1.1. Software Testing

Software testing is a technique in which the software is processed in controlled conditions to evaluate its behavior, to detect the errors and to check that the user requirements are fulfilled and the system includes all the functionalities as were specified by the user.

Agility, shorter time to market, rising complexity, market competition and many other factors compel the organizations to ensure the quality, performance and the reliability of software. Modern software testing includes complete software analysis including review of the source code, design and the architecture. There are multiple types of testing and several open source software testing tools are available that are used for performing these different types of testing as shown in the Figure 1.1for example JMeter is used for load testing and measuring performance, JUnit for unit testing of the code etc.

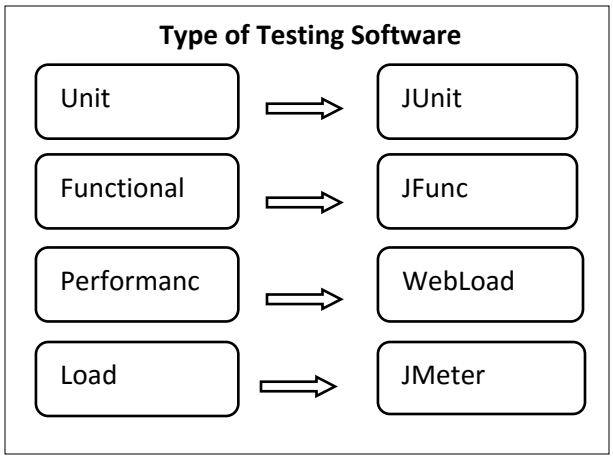


Figure 1.1: Testing Types and respective tools

1.2. Cloud Computing

"Cloud computing" which actually means "internet computing" or "utility computing" makes it possible to access the resources via the Internet by the user from anywhere, for as long as they need, without worrying about any maintenance or management of actual resources. For years, many organizations have suffered the difficulties in maintaining and upgrading different types of software on their own machines. Actually now large organizations are becoming frustrated with the long deploying cycles, high costs, and complicated upgrading processes.

Cloud Computing allows vendors to deliver computing resources on a large scale to the developer and the end users. It actually provides the alternative way of delivering and managing different IT services. The prosperity of provisioning the resources, storage and computing capabilities on demand with pay as you go method is actually the driving feature of the cloud computing that is fetching the attraction of the market place.

There are lots of advantages because of that companies are using cloud computing. One of the major ones is the flexibility that it offers. By using the cloud computing resources can be accessed from anywhere, remotely and tasks can be performed without being at some specific location. As long as people can get on the Internet, People can access information from home, on the road, even from a smartphone. People can also work collaboratively on files and documents, even when they're not actually together. Documents can concurrently be viewed and edited from numerous locations.

Cloud computing provides ease of access and is very quick all we need is just a computer with very simple specification and a good internet connection.

Cloud computing is cheaper as it reduces the costs of buying and managing the software and their licenses, because it's already installed online remotely and we can run them from anywhere, not to mention the fact that many cloud computing applications are offered free of charge. With cloud computing, we subscribe to the software, rather than buying it outright. This means that you only need to pay for it when you need it, and it also offers flexibility, in that it can be quickly and easily scaled up and down according to demand. This can be particularly advantageous when there are temporary peaks in demand [1].

A major advantage of using cloud computing is that a large number of hardware resources are available on demand, user doesn't have to get worried about their management and the maintenance, everything is managed by the cloud vendor and the users pays for what they actually use.

1.3. Testing Modern Software Systems

Now-a-days the requirements of the market are changing so rapidly that modern software systems are also manipulating the changes to They are getting combined within and across the enterprise thereby changing into ecosystems. This has resulted in complexity which seems to be ever increasing and also in a high number of changes these stems have to go through. Complexity is actually directly related to the risks and costs.so when complexity increases the costs and risks automatically touches the upper limits.

There are stresses on time to market and there is a need to ensure the agility of the enterprise at several levels: Technical, Organizational and Business. The cost of finding

&fixing defects is becoming exponential. There's a high premium for Continuing Quality in Modern Software Development.

The above mentioned challenges can be tackled by working in a stable way. We need to exaggerate the Continued Integration discipline by embedding & integrating the following activities:

- Acceptance Testing
- Source Code Analysis
- Common Coding Mistakes
- Unit & Integration Testing
- Load Testing

These activities ensure rapid, immediate, useful feedback throughout the duration of software development thereby transforming a spiteful circle into a virtuous cycle [2].

1.4. Software testing on cloud

In order to reduce the expenses and to achieve the better control on the resources, many organizations are starting using the cloud computing delivery model. As by using cloud computing resources can be accessed on demand, from anywhere, for as long as they are needed without worrying about the maintenance and the management.

Traditionally, when the organizations have to perform large scale testing they have to purchase additional servers and require extra staff which ultimately end in the increase of expenditures but by using cloud computing, test environments can be now easily and

quickly replicated to the cloud and the resources can be used on demand and for as long as they are needed. This is a relatively more cost effective way [3].

The software testing on cloud becomes complex as it should be used to test the SaaS and non SaaS applications on cloud as well as the cloud itself needs testing as shown in the Figure 1.2.

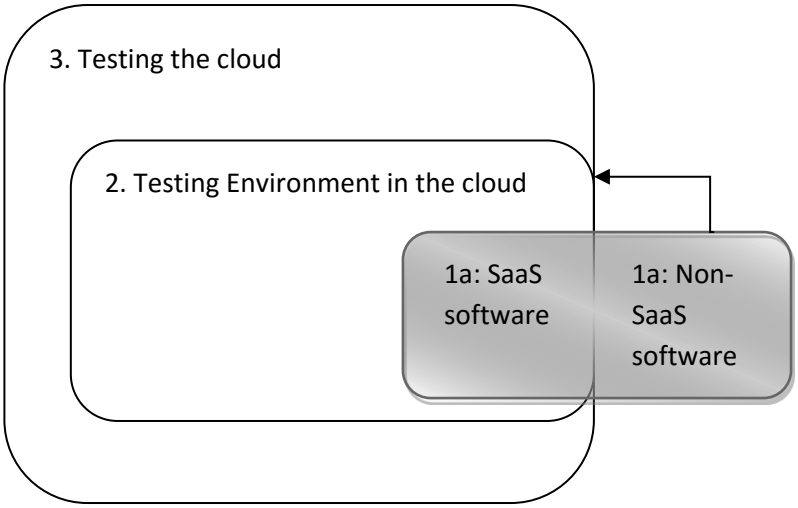


Figure 1.2: Cloud Based Testing

1.5. Thesis organization

This thesis is ordered into eight different chapters. **Chapter 2** provides the background of the software testing, cloud computing and the issues faced by the users in traditional environment. **Chapter 3** presents literature survey about the testing services being provided on cloud there limitations and issues still there **Chapter 4** Presents the proposed solution with aims and objectives **Chapter 5** presents the proposed solution and its system architecture. **Chapter 6** presents the proposed system design and implementation. **Chapter 7** presents the Test results. **Chapter 8** presents the conclusion and future work.

Chapter 2: Literature Review

2.1. Cloud Computing Architecture

There are many cloud providers and each of them provides cloud services at different layers of the cloud. There are basically three layers of the cloud Architecture, the details are given below.

2.1.1. Software as a Service (SaaS)

The customers can make the application of their interest that run on the cloud, usually through the web browser. The user doesn't have any concern with the infrastructure being used to run the application, they don't have to manage and control the infrastructure, an example of SaaS in Cloud Computing is Google docs.

2.1.2. Platform as a Service (PaaS)

In this case, in order to run and access the applications of their interest the customers are provided with programming and execution environments. Just like SaaS model, customers cannot control the underlying cloud infrastructure but have control over the applications they create and to a certain degree, configuration settings of the hosting environment. An example of a PaaS is Google app engine

2.1.3. Infrastructure as a Service (IaaS)

This is where computing services such as storage, processing and networks are provided by the IaaS provider for the customers to deploy and run their applications. IaaS gives a customer the flexibility to control and run software over the computing environment. A popular example is Amazon's EC2.Figure 2.1 shows the architecture of the cloud.

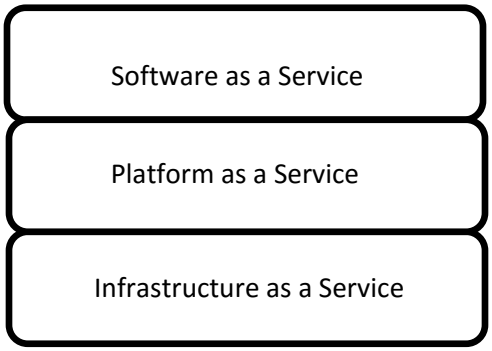


Figure 2.1: Cloud Architecture

2.2. Selecting Cloud Vendor

As there are multiple cloud vendors available to the market for example MS Azure, Google App engine, sales force etc. so making a right decision in choosing the correct vender is based on multiple factors. There were several questions that arise in one's mind while selecting the cloud vendor like security of data and how much reliable the cloud vendor is?

Microsoft Azure, Google App Engine or Amazon Elastic Cloud, all these cloud computing systems have been designed to provide facilities required to support the complete life cycle of building and delivering software applications and making them available over the Internet.[5]

Microsoft azure is directly comparable to Google's App Engine. On same lines as App Engine, Azure also provides environment for building applications for the cloud (PaaS) plus providing finished applications as service over the cloud (SaaS). From development point of view, building applications in Azure environment would be much easier as most

of the IDEs used by developers are now capable of developing and testing applications for cloud. But the backline of Azure is that it is not supporting the Java language yet [6].

Google App Engine is the cloud computing platform for developing and hosting web applications on Google's data centers. It is free up to a certain level of used resources. Fees are charged for additional storage, bandwidth, or CPU cycles required by the application. The supported programming environments that Google currently offers for App Engine are Python and Java, but there's no roadmap for future language support [5].

Of all three cloud services providers mentioned here Amazon is the only one that provides true IaaS, this inference is based on capability of Amazon's Elastic Compute Cloud or EC2 to allow users to use web service interfaces to launch instances with a variety of operating systems. EC2 also enables users to load these Operating systems with custom application environment plus gives them ability to manage network's access permissions and run image using as many or few systems as desired. The Table 2.1 shows the comparison between the world three rocking cloud vendors.

Table 2.1: Comparison of cloud Venders

Feature	**Microsoft**	**Google**	**Amazon**
Offerings	Windows Azure , SQL Azure, AppFabric	Google App Engine	S3, Ec2, SQS, SDB, FWS, Cloud Front
Load Balancing	Yes	Yes	Yes
Storage	Yes	Yes	Yes

Message queuing for machine communications	Yes: Queues in Windows Azure storage	No	Yes: Simple Queue Service(SQS)
Tied to the vendor datacenter	Yes	Yes	Yes
Pricing	Flat rate (including connection packs on AppFabric) or pay-per-use pricing compute + storage + transaction + bandwidth	Free resources up to a level After that enable billing for cost of the resources, bandwidth, CPU, data store, emails etc	Pay as you go + EC2 spot Instances
Language support	Java (Not Completely), Python, Ruby, PHP, .Net, C#, C++, VB REST, SOAP, XML,	- Java/AJAX/JavaScript -Java 6 runtime (white listed JRE classes) Python 2.5	- SOAP & REST - Developer libraries: C++, Ruby, Python, Java, PHP - Command line utilities for EC2
Elasticity	No	No	Yes

As depicted by the Table2.1, Amazon is best for providing testing softwares as a service because of mainly two reasons firstly as it provides the elasticity i.e. resources can be utilized according to the need of time. We can scale up and down on demand. Secondly the language support as it supports multiple languages.

2.3. Amazon Elastic Compute Cloud (EC2)

Amazon EC2 is a web service that enables user to launch and manage server instances in Amazon's data centers using APIs or available tools and utilities. User can use Amazon EC2 server instances at any time, for as long as you need, from anywhere.

2.3.1. Service highlights

Amazon Elastic compute cloud provides multiple services some of them are as following.

- Elasticity: Resources can be used when needed and can be released when they are free.
- Complete Control: The user have complete control on the resources
- Flexible: Provides Flexibility of usage
- Reliable: Provides reliable mechanisms of data storage and manipulation
- Secure: Provides security of the data
- Inexpensive: Provides a cost effective way for using large number of resources [8].

2.3.2. Amazon Machine Image and Instance

An Amazon Machine Image (AMI) contains all information necessary to boot instances of software. For example, an AMI might contain all the software to act as a web server or it might contain all the software to act as a Hadoopnode.

User launches one or more instances of an AMI. An instance might be one web server within a web server cluster or one Hadoop node [8].

2.3.3. Instance Storage

Every instance includes a fixed amount of storage space on which you can store data. Within this document, it is referred to as the "instance store" as it is not designed to be a permanent storage solution. If an instance reboots the data on the instance store will survive. If the underlying drive fails, the instance is terminated, or the instance is stopped, the data is lost. If you need a permanent storage solution the Amazon Elastic Block Store (Amazon EBS) is used.

EBS is a type of storage designed specifically for Amazon EC2 instances [8]. Amazon EBS allows you to create volumes that can be mounted as devices by Amazon EC2 instances. Amazon EBS volumes behave like raw unformatted external block devices. They have user supplied device names and provide a block device interface

2.3.4. Getting Started with Amazon Cloud

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that enables user to launch and manage Linux/UNIX and Windows server instances in Amazon's data centers. User can get started with Amazon EC2 by following the tasks shown in the Figure 2.2.

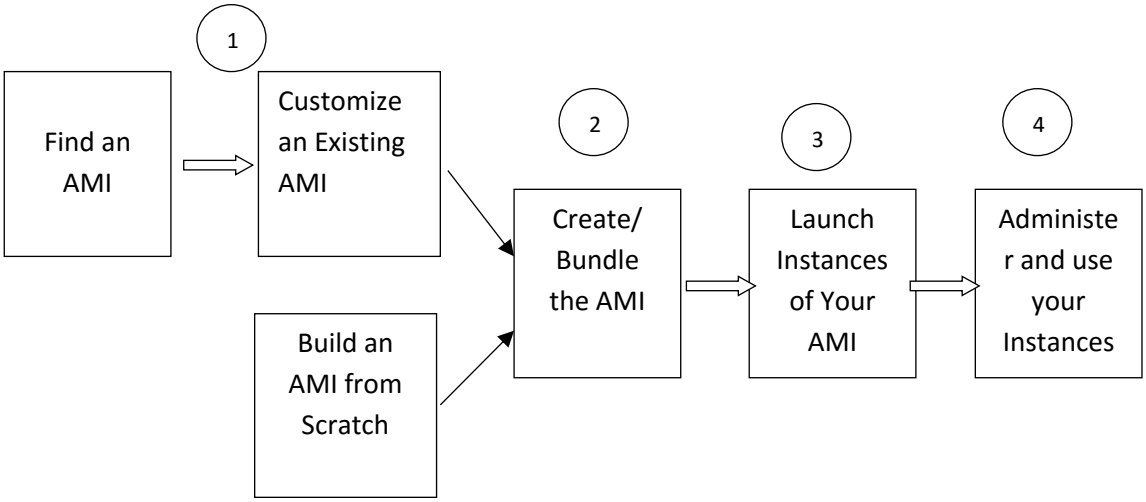


Figure 2.2 Getting started with EC2

2.4. Future of Software Testing

The testing of an application becomes more crucial and time consuming with increase in complexity. Trends in the industry suggest that software testing will become more difficult in future. In order to drive the changes in effective and efficient way, more emphasis must be placed on analysis and design phases. Emphasis should be towards the 4Rs, repeatability, reliability, re-use and robustness.

Testing and Quality Assurance will become more important and add more value as we move into architectures and technologies that support the business in their goals of bringing products and services to the market as rapidly as possible, with minimal risk [10].

Recent studies show that the software testing is now moving into cloud as more and more companies understand the importance and value of cloud [11] and want to get of the problems faced because of physical storage and computation power's limitation.

2.5. Traditional vs. Cloud based Testing

In order to reduce the expenses and to achieve the better control on the resources, many organizations are starting using the cloud computing delivery model. As by using cloud computing resources can be accessed on demand, from anywhere, for as long as they are needed without worrying about the maintenance and the management.

Traditionally, when the organizations have to perform large scale testing they have to purchase additional servers and require extra staff which ultimately end in the increase of expenditures but by using cloud computing, test environments can be now easily and

quickly replicated to the cloud and the resources can be used on demand and for as long as they are needed. This is a relatively more cost effective way [12].

There are several service providers that are providing different services of software testing on cloud; a comparison on the base of services provided is shown in the Table 2.2.

Table 2.2: comparison between providers of software testing in cloud

Providers	Performance Testing	Load Testing	Unit Testing	Functional Testing
Cloud-Intelligence[13]	Yes	Yes	No	No
RTTS[14]	Yes	Yes	No	Yes
Attenda [15]	Yes	No	Yes	No
HP [16]	Yes	Yes	No	No

It is concluded from the Table 2.2 that although the power of cloud computing is being utilized in the software testing but there is still lack of some common platform for accessibility of multiple software testing services on demand.

2.6. Benefits of Testing in the Cloud

In term of resources and cost, t**e**sting on cloud have many benefits, some are as followed

- To perform the load and stress testing when numerous resources are needed, cloud computing helps to reduce the infrastructure cost.
- It provides the facility of scaling up and down the resources according to the need; means we have flexibility of using the resources when they are needed while paying as we go.

- We can skip the time consuming process of setup and procurement by simply using the preconfigured cloud resources.
- The service provider is supposed to ensure that latest version of the tool is provided. So instead of paying a heavy amount for buying a tool, keeping track of latest version, and updating the software accordingly; we just need to pay-as-we-use.
- It somehow reduces the need of hiring the individual tool experts [17].

Because of such benefits, the cloud computing is becoming the best suited environment for the software testing and many organizations have started using it to save their Time, Money and resources.

2.7. Review of papers

This section provides the review of some research papers that provides the details about the cloud computing, its utilization in software testing and benefits that how it improves the test results credibility.

2.7.1. When to Migrate Testing on Cloud [18].

In order to stays up-to-date with changing market requirements software is constantly evolving. The testing process needs to constantly evolve as well to stay in sync with the development process. Migration is one form of software development. As software becomes mature, so does its testing process. To make the testing process more efficient new techniques needs to be in place as the test cases become outdated. There is a noticeable paradigm shift with the introduction of technologies such as SOA, cloud computing, and SaaS. The testing is now being shifted to on demand, pay as you go from

having standalone in-house development and testing. With this paradigm shift in software development, it is time to hold this change and introduce it to the testing process as well.

However, like all movement process, immigration of testing to the cloud does not have to be an “all or nothing” big bang approach. Some parts of the testing process can be migrated to the cloud and others may not. As with all engineering deliberations, the decision of when to do so rests on a number of business, technical, and operational aspects. This paper has laid out some of the characteristics to consider when to migrate testing to the cloud from the application point of view. There are many other aspects to consider, such as the cloud infrastructure, the test environment, the quality of the cloud service provider, and security and privacy concerns of test assets.

2.7.2. Research Issues for Software Testing in the Cloud [19].

This paper provided an overview of cloud computing and discussed various research issues within the context of testing in the cloud. The research issues are presented in four categories Cloud computing provides large business and technical assistances to software testing. As it becomes more common practice and availing a myriad of cloud solutions, services and applications, we observed that organizations seem to be on the lookout for ways to enhance the testing process. There is an anticipation of an increase in testing solutions in the cloud, providing flexibility and cost benefits. We believe cloud computing promises a lot of potential for testing. Additionally, vendors will provide testing services through 562crowdsourcing as well as testing platforms and infrastructure hosted in the cloud.

As the shift to the cloud continues to grow, it will also increase the need for testing in and of the cloud. The problems associated with migration to the cloud are many: security, reliability, performance, scalability and manageability among others. Organizations looking to shift their systems or applications to the cloud need to understand the problems and risks involved in doing so and take the appropriate precautions. Therefore, comprehensive testing becomes necessary as a means to address these problems and risks.

There is a need for pilot projects to see how delivery of testing in the cloud will work in practice. An example is discussed in where so getting started a "proof of concept" with

IBM Development and Test Cloud to experiment with test tooling and infrastructure in the cloud [34]. Another effort is Open Cirrus (TM) - an open cloud-computing research

testbed aimed at supporting research in various aspects of cloud computing e.g. design and management of services [35]. The testbed attempts to encourage collaboration among

a community of interest to share and exchange knowledge. Our current work-in-progress is looking at how cloud computing, service oriented architecture, open source development technologies and crowdsourcing affect an organization's testing process. In general, we are studying how different software organizations adopt to new technology methods and concepts, specifically within their testing activities. We also plan to explore how cloud software development and testing will affect quality requirements in the future addressing the interdependency of cloud software development, cloud testing and overall quality assurance. The research issues for testing in the cloud are many. We believe

different research approaches and methods will suit different problems and scenarios. Suitable research methods could be action research, with survey and the grounded theory method among others. Due to the industry-specific problems of cloud computing, we recommend research methods that encourage a high collaboration between the industry and researchers, e.g. action research, design research and case studies research. This interplay would enable real-life problems to be addressed in a scientific and methodological manner. Cloud computing is growing and there is need for academic research to address different research issues associated with it. We hope that this paper can act as a resource

2.7.3. Testing as a Service over Cloud [20].

This paper proposed an automated testing platform TaaS on a cloud. This platform adopts cloud computing technique to build the elastic resource infrastructures, and provide various kinds of testing services to testing users. To validate TaaS platform, we used unit testing services to perform the experiments. This platform helps testers to set up unit testing environment, select a suitable unit testing method and testing service for the test task, automatically generate test cases, automatically execute test cases, at last collect the test results, and report to testers. The process is automatically completed, thus maximally saves tester's effort for performing a unit testing task. In the future, we plan to deploy more testing services on TaaS cloud platform, and collect a variety of runtime information to perform corresponding analysis on scalability and reliability

2.8 Summary

Before going into the development phase one has to select a well suited cloud vendor. Selecting a well suited cloud vendor is elementary building block of starting using cloud. The selection of well suited cloud vendor is based on multiple factors discussed above. As there are multiple cloud vendors available in the market for example MS Azure, Google App engine, sales force etc. that are providing different facilities according to their architecture so making a right decision in choosing the well suited vender is a quite difficult task.

Chapter 3: Motivation and Scope

3.1. Research motivation

Agility, shorter time to market, rising complexity, market competition and many other factors compel the organizations to ensure the quality, performance and the reliability of software. Modern software testing includes complete software analysis including review of the source code, design and the architecture. There are several open source software's testing tools available in the market that are used for performing different types of software testing like JMeter is used for load testing and measuring performance , JUnit for unit testing of the code etc. but the matter is that organizations have to manage and maintain these multiple software for testing tenacity. And even when they succeed in implementing the tools and instilling the best practices, these things takes Time, Money & Resources.

Cloud Computing allows vendors to deliver computing resources on a large scale to the developer and the end users. It actually provides the alternative way of delivering and managing different IT services. The prosperity of provisioning the resources, storage and computing capabilities on demand with pay as you go method is actually the driving feature of the cloud computing that is fetching the attraction of the market place.

To get rid of all these issues of managing and maintaining the tools and resources and for tackling the issues of storage and computation powers, the industry is locating for a common platform for multiple software testing services so we proposed STFC. It lays at the intersection of these key areas i.e. software testing and cloud computing for providing

multiple open source software testing tools on one platform, with a range of maintainable resources available on demand.

3.2. Research Scope

Providing a Common software testing framework that have multiple open source tools available on demand can resolve the multiple issues faced by the software tester, such as they will not have to manage multiple software for performing different types of testing, they will not have to get worried about the management and the maintenance of the tools and resources, they can access the tools on demand from anywhere, they can access bundle of resources with capability of scaling up and down the resources on demand. Along with all these features the users can get more accurate and credible results over the cloud.

3.3. Aims of proposed Solution

The Proposed solution has multiple aims objectives listed below

- Providing a common platform of testing, from where a user can access multiple open source software testing tools to perform multiple tests of applications.
- To provide the bundle of resources on demand that can be accessed from anywhere with ease of scalability.
- To provide more credible and accurate results.
- To reduce the management and maintenance efforts of the users
- To reduce the costs.

3.4. Summary

This chapter illustrates that are several open source software testing tools available on the market that are used to perform different kind of testings but the a user willing to test all aspects of his application, have to install and manage multiple open sources tools, he has to allocate resources for it, but at the end he has no grantee of getting credible results, so a software testing frame on cloud is proposed that will have property of performing all kind of software testing on demand, with a range of resources available on demand.

Chapter 4: Methodology

4.1. Software Testing Framework on Cloud (STFC)

The STFC provides multiple open source software testing tools under one roof. The STFC provides several facilities as increases the processing capabilities, saves time and generates the accurate and credible results. Traditionally while testing an application in the local environment, results are not credible and accurate all the time because of the involvement of some hardware and software factors such as processing, computing and transmission time or storage differences etc. also on the other hand if an application needs to be tested under certain load for rationale of performance testing, in some cases not only the personal computers but even the private clusters become unsuitable. They usually prohibit deficiency of lack of resources for generating the large scale load so these issues can be tackled easily by deploying testing tools in the cloud environment and utilizing the range of resources available on demand as shown in Figure 4.1.

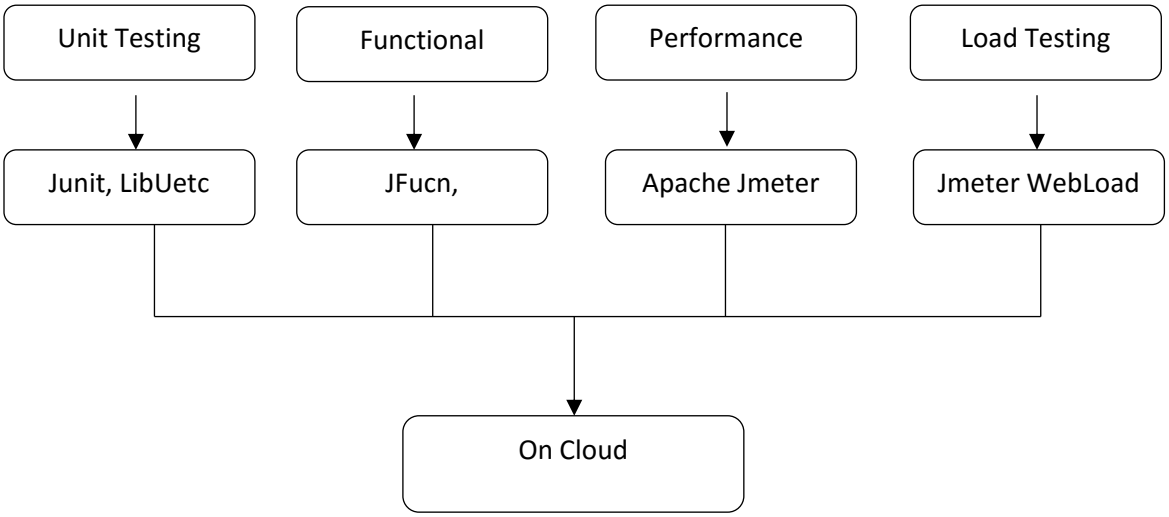


Figure 4.1 Software Testing Framework on Cloud

4.2. Abstract level Architecture

On abstract level the STFC interacts with all the three levels of the cloud architecture, i.e. IaaS, PaaS and SaaS. The user can interact with the SaaS layer through a browser, IDE or program, the SaaS layer interacts with the PaaS and IaaS. Figure 4.2 shows the STFC structure at abstract level.

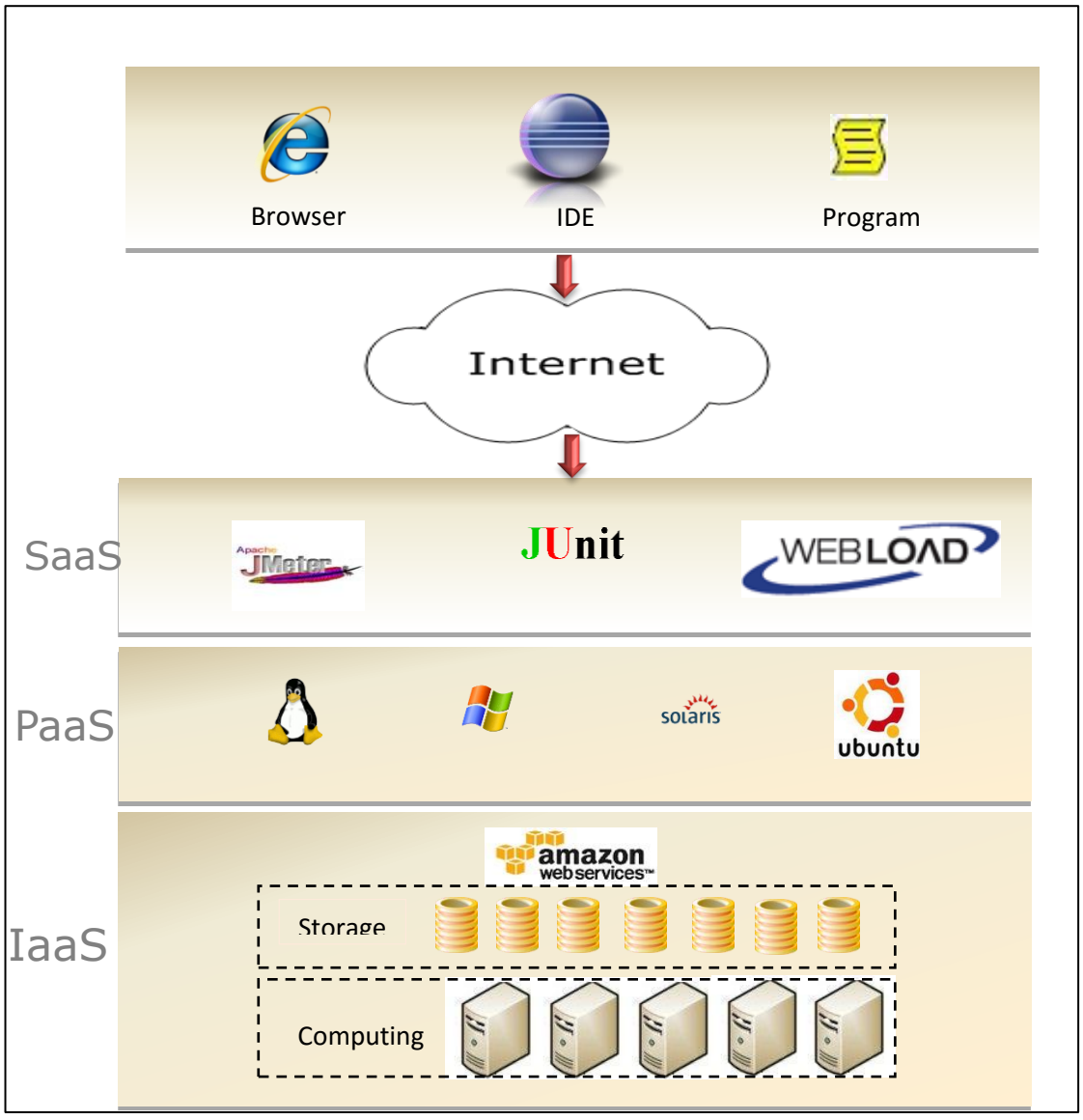


Figure 4.2: STFC Abstract level architecture

4.3. Interaction with STFC

The STFC provides multiple open source software testing tools over on platform, i.e. Cloud. User can interact with the cloud through the internet. The tools are deployed over the cloud and they use the storage of the vendor as shown in Figure 4.3.

STFC eliminates the need of installing and configuring multiple tools on the local machines and also reduces the time and cost needed in allocating resources and man power for performing different types of testing. User can access all the services through a browser, IDE or a program.

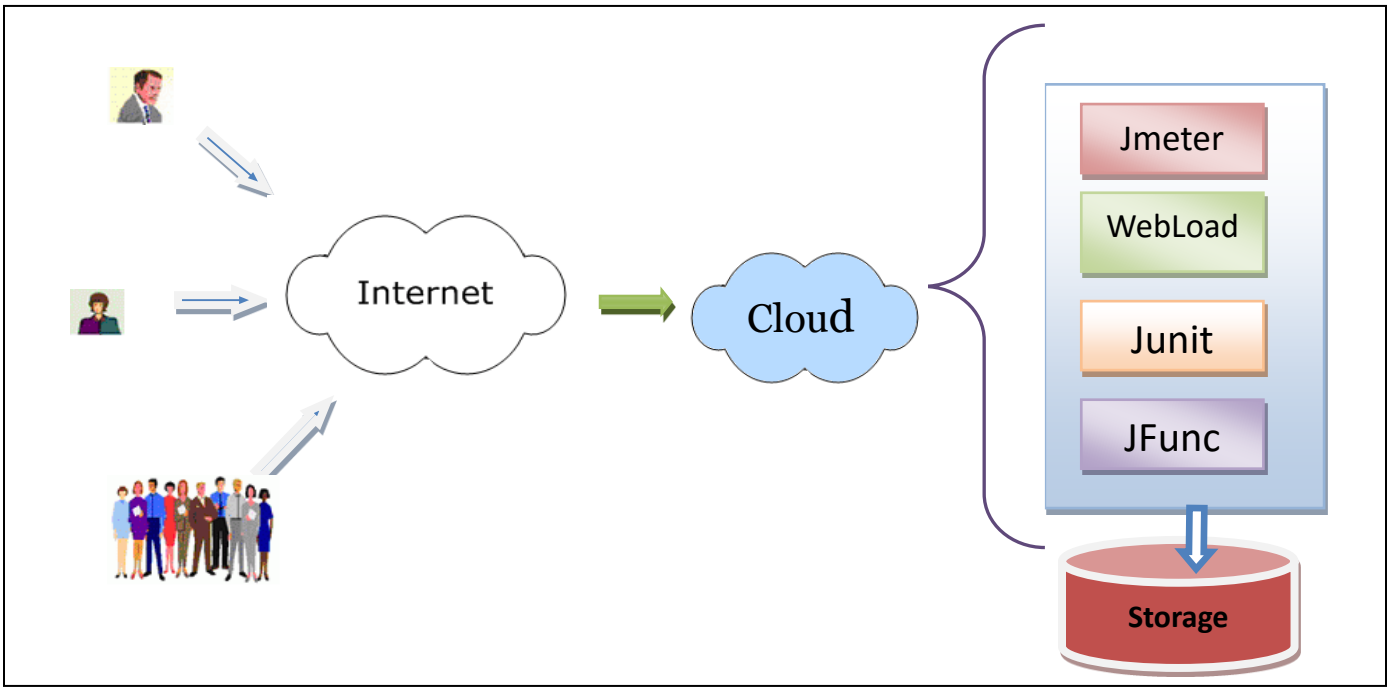


Figure 4.3: Interaction with STFC

4.4. Detailed Architecture of the STFC

The Architecture of the STFC is shown in the Figure 5.4It shows that a user gets connected with the amazon elastic compute cloud through the internet and after getting connected he can launch the instances of the Amazon Machine Image (AMI), that contains all the necessary information of booting the instance and providing multiple open source software testing tools to the end user. A user can launch multiple instance of AMI and run can use the tool of interest according to the need, for example a user wanting to perform the unit testing of the application can launch a separate instance of the same AMI and can access the JUnit to perform the Unit Testing and at the same time if a user is interested in the Load testing of the application he can launch the separate instance of the same AMI and can perform the load testing of the application by initiating the slaves instances etc.

The Figure 4.4 shows the architecture of the STFC, there is an AMI that contains all the necessary information of the software testing tools, their configurations etc. and a user have to simply launch the instance of that AMI, after launching the instance of the AMI the user can run the tools of his interest and can scale up and down the resources according to the need of the time.

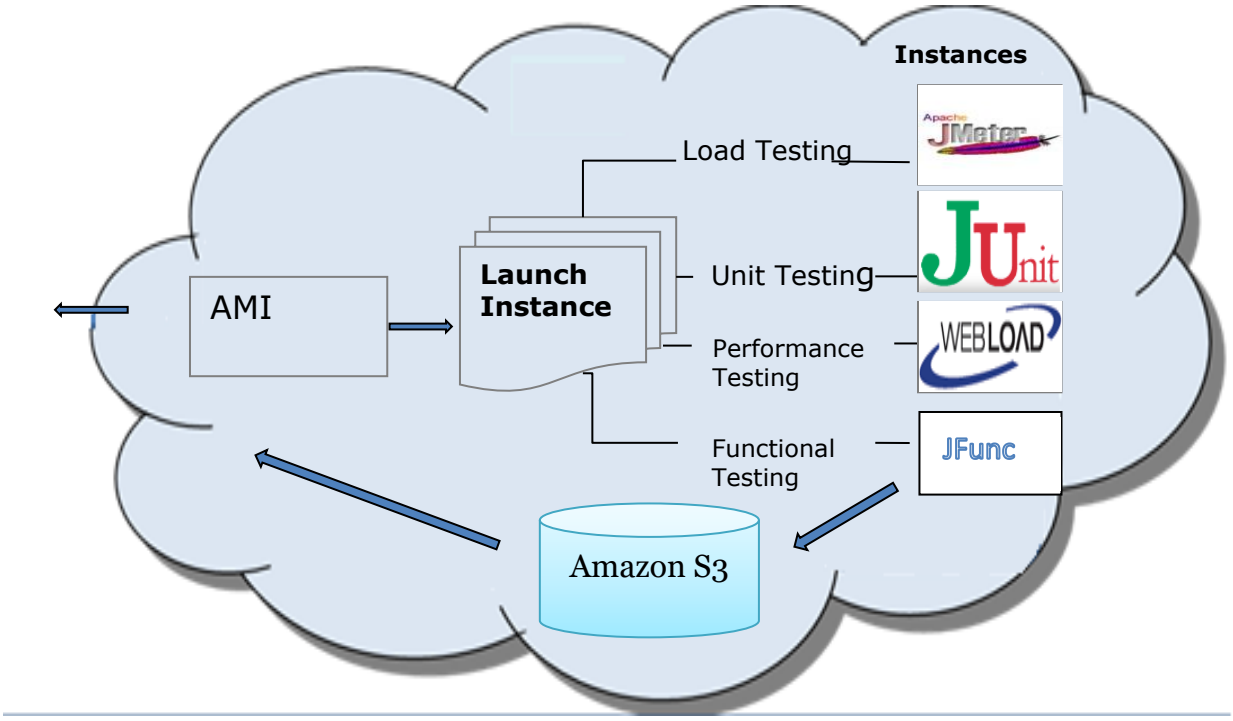


Figure 4.4: Architecture of STFC

As a start point Jmeter is deployed on the cloud, The Amazon machine image is created that contains all the necessary information about the environment and the Jmeter. A user after getting connected with Cloud can launch the instance of the AMI and can perform the load testing of his application without going into the installing and configuration phases of the Jmeter.

The different Modules of the System interact with each other. The Completer system Module Diagram is shown in the Figure 4.5

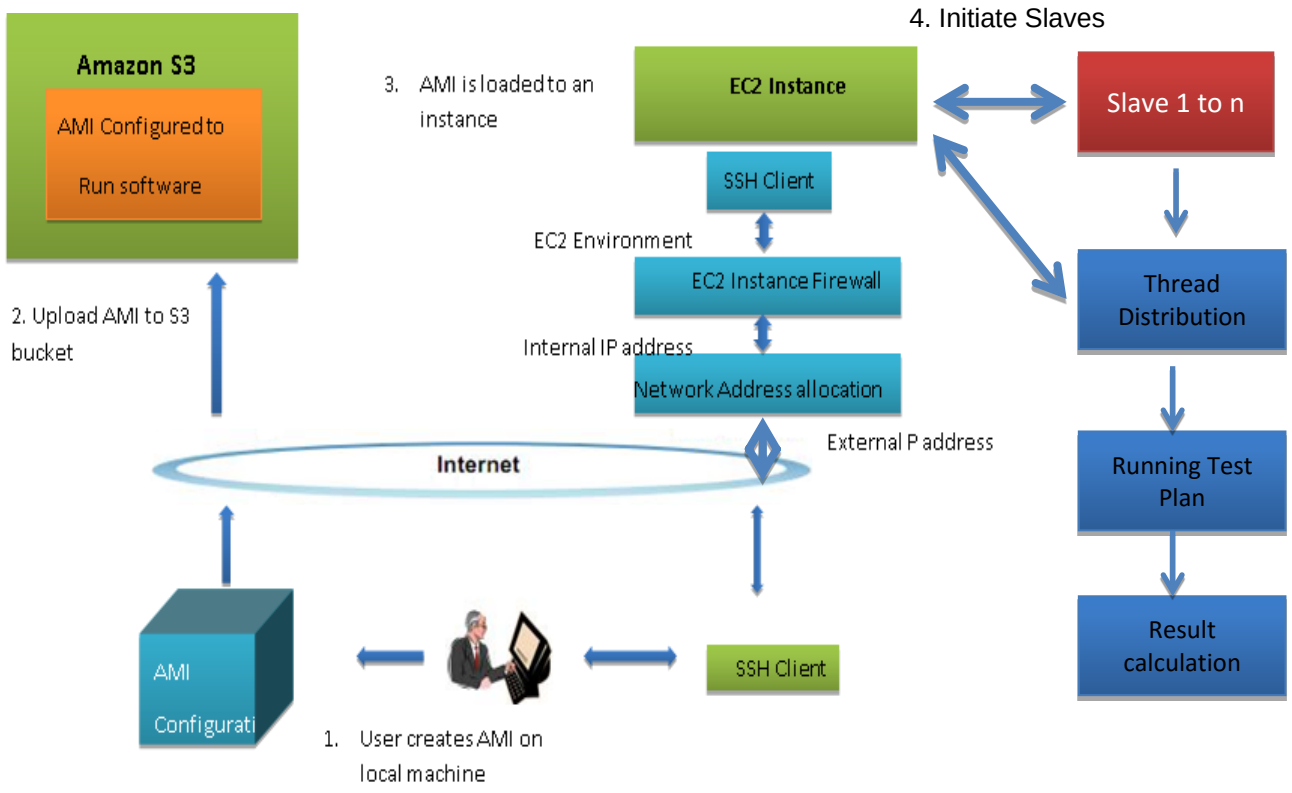


Figure 4.5: System Modules' Diagram

The implementation workflow of the system is shown in the Figure 4.6. It shows the different states of the Master and Slaves and how they interact with each other. The user launches the Master instance, the Master instance stays in Pending state in the start and after some time it becomes active. When the Master becomes active it initiates the slaves, distributes the threads on the slaves. The slaves run the test plan and calculate the results and then test results are shifted back to the user.

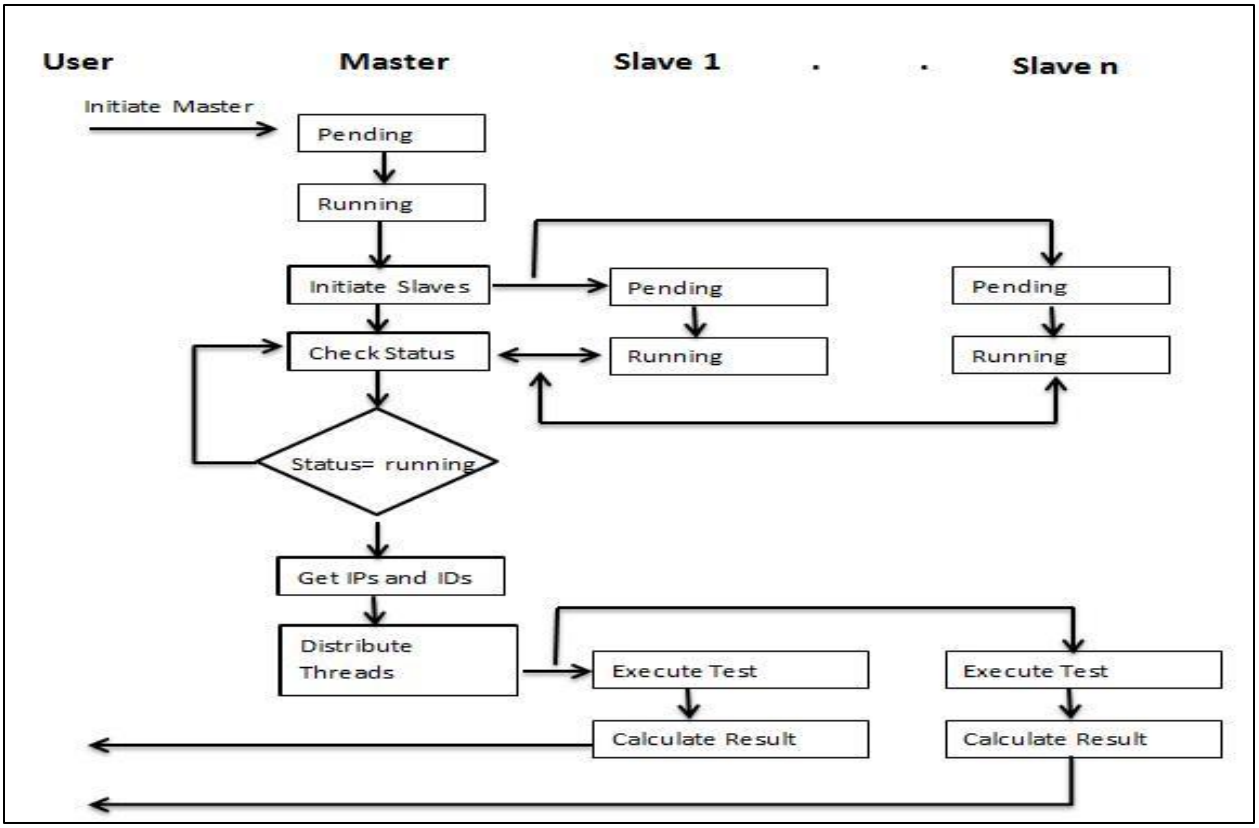


Figure 4.6: System Work Flow

4.5. Amazon Machine Image (AMI) Creation

Amazon machine image contains all the necessary information to boot the instance, with required environment and the tools. The Amazon provides EC2 API tools to interact with their cloud, there are several other ways provided that are used to interact with the amazon elastic compute cloud e.g. AWS console through which instances can be launched and monitored effectively. A user uses SSH connection to get connected with an instance, when a user launches an instance; the instance is assigned with a IP address and Public DNS. This IP address and public DNS are used to get connected with the

instance. When a user creates his own Amazon machine image, the AMI is stored in the S3 storage so whenever the user wants to use the AMI he can launch the instance of this AMI.

Initially we started with Jmeter, we had to create such an AMI that have Jmeter in it as a preconfigured tool, and with all the environmental setups, a user can simply launch the instance of this AMI and can get started with running his test plans without worrying about the management and the maintenance of the tool and without going through the installation and configuration phases. Jmeter is selected as the very first tool because it can actually provide us the overview of the memory utilization, resources and other factor like this, because it is used to load test the applications which means testing the application when a large number of users access the application concurrently, so under such conditions when an application is being accessed by the multiple users at a time then resources are also utilized accordingly which gives an actual and realistic overview of benefits availed through the software

The Amazon machine can be created by the following steps.

- Select an AMI: An AMI is selected that well suits our needs, and that have the required environment
- Launch and Connect to the instance; The instance of the AMI is launched and connection to the instance establishing SSH connection through the Putty
- Install the software; the software of the need is installed on the instance, if the software is not found in the online repository, then it can be transferred from local machine to the instance by using the PSCP.

- Configure the software; the software is configured on the instance according to its need
- Set the environment; the environmental variables are added to the instance's .bashrc file and that .bashrc file is set as a source file for the booting time.
- Create S3 Bucket; S3 is the amazon simple storage system, a S3 bucket is just like a hard disk drive, in which the data of the AMI is stored
- Bundle the Image; The image of the running instance is bundled to create the manifest file, the images is first compressed, signed for integrity, divided into small parts and then the manifest file is created.
- Upload the bundled image to the S3 storage; the image is uploaded to the S3 storage so that it can be accessed by the user whenever needed.
- Register the Amazon Machine Image; the image is registered with the Amazon so that it can get a unique identification in the amazon cloud.

The Figure 4.7 shows the AMI creation and bundling process respectively.

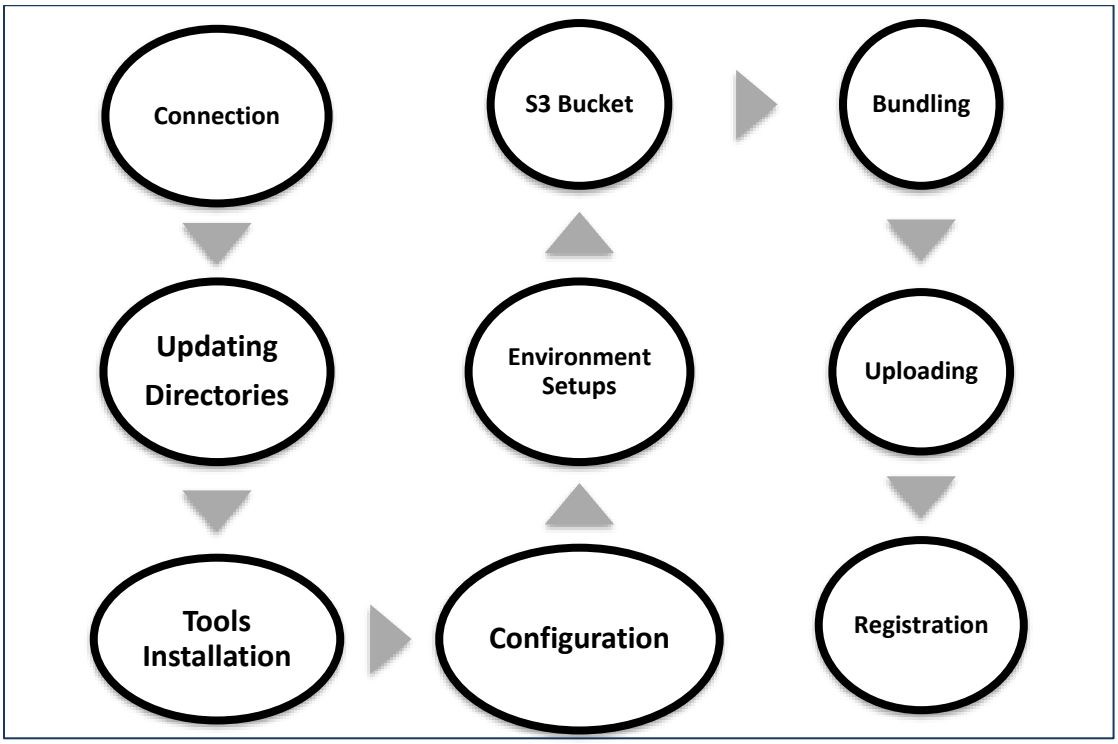


Figure 4.7: AMI Creation Steps

4.5.1 Connection

The SSH secure connection mechanism is used to get connected with the instance running on the Amazon cloud. When instance is launched it is assigned with the IP address and Public DNS. Using these IP address and Public DNS a connection is established with the instance. There are several other AWS credentials that are used for secure connection.

4.5.2. Updating Directories

The directories of the running instance are needed to be updated so that repositories can be upgraded with latest software versions and latest versions of the software can be

installed whenever needed. There are basically two utilities of APT that are used to update and upgrade the repositories.

4.5.3 Tools Installation

The software testing tools are installed to the running instance, if software version needed is not found in the online dictionary of the instance then using any secure copy mechanism the setup files can be transmitted from the local machine to the running instance and then installation can be performed according to the setup requirements. Normally PSCP is used to transfer the files to and from the running instance.

4.5.4 Configuration and Environment Setups

Mostly when any tool is installed on the system it needs some configurations in the system and also requires some other software to be installed or needs some environmental variables to be set so that the system can actually allocated the installation directory of the software. Most of the time its considered more accurate way to add the environmental variables in the .bashrc file and then this .bashrc file is set as the source file so that whenever the instance reboot it first locate the .bashrc for environmental variables.

4.5.5 Simple storage system bucket

Amazon S3 stores data objects in buckets, which are similar in concept to directories, a bucket can be created through AWS console, each bucket have a unique name and identification.

4.5.6 Bundling the Image

Amazon S3 stores data objects in buckets, which are similar in concept to directories. User has to specify a bucket name in the following example as <your-s3-bucket>. Buckets have globally unique names and are owned by unique users. If user has used Amazon S3 before, user can use any of its existing buckets or just give ec2-bundle-instanceany name that makes sense. The ec2-bundle-instance utility uploads the bundled AMI to a specified bucket. If the specified bucket does not exist, it creates it. If the specified bucket belongs to another user, ec2-bundle-instance fails. The bundling process is shown in the Figure 4.8

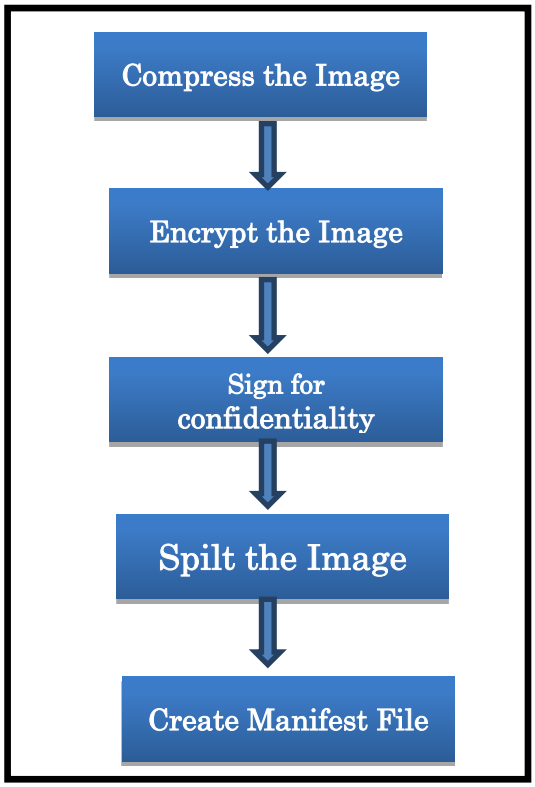


Figure 4.8: Bundling Process

Amazon EC2 provides the ec2-bundle utility for bundling the instance. This utility has parameters are shown in the Table 4.1[22]

Table 4.1: ec2-Bundle Utility parameters

Option	**Definition**	**Required?**	**Example**
-k, --privatekey KEY	The path to the user's PEM encoded RSA key file.	Yes	-k $HOME/pk-234242DEADCAFE.pem
-u, --user USER	The user's EC2 user ID (a.k.a. AWS account number).	Yes	-u 123456789
-i, --image PATH	The path to the image to bundle.	Yes	-i /var/spool/my-image/version-2/debian.img
-d, --destination DESTINATION	The directory in which to create the bundle. Defaults to the current directory.	No	-d /var/run/my-bundle
-p, --prefix PREFIX	The filename prefix for bundled AMI files. Defaults to "image".	No	-p my-image-is-special
--help	Display the help message.	No	--help

4.5.7 Upload the Image

We must upload the bundled AMI to Amazon S3 before it can be accessed by Amazon EC2. ec2-upload-bundle is used to upload the bundled AMI. Amazon S3 stores data objects in buckets, which are similar to directories. Buckets must have globally unique names. The ec2-upload-bundle utility uploads the bundled AMI to a specified bucket. The utility parameters are shown in the Table 4.2 [22]

Table 4.2: ec2-upload Utility parameters

Option	Definition	Required?	Example
-b, --bucket S3-BUCKET	The name of the Amazon S3 bucket in which the bundle will be stored. If the bucket doesn't exist it will be created (provided the bucket is available of course).	Yes	-b aes-cracker-ami
-m, --manifest MANIFEST-PATH	The path to the manifest file. The manifest file is created during the bundling process and can be found in the directory containing the bundle.	Yes	-m /var/spool/my-first-bundle/Manifest
-a, --access-key USER	The user's AWS access key ID.	Yes	-a *******
-s, --secret-key PASSWORD	The user's AWS secret access key.	Yes	-s *******
--acl ACL	The access control list policy of the bundled image. It may be either "public-read" or "aws-exec-read" and defaults to "aws-exec-read" if not specified.	No	--acl public-read
--ec2certificate PATH	The path to the EC2 X509 public key certificate. Defaults to "/etc/aes/amiutil/cert-ec2.pem".	No	--ec2certificate $HOME/pk-234242DEADCAFE.pem
-d, --directory DIRECTORY	The directory containing the bundled AMI parts. Defaults to the directory containing the manifest file (see the "-m" option).	No	-d /var/run/my-bundle
--part PART	Start uploading the specified part and upload all subsequent parts.	No	--part ????
--url URL	The S3 service URL. Defaults to https://s3.amazonaws.com.	No	--url https://s3.amazonaws.ie
--retry	Automatically retry failed uploads. Use with caution.	No	--retry
--skipmanifest	Do not upload the manifest.	No	--skipmanifest

--help	Display the help message.	No	--help
--manual	Display the help.	No	--manual

4.5.8 Register the Image

We must register your image with Amazon EC2, so we can locate it and run instances based on it. The ec2-register utility is used to register the AMI and to get the unique AMI ID.

4.6. Execution of Test Plan

The Jmeter is initially deployed over the cloud, the Jmeter is open source software testing tool that is used to test the application under load. The Jmeter working on cloud is shown in the Figure 4

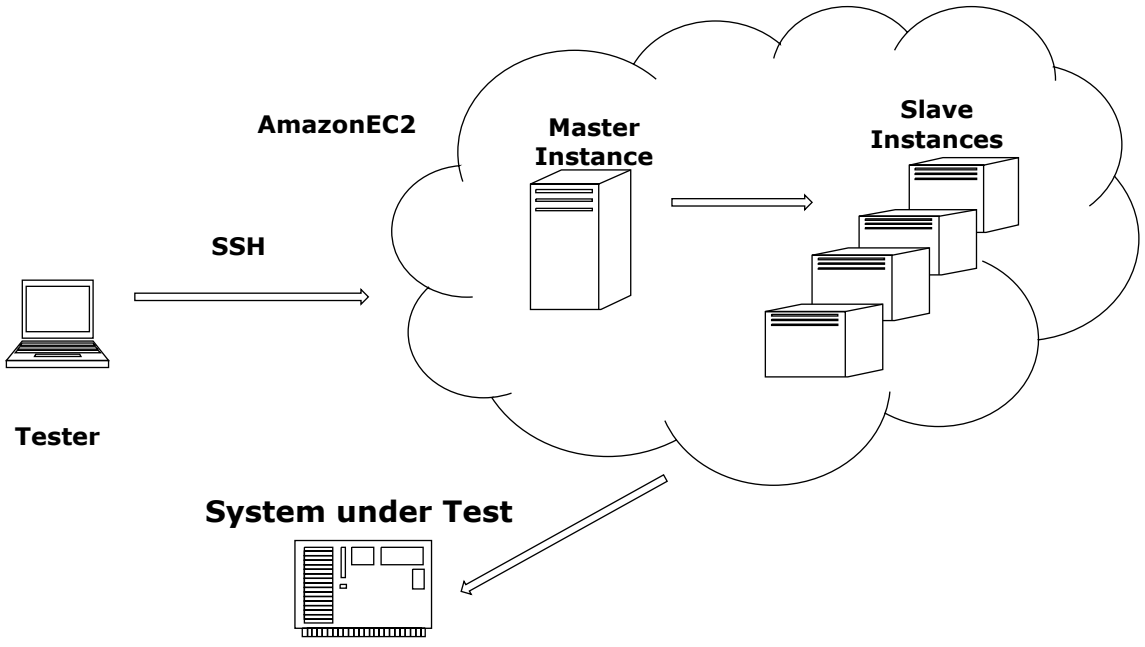


Figure 4.9: Jmeter Working on cloud

The bash programming is done to perform the following tasks.

- Run the slave instance from the master instance
- Continuously checking the state of the instance
- Getting IPs and public DNS of the slaves
- Copying the test plan to the slaves
- Distributing the threads on slaves
- Running the test plan on slaves
- Saving the result files

4.7. Credibility of Results

Latency and throughput are the key considerations for any website. These two KPIs are extremely important to developers as the response time of the web site and the skill to handle large amounts of traffic are directly related to the user experience.

Primarily due to the propagation and transmission delays on the internet the delay comes as the latency of the web site. There are many donors to this latency starting from the DNS lookup, to the link bandwidth etc.

The throughput increases linearly when the traffic to a web site is increased the and finally reaches to the threshold value as shown in the Figure 4.10. While on the other hand at low traffic the response time is low it starts to increase non-linearly with increasing load and continues to increase as it maxes out system resources like the CPU and memory.

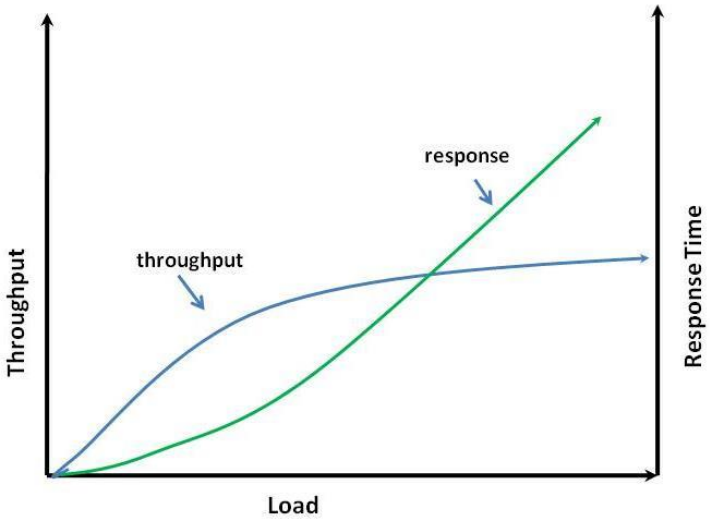


Figure 4.10: Throughput vs. response Time

The latency and throughput are key considerations while deploying applications on the cloud, which are needed to determine the kind of computing resources that are needed in the cloud. Supposing that the web application has been optimized and performance tuned for optimum performance what needs to be done is run load testing of the application on the cloud using different CPU instances. For this a small and medium instances can be utilized to plot the throughput and response time on both. And then analysis can be done to test the adequate behavior of the application [22].

Now whether to use the small instance or medium, this decision can be taken by using this formula.

Capacity of Small Instance=c

Capacity of Large Instance=C

Traffic to be handled = T

Then Instance needed

Small CPU instance it will be n= (T/c) + 1

Medium CPU instance it will be N =(T/C)+1

If

r1 = cost per hour of the small CPU instance

R1= Cost of the medium CPU instance

Then we can compare the both costs to make the choice by using following technique

r1 *n < R1 *N

By using these formulas we can measure the number of instances needed to make the adequate throughput available for the application. So in this way we can have more credible results on as we can compare the results of application test by varying the throughput.

Chapter 5: Test Results and Evaluation

5.1 Test Plan

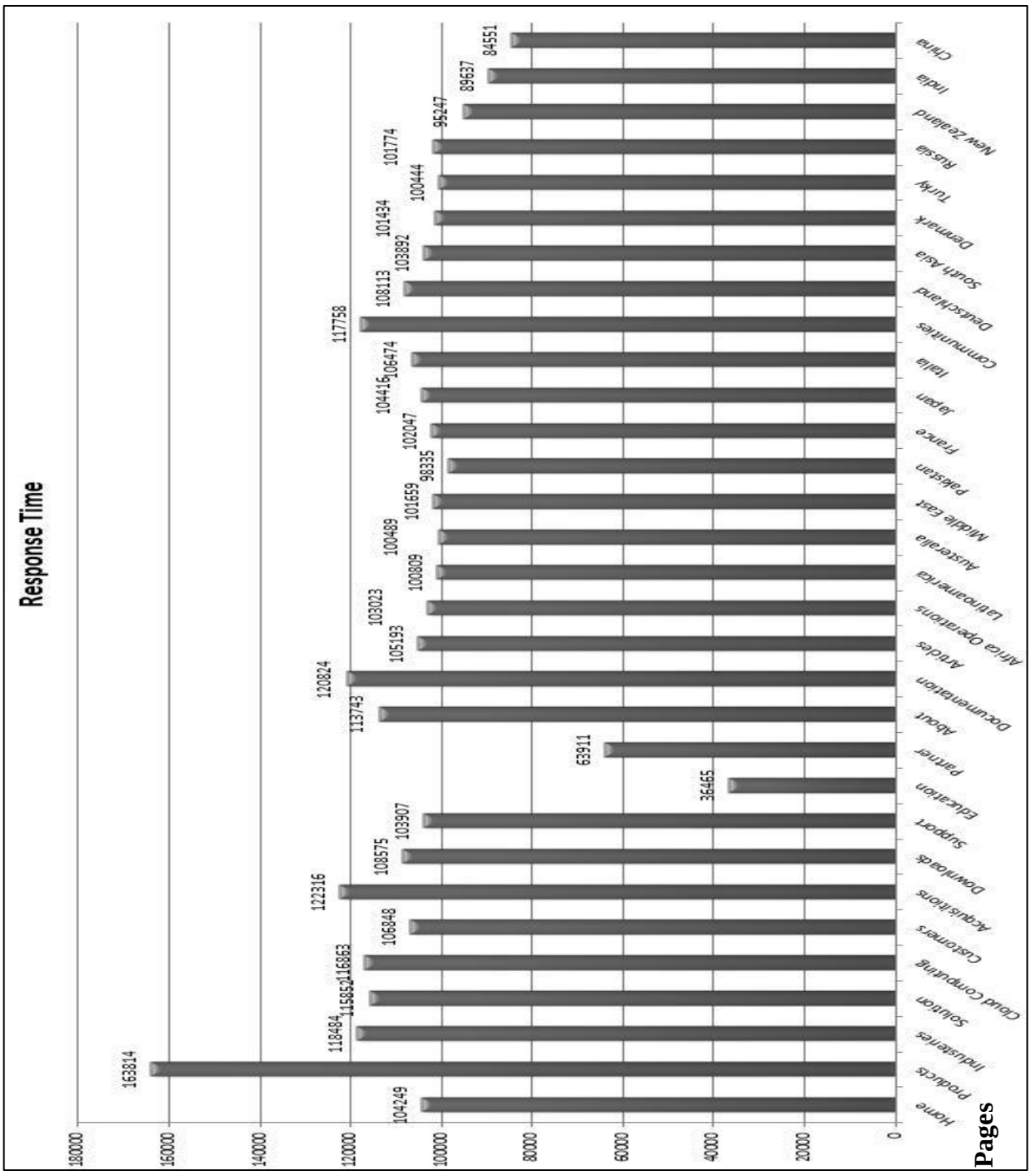
The test plan used to test the working of the STFC included the web pages of the oracle. There were 31 pages added into the test plan with the number of 100 concurrent users and loop count was set as 2 so a total of 6200 samples were tested.

The test plan was first executed in the local environment then on the cloud, the results had the remarkable difference, as on the cloud we have range of resources available on the demand so the throughput high. The Throughput and the response time are inversely proportional to each other so the response time reduces with increase in the through put. The test results are discussed in next section.

5.2. Test Results

5.2.1 Response Time

The response time of the both scenarios i.e. local environment and the cloud have remarkable difference this difference is basically because of throughput, bandwidth, transmission time and storage differences. Figure 5.1And Figure 5.2shows the average response time of the test in the local environment and on cloud respectively.



Response Time (Msec)

Figure 5.1: Response Time in Local Environment

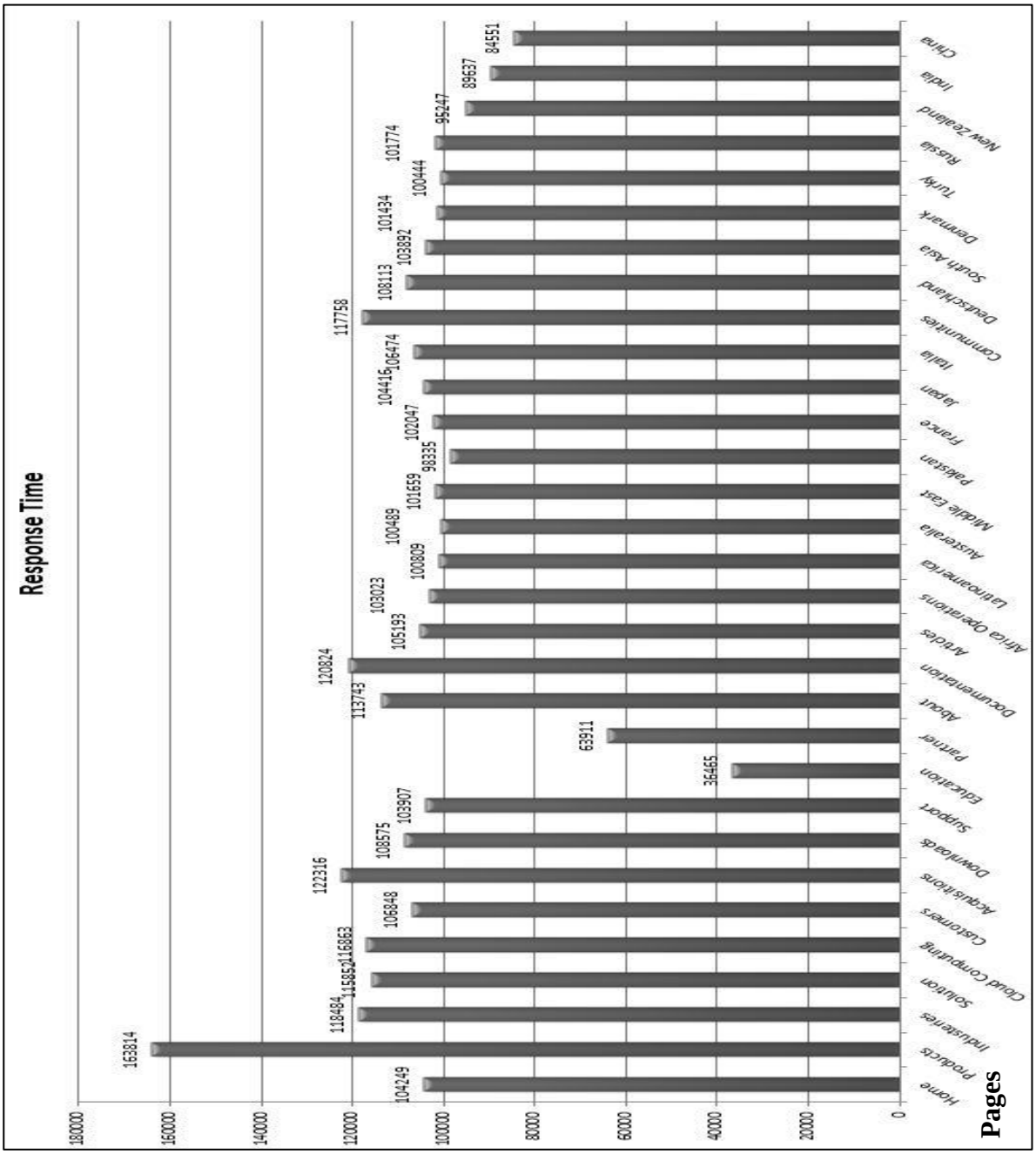


Figure 5.2: Response Time on Cloud

5.2.2 The response Time comparison

The Figure 5.3 shows the comparison of the response time on cloud and local environment.

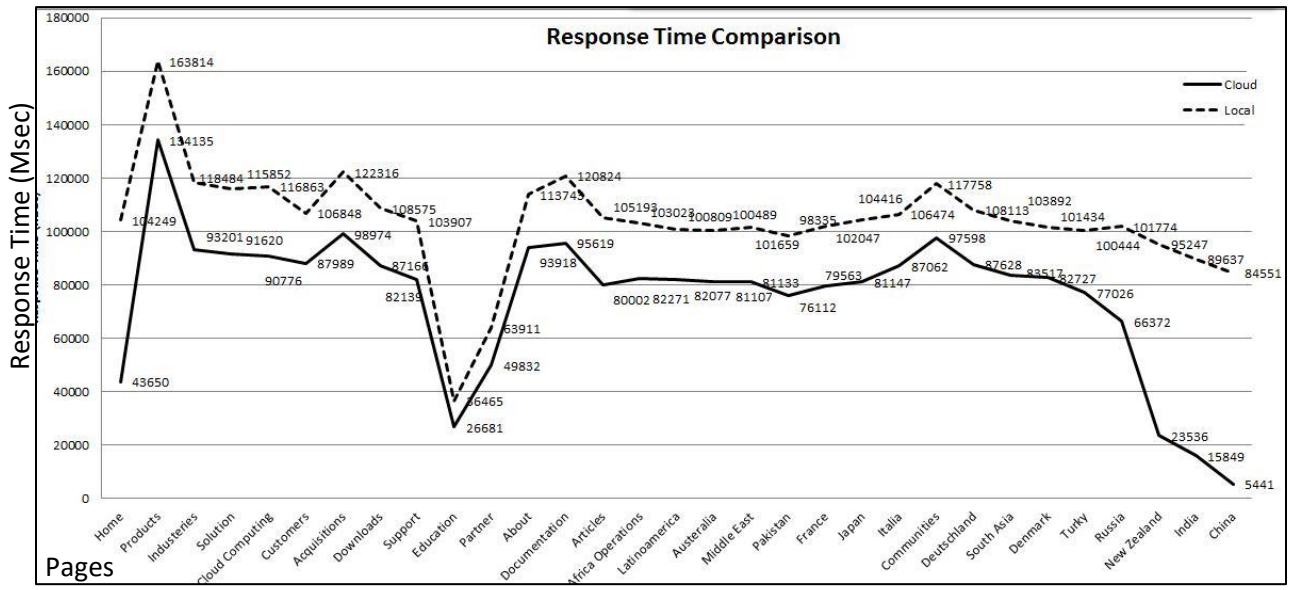


Figure 5.3: Response Time Comparison local vs. Cloud

5.2.3Summary Report

The summary report in the Jmeter is generated to provides the details of the response time and throughput

Table 5.1: Summary report in local Environment

Pages	**Sample#**	**Average**	**Minimum**	**Maximum**	**Error%**	**Bandwidth**	**Average bytes**
Home	200	104249	33650	181343	0	7.117665	132962
Products	200	163814	114135	247388	0	10.73916	206148
Industries	200	118484	95201	204188	0	7.563149	144921.4
Solution	200	115852	90620	192839	0	7.357057	141160.5
Cloud Computing	200	116863	99776	218815	0	7.418434	142812

Customers	200	106848	87789	181885	0	6.906394	132694.8
Acquisitions	200	122316	98974	223869	0	7.845371	151455.5
Downloads	200	108575	87166	170688	0	7.211207	138214
Support	200	103907	82139	219515	0	7.021908	135076
Education	200	36465	26681	94874	0	1.854045	35017
Partner	200	63911	49832	130631	0	3.992039	75848
About	200	113743	93918	218957	0	7.559922	146235.8
Documentation	200	120824	95619	193327	0	7.810304	152830.3
Articles	200	105193	80002	179186	0	6.627403	129063
Africa	200	103023	82271	172572	0	6.352429	125450.8
American	200	100809	82077	179502	0	6.29299	124654
Australia	200	100489	81107	188541	0	6.199369	123063.9
Middle East	200	101659	81133	199316	0	6.38376	127269
Pakistan	200	98335	76112	172085	0	6.042227	120276
France	200	102047	79563	175212	0	6.386633	126822
Japan	200	104416	81147	185114	0	6.268898	124566.3
Italia	200	106474	87062	220422	0	6.518838	129632.8
Communities	200	117758	97598	177726	0	7.455361	148287
Deutschland	200	108113	87628	183888	0	6.735925	133284
South Asia	200	103892	83517	199920	0	6.422258	127279
Denmark	200	101434	72727	164804	0	6.385563	126657
Turkey	200	100444	67326	185255	0	6.336103	125706.6
Russia	200	101774	56372	188473	0	6.40672	125892
New Zealand	200	95247	23636	149411	0	6.532632	126130
India	200	89637	14849	189042	0	6.596035	124551
China	200	84551	5321	193983	0	7.260526	133612

Table 5.2: Summary report on Cloud

Page	**Sample #**	**Average**	**Minimum**	**Maximum**	**Error%**	**Bandwidth**	**Average bytes**
Home	200	43650	1999	91571	0	1008.158	132962
Products	200	134135	2998	122407	0	1623.28	211952
Industries	200	93201	2999	111703	0	1122.39	145729
Solution	200	91620	4998	101989	0	1103.589	142655
Cloud Computing	200	90776	3999	91605	0	1141.47	142812
Customers	200	87989	4998	91903	0	1110.003	133351
Acquisitions	200	98974	2366	81742	0	1327.99	152964
Downloads	200	87166	1999	101639	0	1254.761	138214

Support	200	82139	1197	91658	0	1171.025	135076
Education	200	26681	1000	30153	0	301.7941	35017
Partner	200	49832	2019	53485	0	638.9289	75852
About	200	93918	1074	105669	0	1267.575	147050
Documentation	200	95619	1999	100778	0	1317.304	153588
Articles	200	80002	1999	102334	0	1083.308	129312
Africa	200	82271	1000	91472	0	1031.729	126071
American	200	82077	1000	81456	0	1049.101	124654
Australia	200	81107	2000	91369	0	1064.956	124286
Middle East	200	81133	1000	91416	0	1119.14	127269
Pakistan	200	76112	3509	81438	0	957.8555	120276
France	200	79563	2977	91485	0	999.9565	126822
Japan	200	81147	3113	91396	0	999.7796	125182
Italia	200	87062	2810	81458	0	1065.974	130321
Communities	200	97598	3775	101401	0	1243.398	148377
Deutschland	200	87628	1463	91398	0	1104.035	133284
South Asia	200	83517	1677	81374	0	1094.684	127279
Denmark	200	82727	1871	71379	0	1113.157	126657
Turkey	200	77026	1594	91405	0	1144.78	126328
Russia	200	66372	1737	91407	0	1181.731	125892
New Zealand	200	23536	1567	81351	0	1231.8	126130
India	200	15849	1135	61400	0	1264.101	124551
China	200	5441	1233	61469	0	1413.197	133612

5.2.4. Through Put

Throughput is the number of the requests handled per time unit. The throughput recorded by Jmeter on local and cloud is given in the table 5.3

Table 5.3: Throughput local vs. Cloud

Environment	**Throughput**
Local	56.984/Min
Cloud	9777.124/Min

5.2.5. Graph Results

The Graph Results listener generates a simple graph that plots all sample times. The throughput number represents the actual number of requests/minute the server handled. The advantage of doing the calculation like this is that this number represents something real [23]

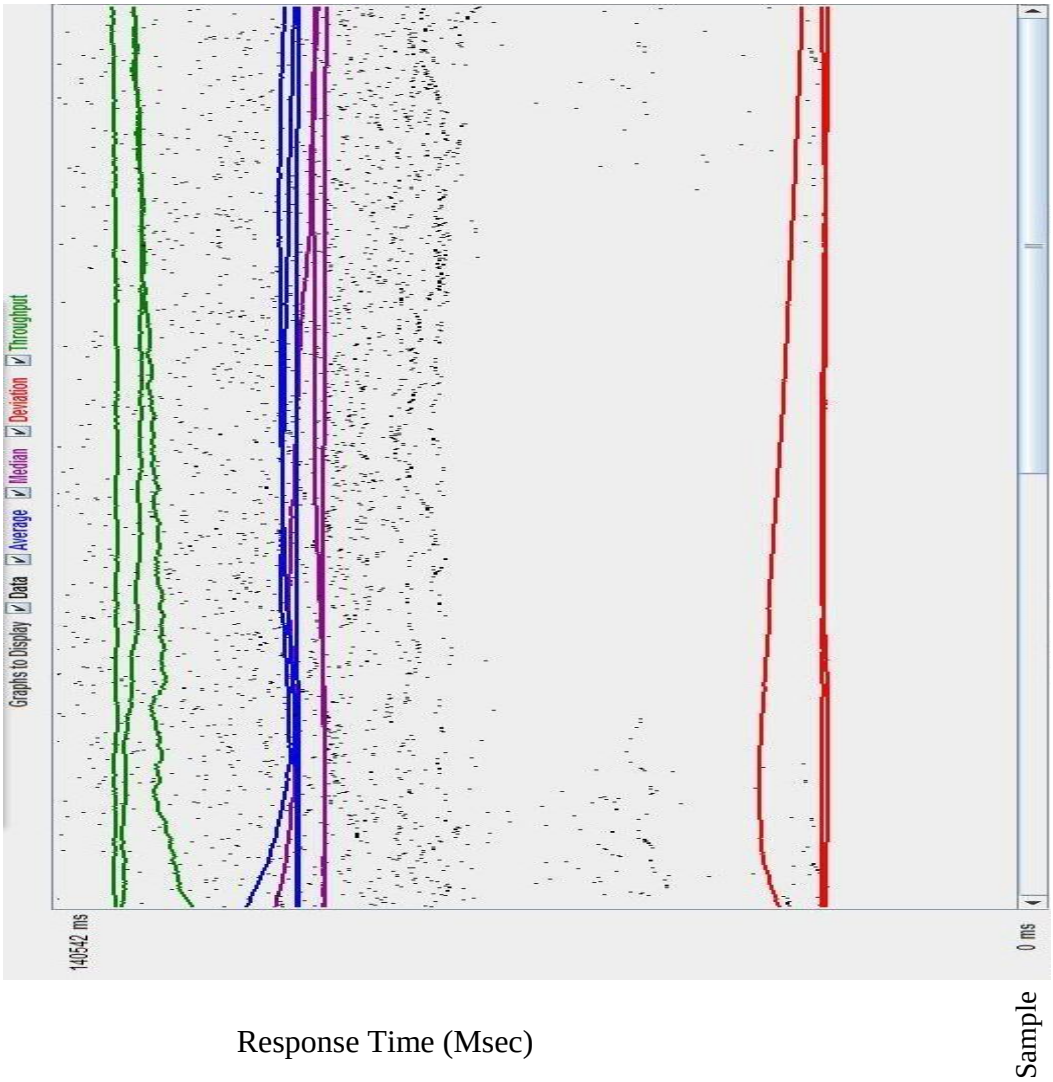


Figure 5.4: Graph results in local environment

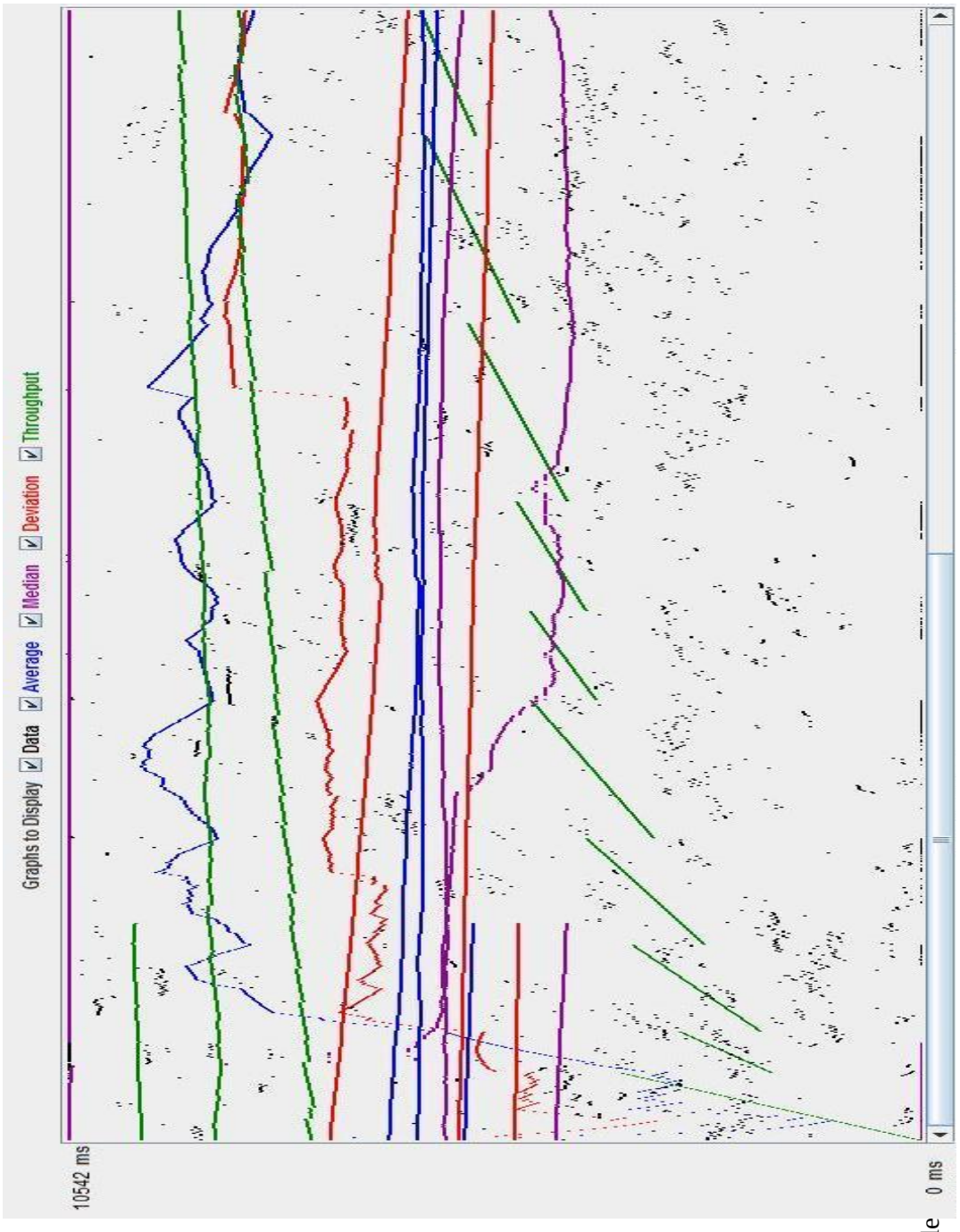


Figure 5.5: Graph results on cloud

5.3. Evaluation

The basic instruction regarding response times has been about the same for thirty years .The response time is idealized according to the user acceptance rate.

- **0.1 second** meaning that no special feedback is necessary except to display the result, user feels that the system is reacting instantaneously,
- **1.0 second** is about the limit for the user's flow of thought to stay uninterrupted, even though the user will notice the delay. Normally, no special feedback is necessary during delays of more than 0.1 but less than 1.0 second, but the user does lose the feeling of operating directly on the data.
- **10 seconds** is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users will want to perform other tasks while waiting for the computer to finish [24].

Table 5.4: Response Time per Request

Pages	Cloud	Local
Home	0.21825	0.521245
Products	0.670675	0.81907
Industries	0.466005	0.59242
Solution	0.4581	0.57926
Cloud Computing	0.45388	0.584315
Customers	0.439945	0.53424
Acquisitions	0.49487	0.61158
Downloads	0.43583	0.542875
Support	0.410695	0.519535
Education	0.133405	0.182325
Partner	0.24916	0.319555
About	0.46959	0.568715

Documentation	0.478095	0.60412
Articles	0.40001	0.525965
Africa	0.411355	0.515115
Latin-American	0.410385	0.504045
Australia	0.405535	0.502445
Middle East	0.405665	0.508295
Pakistan	0.38056	0.491675
France	0.397815	0.510235
Japan	0.405735	0.52208
Italia	0.43531	0.53237
Communities	0.48799	0.58879
Deutschland	0.43814	0.540565
South Asia	0.417585	0.51946
Denmark	0.413635	0.50717
Turkey	0.38513	0.50222
Russia	0.33186	0.50887
New Zealand	0.11768	0.476235
India	0.079245	0.448185
China	0.027205	0.422755

So it is clearly visible from the results that the response time recorded on the cloud is less then local environment that is because bandwidth, throughput, and delays. As on cloud we have range of resources available on demand so it best suits for the testing purpose.

5.4. Summary

The comparison of the Response time and the throughput between local and cloud environment tests shows that as on the cloud we have high throughput so the response is quite down, we can test our application's response time by increasing and decreasing the throughput to get a real time idea of the application's behavior under stress when a large number of concurrent users access it.

Chapter 6: Conclusion and Future work

6.1. Conclusion

Software testing in a technique of evaluating the system behavior, now a day because of agile technologies software is becoming more complex, with increase in the complexity the testing of the applications is also becoming an effort intensive task. User has to manage and maintain multiple open source software testing tools to test different aspects of the application. It becomes hectic to manage and maintain a large range of software on local machines and allocating resources etc. so a software testing framework (STFC) is developed to provide a common platform to the end user from where a user can access all the open source software testing tools on demand, without worrying about their management and the maintenance. User can access them through internet and can run his test plans to get more accurate and credible results as compared with the local environment.

STFC not only provides integrated environments for ease of use and maintainability but also ensures more accurate and credible results. We performed tests in the cloud using Jmeter and the results had a remarkable difference. Our ultimate goal is to place multiple open source software testing tool over one platform so that user can access them on demand without worrying about their maintenance and management.

6.2. Future Work

Software testing is a necessary phase of the software development life cycle. The quality of the product can be enhanced by following the different testing techniques. In future the

work can be done to implement the solution on the smart phones so that the user can get the results of the test plan on the screen of the smart phone. The user should be provided with interactive charts and graphs for the analysis purposes.

REFERENCES

[1] “Advantages of Cloud Computing,” Available: http://www.webhostingreport.com/ learn /advantages-of -cloud-computing.html, [Accessed: Sep. 2010]

[2] “Modern Software Testing,” Available: http://modern-testing.blogspot.com/, [Accessed: Feb. 2010]

[3] “Cloud computing”, Available: http://en.wikipedia.org/wiki/Cloud_computing,[Accessed: Oct. 2010]

[4]L.Riungu,O.Taipale, K.Smolander “Resaerch issues of software testing in the cloud” in proceedings of2nd IEEE International Conference on Cloud Computing Technology and Science, 2010

[5] “Choosing the Right Cloud Platform”, Available: http://pcquest.ciol.com/content/techtrends /2010/110040105.asp, [Accessed: Jun. 2010]

[6] “Microsoft shares SaaS architecture insights” available: http:// www. Zdnet .com/blog /saas /Microsoft-shares-saas-architecture-insights/239, [Accessed: Sep. 2010]

[7] “Comparison - Cloud Offerings” Available: www.blog.kitetail.com,[Accessed: Feb. 2010]

[8] “Amazon Elastic cloud developer’s guide”, available: www.aws.amazon.com, November 30, 2009

[9] “Overview of security process”, White Paper, Amazon Web Services, August 2010.
[10] “Future of Software Testing,” White Paper, App Labs, July 2008
[11] M. Vasilian, “Trends in Software Testing,” SoftwareDev Explained, [Accessed: August 2010].
[12] E.Sha,“Trends in Software Testing for 2011,” information Bible, Jan 2011. Available: http://www.informationbible.com/article-trends-in-software-testing-for-2011-35378.htm, [Accessed: April, 2011]
[13] “PerformanceXpert Pro,” Available: http://www.cloudintelligence.com/applications/j meter/about/, [Accessed: March. 2011]

[14] “RTTS-Performance Testing in Cloud” Available: http://ww w.rttsweb.co m/services/imp lamentation/performance/packages, [Accessed: March. 2011]

[15] “Cost effective web load testing in cloud,” Available: http://www.attenda.net/e-zines /clients /100910/clients-sub1-WebLoadTest-0910.asp,[Accessed: March. 2011]

[16] “HP LoadRunner software in the Cloud,” https://h10079.www1.hp.com,[Accessed: March. 2011]
[17] Jaideep, “Five benefits of software testing on cloud,” Available: http://itknowle dgeexchange.techtarget.com/quality-assurance/five-benefits-of-software-testing-on-cloud/, [Accessed: May 2010]
[18] T.ParveenS.Tilley “When to migrate Testing on Cloud” Third International Conference on Software Testing, Verification, and Validation Workshops, IEEE, July, 2010
[19] L.Riungu,O.Taipale, K.Smolander “Research issues of software testing in the cloud” 2nd IEEE International Conference on Cloud Computing Technology and Science, [Accessed: 5 Jun, 2011]
[20]L.Yu, W.Tsai1, X.Chen, L.Liu, Y. Zhao, L.Tang, W.Zhao “Testing as service over cloud” in Proceedings of Fifth IEEE International Symposium on Service Oriented System Engineering, Dec. 2010
[21] “EC2 Documentation” Available:http://docs.amazonwebservices.com/
[22] Tinniam V Ganesh“Latency, throughput implications for the Cloud”, Available: http://gigadom.wordpress.com/2011/04/13/latency-throughput-implications-for-the-cloud/, April, 13, 2011

[23]“Jmeter User Guide”, Available: Http: //jakarta.apache.org/jmeter/usermanual/component referene.html#GraphResult, [Accessed: Jan. 2011]
[24] “Response Time: The 3 Important Limits”, Available: http://www .useit.com/papers/respon setime.html, [Accessed: May. 2011]