



# **Comparative Study of Performance Testing Tools: Apache JMeter and HP LoadRunner**

**Rizwan Bahrawar Khan**

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

**Contact Information:**

Author:

Rizwan Bahrawar Khan

riba09@student.bth.se

University advisor:

Dr. Mikeal Svahnberg

School of Computing

Faculty of Computing  
Blekinge Institute of Technology  
SE-371 79 Karlskrona, Sweden

Internet : [www.bth.se](http://www.bth.se)  
Phone : +46 455 38 50 00  
Fax : +46 455 38 50 57

## **ABSTRACT**

Software Testing plays a key role in Software Development. There are two approaches to software testing i.e. Manual Testing and Automated Testing which are used to detect the faults. There are numbers of automated software testing tools with different purposes but it is always a problem to select a software testing tool according to the needs.

In this research, the author compares two software testing tools i.e. Apache JMeter and HP LoadRunner to determine their usability and efficiency. To compare the tools, different parameters were selected which guide the tool evaluation process. To complete the objective of the research, a scenario-based survey is conducted and two different web applications were tested.

From this research, it is found that Apache JMeter got an edge over HP Loadrunner in different aspects which include installation, interface and learning.

**Keywords:** Software Testing, Automated Software Testing, Performance Testing, Web Applications

## ACKNOWLEDGMENT

I am specially thankful to **Almighty Allah** who helped me to complete this thesis.

I would like to thanks my supervisor Dr. Mikeal Svahnberg for his valuable comments and direction which helped me to achieve this milestone. I would like to thanks my all family members and my friends who supported and motivated me for work. In addition, I would like to thank all the participants who helped me in completing my survey.

I would specially thank my parents, my brothers, my nephews (Aliyar and Ibrahim) and my little angel niece (Shazmeen) for their encouragement and support without them I cannot be able to finish my thesis.

I dedicate my thesis to my beloved father Mr. Bahrawar Khan.

## List of Tables

|  |    |
|--|----|
| Table 1: <i>Usability Testing Parameters Definition</i> .....      | 24 |
| Table 2: <i>Technical Requirement Parameters Definition</i> .....  | 24 |
| Table 3: <i>Performance Testing Parameters</i> .....               | 24 |
| Table 4: <i>Sample Web Services</i> .....                          | 25 |
| Table 5: <i>Hardware Specification</i> .....                       | 26 |
| Table 6: <i>Scenario Based Questionnaire</i> .....                 | 29 |
| Table 7: <i>Scenario Based Questionnaire</i> .....                 | 30 |
| Table 8: <i>Set of Questions</i> .....                             | 31 |
| Table 9: <i>Test Environment Criteria</i> .....                    | 33 |
| Table 10: <i>Result for Technical Requirement Parameters</i> ..... | 40 |
| Table 11: <i>Result for Usability Test Parameter</i> .....         | 41 |
| Table 12: <i>Average Response Time for WS1</i> .....               | 42 |
| Table 13: <i>Average Response Time for WS2</i> .....               | 42 |
| Table 14: <i>Average Bytes Processed WS1</i> .....                 | 42 |
| Table 15: <i>Average Bytes Processed WS2</i> .....                 | 42 |

## List of Figures

|   |           |
|---|-----------|
| <i>Figure 1: Research Question Flow Chart .....</i>           | <i>23</i> |
| <i>Figure 2: Web Application Architecture .....</i>           | <i>33</i> |
| <i>Figure 3: Apache JMeter Front .....</i>                    | <i>35</i> |
| <i>Figure 4: Apache JMeter Thread Group .....</i>             | <i>36</i> |
| <i>Figure 5: Apache JMeter Test Script Recorder .....</i>     | <i>36</i> |
| <i>Figure 6: Apache JMeter Summer Report .....</i>            | <i>37</i> |
| <i>Figure 7: HP LoadRunner – Virtual User Generator .....</i> | <i>37</i> |
| <i>Figure 8: HP LoadRunner – Controller .....</i>             | <i>38</i> |
| <i>Figure 9: HP LoadRunner Analysis Summary.....</i>          | <i>38</i> |
| <i>Figure 10: Usability Test Parameters Result.....</i>       | <i>46</i> |
| <i>Figure 11: Average Response Time Result for WS1.....</i>   | <i>47</i> |
| <i>Figure 12: Average Response Time Result for WS2.....</i>   | <i>47</i> |
| <i>Figure 13: Average Bytes Processed for WS1 .....</i>       | <i>48</i> |
| <i>Figure 14: Average Bytes Processed for WS2 .....</i>       | <i>48</i> |

# TABLE OF CONTENTS

|   |    |
|---|----|
| Abstract.....   | i  |
| Table of Contents .....   | v  |
| 1. Introduction.....  | 7  |
| 1.1. Proposed Solutions.....  | 8  |
| 1.2. Research Questions .....   | 9  |
| 1.3. Motivation for Research Questions.....                             | 9  |
| 1.4. Outline of the Research .....                                      | 10 |
| 2. Background and Related Work .....                                    | 11 |
| 2.1. Importance of Software Testing.....                                | 11 |
| 2.2. Methods of Software Testing.....                                   | 12 |
| 2.3. Performance Testing Tools.....                                     | 14 |
| 2.4. Related Work .....   | 16 |
| 3. Methodology .....  | 19 |
| 3.1. Selected Tools .....   | 19 |
| 3.2. Evaluation Parameters.....   | 22 |
| 3.3. Parameters Overview .....  | 23 |
| 3.4. Suitability of Selected Parameters .....                           | 25 |
| 3.5. Targeted Websites.....   | 25 |
| 4. Empirical Work .....   | 27 |
| 4.1. Identifying Usability requirements of Software Testing Tools ..... | 27 |
| 4.2. Pilot Test and Survey Questionnaire.....                           | 28 |
| 4.3. Survey Information .....   | 29 |
| 4.4. Scenario Design.....   | 29 |
| 4.5. Questionnaire for Usability Evaluation.....                        | 31 |
| 4.6. Validation of Scenario.....  | 31 |
| 5. Web Application Performance Testing.....                             | 33 |
| 5.1. Identifying Test Environment .....                                 | 33 |
| 5.2. Identify Performance Acceptance Criteria .....                     | 34 |
| 5.3. Implementation of Experiment .....                                 | 34 |
| 5.4. Data Collection .....  | 34 |
| 5.5. Front End Analyses .....   | 34 |
| 5.6. Independent Variables and Dependent Variables .....                | 39 |
| 6. Results.....   | 40 |
| 6.1. Result of two testing tools .....                                  | 40 |
| 7. Validity Threats .....   | 43 |

|   |    |
|---|----|
| 7.1. Internal Validity .....                | 43 |
| 7.2. External Validity .....                | 43 |
| 7.3. Construct Validity .....               | 44 |
| 7.4. Conclusion validity .....              | 44 |
| 8. Analysis and Discussion .....            | 45 |
| 8.1. Technical Requirement Parameters. .... | 45 |
| 8.2. Usability Test Parameters.....         | 45 |
| 8.3. Performance Test Parameters .....      | 47 |
| 9. Conclusion and Future Work .....         | 49 |
| References.....                             | 50 |
| Appendix I.....                             | 53 |

# 1. INTRODUCTION

Software testing is required to improve the quality of software because if software is error-free, user-friendly and provide client satisfaction is considered to be high-quality software [1]. Software testing is a vital part of Software development in providing a quality software product i.e. bugs and defects free [2]. Software testing is the process of working out or calculating a system or system component to identify if it satisfies specified requirements or to identifies the difference between expected or actual results [3]. Software testing is also defined as a process in which the business and technical requirement of the product are accordingly validated and verified [4]. Testing of software is important to be done carefully and efficiently for deployment in order to retain existing customers and also attract new customers [5]. Testing is important because the reliability of software is defined by testing and approximately fifty percent of the software development budget is spent on testing [5]. Software testing is important because errors are done unintentionally as it is designed and constructed [2].

Nowadays, the software developed are more complex which results in more line of code and more thorough testing are needed to be done [2]. Software testing is a wide and ongoing activity which is performed during the software development and maintenance process [6]. The Software Development Life Cycle includes: Requirements, Design, Code, Test and deployment and maintenance [7]. Software testing is considered as an important part of software development life cycle since it determines whether the software is ready and to be released.

There are two types of Software Testing i.e. Manual Testing and Automated Testing [2] [3]. The basic method of software testing is Manual Testing. Manual testing is a method where software tester follows a test plan written and these written test plans lead them to a set of important test cases [3]. Manual testing has a lot of disadvantages such as consuming time and cost, require experience, less efficient, complex reusing and no scripting facilities for coding. On another hand, Automated Testing reduces the need for manual or human involvement in software testing. Automated testing covers all the problems which are in manual testing [3] for

example consuming time and cost, require experience, complex reusing, less efficiency and not provide scripting facility for code. [8].

According to K. Shaukat et al [9], many researches have been done on software testing tools but there are no precise criteria. K. Shaukat et al compared 32 software testing tools based on their attributes such as operating system, browser support, license, cost and some other attributes and presented disadvantages of automation testing tools such as knowledge and learning required to use the tools, some automation tools are not free and are expensive and it is difficult task to maintain and is expensive.

In this research, two performance testing tools will be compared to determine their usability. This research will also show if there is any difference between a licensed or open source performance testing tools, therefore, one licensed based performance testing tool and one open source performance testing tool will be selected for the comparison.

### **1.1. Proposed Solutions**

The objective of this research is to evaluate and compare two automated software testing tools to identify their usability and effectiveness. To achieve these objectives, the researcher will:

- Select two automated software testing tools to be evaluated.
- Development of a parameter suite used to evaluate the tools.
- Select target application to be tested with the tools.
- Implement feature assessment for the selected tools with the aim of ranking the tools based on their features.
- Test the selected applications by the selected automated testing tools.
- Interpretation and evaluation of the results.

## **1.2. Research Questions**

*RQ1 What are the testing capabilities of JMeter and Loadrunner with respect to usability?*

*RQ2 What are the testing efficiencies of JMeter and Loadrunner with respect to average response time?*

## **1.3. Motivation for Research Questions**

### **1.3.1. Research Question 1**

Jackob Nelsion defines usability as ““Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process” [10]. Usability testing is a vital method for testing how system users understand and use a system to achieve a specific work [11]. Usability as an objective is identical with quality of use i.e. that the product can be used in real world. Thus usability has two complementary roles i.e. a product designed at a highest level of quality objective and necessary to set and evaluate measureable targets for usability to identify and resolve usability defects [12]. By conducting usability test of Apache JMeter and HP LoadRunner, selected the user satisfaction [13] will be compared.

### **1.3.2. Research Question 2**

Response time measure how long it takes for a response to return back from the application when a request is sent [14]. In performance testing, response time is important because it represent how long a user must wait for a request to be processed by the application. If the response time is slow, the user experiences a delay in getting the response from the application. The response time parameter are evaluated to find the performance of the tools and the result obtained will analyzed to determine the efficiency of the testing tools [15].

#### **1.4. Outline of the Research**

Chapter 2 presents the related work. Chapter 3 represents the methodology including the selected tools and evaluation parameters. The result is presented in chapter 4. Chapter 5 presents the discussion and lessons learned. In chapter 6, conclusion and future work is discussed.

## **2. BACKGROUND AND RELATED WORK**

### **2.1. Importance of Software Testing**

Humans can commit errors in any process, some of the errors can be ignored as those errors do not have much impact on our daily routine, but some errors are very important and they can break the whole system or software. In such situations, precautions should be taken to catch them in advance before deployment of the system/software in production environment [16].

Testing is the important process in software engineering [17]. As the digital computers spread and increased with a parallel increase in the cost of software failure, the important of test grew [18]. For highly reliable systems, software testing plays a significant role in the process of software development [19]. The purpose of testing is to identify problems and fix them to improve quality [20]. Software testing normally allocated 40% of a software development budget [20]. The four main objectives of testing are Demonstration, Detection, Prevention and Improving Quality [20].

Software testing plays an important role in the Software Development-Life Cycle in order to identify the difficulties in the process [21]. If, the problems and bugs are identified during the development process are easier to clear and can make the software more reliable.

The selection and combination of the most proper testing techniques is a problem in software engineering communities and in industrial practice as well [22]. The challenge is to identify what are the most effective methods for testing of specific software and what methods are more reliable to detect a specific type of faults [22].

## **2.2. Methods of Software Testing**

There are several types of testing methods that have been designed [20]. Each method is used for the different purpose of testing different artifacts like designing, coding, planning, requirement specification [20].

### **2.2.1. Manual Testing**

The basic method of software testing is Manual Testing. Manual testing is a method where software tester follows a test plan written and these written test plans lead them to a set of important test cases [3]. It is not required to have knowledge of any testing tool in Manual Testing. It is not 100% possible to automate a basic software testing fundamental and because of this, manual testing is vital [23].

### **2.2.2. Automated Testing**

Automated testing uses an automation tool to run the test case suite [23]. Manual testing has a lot of disadvantages such as consuming time and cost, require experience, less efficient, complex reusing and no scripting facilities for coding. On another hand, Automated Testing reduces the need for manual or human involvement in software testing. Automated testing covers all the problems which are in manual testing [3] [8]. Automated testing can expressively decrease the cost of software development and maintenance [24]. It is possible to record and replay the test suite by using a test automation tool and if the test suite is automated, no human involvement is required [23].

### **2.2.3. Regression Testing**

Software has to be modified in order to remove bugs or if there is any change in the user specification. . Regression testing must be done to identify if recent program changes have not affected existing features. All the test cases generated at earlier stages are re run by the tester to ensure that the program behaves as expected [25]. Regression testing compare the performance of new version to the performance of an older version to make it sure that no fault are made after modification. If the result produced by two version are different, regression faults are identified [26].

#### **2.2.4. Unit Testing**

Unit testing is a testing method by which individual components of source code are tested to decide if they are fit for use. The smallest testable part of an application is a unit and it can be an individual function or procedure [27]. A unit test is a technique without parameters that execute a sequence of methods that run the code under test and shows the expected behavior of the code [28].

#### **2.2.5. Integration Testing**

Integration testing is done after Unit Testing. The objective of integration testing is to put the units in their proposed environment and execute their interactions as completely as possible [29].

#### **2.2.6. System Testing**

System testing is define as “Testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements” [30]. System testing is considered as an important step in quality management process which allows tester to verify and validate the application architecture and their requirements. In system testing is used not only to identifying and debugging failures but also preventing potential failures [31].

#### **2.2.7. Acceptance Testing**

Acceptance testing is defined as ““Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component” [30] [State-of-Practice in GUI-based System and Acceptance Testing: An Industrial Multiple-Case Study].Acceptance testing is one of the important phase in software development life cycle. The essential part of acceptance testing is the verifying the external behavior of the software. The aim of acceptance testing is to validate how well the developed software system facilitates the customers” requirements. The focus of acceptance testing is on verifying man-machine interactions, required function features and specified system constraints [32].

#### **2.2.8. Performance Testing**

Non-functional testing is used to measure the software characteristics such as response time, page load times, peak load limit, threshold limit for best performance

of the software product. [16]. Performance Testing is a type of a non-functional testing which tests the performance of software under the satisfactory and unsatisfactory conditions. This includes all time-related parameters for example load time, access time, run time, execution time, success rate, failure frequency, mean time between failures and overall reliability of software [33]. Performance testing involves testing software applications to confirm they will perform under their expected workload [16].

#### **2.2.9. Load Test**

Load Test is a process of testing the capabilities of system in peak load where maximum number of concurrent users accessing system at same time [34]. The purpose of Load Testing is to obtain the web application response time while the number of users are changing and long with this, simulate the user access for web performance evaluation. This assessment can be done as number of web user at time or amount of data processed online [35].

#### **2.2.10. Stress Testing**

The Stress testing method was mostly used by those who are responsible to develop and manufacture hardware for example, the stress mostly include temperature extremes, thermal cycles, vibrations etc. The stresses are effective in quickly identifying hardware defects from degradable, minimum or intermittent failures to hard failures so that the root cause analyses and helpful actions may be made [36].

### **2.3. Performance Testing Tools**

The automated performance software testing tools which are currently available are [9];

#### **2.3.1. HP LoadRunner**

HP LoadRunner [37] is the product of HP. It is an industrial standard based performance automated tool for load testing of applications which is used for studying system performance and behavior. It works by using virtual users. It simulates thousands of real-time users to put the application through user loads and thoroughly analyses the results to identify the particular behavior.

### **2.3.2. Apache JMeter**

Apache JMeter [38] is a product of Apache Software Foundation designed to load test functional behavior and measure performance. It is an open source java application. Apache JMeter was basically designed for testing web applications but now it can be used for performance test both on static and dynamic resources. Distributed testing feature is one of the strong points of Apache JMeter.

### **2.3.3. Selenium**

Selenium [9] is used to test web application. It consists of four tools i.e. Selenium IDE, Selenium RC, Selenium Web Drive and Selenium Grid. Selenium is an open license and supports different web browsers.

### **2.3.4. NeoLoad**

NeoLoad [38] is used to measure the performance of the web application. NeoLoad provides pragmatic solution facilitating design and development of the optimized websites. Neotys is a French company which owned, maintained and developed NeoLoad. It provides monitoring of the user response times and infrastructures statistics. It can run on most of the operating systems. A non-specialist can easily create a test by using the NeoLoad automation testing tool.

### **2.3.5. WebLoad Professional**

WebLoad Professional [38] is developed by RadViews for Performance testing of internet applications. WebLoad Professional include built-in support for Ajax technologies, JSON data types and different types of SOAP and XML web services. It supports Windows operating systems as well as Linux machines. The proxy recorder is used for recording business processes. The WebLoad result test script is written in JavaScript. It is an open source project.

### **2.3.6. LoadUI [9]**

LoadUI is an open source performance testing tool used for load testing. LoadUI is used to check the performance of web application. LoadUI can also work with

SoapUI really efficiently. LoadUI is the most flexible and interactive performance testing tool.

### **2.3.7. WAPT [39]**

Web Application Performance Tool or WAPT, is used to test web application and the interface of web. WAPT is used for performance, Load and Stress Testing of web application. WAPT specify to the tester that how many virtual users are using the testing environment i.e. either increasing, constant or interrupted users load.

### **2.3.8. Loadster [40]**

Loadster is a Load Testing Tool which is used for test solutions for websites, web application and web services. Loadster is built for real web applications and handling cookies, user sessions, custom header, dynamic form data. Loadster create an testing environment where single user state and Loadster gathers statistics for each virtual user individually.

### **2.3.9. LoadImpact [41]**

LoadImpact is a performance testing tool where it simulate traffic exactly how it would happen in real life in a testing enviroment. LoadImpact generate scripts automatically and there is no need of programming. LoadImpact also measure the usage of CPU, Memory Usage, Disk I/O and Network I/O.

## **2.4. Related Work**

Many research articles are found regarding the comparison of different software testing tools. [42] performed an empirical comparative study with two Visual GUI Testing (VGT) tools i.e. Sikuli and CommercialTool which are used for acceptance testing, were compared based on their static properties and their ability to automate industrial test cases for a safety critical air traffic management system. According to their study, there is no statistical significant differences between the tools and both selected tools are fully capable of performing the automation with equal defect identification as the manual test cases.

Emil Alegroth, Michel Nass and Helena H. Olsson [43] compared Jautomate which is a Visual GUI Testing (VGT) tool with two VGT tools i.e. Sikuli and CommercialTool used for acceptance testing, based on tool static properties. The comparison shows that JAutomate has several benefits over the other tools but the tools have different properties which are suitable in different contexts.

Research done by [44] , presented a comparative study of open source web services testing tools with technical overview and features. They did the comparison on the basis of response time. Tools selected are used for Performance testing i.e. SoapUI, Storm, SoapSonar Personal and .Net Webservice Studio. Web applications were tested using the selected tools and comparison is made on the basis of response time. From their research, it is found that SoapSanor Personal is better than other selected tools.

[2] compared two Performance testing tools QTP and Ranorex on the basis of technical overview. They compare different parameters like cost, environment support, browsers support, online support and coding support. They also present the comparison of strength and weakness of the selected tools and resulted that Ranorex is better as compare to QTP.

Meenu and Yogesh Kumar [3] compared four Performance testing tools i.e. Selenium, SoapUI, HP QTP/UFT and TestComplete on the basis of different features. They use 3-point scale i.e. 3, 2, 1 as Good, Average, and Bad respectively for the purpose of rating the comparison parameters. It results in a different value for different parameters for the selected automated testing tools. For comparative study, the calculated values of parameters are used for result and analysis. They show the result in the form of graph based on the value calculated for the selected automated testing tools and consider SoapUI as the best among the selected tools on the basis of usability.

Bhoomit Patel, Jay Parikh and Rushabh Shah [45] compare two Performance testing tools i.e. LoadRunner and JMeter on the basis of technical over. They compare the parameters i.e. Load Generating Capacity, installation, Download proficiency, Result

Reporting, Cost, Technicality of software and Reliability. They compared the selected automated testing tools for SQL Performance testing.

### **3. METHODOLOGY**

In this research, mix methodology will be used which include case study and experiment. The types of research methodology used to conduct research are Interview, Case Study, Implementation, Survey and Experiment [46]. In case study, a case can be individual, group, institutions or community [46]. In this research, there are two case study i.e. Apache JMeter and HP LoadRunner. In experiment, the results obtained is compared from an experimental group and a control group [46]. An experiment will be conducted to check the performance efficiency of the selected tools, i.e. Apache JMeter and HP LoadRunner.

#### **3.1. Selected Tools**

For selection of the tools, few steps were consider which are listed below,

1. Those tools are selected which are used for Performance testing.
2. Those tools are selected which are widely used in the industry.
3. Those tools are selected which satisfy the selected parameters.
4. Which generate auto scripts by using Record Playback function.

After considering the selection steps, two testing tools selected for the comparison i.e. Apache JMeter and HP LoadRunner. In the start, different software testing tools were selected but they were rejected because of many reasons. For example, Unified Functional Testing (UFT) and Load Impact Tool were selected but due to the complex setup of Unified Functional Testing tool, it was rejected. UFT put lot of load on CPU and RAM usage and support limited number of browsers which can impact on the usability of the tool. Later on, Load Impact tool was also rejected because it was cloud based load testing tool and no installation was required on system so the comparison was not able to be done.

The first tool selected is Apache JMeter. The reason for selecting this tool is that it is widely used in industry for performance testing and it is open source. Apache JMeter is available for free and can be downloaded from its website. The second tool selected i.e. HP LoadRunner is also widely used for performance testing. The reason

for selecting HP LoadRunner is that it is license based and users have to purchase this automation tool.

### **3.1.1. Apache JMeter**

Apache JMeter is an Apache product that is used as a load testing tool for analyzing and measuring the performance of web applications.

Apache JMeter is a testing framework from Apache and used for testing web applications. Apache JMeter is used to examine overall server performance under heavy load.

Apache JMeter features are [47];

- Apache JMeter has the ability to load test and performance test on different server/protocol like:
  - Web - HTTP, HTTPS
  - SOAP / REST
  - FTP
  - Database via JDBC
  - LDAP
  - Message-oriented middleware (MOM) via JMS
  - Mail - SMTP(S), POP3(S) and IMAP(S)
  - MongoDB (NoSQL)
  - Native commands or shell scripts
  - TCP
- Apache JMeter is complete portable and 100% java pure.
- Apache JMeter has full multithreading framework which allows concurrent sampling by many threads and simultaneous sampling of different functions by separate thread groups.
- Its GUL is carefully designed which allows faster test plan building and debugging.
- Apache JMeter has pluggable samples which allow unlimited testing capabilities.

- With pluggable timers, the tester can choose several load statistics.
- Data analysis and visualization plugins give great extensibility and personalization.

Disadvantages of Apache JMeter are;

- Apache JMeter is only for web base application and is not for windows base application.
- Apache JMeter does not control Graphical User Interface elements for example simulate pressing a button or scrolling a page.
- Apache JMeter cannot be used for desktop applications for examples MS office.
- Apache JMeter is not a browser and cannot execute javascript.

### 3.1.2.. HP LoadRunner

HP LoadRunner is a software testing tool, developed by Hewlett-Packard and it is used to test applications measuring system behavior and performance under load.

Hp LoadRunner testing tools are used to test the internet applications, web 2.0 technologies, ERP and CRM application and legacy applications. HP LoadRunner gives you the idea of an end to end system performance before going live so that it can be verified that the new or upgraded applications meet the performance requirements. HP LoadRunner predicts the behavior and properties of the system. Load Runner is used for real-time performance and fastens the release cycle of the application system [48].

Components of LoadRunner are [49];

- The **Virtual User Generator** which is also known as VuGen record end-user business models and develop an automated performance testing script which is known as Vuser Script.
- **Controller:** The Controller is used to organizes, drives, manage and monitors the load test.
- **Analysis:** Analysis helps the tester to view, dissects and compare the results of the load tests.

- **Load Generators:** the computer that runs the Vusers to execute a load on the system

Features of HP LoadRunner are;

- LoadRunner Support Rich Internet Applications, Web 2.0(HTTP/HTML, Ajax, Flex and Silverlight etc.), Mobile, SAP, Oracle, MS SQL Server, Citrix, RTE, Mail and above all, Windows Socket.
- LoadRunner support all browsers.
- LoadRunner support VB, VBscript, java, javaScript, c# as programming languages.
- The HP LoadRunner Controllers let the tester to easily and effectively control all the Virtual Users from a single point of control.
- HP LoadRunner help the tester to improve the performance by monitoring the network and server resources.

### 3.2. Evaluation Parameters

For comparison of the selected automated testing tools, different parameters were selected. Parameters are important because it helps in comparison of different tools. These selected parameters are compared with each in order to answer the research questions. The selected parameters are divided into three categories i.e. Usability Test Parameters, Technical Requirement Parameters and Performance Testing Parameters.

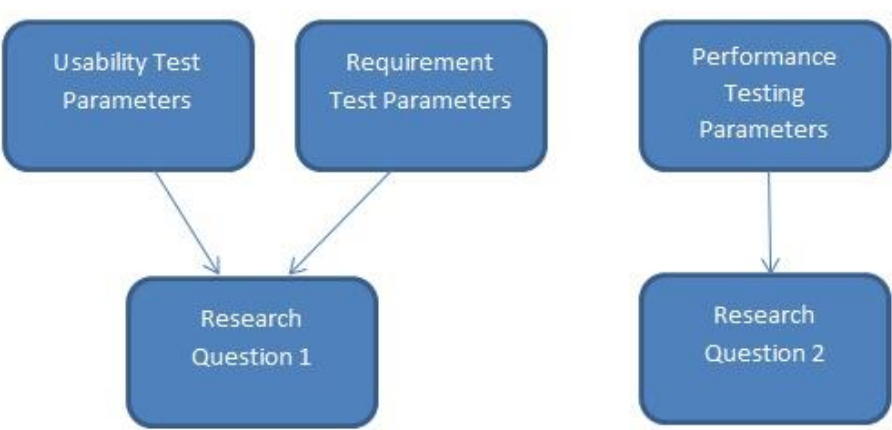
The rating for the Usability Test Parameters for comparison of both testing tools, a scenario-based survey will be conducted. Total numbers of participants in this survey will be 20. The Post-Study System Usability Questionnaire (PSSUQ) [50] method is used for the selecting the questions.

Technical Requirement Parameters are satisfied by getting the answer from literature and information available on the websites of the respective testing tools.

In order to get the result for the Performance Testing Parameters, an experiment will be conducted. Two websites are selected and will be tested with both tools. The details of the website are available in table 4. Data will be collected and further analyses will be done in chapter 4.

Comparison of Usability Test Parameters and Technical Requirement Parameters will answer the Research Question 1 while Performance Testing Parameters will answer the Research Question 2.

Figure 1: Research Question Flow Chart



### 3.3. Parameters Overview

Usability Test Parameters will show how easy the tool can be installed, configured and used. This category is important because it will help in understanding and learning of the tool. The Usability Test Parameters along with their definition are listed below in the table.

Table 1: *Usability Testing Parameters Definition*

| Parameter                      | Definition   |
|--------------------------------|--|
| Easiness in Installation       | This parameter will tell if it is easy or difficult to install the software.   |
| Friendly Interface             | This parameter will show the difficulty level of using the tool and its understandability.                           |
| Benefits of error message      | This parameter will show how easy or difficult it is to understand error message occurred while using the tool.      |
| Tutorial Availability          | This parameter will show if there are tutorial available and how easy or difficult it is to understand the tutorial. |
| Technical Support Availability | This parameter will show how easy or difficult it is to get help if a problem occurs while using the software.       |
| Terminology                    | This parameter will show if the terminology used in the application are easy to understand or not.                   |

The second category is Technical Requirement Parameters. This will give a quick overview of the tools and can help the user to decide which tool to use on basis of technical requirement.

Table 2: *Technical Requirement Parameters Definition*

| Parameter               | Definition   |
|-------------------------|--|
| Browser Support         | This parameter will show that which web browser can support the tool.                                      |
| Coding Language Support | This parameter will determine which coding language can be supported by the tool.                          |
| Pricing                 | This parameter will determine the cost of the tool.  |
| Record Playback         | This parameter will show the ability of the tool to record the scripts and run under different situations. |

The third category is Performance Testing Parameters. The definition of each parameter is listed below.

Table 3: *Performance Testing Parameters*

| Parameter               | Definition   |
|-------------------------|--|
| Average Response Time   | Average Response time is the average amount of time taken to response a request for service. |
| Average Bytes Processed | Number of total average Bytes processed during a request for service                         |

### 3.4. Suitability of Selected Parameters

The parameters selected for this research are based on functionality, architecture of the tools. The selected parameters are the most common parameters available in literature [2], [3], [45], [8] and these parameters cover all the aspects of the tools usability.

There were other parameters e.g. Cross platforms, Programming skills. Cross platforms parameter run the tool on different operating systems. This parameter was rejected because the researcher was not having resources and time to install different operating systems and conduct survey on participants. Programming skills parameter was rejected because there was a threat that the researcher might not find those participants who are experience in those programming languages which are supported by the selected tools. Other parameters such as Integration with QA and Test Result Reports were not selected on the basis that their usability test was not possible.

### 3.5. Targeted Websites

In the experiment, two websites are selected for comparison. The table below give the details of the selected websites,

*Table 4: Sample Web Services*

| Web Service Name | Description                          | Publisher            |
|------------------|--------------------------------------|----------------------|
| Loan Calculator  | Calculation of monthly loan payment. | Javascript kit       |
| BMI Calculator   | Calculation of Body Mass Index       | Smart BMI Calculator |

Reason for selecting these websites are;

- Load Calculator and BMI Calculator are proper web base applications.
- There is not enough traffic on these web application and there is not enough load on the web application which can affect the result of test.
- Smart web application e.g Yahoo weather, has tight security protocol and there is always possibility that the server block the testing if user load is applied on such websites.

- Most of web applications use firewalls to segment resources with different security requirements to prevent any load on their databases. There are no such firewalls on the selected web applications.

The hardware specification of the machine used for the experiment is below;

*Table 5: Hardware Specification*

|                  |   |
|------------------|---|
| Manufacture      | Dell                                    |
| Model            | Vostro 3360                             |
| Operating System | Windows 10                              |
| Processor        | Intel Core i5 (3rd Gen) 3317U / 1.7 GHz |
| RAM              | 4 GB                                    |
| Edition:         | Windows 10 Pro                          |
| Version:         | 1511                                    |
| OS Build         | 10586.218                               |

## **4. EMPIRICAL WORK**

In this chapter, the author discussed how the usability survey and the experiment will be conducted and on based on the data obtained from survey and experiment, the results will be concluded. Pilot Usability Test is conducted first which is used to refine the usability scenario and questionnaires. Scenarios are designed to test the usability of software testing tools i.e. LoadRunner and JMeter. The Post-Study System Usability Questionnaire (PSSUQ) [13] method is used to design the questionnaires. Completing Post-Study System Usability Questionnaire (PSSUQ) let the participants give an overall evaluation of the system they used [13].

### **4.1. Identifying Usability requirements of Software Testing Tools**

According to Andreas Holzinger [51] , usability is defined as the comfort of use and suitability of a system for a particular class of users carrying out precise work in a precise environment. Author specified criteria level for ease of use, ease of learning, simplicity, effectiveness, information, and the user interface as usability requirements for the evolution of selected software testing tools.

#### **4.1.1. Selection of Participants**

For conducting the usability studies, the author selected a total of 20 participants. These participants are students of Blekinge Institute of Technology who are currently studying in the field of Computer Science and Software Engineering. Author divide the participants into two groups. Group 1 will conduct the usability test using LoadRunner while group 2 will conduct the usability test using JMeter.

The participants were selected randomly. Initially 28 Students were contacted for the survey but only 23 students agreed. Later on, 2 students informed the author that they are busy and would not be able to attend the survey and 1 participant didn't show up for the survey.

In these 20 participants, 4 participants have work experience in software testing and are familiar with automated testing tools. 6 students have already the experience of conducting the usability survey and they are equally divided into the groups. Remaining 10 participants have no experience in usability survey and are divided equally into two groups.

#### **4.1.2. Scenarios based Tasks**

Tasks selected for the usability test are a scenario based and is written in the English language. These tasks cover all the functionalities of the selected parameters of the selected tools. It is not easy to test all the functionalities of the selected tools so author designed the scenarios in such way that the objective of this research is achieved.

#### **4.1.3. Test Location**

The usability test is conducted in Blekinge Institute of technology group rooms. All the tasks were provided on paper and participants were left alone so that they can conduct the test without any interruption.

#### **4.1.4. Usability Test Equipment and Material**

To have a control experiment, author provides computer to every participant. The Same computer was provided to each participant. After the completion of the test, the author made sure that the software testing tool is completely uninstalled. All the tasks and questionnaires were written on paper and were provided to each participant individually.

### **4.2. Pilot Test and Survey Questionnaire**

The pilot survey is the feasibility studies which is done for the preparation of major study in small-scale version or on a trial basis [52]. The objective of conducting Pilot survey are to create and test the suitability of research instruments, assessing the feasibility of study/survey, collecting initial data, developing a research question and

research [52]. The author conducted the pilot survey with a PhD Student in the field of Computer Science with main focus on usability testing.

### 4.3. Survey Information

Before the start of the survey, the author explained the purpose of the usability, the purpose of the test and usability evaluation of the software testing tools i.e. HP LoadRunner and Apache JMeter to the participants. After the completion of scenarios, participants completed the survey by circling the rating which was provided to them on paper.

### 4.4. Scenario Design

Author designed different scenarios for both testing tools. There are total of 7 scenarios for each tool. These scenarios were designed by keeping the selected parameters in mind so that participants can answer the questionnaires and on the basis of data collected, the author can answer the research question. The author first learns both tools and created the scenarios keeping all the challenges he faced while using the tools for the first time. After the scenarios were created, the author performs the scenarios on both tools to check if anything is missing. Simple English was used rather than using complex words. These scenarios were reviewed by a Phd student and based on his feedback, the scenarios were improve.

The scenarios are presented as below.

*Table 6: Scenario Based Questionnaire*

|                             |  |
|-----------------------------|--|
| <b>Survey Questionnaire</b> |  |
| <b>Name of Participant:</b> | <u>XYZ</u>   |
| <b>Name of Tool:</b>        | <u>Apache JMeter</u>   |
| I.                          | Download and install the tool on the computer.   |
| II.                         | Check all the menus in installed tool. Try to understand the interface of the tool.                        |
| III.                        | Open new test plan by clicking “New”. Click on “Run” and select “Remote Start All”. (This will give error) |
| IV.                         | Now understand the Error by taking the help (online or if any help provided                                |

|  |
|--|
| <p>in the help menu)</p> <p>V. If you cannot solve the error, try to take the help from technical support(if available).</p> <p><b>Test a website by following the instructions.</b></p> <ol style="list-style-type: none"> <li>1. Click New and Save the Test Plan.</li> <li>2. Click Test Plan, Right click “Add” select “Threads” and select “Thread Group”.</li> <li>3. Click “Workbench”, Click “Add”, Select “Non-Test Elements”, Select “HTTP Test Script Recorder”</li> <li>4. Change Port to 7070. Select “Test Plan &gt; Thread Group” in Target Controller.</li> <li>5. Click on Start and click OK.</li> <li>6. Open Mozilla FireFox browser and type <a href="http://www.google.com">www.google.com</a> and press enter.</li> <li>7. Open Jmeter and click Stop.</li> <li>8. Double click Thread Group on the left side of the window.</li> <li>9. Right click “Thread Group”, select “Add”, select “Listener” and “Select Summery Report”.</li> <li>10. Click “Run” from Menu and select “Start”.</li> <li>11. Save the Result.</li> </ol> |
|--|

Table 7: Scenario Based Questionnaire

|   |
|---|
| <p><b>Survey Questionnaire</b></p> <p><b>Name of Participant:</b> <u>XYZ</u></p> <p><b>Name of Tool:</b> <u>HP LoadRunner</u></p> <ol style="list-style-type: none"> <li>I. Download and install the tool on the computer.</li> <li>II. Open Virtual User Generator. Check all the menus in it. Try to understand the interface of the tool.</li> <li>III. Open new test plan by clicking “New”. Click on “Run” and select “Remote Start All”. (This will give error)</li> <li>IV. Now understand the Error by taking the help (online or if any help provided in the help menu)</li> <li>V. If you cannot solve the error, try to take the help from technical support (if available).</li> </ol> <p><b>Test a website by following the instructions.</b></p> <ol style="list-style-type: none"> <li>1. Click the file and select “New Script and Solution”.</li> <li>2. Select web HTTP/HTML, Give Script Name, Give Solution Name and click Create.</li> <li>3. Click on “Record” from the menu and select Record.</li> <li>4. Select Mozilla FireFox as Browser and write <a href="http://www.google.com">www.google.com</a> in URL. Click</li> </ol> |
|---|

|   |
|---|
| Start Recording.  |
| 5. Write down HP LoadRunner in Browser and press enter.                                   |
| 6. Stop Recording.  |
| 7. Click “Tool” from the menu and select “Create Controller Scenario”. Click OK.          |
| 8. In Controller window, click “Scenario” from the menu and click “Run”.                  |
| 9. After it is executed, click on “Result”’s from the menu and click on “Analyze Result”. |

#### 4.5. Questionnaire for Usability Evaluation

After the completion of scenarios, participants completed the questionnaire in order to collect the data. The questionnaire is designed according to The Post-Study System Usability Questionnaire (PSSUQ) [13]. The questionnaire includes total of 15 questions, set on user agreement for usability requirements on ease of use, ease of learning, simplicity, effectiveness, information, and the user interface. The questions divided into sets are as below;

*Table 8: Set of Questions*

| <b>Parameter</b>               | <b>Number of Questions</b> |
|--------------------------------|----------------------------|
| Easiness in Installation       | 5                          |
| Friendly Interface             | 3                          |
| Benefits of error message      | 2                          |
| Tutorial Availability          | 2                          |
| Technical Support Availability | 1                          |
| Terminology                    | 2                          |
| <b>Total</b>                   | <b>15</b>                  |

#### 4.6. Validation of Scenario

To validate the scenarios created which was used by the participants, the author first conducted the survey by himself and compared the result with answers gathered by

the participants. This is a criteria to validate a survey. If results are matching, this mean the scenario made are understandable and accurately interpreted by the participants. If the result is not matching, the participants are not able to interpret the scenarios accurately which means, scenarios are needed to be modify and made more generic so that all the participants can interpret the scenarios.

## 5. WEB APPLICATION PERFORMANCE TESTING

This chapter discusses the test preparation, execution and reporting of web application performance testing implementation. The main focus of this experiment is load testing. The below figure show the web application architecture.

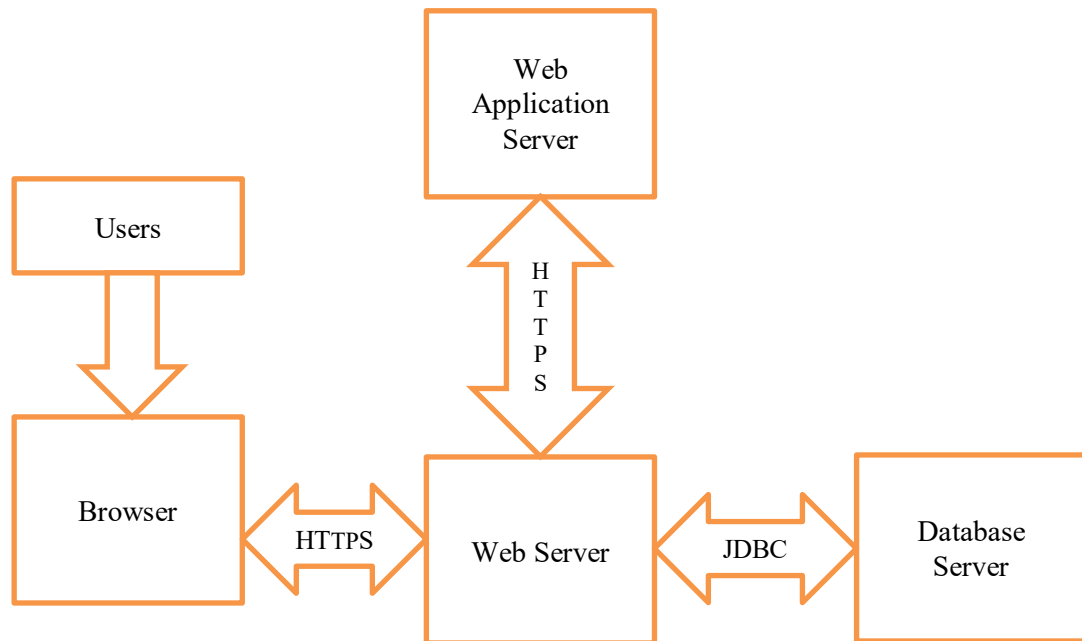


Figure 2: Web Application Architecture

### 5.1. Identifying Test Environment

The objective of system evaluation is to collection information which can be helpful for performance testing to get the specific needs of the project. This information is very much important for the collection of performance requirements and goals, generating strategies and plans for the performance testing.

Test environment can be explained as an environment where performance tests are executed with all the tools and hardware. Some important criteria considered are;

Table 9: Test Environment Criteria

|          |   |
|----------|---|
| Hardware | Computer Hardware (like RAM, Processor )                  |
| Network  | Network Architecture, Load Balancing                      |
| Software | Software installed in environment (like Operating System) |

## 5.2. Identify Performance Acceptance Criteria

For establishing performance (acceptance) criteria for an application, requirement elicitation must include performance consideration such as projected user base and number of simultaneous user, typical usage scenarios, maximum response times and maximum server resource utilization [53]. The parameter criteria selected for this research are;

**Response Time:** Response time is the average amount of time taken to response a request for service.

**Bytes Process:** Number of total average Bytes processed during a request for service.

## 5.3. Implementation of Experiment

The experiment is conducted at the Blekinge Institute of Technology computer lab.

The researcher uses his own personal laptop with all the tools installed.

The reason for selecting the Blekinge Institute of Technology computer lab because of its stable internet speed.

## 5.4. Data Collection

The experiment is conducted at 3 different times of a day, keeping 8 hours difference between each experiment. First Tool 1 i.e. Apache JMeter is used for testing the web application and then Tool 2 i.e. HP LoadRunner. All the data is collected and saved for analysis which is done in next chapter.

## 5.5. Front End Analyses

This section represents a simple frontend analysis of the selected tools. This section will help in understanding the layout of the tools. Front end of each is represented as below;

### 5.5.1. Apache JMeter

The startup windows of Apache JMeter is present in figure 3 below. Here, the tester creates the scenario and record the test script.

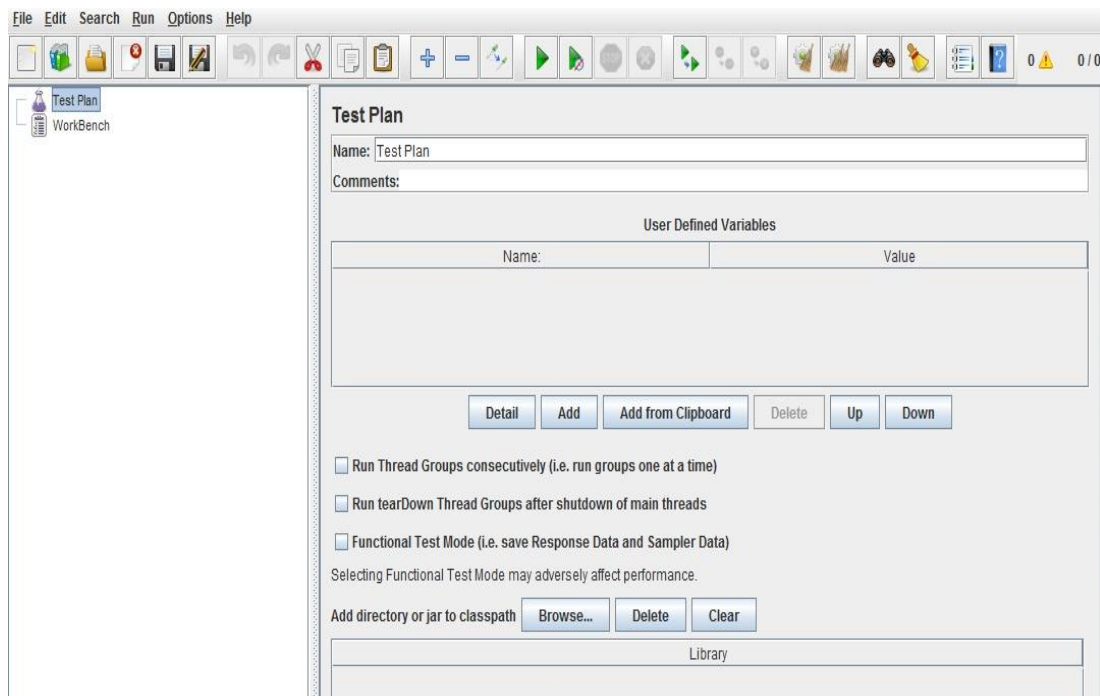


Figure 3: Apache JMeter Front

In Test Plan, the thread group is selected. In Tread Group, a number of the user, how many times the user will hit the site and time is defined.

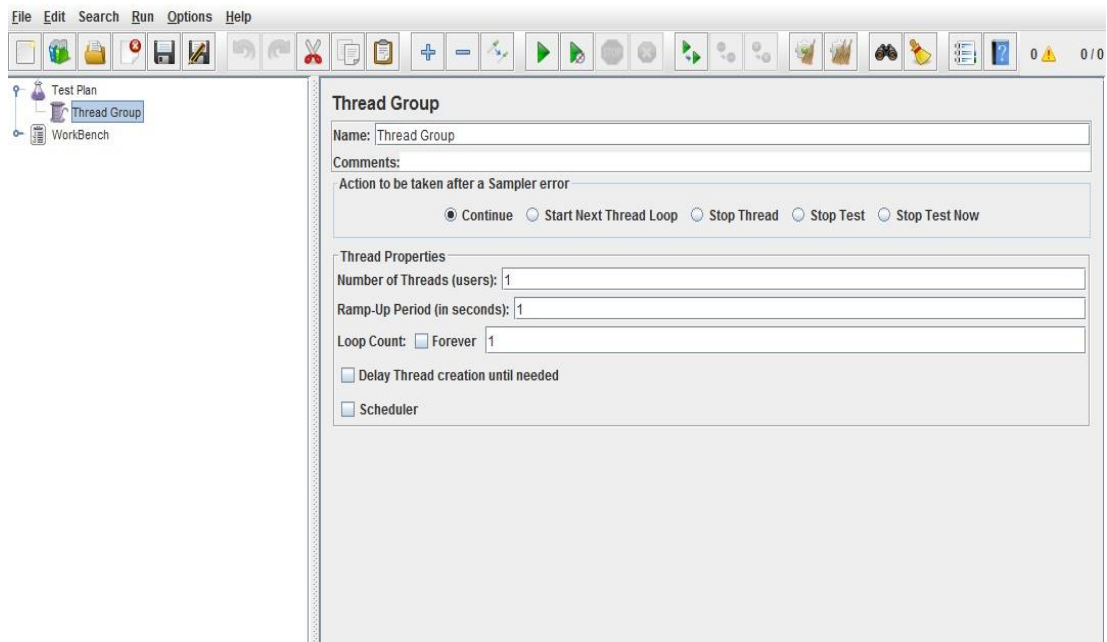


Figure 4: Apache JMeter Thread Group

WorkBench in Apache Jmeter is like an environment used for rough work. Before saving the test plan, the test script is recorded in WorkBench and after recording, it transfers all the work to Test Plan section from where the script is saved. In WorkBench, the script is recorded and the output is saved in Thread Group.

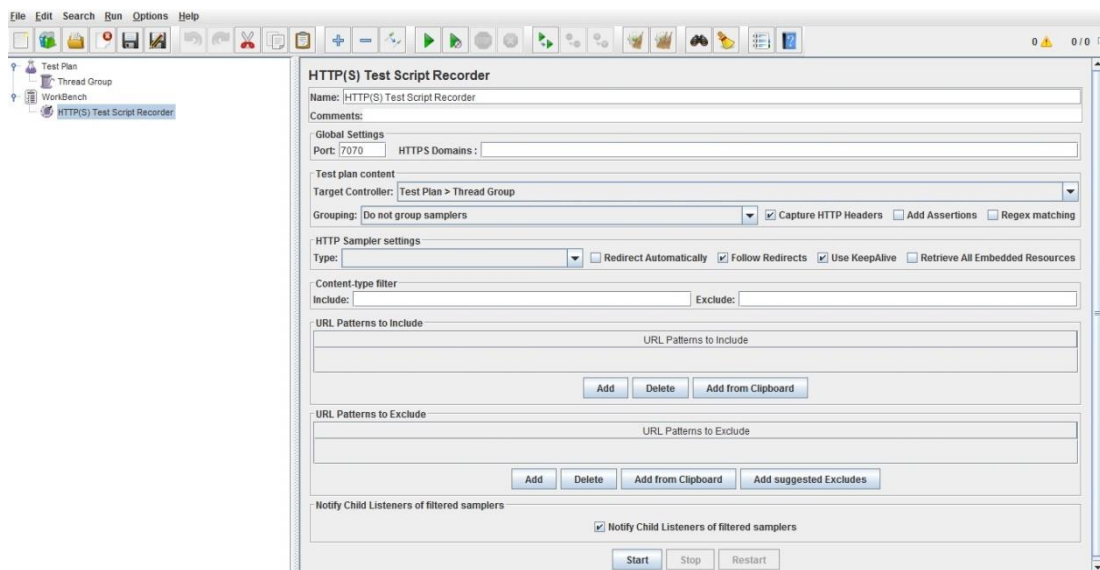
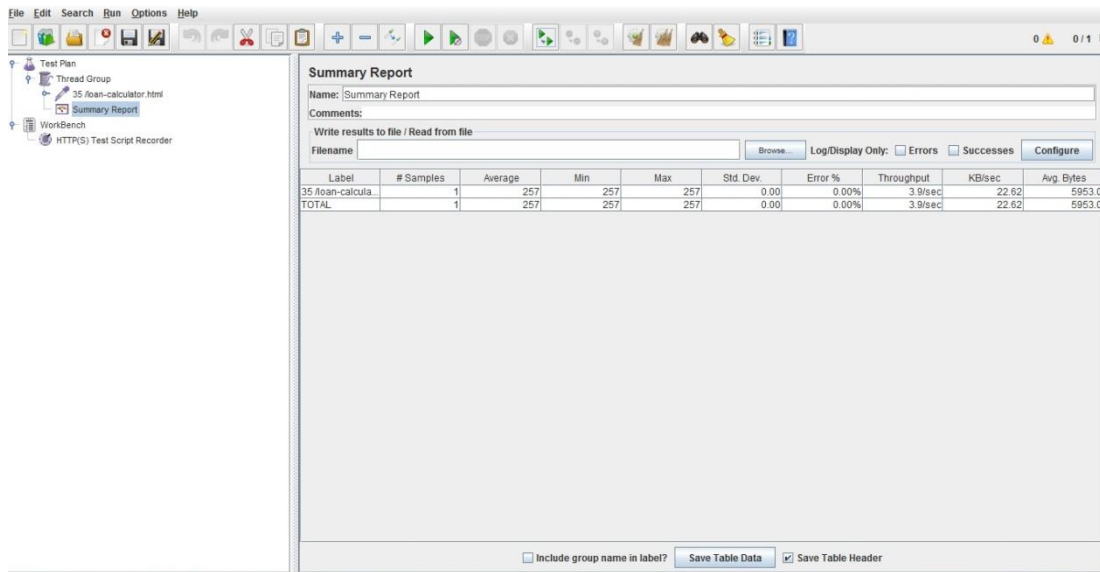


Figure 5: Apache JMeter Test Script Recorder

The summary of the recorded script is displayed in Test Plan section as below.



The screenshot shows the Apache JMeter Summary Report window. The left sidebar lists the Test Plan, Thread Group, 35 loan-calculator.html, Summary Report, WordBench, and HTTP(S) Test Script Recorder. The main window displays the Summary Report for the selected test.

**Summary Report**

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only: ☐ Errors ☐ Successes

| Label           | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | KB/sec | Avg. Bytes |
|-----------------|-----------|---------|-----|-----|-----------|---------|------------|--------|------------|
| 35 loan-calcula | 1         | 257     | 257 | 257 | 0.00      | 0.00%   | 3.9/sec    | 22.62  | 5953.0     |
| TOTAL           | 1         | 257     | 257 | 257 | 0.00      | 0.00%   | 3.9/sec    | 22.62  | 5953.0     |

☐ Include group name in label?  ☒ Save Table Header

Figure 6: Apache JMeter Summer Report

### 5.5.2. LoadRunner

There are three main components of LoadRunner that are used in this research i.e. Virtual User Generator, Controller and Analyser. The Virtual User Generator is used to record the script.

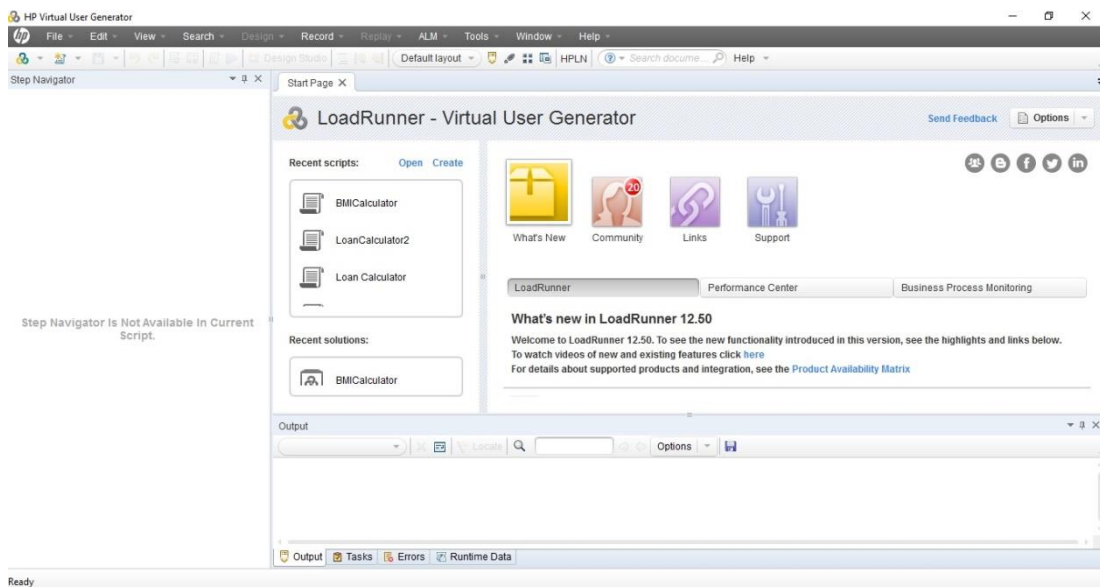


Figure 7: HP LoadRunner – Virtual User Generator

For managing and maintaining the scenarios and controlling all the VUser in the single work environment, Controller is used. It is an important component of

LoadRunner because LoadRunner uses Controller to emulate the real-time users. Number of VUsers, run-time setting, Load test duration, start and end of execution of scripts are configured in Controller.

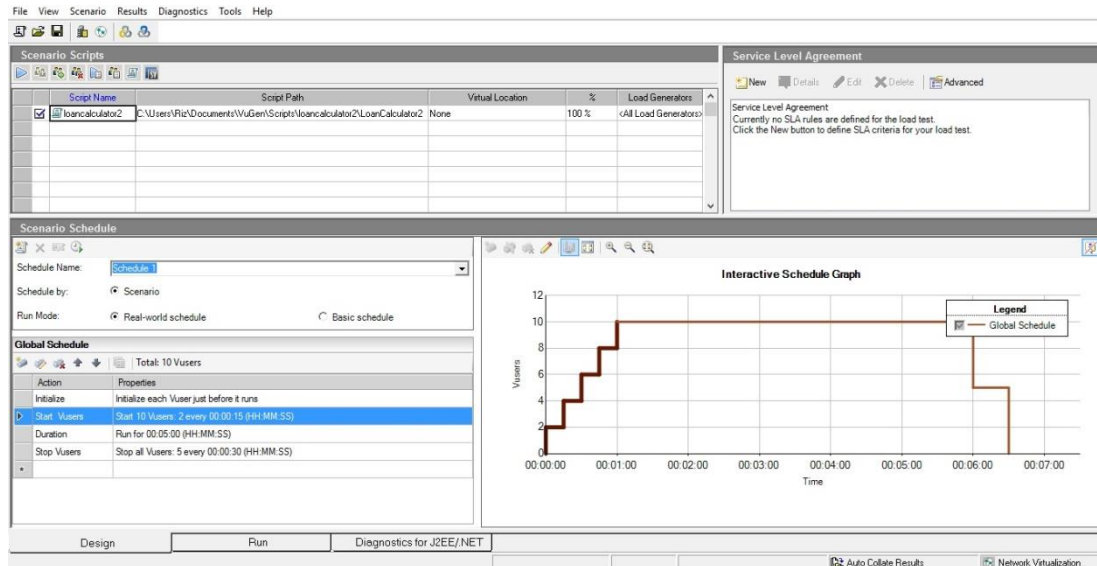


Figure 8: HP LoadRunner – Controller

The detail analyses of the performance test are done in Analyzer Component of LoadRunner. Controller creates the dump during the execution of load test. This dump contains all the information in a raw format and this dump is parsed by Analyzer to generate results and graphs.

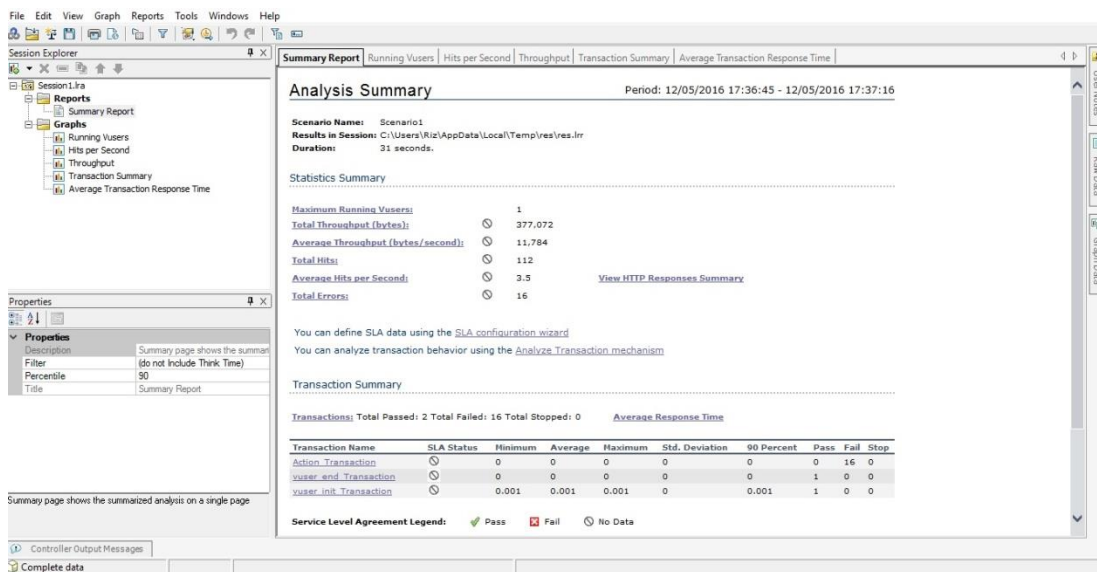


Figure 9: HP LoadRunner Analysis Summary

## **5.6. Independent Variables and Dependent Variables**

The Independent variables and dependent variables [54] selected in this experiment are,

### ***Independent Variables***

Software Testing Tools (Apache JMeter and HP LoadRunner)

Network Infrastructure

Personal Experience

### ***Dependent Variables***

Average Response Time

Average Bytes Process

## 6. RESULTS

After the completion of each test, the parameters were outlined and results found for each parameter.

### 6.1. Result of two testing tools

The results are presented and outlined in this section. Both tools have been tested, evaluated and compared with each other. The three evaluation parameter, Technical Requirement Parameters, Usability Test Parameters and Performance Testing Parameters has results that are detailed below.

#### 6.1.1. Result for Technical Requirement Parameters

Technical Requirement Parameters determining if the tools are feasible with the hardware system that is going to run the tools. For getting the result for Technical Requirement Parameters, a literature review is conducted and information available on tools website.

*Table 10: Result for Technical Requirement Parameters*

| Parameter               | JMeter                          | LoadRunner                            |
|-------------------------|---------------------------------|---------------------------------------|
| Browser Support         | Chrome, Firefox, Opera, Ms Edge | Chrome, Firefox, Opera, Ms Edge       |
| Coding Language Support | Java                            | C, JavaScript, VB, VBscript, java, c# |
| Pricing                 | Open Source                     | License                               |
| Record Playback         | Yes                             | Yes                                   |

#### 6.1.2. Usability Test Parameters

The next Evaluation Parameter is Usability Test Parameter of the tools. This was very important evaluation parameter to determine which tool would be best to use. If time is the factor, the ease of installation is the important factor. The table 11 shows the rating system from 1 to 7 [13] on the basis of the survey conducted on a number

of participants and the result is obtained by a questionnaire completed by the participants. Table 11 shows the results observed by the usability test.

As the questionnaire of usability test are divided into a set of 6 as shown in table 8. The data is in numeric form, all the numbers are added and the result is generated on the basis of average which is shown in table 11 where 1 (strongly agree) to 7 (strongly disagree).

*Table 11: Result for Usability Test Parameter*

| <b>Parameters</b>              | <b>JMeter</b> | <b>LoadRunner</b> |
|--------------------------------|---------------|-------------------|
| Easiness in Installation       | 2             | 5                 |
| Friendly Interface             | 3             | 5                 |
| Benefits of error message      | 3             | 3                 |
| Tutorial Availability          | 2             | 2                 |
| Technical Support Availability | 6             | 1                 |
| Terminology                    | 4             | 5                 |

#### **6.1.2.1. Validity of Scenarios**

As the results of participants are matching, this indicates that the participants were able to understand and interpret the scenarios same way, considering that, the background of participants were not familiar with the selected tools hence scenarios are validate.

#### **6.1.3. Performance Testing Parameters**

The result obtained for Performance Testing Parameters are from the experiment conducted on the two selected websites. This testing is very much important to get the comparison between the two tools based on their performance.

*Table 12: Average Response Time for WS1*

| Tool       | Web Service ID | Average Response Time (ms) |         |          |
|------------|----------------|----------------------------|---------|----------|
|            |                | 8:00 AM                    | 4:00 PM | 12:00 AM |
| Jmeter     | WS1            | 257                        | 256     | 333      |
| LoadRunner | Ws1            | 2682                       | 2549    | 2601     |

*Table 13: Average Response Time for WS2*

| Tool       | Web Service ID | Average Response Time (ms) |         |          |
|------------|----------------|----------------------------|---------|----------|
|            |                | 8:00 AM                    | 4:00 PM | 12:00 AM |
| Jmeter     | WS2            | 559                        | 550     | 554      |
| LoadRunner | WS2            | 2241                       | 2304    | 2267     |

*Table 14: Average Bytes Processed WS1*

| Tool       | Web Service ID | Average Bytes Processed |         |          |
|------------|----------------|-------------------------|---------|----------|
|            |                | 8:00 AM                 | 4:00 PM | 12:00 AM |
| Jmeter     | WS1            | 5953                    | 6881    | 8754     |
| LoadRunner | Ws1            | 16616                   | 15521   | 18442    |

*Table 15: Average Bytes Processed WS2*

| Tool       | Web Service ID | Bytes Processed |         |          |
|------------|----------------|-----------------|---------|----------|
|            |                | 8:00 AM         | 4:00 PM | 12:00 AM |
| Jmeter     | WS2            | 6540            | 9873    | 6803     |
| LoadRunner | WS2            | 15301           | 16460   | 15903    |

## **7. VALIDITY THREATS**

In this chapter, different validity threats related to the case study and experiment are discussed. The author used Creswell [55] to explain different validity threats in the research.

### **7.1. Internal Validity**

Internal validity focus on how sure we can be that the treatment actually caused the outcome. There can be other reasons that have caused the result on which we do not have control over or have not measure [56].

The internal validity threats in this research are;

- There was a threat that the questions asked in the survey can really explain the outcome the author wants to research. To overcome this threat, a pilot survey was conducted before the execution of survey. The researcher used both tools and tests the scenarios to validate the result of participants.
- The author had not used the selected tools before. To overcome this threat, the author learned how to conduct the performance testing by taking help from the online tutorials. After the author learned the tools, the experiment was conducted.

### **7.2. External Validity**

External validity is related with whether the results can be generalized outside the scope of the study [56].

- There was threat about the selection of participants and their knowledge of the selected tools which can affect the result of this research. To overcome this threat, researcher divided the participants into two groups for each tool according to their knowledge and background of testing experience.
- There was the threat of network infrastructure with unstable internet speed. This experiment was done in a computer lab of Blekinge Institute of Technology where the internet speed was stable.

### **7.3. Construct Validity**

Construct validity motivation is on the relation between the theory behind the experiment and the interpretations. The interpreted result might not correspond to the effect what is being measured [56].

- There was a threat that the participants might misunderstand the scenarios. To minimize this threat, the author was there to answer all the problems they faced. The data gathered was double checked by the author in order to minimize this risk.
- There was a threat that the selected tools can answer the selected parameters for the experiment. To overcome this threat, different literature and the official websites of the selected tools were studied and confirmed that the tools can satisfy the selected parameters.

### **7.4. Conclusion validity**

Conclusion validity concentrate on how sure the treatment used in an experiment really is related to the actual result obtained [56].

- Conclusion validity is a threat that can lead the research to an incorrect conclusion. To overcome this research, those participants were selected who had the background of Software Engineering or Computer Science and had studied the course Verification and Validation. To validate the result, each participants survey result was compared with the result of researcher's when he conducted the survey and with the result achieved from the pilot survey.
- The author used two web application for testing. There was a threat that if what any web application is down from hosting side. To overcome this threat, the experiment was conducted in 3 different times which helped in getting accurate data. Each result was compared with each other to validate the result.

## **8. ANALYSIS AND DISCUSSION**

In this chapter, the analysis and discussion of the research is done. All the observation made during the survey and experiment which lead to the answers to the research questions are discussed with the help of graphs. Mean and median are a suitable representation of „average“ value which is set for a continuous variable which can take any value [57]. It is also suitable to category data, for example, an even or response is rated on a scale from high to low or easy to difficult [57].

### **8.1. Technical Requirement Parameters.**

Table 10 summarize the Technical Requirement Parameters of Apache JMeter and HP LoadRunner testing tools.

### **8.2. Usability Test Parameters**

The graph below, generated by the data from the table 11 represent the Usability Testing Parameters. As the rating is from 1 – 7 where 1 is Strongly agreed to 7 which strongly disagrees [13]. The questionnaire can be found in Appendix I.

#### **8.2.1. Easiness in Installation**

From the graph, it is found that Apache JMeter is easy in installation and it takes very less time as compared to HP LoadRunner where installation took a lot of time.

*Winner: Apache JMeter*

#### **8.2.2. Friendly Interface**

From participants, it is observed that the interface of Apache JMeter is easier to work on as compared to HP LoadRunner.

*Winner: Apache JMeter*

#### **8.2.3. Benefits of Error Message**

It is found that both tools have well explained the error messages as participants rated same for this parameter.

*Winner: Draw*

#### 8.2.4. Tutorial Availability

A lot of tutorials are available online to learn and use the both software testing tools.

*Winner: Draw*

#### 8.2.5. Technical Support Availability

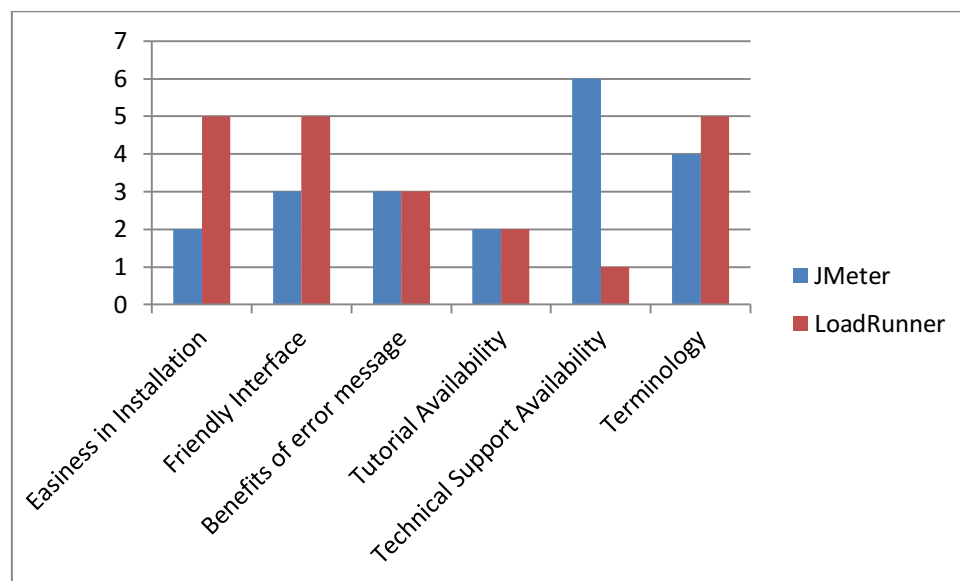
A significance difference is seen Technical Support Availability as participants found hard to get the support from Apache JMeter as compare to HP LoadRunner.

*Winner: HP LoadRunner*

#### 8.2.6. Terminology

From the graph, it is found that Apache JMeter is easier to understand the terminology used as compared to HP LoadRunner.

*Winner: Apache JMeter*



*Figure 10: Usability Test Parameters Result*

### 8.3. Performance Test Parameters

In this experiment, two different web servers are used. The author presented the data in different tables in order to show it more easily to understand.

#### 8.3.1. Average Response Time

From table 12 & 13, it is analyzed that the average response time for Apache JMeter is better than HP LoadRunner. The observed data is also presented in graph which is shown in figure 11 for web server 1 and Figure 12 for web server 2

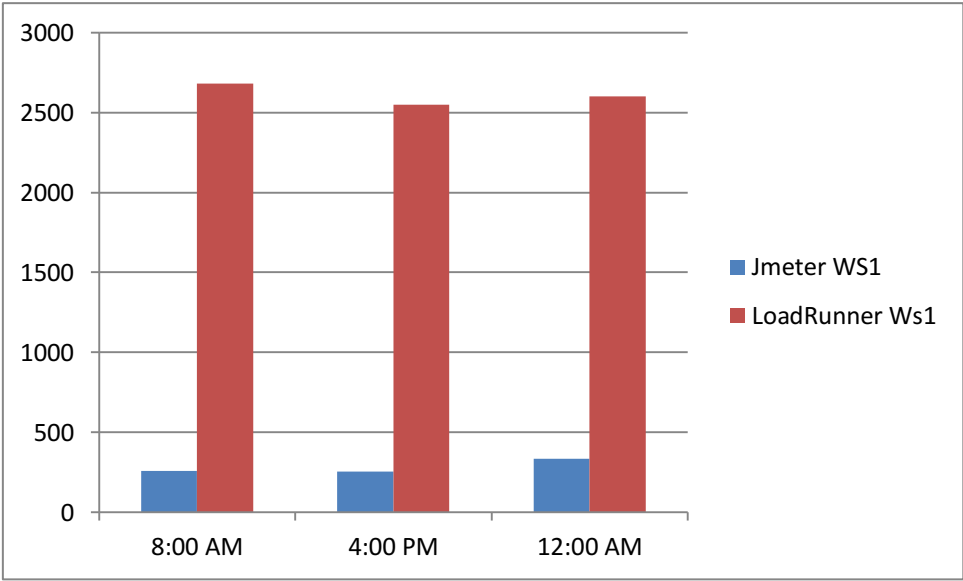


Figure 11: Average Response Time Result for WS1

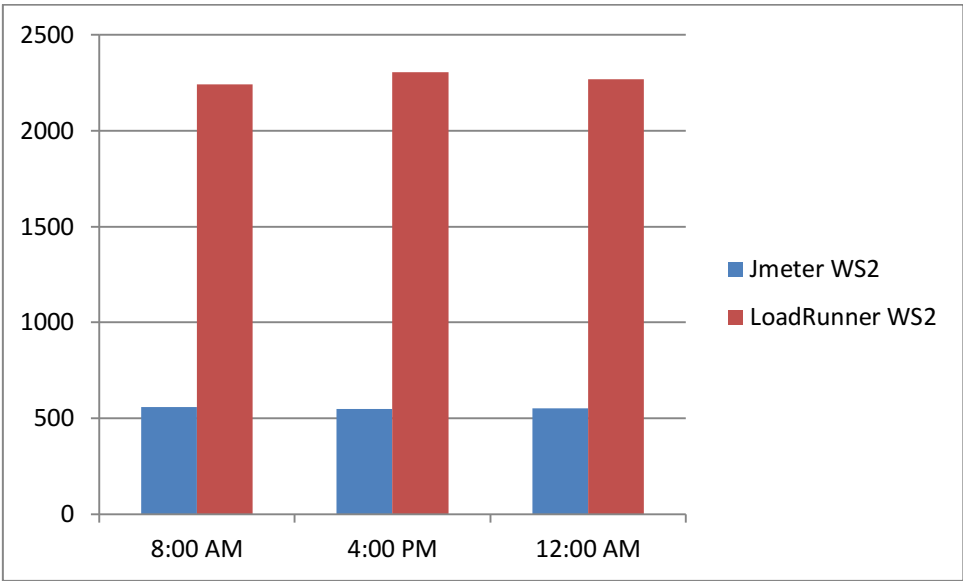


Figure 12: Average Response Time Result for WS2

8.3.2. Average Byte Processed

Average Byte Process is the second parameter for the comparing tools. From table 14 and 15, it is observed that HP LoadRunner takes number of bytes to process the test as compared to Apache JMeter. This shows that HP LoadRunner checks more options or attributes during request and response which is good as compare to Apache JMeter which took very less number of bytes to process the test. It can be observed from the graphs which are generated from data of table [14] and [15].

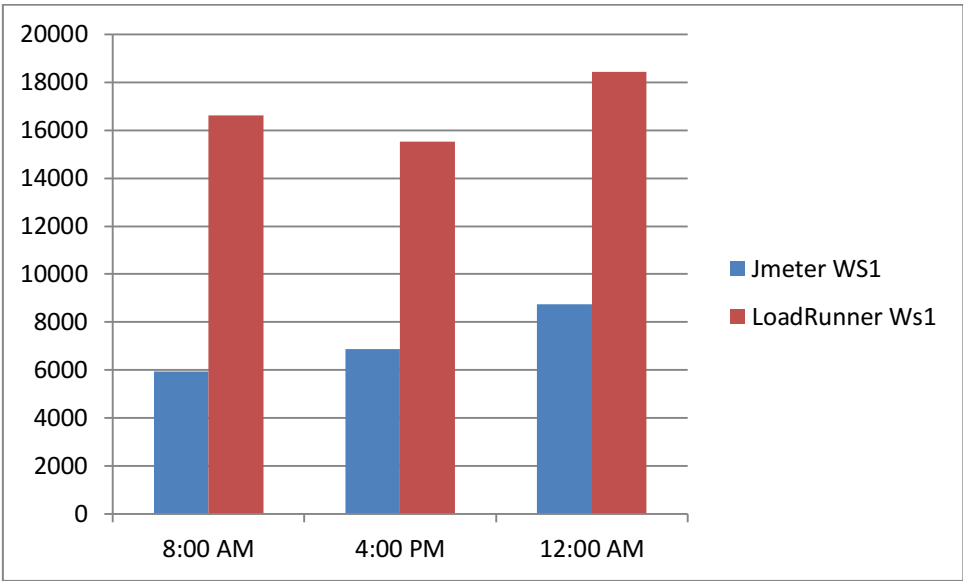


Figure 13: Average Bytes Processed for WS1

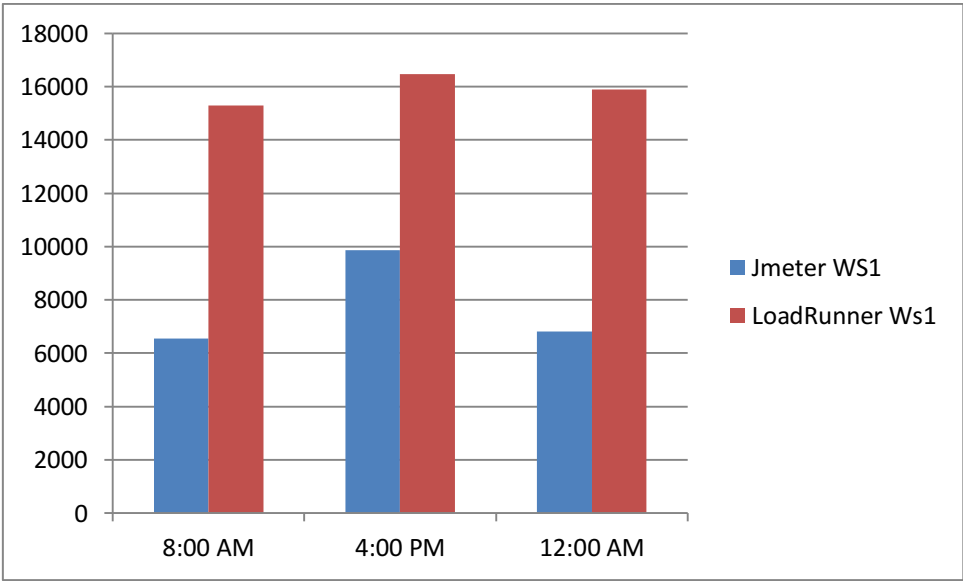


Figure 14: Average Bytes Processed for WS2

## **9. CONCLUSION AND FUTURE WORK**

In the conclusion of this research, I have learned that software testing tools are very much different to use as compare to other software. It is not easy to select a software testing tools because it a lot of time and effort to understand the software testing goal and software testing needs. There are a lot of software testing tools available and such variety makes it more difficult in the selection of software testing tool.

The two software testing tools compared in this research on the basis of different parameters are Apache JMeter and HP LoadRunner. Apache JMeter is open source software and is 100% java application. HP LoadRunner is not open source and is the industry standard for application performance testing. Both tools are very good for performance testing but Apache JMeter has edge over HP LoadRunner. The user interface of Apache JMeter is very simple and it is very easy to understand and use the tool while on another hand, HP LoadRunner interface is complex to use. Apache JMeter installation is very simple as it is only to unzip a tar file without any installation. HP LoadRunner takes a lot of time in installation and occupies lot of disk space.

In my view, I will recommend that tool which is easy to install and easy to learn how to use it. Hence, in comparison of Apache JMeter and HP LoadRunner, I will recommend Apache JMeter.

For future work, this research work can be extended to more software testing tools with different web services and different parameters which can be responsible for more realistic results.

## REFERENCES

- [1] G. Singh, S. Singh, S. Singh, "Software Testing," *International Journal of Advance Research in Computer Science*, vol. 1, no. 3, Sept-Oct 2010.
- [2] S. Dhankar, A. Jain, M. Jain, "A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing," *International Journal of Engineering, Management & Sciences*, vol. 1, no. 1, January 2014.
- [3] Meenu, Y. Kumari, "Comparative Study of Automated Testing Tools: Selenium, SoapUI, HP Unified Functional Testing and Test Complete," *JETIR*, vol. 2, no. 9, Sept 2015.
- [4] J. Parikh, R. Shah, B. Patel, "A Review Paper on Comparison of SQL Performance Analyzer Tools: Apache JMeter and HP LoadRunner," *International Journal of Current Engineering and Technology*, vol. 4, no. 5, Oct 2014.
- [5] P. J. Clarke, D. C. Crowther, "Examining software testing tools," *Dr. Dobbs' s Journal: Software Tools for the Professional Programmer*, vol. 30, no. 6, June 2005.
- [6] J.W. Moore, P. Bourque, R. Dupuis, L.L. Tripp, A. Abran, "Guide to the Software Engineering Body of Knowledge: 2004 Edition-SWEBOK," *IEEE Computer Society*, 2005.
- [7] E. H. Bersoff, E. R. Comer A. M. Davis, "A strategy for comparing alternative software development life cycle models," *IEEE Transactions on Software*, vol. 14, no. 10, Oct 1988.
- [8] M. Kaur, R. Kumari, "Comparative Study of Automated Testing Tools: TestComplete and QTP," *International Journal of Computer Application*, vol. 24, no. 1, June 2011.
- [9] U. Shaukat, F. Feroz, S. Kayani, A. Akbak K. Shaukat, "Taxonomy of Automated Software Testing Tools," *International Journal of Computer Science and Innovation*, vol. 2015, no. 1, pp. 7-18, 2015.
- [10] J. Nielsen., "What is Usability," *Usability Engineering: An Imprint of Academic Press*, p. 26, 1993.
- [11] Wan Abdul Rahim Wan Mohd, et al. Isa, "Usability testing research framework: Case of handicraft web-based system," *Information and Communication Technology (ICoICT)*, pp. 199-204, 2014.
- [12] N. Bevan, "Measuring usability as quality of use," *Software Quality Journal* 4, no. 2, pp. 115-130, 1995.
- [13] J. R. Lewis, "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use," *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57-78, Jan 1995.
- [14] Catalin, et al. Dumitrescu, "Diperf: An automated distributed performance testing framework.," *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on. IEEE*, pp. 289-296, 2004.
- [15] R. Kumar and A.J. Singh, "A Comparative Study and Analysis of Web Service Testing Tools," *IJCSMC*, vol. 4, no. 1, pp. 433-442, January 2015.
- [16] K Rai G Saini, "An Analysis on Objectives, Importance and Types of Software Testing," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 9, pp. 18-23, Sept 2009.
- [17] "Software testing research: Achievements, challenges, dreams," *2007 Future of Software Engineering. IEEE Computer Society*, pp. 85-103, May 2007.
- [18] B. Hetzel D. Gelperin, "The growth of software testing," *Communications of the ACM* 31, vol. 31, no. 6, pp. 687-695, 1988.
- [19] T. Tsuchiya, T. Kikuno, N. Kobayashi, "Applicability of Non-Specification-Based Approaches to," *In Dependable Systems and Networks, IEEE*, pp. 337-346, 2001.
- [20] G. Dubey M. Tuteja, "A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle (SDLC) Models," *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307*, vol. 2, no. 3, July 2012.
- [21] F Khan, ME Khan, "Importance of Software Testing in Software Development Life Cycle," *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 2, p. 120, 2014.
- [22] R. Pietrantuono, S. Russo, D. Cotroneo, "Testing techniques selection based on ODC fault types and software metrics," *The Journal of Systems and Software*, vol. 86, no. 6, pp. 1613-1637, June 2013.

- [23] P. Kaur, G. Gupta, "Software Testing – Levels, Methods and Types," *IJAIR*, ISSN: 2278-7844, 2013.
- [24] S. Khurshid, D. Marinov, C. Boyapati, "Korat: Automated testing based on Java predicates," *ACM SIGSOFT Software Engineering*, vol. 27, no. 4, pp. 123-133, 2002.
- [25] W. Eric, et al. Wong, "A Study of Effective Regression Testing in Practice," *Software Reliability Engineering, 1997. Proceedings., The Eighth International Symposium on. IEEE*, 1997.
- [26] and D. Notkin, T. Xie, ""Checking inside the black box: Regression testing based on value spectra differences." , " *In Software Maintenance, 2004. Proceedings. 20th IEEE International Conference* , pp. 28-37, 2004.
- [27] A. Almalaise M. Wahid, "JUnit framework: An interactive approach for basic unit testing learning in Software Engineering," *Engineering Education (ICEED), 2011 3rd International Congress on. IEEE* , pp. 159-164, 2011.
- [28] J. de Halleux, T. Xie N. Tillmann, "Parameterized Unit Testing: Theory and Practice," *In 2010 ACM/IEEE 32nd International Conference on Software Engineering*, vol. 2, pp. 483-484, 2010.
- [29] J.C. Maidonado M.E. Delamaro, "Interface Mutation: An Approach for Integration Testing," *IEEE Transactions on Software Engineering*, vol. 27, no. 3, pp. 287-247, 2001.
- [30] E. Alégroth, R. Feldt G. Liebel, "State-of-Practice in GUI-based System and Acceptance Testing: An Industrial Multiple-Case Study," *39th Euromicro Conference on Software Engineering and Advanced Applications, IEEE*, pp. 17-24, 2013.
- [31] M. Demirer A. Avcioglu, "Implementation of System Testing Automatization on Computer Aided Systems for Hardware and Software," *IEEE AUTOTESTCON* , pp. 127-133, 2015.
- [32] J Gao, J Samuel, D Kung P Hsia, "Behavior-Based Acceptance Testing Of Software Systems: A Formal Scenario Approach," *Computer Software and Applications Conference, 1994. COMPSAC 94. Proceedings., Eighteenth Annual International*, pp. 293-298, 1994.
- [33] R.S. Chhillar, I. Hooda, "Software Test Process, Testing Types and Techniques," *International Journal of Computer Applications*, vol. 111, no. 13, Feb 2015.
- [34] U Qamar, S Hassan M Arslan, "automatic Performance analysis of cloud based load testing of web application and its comparison with traditonal load testing," *Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference. IEEE.*, pp. 140-144, 2015.
- [35] N Agheli, B Hosseini A Shojaei, "Cloud-based load testing method for web services with VMs Management," *2nd International Conference on Knowledge-Based Engineering and Innovation (KBEL), IEEE*, 2015.
- [36] HA Chan, "Accelerated Stress Testing for Both Hardware and Software," *Reliability and Maintainability, 2004 Annual Symposium-RAMS. IEEE.*, pp. 346-351.
- [37] R. Angmo, M. Sharma, "Web based Automation Testing and Tools ," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 908-912, 2014.
- [38] A. Grguri, M. Mosmondor, P. Lazarevski, J. Krizani, "Load testing and performance monitoring tools in use with AJAX based web applications," *MIPRO, 2010 Proceedings of the 33rd International Convention. IEEE*, pp. 428-434, May 2010.
- [39] (2016, July) WAPT. [Online]. <http://www.loadtestingtool.com/>
- [40] (2016, July) Loadster. [Online]. <https://www.loadsterperformance.com>
- [41] (2016, July) Load Impact. [Online]. <https://loadimpact.com>
- [42] E E. Borjesson and R. Feldt, "Automated System Testing using Visual GUI Testing Tools: A Comparative Study in Industry," *IEEE Fifth International Conference on Software Testing, Verification and Validation*, pp. 350-359, 2012.
- [43] M. Nass, H. H. Olsson, E. Alegroth, "JAutomate: a Tool for System- and Acceptance-test Automation," *IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013.
- [44] Dr. A. K. Sharma, S. Sharma, "Empirical Analysis of Web Service Testing Tools ," *International Journal of Engineering Technology, Management and Applied Sciences* , vol. 3, no. 4, April 2015.
- [45] J. Parikh, R. Shah, B. Patel, "A Review Paper on Comparison of SQL Performance Analyzer Tools:

- Apache," *International Journal of Current Engineering and Technology* , vol. 4, no. 5, Oct 2014.
- [46] Jürgen Börstler and Martin Boldt, Research Methodology 1. Thesis Lecture 2, 2015.
  - [47] Apache JMeter. (2016, January) <http://jmeter.apache.org/>. [Online]. <http://jmeter.apache.org/>
  - [48] M. Kaushik P. Fageria, "Research of Load Testing and Result Based on Loadrunner ," *International Journal of Civil Engineering (SSRG-IJCE)*, vol. 1, April 2014.
  - [49] M. A.Hebaishy, A. S. Alotaibi M. Imran, "A Comparative Study of QTP and LoadRunner Automated Testing Tools and their Contributions to Software Project Scenario," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, no. 1, January 2016.
  - [50] J. R. Lewis, "IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use," *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57-78, Jan 1995.
  - [51] A. Holzinger, "Usability Engineering Methods for Software Developers," *Communications of the ACM*, vol. 48, no. 1, pp. 71-74, 2005.
  - [52] M. L. Mursidi, N. A. A. Husin, R. Raihana, "Reliability analysis for pilot survey in integrated survey management system," *In Software Engineering (MySEC), 2011 5th Malaysian Conference, IEEE* , pp. 220-222, 2011.
  - [53] Jukka Palomäki, Web Application Performance Testing , 2009, Master's Thesis.
  - [54] C. Wohlin, "Experimentation in software engineering: an introduction," *Springer*, 2000.
  - [55] J.W. Creswell, "Research design: Qualitative, quantitative, and mixed methods approaches," *Sage Publications, Inc*, 2009.
  - [56] A Magazinius, R Feldt, "Validity Threats in Empirical Software Engineering Research-An Initial Survey," *SEKE.*, pp. 374-379, Jul 2010.
  - [57] Anglia Ruskin University. (2011, jan) <http://web.anglia.ac.uk/>. [Online]. [http://web.anglia.ac.uk/numbers/biostatistics/desc1/descriptive\\_1.pdf](http://web.anglia.ac.uk/numbers/biostatistics/desc1/descriptive_1.pdf)
  - [58] Dmitri Nevedrov. (2006, Feb) <http://dev2dev.bea.com>. [Online]. <http://dev2dev.bea.com/pub/a/2006/08/jmeter-performance-testing.html>
  - [59] HP. (2013, April) <http://snustaging.webapps.snu.edu.in/>. [Online]. [http://snustaging.webapps.snu.edu.in/sites/default/files/homeslider\\_pdf/HP\\_LoadRunner\\_tutorial\\_T7177-88041.pdf](http://snustaging.webapps.snu.edu.in/sites/default/files/homeslider_pdf/HP_LoadRunner_tutorial_T7177-88041.pdf)
  - [60] B.M. Subraya, *Integrated Approach to Web Performance Testing: A Practitioner's Guide: A Practitioner's Guide.*: IGI Global, 2006.
  - [61] Jürgen Börstler and Martin Boldt, Research Methodology 1. Thesis Lecture 2., 2015.
  - [62] Jürgen Börstler and Martin Boldt, Research Methodology 1. Thesis Lecture 2, 2015.

## APPENDIX I

**Instructions and Items.** The questionnaire's instructions and items are:

This questionnaire, which starts on the following page, gives you an opportunity to tell us your reactions to the system you used. Your responses will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you.

To as great a degree as possible, think about all the tasks that you have done with the system while you answer these questions.

Please read each statement and indicate how strongly you agree or disagree with the statement by circling a number on the scale. If a statement does not apply to you, circle N/A.

After you have completed this questionnaire, I'll go over your answers with you to make sure I understand all of your responses.

Thank you!

1. Overall, I am satisfied with how easy it is to use this system.

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

2. It was simple to use this system. 1

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

3. I could effectively complete the tasks and scenarios using this system. 1

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

4. I was able to complete the tasks and scenarios quickly using this system. 1

Strongly Agree: 1 2 3 4 5 6 7 Strongly  
Disagree. N/A

5. I was able to efficiently complete the tasks and scenarios using this system. 1

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

6. I felt comfortable using this system. 6

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

7. It was easy to learn to use this system. 6

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

8. The system gave error messages that clearly told me how to fix problems. 3

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

9. Whenever I made a mistake using the system, I could recover easily and quickly. 3

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

10. The information (such as on-line help, on-screen messages and other documentation) provided with this system was clear. 5

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

11. It was easy to find the information I needed. 4

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

12. The information provided for the system was easy to understand. 4

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

13. The interface of this system was pleasant. 2

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

14. I liked using the interface of this system. 2

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A

15. Overall, I am satisfied with this system. 2

Strongly Agree: 1 2 3 4 5 6 7 Strongly Disagree.  
N/A