CHAPTER 1 : INTRODUCTION

E-commerce plays a vital role in the growth of the majority of businesses worldwide. The rapid rise of web based E-commerce applications has led to lots of research on ensuring the high performance of E-commerce applications.

E-commerce websites are specialized web applications that allow commerce to be transacted electronically over the Internet without the need for a "brick and mortar" retail outlet (Ecommerce Website Design, 06). A typical E-commerce solution includes a website with a data backend, a shopping cart, a payment gateway tied to a merchant account.

The accelerated pace of technological growth across all areas of the Internet over the past several years have mandated that businesses keep abreast of new developments and technologies and leverage these to grow their businesses. E-commerce is one of the most important breakthroughs across all of Information Technology and now plays a critical part as one of the premier marketing and sales channels available to businesses and most will leverage E-commerce in some shape or form.

As such the high performance of E-commerce applications is crucial to ensure the success of E-commerce business.

1.1 RESEARCH MOTIVATION

This dissertation is motivated by the desire to further understand and apply the research of current researchers who have actually done massive research on load testing and developed very comprehensive load testing tools to test the scalability of the web server that host web components of the E-commerce applications. The tool developed is based on these research papers and existing web load testing tools.

Software testing in general is a crucial step in determining whether a software application is viable, ready for market and is free of bugs. No software will be completely free of glitches but thorough software testing can and will make sure that it is as error-free as humanly possible.

There are two broad types of software testing: functional and performance. Each has its merits and both are important. However this research would focus on performance testing and precisely on the load testing of the web servers hosting E-commerce applications and also comparing the web servers which has been tested.

1.2 RESEARCH OBJECTIVE

The following are the objectives of the research for this dissertation :

1. To identify and present the details of the factors influencing the performance of an E-commerce application as a roadmap to understand the big picture of performance testing for E-commerce applications.

2. Develop a prototype performance testing tool for E-commerce applications focusing on real time load testing of web servers hosting E-commerce applications. The tool would enable E-Commerce developers to test and compare the performance of web servers before selecting a web host to host their E-commerce application and also to test the web servers currently hosting their E-commerce application.

1.3 RESEARCH METHODOLOGY

The research methodology approach adopted in this dissertation is by gathering data and analyzing it to design and develop a new tool. After development, evaluation of the tool is done to see if it meets the objective.

1.3.1 DATA GATHERING AND ANALYSIS

The data gathering and analysis of this dissertation involves a lot of research. The types of research which were used are

i) Library Research

Through the library research, lots of information from diaries journals, books, magazines, speeches, interview, letters, memos, manuscripts and other papers can be gathered regarding performance testing for E-commerce. These sources give a good overview of the topic, outline the basic historical context and help identify key participants, dates and publications associated with the research topic.

ii) Internet Research

Information is everywhere on the internet, existing in larger quantities and continuously being created and revised. These information exist in a large variety of kinds (facts, opinions, stories, interpretations, and statistics) and is created for many purposes (to inform, to persuade, to sell, to present a viewpoint and to create or change an attitude or belief). For each of these various kinds and purposes, information exists on many levels of quality or reliability. It ranges from very good to very bad and includes every shade in between. It is the researcher's job as a searcher, then to evaluate what's located, in order to determine whether it suits the researchers need

iii) Observations

A great deal of useful information is obtained by simply observing and studying the current performance and web load testing tools rather than conducting interviews.

1.3.2 DESIGN

The designing of the tool is based on the study on the existing tools and the information gathered from the literature review.

1.3.3 DEVELOPMENT

The development of web server load test and comparison tool was done using the Rapid Application Development (RAD) methodology. The tool is developed on a Visual Basic environment but the backend of the tool which actually does the load test is done in Visual C++.

1.3.4 EVALUATION

Evaluation is a very important phase of the research methodology as it helps to evaluate the tool. First the tool is tested by the researcher itself to validate the tool. The functionality of the new tool is tested on different web sites with different load parameters. Next the tool is sent out to E-commerce developers for testing and their comments on the tool are used for evaluation. The result from the testing and evaluation is used to conclude if the research objectives are achieved.

1.4 SIGNIFICANCE OF THE RESEARCH

The quality of the E-commerce application in terms of performance is very important to determine its success in the online business. Performance testing is perhaps the most important form of testing as it is the way in which the response time of certain application under certain load can be detected. Performance testing can be broken down into four basic types which are classic, load, stress, and volume.

Load testing gauges how efficiently the application performs when its functions are tested with varying workloads.

1.5 SCOPE OF STUDY

The scope of this research is to identify and present all the contributing performance factors of the E-commerce applications. However most of the performance factors are not measured by the application which is to be developed as there a lot of previous researches and application measuring these factors.

The purpose of this research is to develop a tool to do a real time test on web servers to get the response time and the transaction details with predetermined parameters such as amount of users and the time limit for a specified web server. By analyzing and comparing these features on the web servers, a developer of an E-commerce application would be able to select the appropriate web server to host their E-commerce application.

1.6 ORGANIZATION OF THE DISSERTATION

This dissertation has been subdivided to seven chapters to give a comprehensive report on the research done.

Chapter 1 : Introduction

Gives an overview of the whole dissertation starting from research motivation, to research objectives, research methodology, significance of research, scope and organization of dissertation

Chapter 2 : Literature Review

In this chapter the performance testing of E-Commerce applications is further researched by analyzing current researches on performance testing and load testing of web servers. The out come of the research is discussed in this chapter.

Chapter 3 : Requirement Analysis and Specification

In this chapter existing web server load test tools are studied and requirement specification for the new tool is gathered from this study and also from the information from Chapter 2.

Chapter 4 : System Design

In this chapter, the top level components of OpenWebLoad, an existing web server load test tool is discussed in detail, as the new load test tool is actually an enhancement of OpenWebLoad. The new load test tool is designed with the proper understanding of the infrastructure of E-commerce applications and also of

OpenWebLoad. A prototype of the graphical user interface of the new tool is established in this chapter.

Chapter 5 : Implementation

In this chapter, the tools used to develop the new load test tool and how the new tool is implemented is discussed. The incorporation of the new features into OpenWebLoad is discussed in detail. The exact process flow of the tool along with the source code for tasks done by the tool is elaborated in this chapter.

Chapter 6 : Testing and Evaluation

The developed tool is tested with various parameters to determine the validity of the tool. Then the new tool was sent out to E-commerce application developers for them to test the web server they were considering to host their application or already hosting their application and also to test the web servers hosting E-Commerce applications available on the net. The results of the testing and user acceptance test are discussed in this chapter.

Chapter 7 : Conclusion and Future Framework

In this chapter the contribution of this tool to the developer community is discussed and the framework for future development and research is proposed.

CHAPTER 2 : LITERATURE REVIEW

2.1 E-COMMERCE

E-commerce or Electronic Commerce means any business that is transacted electronically, or over the Internet (Rosen, 2000). An E-commerce web site is needed to conduct business on the internet. An E-commerce web site can be as simple as a catalogue with an order form, or it can be a complex multimedia enabled web site with real-time payment processing capabilities. E-commerce offers customers the opportunity to browse, select and pay for goods in real time from the comfort of their own home or office. Business of all sizes can equally share the benefits of E-commerce.

According to Chaudry et al (2002), E-commerce has been around a little longer than most people think. Automatic teller machines (ATMs) have long allowed people to manage and transfer funds between accounts. When an ATM transaction takes place, banks communicate using standard electronic funds transfer (EFT) formats.

Electronic data interchange (EDI) is an even older form of E-commerce which allows organizations to exchange information through dial-up lines or leased lines. Today, EFT and EDI are used with the Internet or virtual private networks (VPNs) to enable online transactions between organizations. The buyer can send a purchase order over the Internet in EDI format and pay by using EFT to transfer funds to the seller's bank account.

As E-commerce use spreads across the business world, organizations will be connected to one another in an unprecedented way through the Internet.

Organizations of all sizes will look to the Internet to find buyers and sellers and build the efficient value chains. In this dynamic community of buyers and sellers, information will be the key to finding the most promising opportunities (Brain, 2006).

2.1.1 DIFFERENCE BETWEEN A WEB SITE AND AN E-COMMERCE SITE

An ordinary web site offers customers with only information and purchasing options are available off-line. It is like a copy of printed catalogue. All orders are manually entered into the system and processed off-line.

An E-commerce web site presents information and also provides secure options to complete the transaction online. An E-commerce web site can complement the existing traditional selling channels and provides high level security by encrypting data and transferring it over secure servers. It integrates the existing system for inventory control, order processing and order fulfillment.

2.2 IMPORTANCE OF E-COMMERCE

The Internet users are consistently increasing. Internet use is growing faster than use of any other technology in history. Each year the number of computers connected to the Internet increases worldwide and the number of Internet users has also increased. According to IDC Malaysia(2006), in Malaysia alone the users of internet is expected to increase each year, indicating huge number of prospective online customers.

The intensity of Internet use reflects levels of economic development. IDC Malaysia (2006) stated that the United States of America, Canada and Nordic countries adopted Internet technologies most rapidly, while the European Union, Australia, New Zealand, Japan, the Republic of Korea, Singapore, Hong Kong and China have caught up with the early adapters. Individuals, organizations and firms in developing countries in Southeast Asia, India, China, Brazil and Argentina have had access to the Internet since 1996, while it appears that the Internet will be slower in having an impact on least developed countries in Asia and Africa, or in countries that shun Internet use, such as Syria.

The Internet is a powerful tool for trade. Use of the Internet lowers communication costs, reduces time-to-market for goods and services, allows delivery of many kinds of information in a digital format, reduces transport and distribution costs, and allows for more fully integrated and broader business alliances.

The Internet is an emerging global trading platform. As Internet technology advances and overcomes problems with reliability and speed, it is likely to be used in almost every conceivable way to exchange information and trade goods and services. Many large firms now integrate online technology and building new Internet-based systems for supply chain management and other inventory control. In addition to selling their products and services, firms are also using the Internet to survey customers, supply mass and trade media with information and pictures, submit regulatory compliance documents, and manage and transfer funds.

Growth areas are emerging. The strongest growth in business-to-business (B2B) and business-to-consumer (B2C) E-commerce is in services such as publishing, media, finance, education, entertainment, provision of specialized software, and retailing of consumer products such as clothing, footwear, processed foods, health products and commodities.

Small and rural businesses may be among the biggest beneficiaries. By reducing transaction costs, the Internet provides unprecedented opportunities for small-and medium-sized firms to trade across borders. Lower transaction costs also provide opportunities for many rural and regional communities to revitalize their economic bases. The Internet can give farmers, small enterprises and communities the ability to present a regional image to the world, and create focal points (or portals) for inquiries about local businesses and their offerings.

The benefits of E-commerce help to define what E-commerce is about. E-commerce provides efficient ways to manage supply chains and value chains thus enhancing manufacturing, logistics, distribution and collaboration between business partners. E-commerce also cuts customer service costs while improving the quality of service and increasing manageability of customer relationships. It offers the opportunity to buy and sell products, services, and information on the Internet and automates business activities and the flow of information within and between organizations.

However it must be understood that all these benefit cannot be materialized if the concept of the whole E-commerce application is not well understood. Common misunderstandings and mistakes regarding E-commerce is, believing that every business belongs on the Internet and that having a web site will lead to instant sales. Even though using the Internet is an easy way to make a business and its products known worldwide, it must be remembered that there are about a third of a trillion pages visible on the World Wide Web and the number is growing very rapidly. Being present on the web does not necessarily mean success is ensured. Sufficient time must be spent on web site navigation, ease of use and content layout. Dibachi (2005) has said that the fastest way to lose potential customers is by making the web site complicated and difficult to use with lengthy load and processing time. Another reason to loose customers is a site that makes it difficult for the customers to identify the advantages of the product, or if the site is used for online transactions, a site that makes it difficult for the customers to buy. The web site must be updated frequently.

Perfect design for the site does not automatically lead to sales (Dibachi, 2005). Dibachi (2005) has stated that when designing and deploying an E-commerce system it is important to understand that it is not just another n-tier client server system because there are more risk involved. Firstly there are no second chances as websites sometimes form the customer’s crucial first impression of the business. The web site is often the first point of contact with customers. No second chances means the site must be up all of the time, key features must be accessible, and performance must be adequate. At least, this level of functionality must be provided or one has to be prepared to lose customers or viewers even before having the chance to sell to them.

It must also be understood that there is minimal control over the customers' environments. In E-commerce, end users can use PCs, Macs, Unix boxes, internet-enabled phones, 1024 x 760, 800 x 600 or they can use different web browsers such as Netscape Navigator, various Microsoft Explorer versions and different security settings. By far the most difficult part of ensuring the quality of an E-commerce application is ensuring the quality of the client environment.

Another risk is that the fact that we do not know the customer. There is no idea that will come into the web store. Most software products have been designed for a specific type of user. And most testers have tested products for a very specific type of customer. But with E-commerce applications this is very different. Since the customers are anywhere. For instance, if a customer is from Congo, does the one day shipping guarantee still hold, can their addresses even fit in the forms, or what currency and measurement system to be used? There are few opportunities for training and documentation. Training is not possible. Printed manuals are out. As such the website must be usable even in the non usual circumstances.

Another big risk involved is Web Time, more maintenance mode testing has to be done with an internet application than with a normal client server application. With an E-commerce site, the content will be changing frequently. The software development process begins to more closely resemble a publishing process. Typically there are many small changes to the user interface throughout the development cycle. Release cycles are measured in days, not months. The technologies used are uncertain, unstable technologies, and are constantly changing.

Since the E-commerce applications are entirely different from the conventional systems, there is a need to ensure the quality of the final deliverable product by overcoming the risks mentioned above. Each of the risk mentioned above has to be overcome with different approaches.

Testing process is one way of ensuring the quality. In conventional system, testing is followed as per the guidelines of the test plan. The test plan is mainly concentrating on the functionality of the application and is guided by the requirement specification. However, test plan alone is not sufficient to test the E-commerce applications due to its dynamic nature. Accordingly, the quality of the site may be measured in terms of time, structural, content, accuracy and consistency, response time and latency, and performance (Menasec, 2000).

However, an important quality parameter, performance, has to be addressed in order to ensure the reliability and speedy accessing of E-commerce sites. Performance problems can bring all sorts of undesired consequences, including financial and sales loss, decreased productivity, and a bad reputation for a company. It is important to plan ahead of the system deployment and capacity of networked systems (Barker, 2000). Capacity planning involves being able to predict when the existing applications will fail to meet the required performance levels.

Performance testing is directly reflecting the behavior of the complete E-commerce web site. Visitors expect the fast response with in a short period of time. Therefore, a rigorous performance testing must be carried out on each site. Performance testing can be viewed as a "Black Box" testing, which focuses on application and system behavior from outside, with no knowledge of the program code that supports the system.

2.3 UNDERSTANDING THE ARCHITECTURE OF E-COMMERCE APPLICATIONS

Performance testing of an E-commerce application needs a thorough understanding of the E-commerce architecture. Figure 2.1 shows the architecture of E-Commerce as elaborated by Dumm (2003).

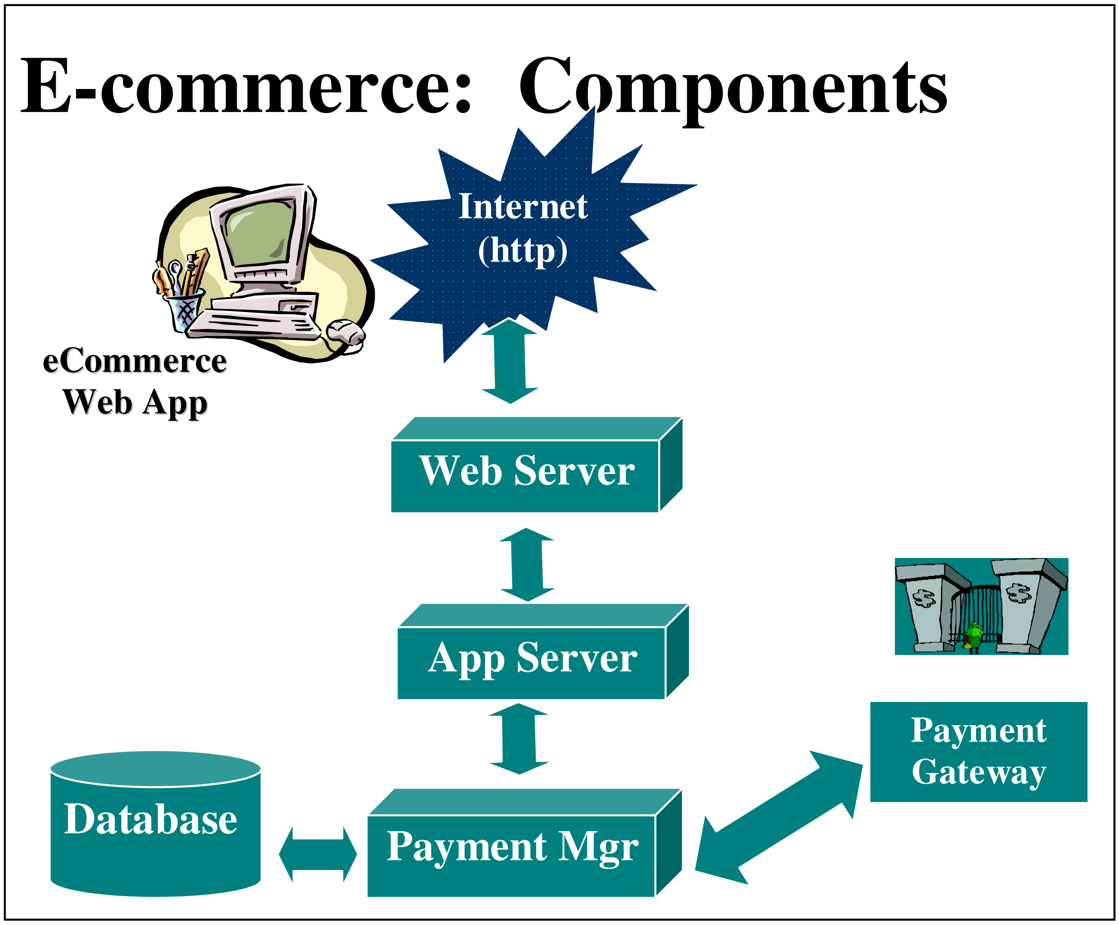


Figure 2.1 : Architecture of E-Commerce Application (Dumm,2003)

Each component in the architecture plays its own role in determining the performance of the web application. As such each component must be analyzed thoroughly before being applied in the architecture.

In order to develop a performance testing tool it is crucial to understand the architecture of the E-commerce applications. Electronic Commerce Websites incorporate many components familiar to software testers, including GUIs and client-server architectures. Automated testing tools do exist to help with the testing of these sites, but the real challenge is dealing with new and buggy technologies married to a mission critical retail application (Dibachi, 1999). As such proper understanding of the E-commerce architecture is important to test an E-commerce application. Most of today's E-commerce environments are based on distributed, multi-tiered, component-based architectures. The inherent complexity of such architectures makes it extremely difficult for system deployers to estimate and plan the size and capacity needed to provide the desired service levels. Thus the understanding of large distributed E-commerce applications is important for system deployers to estimate and plan the size and capacity needed to provide the desired service levels. The process of analyzing the performance of web servers is important in determining the right web server to host E-commerce applications. Figure 2.1 gives a clear perception of the architecture of a typical E-commerce application and the components in the architecture.

According to Monchai Sopitkamol (2004), typical E-commerce sites are complex and are composed of several connected servers. Each server generally has several configuration parameters. Settings of these parameters can have significant performance of individual servers and the whole site.

One of the typical architectures employed by many E-commerce sites is a three-tiered architecture as shown in Figure 2.2. This architecture consists of three logical layers of servers running different applications to provide support for various functionalities offered by those sites. These layers include Web server, application server, and database server. This architecture which was used by Monchai Sopitkamol to conduct his research on workload characterization is the architecture which would be used through out this dissertation.

A web server serves as a front end of an E-commerce site, where HTTP requests are received from the clients, processed according to the type of request, or, if necessary, forwarded to the next tier, the application layer, and replied back to the clients. (Subrahmanya, 2000)

A web server is a server program implementing HTTP protocol. A typical server would:

i) Listen on a known port (usually 80)

ii) When a request arrives, create a new socket, hand it over to a helper process

iii) Go back to the listening mode

iv) The helper process would communicate with the client and serve the request

At the middle tier, application servers implement the business logic by invoking server-side applications. Examples of application server implementations are CGI scripts, FastCGIs, server-applications, server-side scripts, Active Server Pages (ASP), Java Server Pages (JSP), and Enterprise Java Beans (EJB) (Subrahmanya, 2000). Application servers may interact with the next tier, transaction and/or database servers, if they need transaction processing to be performed, and/or need to access a database.

Finally, transaction and/or database servers take the requests, perform the corresponding transaction processing, and/or database queries or updates, and then send the results back to the application servers, which, in turn, generate HTML pages that are sent back to the client by the Web servers.

These three logical layers can be implemented on one machine or can be distributed to different machines, depending upon the workload intensity level and the system capacity.

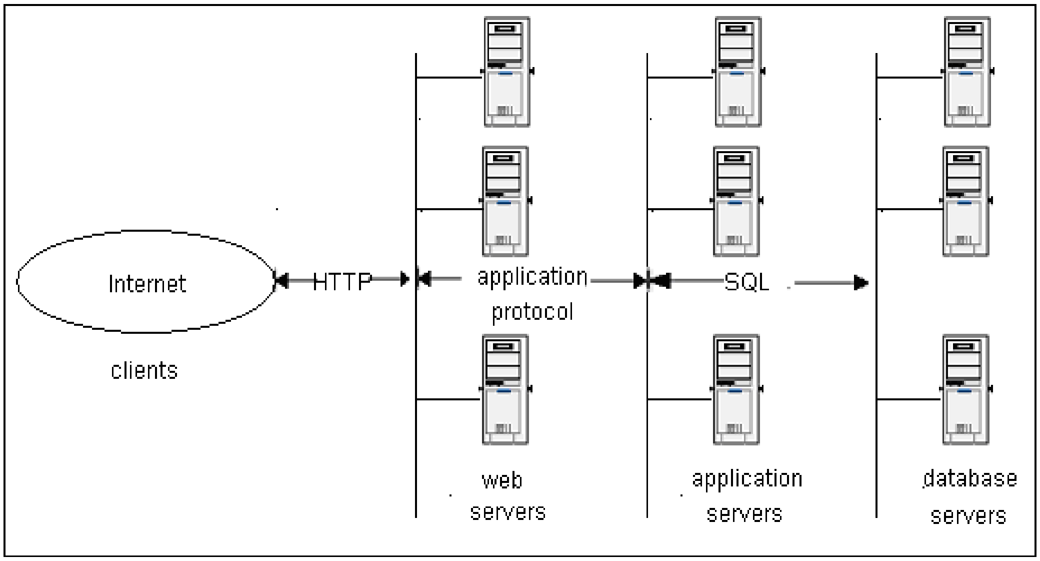


Figure 2.2 : Multi-tiered architecture of an E-commerce application (Sopitkamol, 2004)

2.3.1 MULTI-TIERED ARCHITECTURE

Some designs are more sophisticated and consist of three different kinds of nodes: clients, application servers which process data for the clients, and database servers which store data for the application servers. This configuration is called a three-tier architecture, and is the most commonly used type of client-server architecture. Designs that contain more than two tiers are referred to as *multi-tiered* or *n*-tiered.

The advantages of *n*-tiered architectures is that they are far more scalable, since they balance and distribute the processing load among multiple, often redundant, specialized server nodes. This in turn improves overall system performance and reliability, since more of the processing load can be accommodated simultaneously.

The disadvantages of *n*-tiered architectures include:

1. More load on the network itself, due to a greater amount of network traffic.

2. More difficult to program and test than in two-tier architectures because more devices have to communicate in order to complete a client's request.

In Technical Encyclopedia of Techweb (2006) a client is defined as an application or system that accesses a (remote) service on another computer system known as a server by way of a network. The term was first applied to devices that were not capable of running their own stand-alone programs, but could interact with remote computers via a network. These terminals were clients of the time-sharing mainframe computer.

Client-server, according to Technical Encyclopedia (2006) is a computing architecture which separates a client from a server, and is almost always implemented over a computer network. Each client or server connected to a network can also be referred to as a node. The most basic type of client-server architecture employs only two types of nodes: clients and servers. This type of architecture is sometimes referred to as *two-tier*. It allows devices to share files and resources.

2.4 PERFORMANCE TESTING FOR E-COMMERCE APPLICATIONS

Over the past several years interest in E-commerce has grown significantly. In order to be successful, a business must not only attract new customers but retain existing ones, as profits rise with extended customer relationships (Krishnamurthy, 2001). The rapid rise of web based E-commerce applications has led to a plethora of design and development tools designed to minimize the time to market for many E-commerce applications. Designing and developing, however, are only two phases of application development. An E-commerce application demands rapid development and deployment. Testing, a critical third phase occupies the dominant role in the life cycle of the E-Commerce application (Subrahmanya, 2000).

During the deployment of an E-commerce application, one of the most important issues need to be addressed, is application's performance. Unlike traditional client/server applications, where user loads can be easily predicted, performance problems are inherent to E-Commerce systems, where the volume of users and traffic levels can change in a matter of seconds. (Meier, 2004)

Online merchants desire to maintain a continuous, consistent presence on the Web in order to keep customers satisfied and maximize revenues and returns on their infrastructure. Two problems are typically encountered with deploying E-Commerce web sites. First is overload, where the volume of requests for content at a site temporarily exceeds the capacity for serving them and renders the site unusable. Second is responsiveness, where the lack of adequate response time leads to lowered usage of a site, and subsequently, reduced revenues. Both issues are instances of a larger problem, given the unpredictability of web accesses. (Elnikety, 2004)

To an online customer, quality of service means fast, predictable response service level of a web site noted in a real time. User measures the quality by response time, availability, reliability, predictability, and cost.

According to (Subraya and Subrahmanya, 2000) the main objectives of the performance testing are:

i) Maximum number of concurrent users that can be supported while offering "acceptable performance.”

ii) Maximum number of concurrent users that can be supported prior to causing a system failure.

iii) Location of bottlenecks within the application architecture.

iv) Impact of a software or hardware change on the overall performance of the application.

v) Scalability issues

Some performance testing is done by recording or scripting the actions that real users perform, and then playing those actions back against the System Under Test (SUT), in an automated and controlled manner where else some performance testing is done at real time. Performance testing also depends on certain issues like, Graphics, Load Time and Connection Methods. (Subraya, 2000)

Performance of many E-Commerce Web sites depends on the load on the site at peak time under varying conditions. Traditionally, client-server testing does not ensure the performance of E-Commerce Web sites, as there are unknown users with many uncertainties. Under these circumstances performance testing must be carried out in such an environment which reflects the real users. Since it is difficult to provide such an environment before the deployment of the application, it is more appropriate to follow a methodical approach that ensures the required performance under varied circumstances. (Subraya, 2000)

2.4.1 Load and Scalability Testing

The purpose of load and scalability testing is to ensure that the E-Commerce application will have a good response time during peak usage. These tests can also be used to test how the E-Commerce application will behave over time as the E-Commerce site contains more and more data in the database. First some testing scripts are written to populate the database with an average amount of data. The performance tests are then run to measure response time. Next the database is populated with an extreme amount of data and performance tests are run again. If response times are significantly larger for the second test, then something is wrong.

Load testing provides insight into the working of the E-Commerce Web site in the day to day conditions (theserverside, 2000). It models the behavior of users in the real world. Test script mimics the behaviors that the users commonly perform, and include think time delays and arrival rates reflective of those in the real world. When selecting think time delays and arrival rates to be used in the load test, it is best to choose values observed during peak load. The behavior of the applications selected will depend on the type of the system under test. In some of the applications, the test may run for a short period of time depending on the functionality being tested. Load test mainly provides insight in to the application under test. Most E-Commerce Web actions should complete in seconds, thus a 10-30 minute test duration is usually adequate to bring the load up to a steady state, exercise the desired functionality, and get a large result set. (Subraya, 2000)

2.4.2 Stress Testing

Stress testing deals with the quality of the application in the environment. The purpose is to create an environment more demanding of the application than the application would experience under normal work loads. This is the hardest and most complex category of testing to accomplish and it requires a joint effort from all teams.

There are two approaches (Nguyen, 2003) which can be used to establish a test environment:

i) Hardware intensive approach

This approach involves the use of multiple client workstations in the simulation of real-world activity. The advantage of this approach is that one can perform load and stress testing on a wide variety of machines simultaneously, thereby closely simulating real world use. The disadvantage is that a large number of workstations are needed to perform such testing.

ii) Software intensive approach

This approach involves the virtual simulation of numerous workstations over multiple e connection types. The advantage of the software-intensive approach is that only a few physical systems are required to perform testing. The disadvantage is that some hardware, software, or network specific errors may be missed.

Race conditions and memory leaks are often found under stress testing. A race condition is a conflict between at least two tests. Each test works correctly when done in isolation. When the two tests are run in parallel, one or both of the tests fail. This is usually due to an incorrectly managed lock.

A memory leak happens when a test leaves allocated memory behind and does not correctly return the memory to the memory allocation scheme. The test seems to run correctly, but after being exercised several times, available memory is reduced until the system fails.

Stress testing is load testing with the user think time delays removed. In stress tests, scripted actions are executed as quickly as possible. Stress testing reflects the worst case scenario likely to be experienced by a E-Commerce Web site. It can bring to light performance issues that might not otherwise be seen, such as database record contention or the impact of too few connection service processes. In some of the applications, it is essential to increase the stress and find out whether the application under test breaks or not. Some of the Web site applications may not require stress testing as intensive because of its nature. However, stress testing is essential for all E-Commerce Web based applications.

2.4.3 Test Plan for Performance Testing

The test plan for performance testing differs from the conventional test plan used in software testing. The conventional test plan tries to bring about the various test cases to test the functionality of the system. In case of performance testing, the functionality of the system is already tested and the correct system is subjected to next level of testing, which is performance testing viewed from the user perspective. Thus the test plan for performance testing mainly deals with performance indicators under varying conditions. The test cases include the stress levels for each performance indicators, number of users, type of grouping the users, architecture which includes client's machines and browsers. Purpose of each test must be specified for each performance indicators and probable expected performance levels have to be provided. The test plan in a distributed environment must include the location wise details such as the system configuration, software platform and performance indicators. Performance indicators differ from location to location and are subjected to the performance requirements of the specified location. Based on the test plan, performance test is conducted and the results are analyzed. The results, if represented in the form of graphs for different stress levels, will reflect the true behavior of the system. Based on the analyzed reports, if the user is not satisfied then the application is modified and subjected to the same cycle of operation. The application modification may include, reconfiguring the servers, fine tuning the database servers, and suitable enhancement to the architecture or software. Each Scenario can be tested individually or as a test suite for specific interval of time, known as test sessions. The test sessions may be one minute or hours or days that depends on the application and the type of test conducted on the application

2.5 PERFORMANCE TESTING TOOL FOR E-COMMERCE APPLICATIONS

Web site performances and analysis is a crucial factor in determining today's online business's success. When testing for performances one has to stimulate a large number of users using the E-Commerce Web site simultaneously. Manual testing of such issues is almost impossible and thus, tools are introduced.

Tools are used to simulate heavy loads and analyze E-Commerce Web sites. Heavy load simulation helps us determine the load our E-Commerce Web site can handle, what needs to be fixed, and where to fix in order to achieve the desired result.

The effectiveness of a testing process may significantly depend on the tools used to support the process. Testing tools usually automate some of the tasks required by the process, such as test case generation, test case execution, and evaluation of the test case results. Several testing tools, for example NeoLoad support the production of useful testing documentation and provide a configuration management of it. This kind of testing is very important for E-Commerce Web applications, for in any time, users may turn to other competitive E-Commerce Web sites due to the intolerable waiting. For this type of testing, automated testing tools are needed to generate the system workload and gather the accurate system performance and load metrics under different levels of user visiting.

The usual method of performance testing is to setup multiple virtual users by Scripts, which visit and interact with the Web application automatically and concurrently, imitating the visiting actions of the actual users, so as to determine the running performance parameters of this system (Swama, 2001). It is important to know under the precondition of reducing the testing cost, how to obtain the actual and effective visiting actions of users, and apply them into the generation process for the virtual users, so as to make the visiting effect is consistent with the realities (Lucca, 2002). It is important to ensure the quality of the virtual users, that is obtaining the visiting actions of actual users firstly, then modeling the users' common interesting so as to obtain the typical visiting behaviors, next founding the stochastic procedure model to guide the actions of a single user, and finally generating virtual users by the two models, so as to obtain more accurate and authentic performance testing results. In this way, we can achieve the actual metrics of the system performance (Xu, 2005).

2.6 WEB SERVER TESTING

As mentioned earlier, E-commerce applications are composed of many components with several configurable parameters that, if properly configured, can optimize system performance. The web server from the E-Commerce system layer significantly impacts overall system performance, and the performance of the most significant web function types. When testing web server for its performance, understanding of the web server is very important.

A web server (program) has defined load limits, because it can handle only a limited number of concurrent client connections (usually between 2 and 60,000, by default between 500 and 1,000) per IP address (and IP port) and it can serve only a certain maximum number of requests per second depending on(Nguyen, 2003):

i) its own settings

ii) the HTTP request type

iii) content origin (static or dynamic)

iv) the fact that the served content is or is not cached

v) the hardware and software limits of the OS where it is working

When a web server is near to or over its limits, it becomes overloaded and thus unresponsive.

Main key performance parameters (measured under a varying load of clients and requests per client) can be determined by testing web servers.

According to Bacigalupo et al, (2005), following are the key performance parameters:

i) number of requests per second (depending on the type of request, etc.);

ii) latency response time in milliseconds for each new connection or request;

iii) throughput in bytes per second (depending on file size, cached or not cached content, available network bandwidth, etc.).

Above three parameters vary noticeably depending on the number of active connections, so a fourth parameter is the concurrency level supported by a web server under a specific configuration. Performance of a web server can be measured using load testing tools.

2.7 WEB SERVER SELECTION

Section 2.5 and 2.6 has given a clear view of the importance of selecting the proper architecture component for an E-commerce application to perform well. There are two options for a developer to configure the architecture of their E-commerce application :

1. Buy and configure the components in the architecture
2. Subscribe the service of e-commerce hosts to host their application

The selection of either way depends on the financial and expertise resource available for the developer as well the performance requirement expected for the E-commerce application. Generally most small scale developers prefer getting a web host or E-commerce host to host their application.

E-commerce hosts are companies that provide space on a server they own for use by their clients as well as providing Internet connectivity, typically in a data center. E-commerce hosts can also provide data center space and connectivity to the Internet for servers they do not own to be located in their data center, called colocation.

There are certain selection criteria generally used to select the suitable E-commerce hosts. Web site hosting can be a complex undertaking. Determining how much space needed, how much transfer, finding a reliable host, and getting everything online is no simple task. Add E-commerce to the mix and things become even more complex.

All of the same considerations to finding general hosting can be applied to E-commerce hosting, there are simply a few additional ones that need some attention :

i) Basics- Disk Space and Transfer

E-commerce application need more space due to the shopping cart module

ii) Shopping Cart

It is important to find a shopping cart that suits the individual user.

iii) Reliability and Support

The most reliable hosting company with good uptime and support must be chosen.

As such the performance of the web server has to be tested before we can actually choose the web server to host the E-commerce applications. By conducting load tests on the web server, the performance of the web server can be determined.

As stated in Globat, other issues that must be resolved and prioritized during the selection of web server host include ongoing maintenance and physical support services, network quality and speed, web server speed, contract terms, and of course the value for money. Prospective online merchants will generally have a good idea how many products they will be selling initially. This will vary wildly from merchant to merchant, and many merchants do not put their entire stocks online. As such the selection of the web server host should be based on individual business needs.

2.8 CONCLUSION

The literature review has given a clear understanding on the components of E-commerce applications and factors that influence the performance of E-commerce applications. This research proceeds based on the knowledge gathered from the literature review. The architecture which is going to be used for further research is based on Sopitkamol (2004) research as shown on Figure 2.1. According to Scott Barber (2006), web server plays an important role in determining the performance of an E-commerce application. As such it is very important to choose the most appropriate web server to host an E-commerce application based on the needs of the business. If heavy load is expected on the E-commerce site, then a web server with a better scalability and fast response time has to be chosen. To determine the scalability and response time of the web server towards a request, the web server has to be load tested. The next chapter would proceed with analysis on the current load test tools available in the market and the result of the analysis would be used to design and develop a new load test tool.

CHAPTER 3 : REQUIREMENTS ANALYSIS AND SPECIFICATION

3.1 INTRODUCTION

The literature review has enabled us to begin thinking about the role of web servers in ensuring the high performance of E-Commerce applications. It has also uncovered valuable research detailing the methods that best support performance testing of E-Commerce applications. Although testing the whole E-Commerce components and parameters are essential in making an E-Commerce application perform well, due to time and resource constrains, this dissertation would focus on the load testing of web servers and also to compare the result of the web server's tested.

However it is important to balance this theoretical base with an understanding of more practical issues surrounding existing load testing tool for web servers hosting E-Commerce applications.

The requirements analysis process enables us to produce a "stable set of requirements that form a sound basis to move forward into thinking about design" (Preece et al, 2002). Moreover, the emergence of a detailed set of requirements provides a framework for effective evaluation of the final system as a solution to the problem space.

3.2 LOAD TESTING TOOL FOR WEB SERVERS HOSTING E-COMMERCE APPLICATIONS

In Chapter 2, a clear understanding of the multi tier E-Commerce architecture was established based on the architecture which was used by Monchai Sopitkamol (2004) in his research on workload characterization. After looking at the three different layers of the architecture, the importance of the performance of the web server has been noticed.

Most websites and web applications run smoothly and correctly as long as only one user (e.g. the original developer) or just a few users are visiting at a given time. But when thousands of users access the website or web application at the same time the performance would be different. A load test tool can simulate various load patterns for the web server which will help to find problems in the web server set up. With steadily increasing loads, the amount of load the server can handle before serious problems arise can be determined.

In order to conduct a performance test on the web server, a load test tool has to be designed and developed to determine the capacity of the web server.

The web server has to handle the http request under various loads and send a response to the request as fast as possible. As such, the capacity of the web server which is to host the E-Commerce application has to be tested with different parameters and the most appropriate web server which can handle the expected load of the E-commerce application being developed must be used to host the application in order for it to give the best performance. With that in mind a prototype of a load test tool has to be developed to test different web servers.

3.2.1 EXISTING WEB SERVER LOAD TEST TOOLS

There are plenty of web load test tool available in the market. For this research, the tools OpenSTA, OpenWebLoad and WAPT 4.0 were studied to get a clearer view on the functions and processes of a web server load test tool. Each tools execution, result and monitoring is analyzed based on previous research and also on the description of the tool on their respective portals. Conclusion for each tool is derived in following sections.

3.2.1.1 OpenSTA

Open STA is an HTTP load test application with scenario creation and monitoring, which can be distributed on different computers. It is an open source software developed in C++, and released under the GPL licence. Buret (2003) has done a review on the tool. The following review is based on Buret's research. OpenSTA provides a script language which permits to simulate the activity of a user. This language can describe HTTP/S scenario, OpenSTA can only build load tests for web applications. Scripts can be written with an editor provided with the application or a module which records the request and response of a browser and generate a script automatically can be used. To simulate a great number of users, this script is executed by each virtual user at the same time.

i) Execution

The entire test configuration is managed in the graphical interface. The composition of the test is very simple (drag & drop of elements). Scripts can be chosen for a test, a remote computer that will execute each test, the number of users, different monitoring possibilities and to launch the test.

During the test execution, parameters like CPU idle time, network traffic, number of processes, disk usage, and error in HTTP response can be monitored. For each user configured in the test, OpenSTA creates one thread which executes the script.

The test can be distributed on several computers. OpenSTA provides a name server for registering the different computers. During the building of the test, which computer each scenario will be launched can be selected. All the distribution is based on CORBA. It uses the omniORB Object Request Broker (ORB) and Naming Service from AT&T Laboratories Cambridge.

ii) Results

A great number of data can be caught during the test. First, all information of system monitoring (WinNT performance monitor or with SNMP). Information on the test, like number of requests, error, throughput of client can be captured. All information can be reviewed with customizable graphics. All information can be exported in a file in order to be used with other software like Excel.

iii) Monitoring

Lots of queries can be made to monitor the computer. Two systems can be used for this are Windows performance monitor or SNMP server. Windows can only be used if the computer uses Windows NT and if OpenSTA is launched (with registration in the name server), it is not really a problem for all the clients because OpenSTA is designed only for Windows. The second solution needs a SNMP server in the network and SNMP agent on each computer monitored.

iv) Conclusion

Positive parts of this software are the following:

-A user-friendly graphical interface.

-The script capture from the browser.

-The monitoring functionality

- A complete test can be done with the script language

Negative parts of this software are the following:

-Only designed for Windows

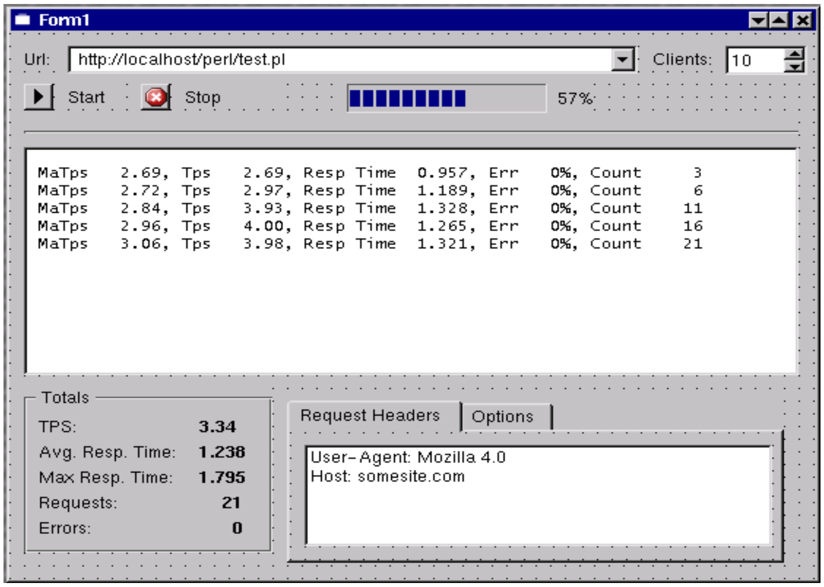
-Only for HTTP

3.2.1.2 OpenWebLoad

OpenWebLoad is a tool for load testing web applications. It aims to be easy to use and providing near real-time performance measurements of the application under test. This is particularly useful when you are doing optimization as you can see the impact of your changes almost immediately. The load test module in the new tool is abstracted from this open source load test tool.

i) Execution

OpenWebLoad is a load testing tool for Web applications. It simulates a number of concurrent users and it measures transactions per second (a transaction is a completed request to the Web server) and response time. Currently it has no proper user inter face to show the results of the test. But there has been some proposal for the user interface. Figure 3.1 shows the first attempt made on the user interface for this tool.



Url: http://localhost/perl/test.pl

Clients: 10

```
MaTps   2.69, Tps   2.69, Resp Time  0.957, Err   0%, Count    3
MaTps   2.72, Tps   2.97, Resp Time  1.189, Err   0%, Count    6
MaTps   2.84, Tps   3.93, Resp Time  1.328, Err   0%, Count   11
MaTps   2.96, Tps   4.00, Resp Time  1.265, Err   0%, Count   16
MaTps   3.06, Tps   3.98, Resp Time  1.321, Err   0%, Count   21
```

Totals

TPS:	3.34
Avg. Resp. Time:	1.238
Max Resp. Time:	1.795
Requests:	21
Errors:	0

User-Agent: Mozilla 4.0
Host: somesite.com

Figure 3.1 : User interface for OpenWebLoad (OpenWebLoad, 2006)

ii) Results

However the reporting of OpenWebLoad is limited to the following parameters:

1) Transactions per second (TPS)

   Is the number of completed requests during that second.

2) Maximum Transactions per second (MaTPS)

3) Total TPS

   The average TPS for the whole run, that is :

   (Total completed requests) / (Total elapsed time).

4) Response time (Resp Time)

   The average response time in seconds for the elapsed second.

5) Average response time (Avg. Response time)

   The overall average response time in seconds.

6) Maximum Response Time (Max Response time)

   The highest response time during this run.

7) Error (Err)

   The percentage of responses that was erroneous, i.e. didn't return a HTTP 200 Ok status.

8) Count

   The total number of completed requests.

iii) Monitoring

As OpenWebLoad does not have features such as saving into database and analysis graphs to compare servers which are being tested, a new tool would be developed to enable an E-Commerce developer to make comparison of the tools and select the web server which would best suit the developers application.

iv) Conclusion

Positive parts of this software are the following:

- Very simple to use and yet gives out results same as the mentioned tools.
- It is free as it is an open source application
- Compatible to other programming languages such as Visual Basic 6.0 to further enhance the features of the tool.

Negative parts of this software are the following:

- It does not have the capability to conduct record and playback test
- There is no monitoring function

3.2.1.3 WAPT 4.0 (Web Application Testing)

WAPT is a very straight forward Web testing tool and requires no training to operate. It uses advanced technique to simulate live load conditions. The approach used is more efficient and in effect produces high load conditions. It sends in requests to the website, creating a simulation of many different users coming from different IP addresses, each defining their own parameters, cookies, input data for various page forms, connection speed, username and password, etc. WAPT can also simulate random page access time to make each virtual user as similar as possible to a human user. WAPT allows you to group the virtual users into separate activities, defining customisable parameters of each of the user and their path through the site.

i) Execution

WAPT has varied test scenario setting:

- Capacity and Overall Performance.
- Performance Characteristics.
- Stress.
- Reliability and Endurance.
- Real User Experience/Background Testing.
- Scalability.
- Bench Marking.

WAPT allows options in load/stress test. The options are to choose either a constant load level for the whole test session or define the steps at the rate that you would want to add virtual users at specific time intervals, along with the virtual user specifics mentioned earlier. WAPT also provides the facility to choose website access bandwidth for the chosen virtual users.

ii) Results

The test results displayed by WAPT are through descriptive reports and graphs. The real time analysis are even available while the test is running, letting the key parameters of the web application to be monitored. It helps to find out how the website is responding to the increasing load volume as the test proceeds. At the end of each test, the summary reports and graphs are available in HTML for determining the presence of any problem in the overall performance of the website. The report and its graphical representations allow study of the different virtual user profiles helping to diagnose and identify bottlenecks. These HTML reports can also be shared with the technical team, the client, etc., making distribution easy.

iii) Monitoring

It has a running graphical display of the real time process and presents optional viewing of the different behavioral characteristics of the various test parameters. The analysis reports are presented in HTML form and are easily exportable to Excel and .csv files.

iv) Conclusion

Positive parts of this software are the following:

- Supports multi user activities, with users coming from different IP addresses having individual parameters, cookies, input data for various page forms, usernames and passwords, connection speed and having own specific path through the site.

- Supports Constant and Ramping patterns of virtual user access to the test website.
- Supports concurrent and individual virtual user access.
- Supports scheduling of Load/Stress test with defined Virtual User parameters.
- Supports testing of dynamic web applications.
- Provides real time graphical display of results as the test is running, with options to view individual parameter behaviour, including page to page representation
- Supports Proxy Servers.
- Supports cookies.
- Comprehensive Reporting.
- Automated Testing for Unit Testing, System Testing

Negative parts of this software are the following:

- Limited option in Load/Stress test pattern design.
- Does not support custom scripts.
- Commercial license

3.2.1.4 Conclusion

The following table shows the comparison of the three tool which has been studied.

Table 3.1 : Comparison of openSTA, OpenWebLoad and WAPT 4.0

Description of Tests	openSTA	OpenWebLoad	WAPT 4.0
Graphical user interface	YES	NO	YES
Transaction Diagnostics	YES	YES	YES
Scalability: Maximum Virtual Users	>1000	>1500	>1000
Custom Scripting	YES	NO	NO
Record and playback test	YES	NO	YES
Analysis of Test Results	YES	YES	YES
Results can be saved to database	YES	NO	YES
Analysis of Log	YES	NO	YES
Test Scheduling	NO	NO	YES
Real time load testing	NO	YES	YES
Comparison of two servers test result	NO	NO	NO
Platform	Windows	Windows and Linux	Windows
Cost of the System	FREE	FREE	USD 210 from 20 to unlimited virtual users

As conclusion, the comparison of the three tools has enabled the researcher to analyze the three tools in terms of functionality, user friendliness, reporting and cost. The WAPT 4.0 is a test tool with automated testing features, comprehensive reporting capability and is able to conduct real time load test. OpenSTA on the other hand is a HTTP load test tool with the scenario creation and monitoring. However one has to have the knowledge of writing test scripts in order to use OpenSTA. OpenSTA does not support real time load test. Where else OpenWebLoad is a real time load test tool, however minimal effort has been done on the development of the user interface of this tool. OpenWebLoad tool only functions as a test tool and has no comprehensive reporting functions.

Although WAPT 4.0 is good tool to load test, it is a commercial tool and not all the E-commerce developers can afford to buy a test tool. OpenSTA on the other hand is an open source tool but it does cannot run a real time test. As real time test has been identified as a very important test for load testing of web servers, the functions of OpenWebLoad can be used to build the prototype in this research. However the reporting technology from WAPT 4.0 and OpenSTA can be used as a guidance to do the reporting feature of the prototype as OpenWebLoad does not have good reporting technology.

3.3 REQUIREMENTS FOR THE NEW TOOL

The web server from the E-Commerce system layer significantly impacts overall system performance, and the performance of the most significant web function types. Following the overview of existing software and applications that deal with load testing in all aspects, the new application is a collection of the advantages and drawback of the tools mentioned. Of course it is not possible to build a tool as superior to WAPT 4.0, but features of the tools are used as an idea to build the new tool.

The performance metrics of interest include the web servers response time and transaction per second. The load test would be done at real time as it will give a clearer perception of what are expected results with certain load. For this purpose the OpenWebLoad component have been chosen to be integrated to the new tool.

3.3.1 LOAD SIMULATION

Load simulation is the most important feature of a load testing tool. Load is simulated to send in http request to the web server. A user can send one or more request at a time.

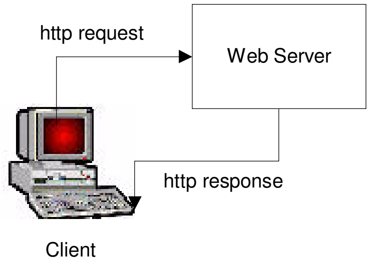


Figure 3.2 : Client web server request and response

3.3.2 REAL TIME TEST

The load test displays the real-time statistic such as the measured response times of all Web pages, the response times of all URL calls, and the current execution steps of all concurrent users.

Thanks to the availability of real-time access to automatically created "frozen" error snapshots, error analysis does not have to wait for the end of a load test, but can begin immediately after an error has occurred. This can be a significant advantage for load tests which last many hours.

3.3.3 WEB SERVER RESPONSE TIME AND TRANSACTIONS

There are many parameters which can be recorded with a load test. However for this tool we would focus on response time of the web server in handling the requests sent in by the users and also transactions which were successful transacted. This tool keeps count of the errors which occurred during request and the requests which got through. The following parameters would also be calculated

i) Average response time (seconds)

ii) Maximum response time

iii) Total number of requests

iv) Transactions per second

v) Total request

vi) Total Error

3.3.4 RECORD INTO DATABASE

The result from the test is saved in the text file. From here the test result is to be saved into the table in the database. This recorded test result would be later used to compare the performance of two web servers.

3.3.5 COMPARE TWO WEB SERVERS

As the purpose of this tool is to compare web servers to host the e-commerce applications, a log display and graph display module has to be added.

3.3.5.1 DISPLAY LOG

The program sorts the log information records that are linked with the respective response results according to a predetermined sort condition and displays the sorted log information records onto a display.

3.3.5.2 DISPLAY GRAPH

The program sorts the log information records that are linked with the respective response results according to a two predetermined web servers to be compared and displays the sorted log information records onto a display along with a graph tp give a clearer view of the comparison of the two servers.

3.4 REQUIREMENTS SPECIFICATION

A set of requirements specification has been created for the new tool based on the requirements which has been identified from section 3.3. The functional specification has been divided into two that is functional specification and non functional specification which would be discussed in section 3.4.1 and 3.4.2

3.4.1 FUNCTIONAL SPECIFICATION

Functional specifications, in the end, are the blueprint for how you want a particular web project or application to look and work. It details what the finished product will do, how a user will interact with it, and how it will look like. By creating a blueprint of the product first, time and productivity are saved during the development stage because the programmers can program instead of also working out the logic of the user-experience.

A **use case** is a technique used in software and systems engineering to capture the functional requirements of a system. Use cases describe the interaction between a primary system actor the initiator of the interaction and the system itself, represented as a sequence of simple steps. Actors are something or someone which exist outside the system under study, and who (or which) take part in a sequence of activities in a dialogue with the system, to achieve some goal: they may be end users, other systems, or hardware devices. Each use case is a complete series of events, from the point of view of the actor.

A **use case diagram** is a type of behavioral diagram defined by the Unified Modeling Language (UML). Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals represented as use cases and any dependencies between those use cases.

Based on the list of requirements which has been identified for the new tool, the requirements has been categorized into three major module. The modules are test load , log information and graph analysis.

3.4.1.1 TEST LOAD

Test Load module is the module which would run the main process of load testing the web servers hosting e-commerce applications. It would consist of the OpenWebLoad component which has been built using C++ programming language, the Graphical User Interface (GUI) component which would be built using Visual Basic 6.0 and the database component which is to be developed using Microsoft Access.

It is very important to understand the functions of the OpenWebLoad component which is used in the new tool to generate load and record response time and transations details. The following chapter would further elobrate the functions and components in the OpenWebLoad component which is part of the test load component in the new tool.

The GUI component is the front end form where the tester enters test paramaters such as the url of the web server which is to be tested, amount of users to be generated for the load test and duration of the test. These parameters are then sent to the OpenWebLoad component which would then genearate the load specifed for the duration specied for the web server which is to be tested. Once the result is generated the result would be saved into the database component to be used in the remaining two modules that is log information viewing and web server comparison analysis. Figure 3.3 shows the data flow diagram for the test load module.

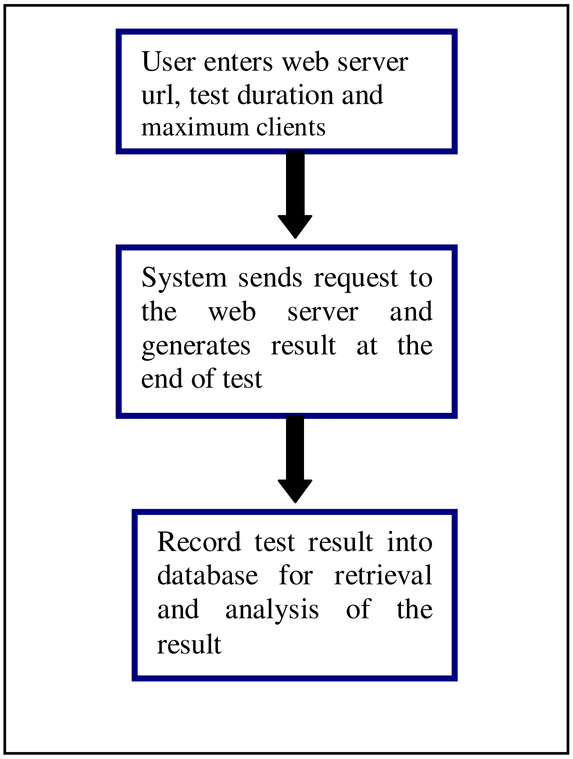

User enters web server url, test duration and maximum clients
System sends request to the web server and generates result at the end of test
Record test result into database for retrieval and analysis of the result

Figure 3.3 : Data flow diagram for test load module

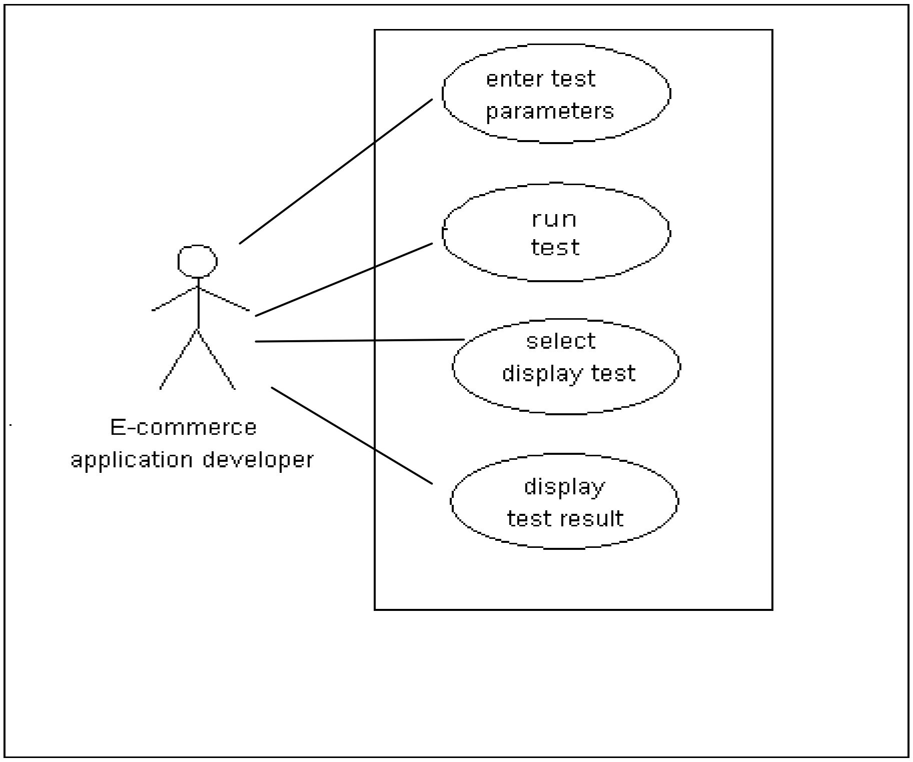


Figure 3.4 : Use case diagram for Test Load component

Figure 3.4 shows a use case diagram for the test load component. It shows the criteria to be keyed or selected by the tester whom would be an e-commerce applications developer and the subsequent output of selected criteria.

The test load module has to be connected to internet as shown on the Figure 3.5 as it is a real time test.

As such for the test to be conducted, the workstation hosting the tool must be connected to the internet. The internet connection speed should be at good level for this tool to generate an acurate result.

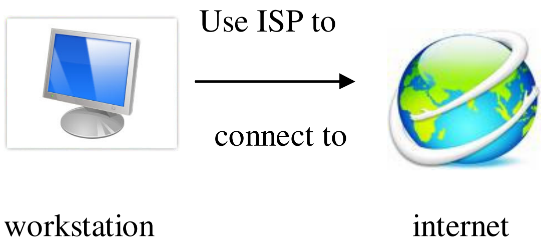


Figure 3.5 : Typical internet environment

The database component for this module is to save the test parameters details and all the test result which has been generated during the test. As such the log information for a particular test which is distinguished by web server url , date and time of test might have multiple entries for response time. As such only one table needs to be created in this database to save the details of the test.

3.4.1.2 LOG INFORMATION

This is the module which is used to retrieve all the information needed for a particular test. The tester would select the date on which the test was conducted and the tool would generate a list of web servers which was tested on that particular date. Then the tester selects the server name he wants and the tool would display the log information and as well as the test details. The following diagram shows GUI use case diagram for Log Information component.

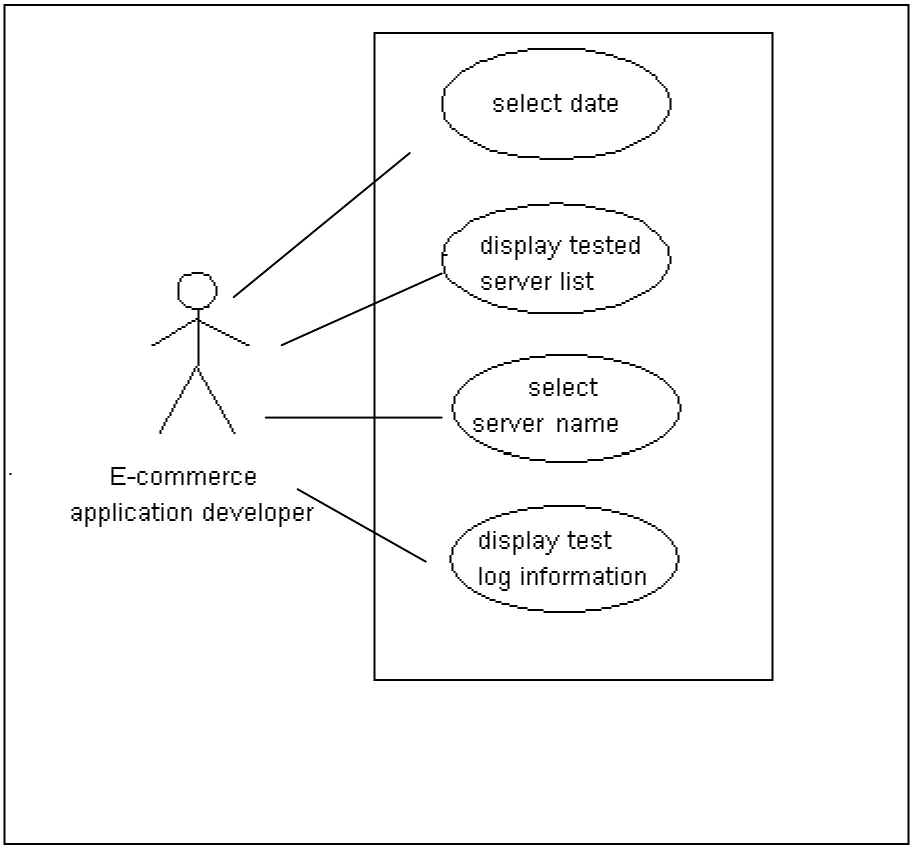

select date
display tested server list
select server name
display test log information
E-commerce application developer

Figure 3.6 : Use case diagram for Log Information component

3.4.1.3 ANALYSIS GRAPH

This is the componet which is used to compare two web servers which have been tested. The tester first would have to select the number of users and list of servers which was tested with that particular load would be displayed by the tool. Then the tester selects the servers to be compared and the test result would be displayed in tables and graph format. Figure 3.7 shows the use case diagram for Analysis Graph component.

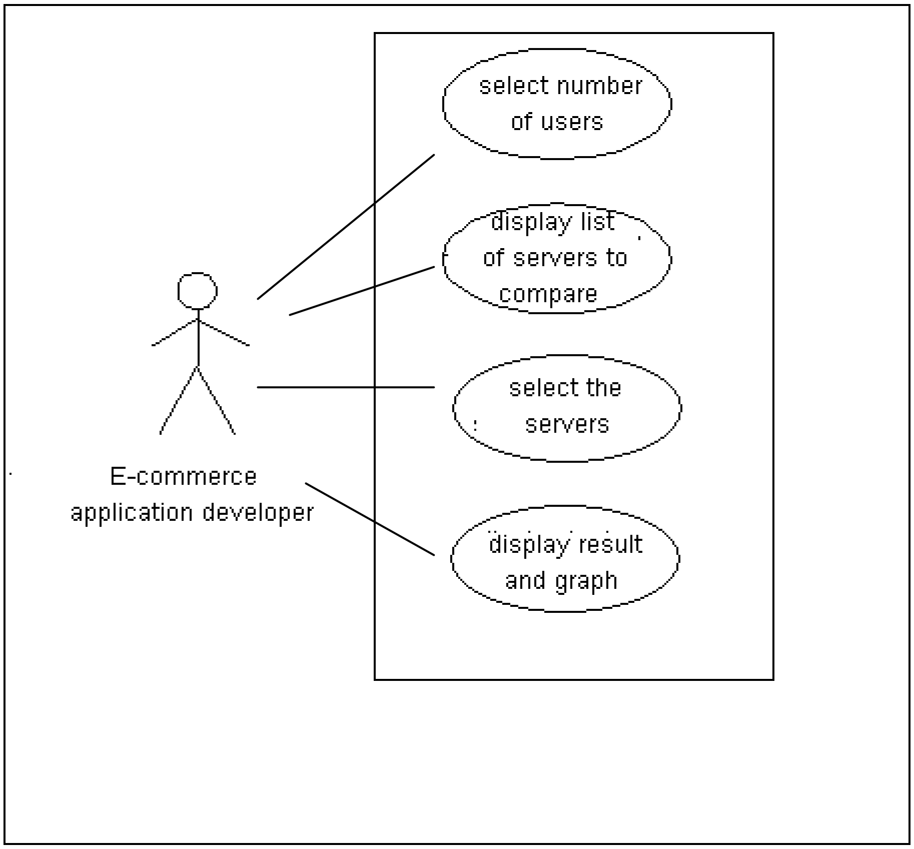

select number of users
display list of servers to compare
select the servers
display result and graph
E-commerce application developer

Figure 3.7 : Use case diagram for Analysis Graph component

3.4.2 NON FUNCTIONAL SPECIFICATION

The non functional specification for this new tool would be as the following:

1) User friendly

The tool must have minimal steps to run the test and also to see the result. It should be simple and straight forward.

2) High performance speed

The time to start the test should be immediately after start or run button is clicked. There shouldn't be a big time lapse for the test to actually start.

3) Ease of distribution

As the tool is not a web application. The tool has to be set up on individual computers running the test. As such the application file should be packaged and sent to the users to be downloaded and installed to the computer. The steps involved in installation should be minimal and straight forward.

CHAPTER 4 : SYSTEM DESIGN

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy the requirements specifications from the previous chapter. Based on the functional specification from the previous chapter, there are three main things to be designed, that is the designing of the test itself, GUI and the database.

4.1 LOAD TEST DESIGN

To design the load test it is very important to understand how OpenWebLoad functions because the test is incorporated with the components of OpenWebLoad to run the load test. Section 4.1.1 gives a clear description of the top level components and the source code involved in each component in OpenWebLoad. Following the analysis of the OpenWebLoad, the new tool which is to be a Visual Basic 6.0 application, has to be designed to interact with the components as described in Section 4.1.1.

4.1.1 OpenWebLoad

The following description (OpenWebLoad, 2006) gives an overview of the top level components of the test tool which is derived from OpenWebLoad. Figure 4.1 shows the top level components in OpenWebLoad.

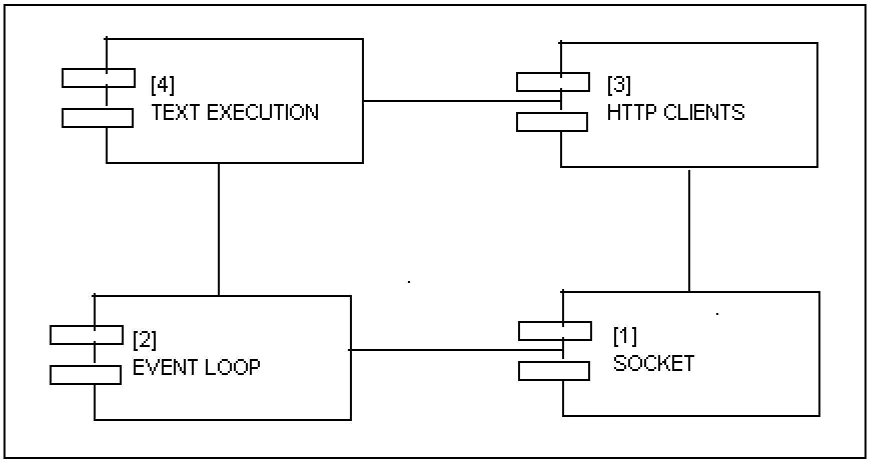


Figure 4.1 : Top level components in OpenWebLoad (OpenWebLoad, 2006)

[1] Socket

The socket component handles the interface to the socket layer (WinSock for Win32). Sockets are used asynchronously, there are no blocking calls. This means that the socket component is implemented as a state machine.

The component provides interfaces for:

i) Connecting to a foreign host.

ii) Sending a buffer or string.

iii) Reading a buffer or line.

iv) Closing the socket

Each of these operations provides a callback at completion.

The operations above are incorporated in the class CTcpSock. The CTcpSock represents a socket used for asynchronous operation. When a CTcpSock has been created, it can be registered with class CEventLoop from file event_loop.h which passes 'events' from a select system call to the socket object, whenever data arrives or the socket is ready to receive more data for writing. Handling of the events is implemented as a finite state machine. Callbacks can be registered for whenever the socket changes state. Callback function pointers are stored in the m_cb* member variables. This class represents an asynchronous TCP socket.

[2] Event Loop

The event loop listens for incoming events from the registered sockets. Events are:

i) Socket connected to remote host.

ii) Data arrives on a socket.

iii) Socket is ready to send more data.

iv) An error occurred on a socket

The event loop continuously receives events (by calling the select function), and dispatches the arriving events to the corresponding socket objects.

The component provides interfaces for:

i) Adding a socket to the event loop

ii) Removing a socket from the event loop

iii) Running the event loop

[3] HTTP Client

The HTTP client implements the HTTP protocol on top of the socket component, where it sends HTTP requests to the web server under test and receives and interpretes the HTTP responses. This is also implemented as a state machine. The component also handles HTTP headers (request and response) and URLs.

The component provides interfaces for:

i) Parsing URLs

ii) Managing a list of HTTP headers, that is adding, removing, finding and iterating over individual headers

iii) Sending a HTTP request and receiving the response. A callback function is called when the request completes with information about the response, status, etc.

The following pseudo code shows how this component works.

create socket and context

```
First add socket to the event loop
Then get ready to send command
Next start connecting
Check what method to use
IF ok
First allocate buffer for the command
Then create the command as 'method path HTTP/1.1CRLF'
```

Set connection header to close

Then check for host header

Next get ready to send the first header

Send the command

And free last send buf

Next allocate buffer for the header line

First create header line as 'name: valueCRLF'

set context to point to next header

if there are no more headers

send an extra CRLF

if this is POST

get ready to send body

otherwise to read status

Next send the header

free last send buf

get ready to read status

send the body

Next get ready to parse status

get ready to read more headers

read first header

IF got an empty line, done with reading headers

delete the line buf

check if body is chunked

set start len to 0

get ready to get chunk len

allocate linebuf

read first chunk len

get content length

handle when content-length header is not set

add header to the response object

read next header

set actual read len

check if we are done

extend buffer

get ready for reading chunk

get the chunk len

delete the line buf

Next check if we are done

allocate space including his chunk

First copy old data if necessary

Then read the chunk

update total size

check if the socket was closed at the other end

allocate linebuf

read chunk len

```
remove sock from event loop
delayed delete (and implicit close) socket
make call back
cleanup
```

[4] Test Execution

This is the main program that actually executes the test. It works by starting the requested number of simultaneous HTTP requests and then entering the event loop. Every time a request completes a callback function is called which updates the statistics (increments the count of transactions and notes the response time) and each second output is generated.

The following pseudo code shows how this component works.

Get request parameters

```
TReqParams* pReqParams = (TReqParams*) pContext->m_pParam;
For TEST mode show body and stop
Show response
Get response time (duration) of this request
Check if we got a HTTP 200 OK response
verify if needed
increment sequence number
Update counters
if it is more than a second since last report, do a new one
Calculate moving average of TPS
check if have reached the time limit
```

```
if verifying is enabled, update the url
Send new request
parse arguments
add request header
Get time limit
Then convert time limit to milisecond
```

Following the description in Section 4.1, the new tools test component would have a new user interface which would enable to key in parameters for the test and see result of the test. The graphical user interface (GUI) of the new tool would be discussed in the next section. The detailed integration of OpenWebLoad and the new features would be discussed in Chapter 5.

4.2 GRAPHICAL USER INTERFACE DESIGN

The graphical user interface design for each module of the proposed tool is based on the specification which has been determined in chapter 3. Figure 4.2, Figure 4.3 and Figure 4.4 shows the design for the test load module, log information module and graph analysis module respectively.

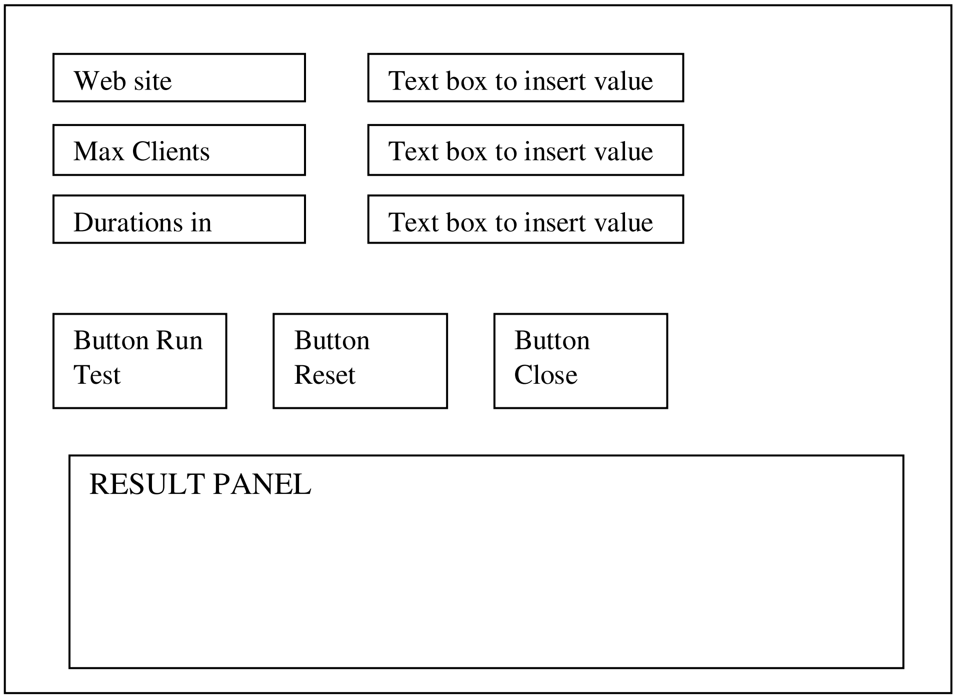


Figure 4.2 GUI design for test load module

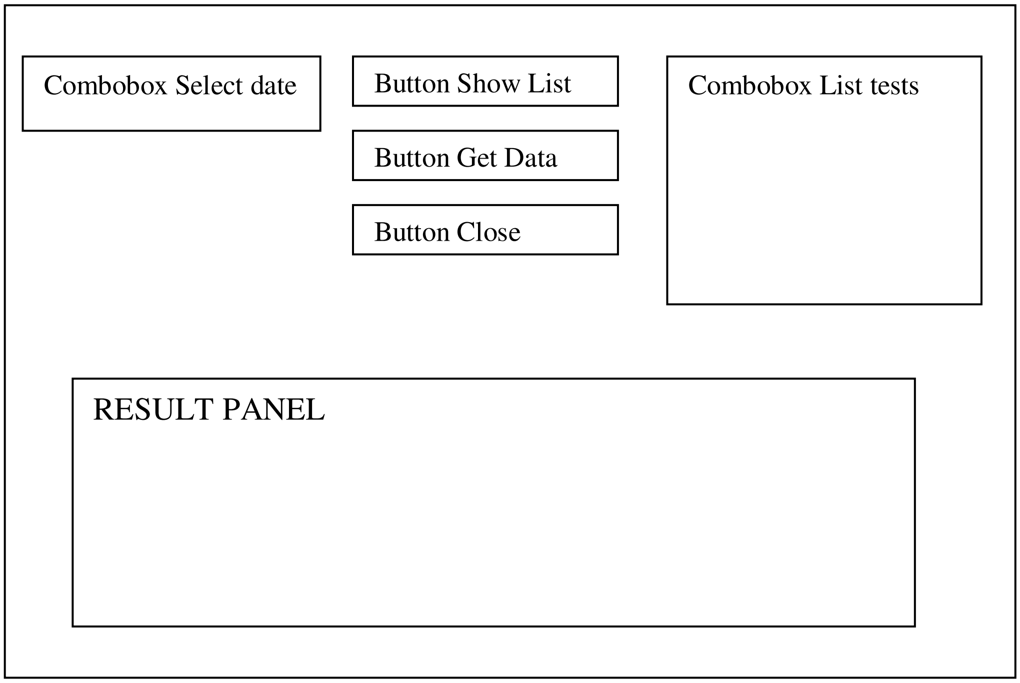


Figure 4.3 : GUI design for log information module

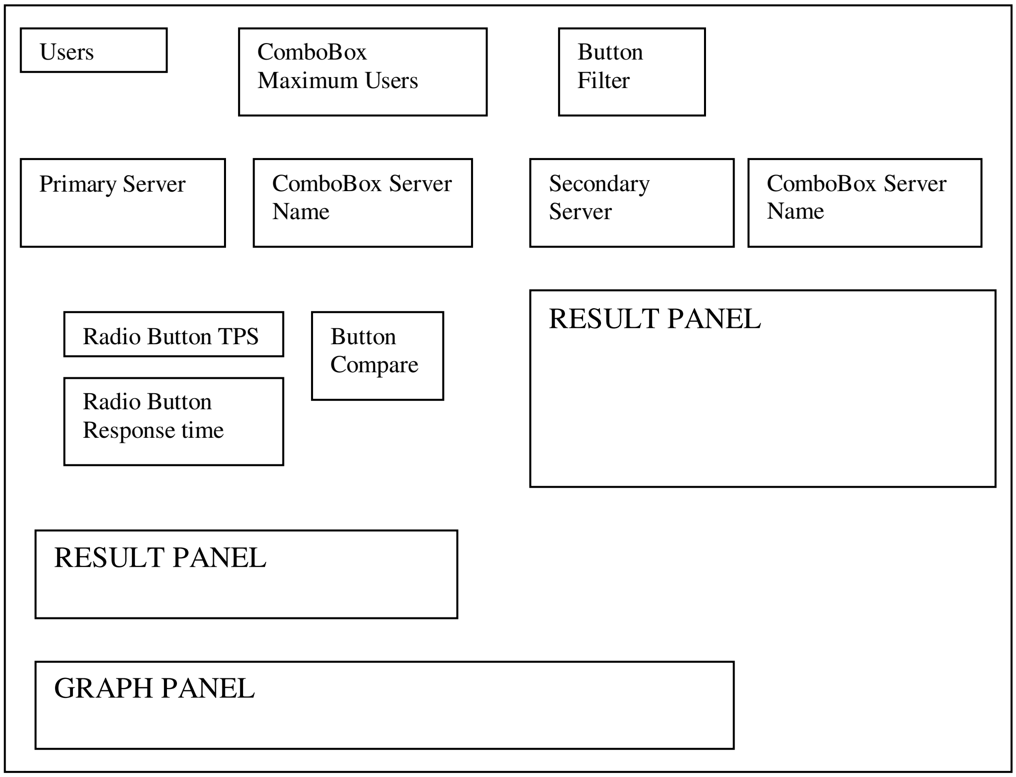


Figure 4.4 : GUI design for log information module

4.3 DATABASE DESIGN

Microsoft Access database is used to save results from the test.

Following are the details of the database structure.

Database Name : ltest

Table Name : History

Field : Table 4.1 shows the field name and data type for fields in table History

Table 4.1 : List of fields for table history

No	Field Name	Data Type	Description
1	server	Text	url of the web server
2	Log Date	Date/Time	Test date
3	LogTime	Date/Time	Test time
4	ClientCount	Text	Number of clients
5	TLimit	Text	Duration of test
6	MaTPS	Text	Maximum Transactions per Second
7	TPS	Text	Transaction per Second
8	RespTime	Text	Response time
9	ErrCount	Text	Number of errors
10	LCount	Text	Number of completed request
11	TTPS	Text	Total Transactions per Second
12	AvgRespTime	Text	Average response time
13	MaxRespTime	Text	Maximum Response time
14	TotReq	Text	Total request
15	TotErr	Text	Total error

CHAPTER 5 : IMPLEMENTATION

Once the system design has been established the proposed prototype load testing tool was developed using the specification and design established in chapter 3 and chapter 4. As this is only a prototype tool, the Rapid Application Development (RAD) methodology was used as RAD involves iterative development, and the construction of prototypes. Traditionally the rapid application development approach involves compromises in usability, features, and execution speed. It is described as a process through which the development cycle of an application is expedited. RAD thus enables quality products to be developed faster, saving valuable resources.

Once the development was completed the tool was implemented and used. This chapter would look into the details of every module available in the new tool. The details include the source code for a particular behaviour of the tool and the screen shots of the forms available in the test.

5.1 IMPLEMENTATION LANGUAGE

In both the requirements document and the design chapter we discussed and specified the need to provide an interface language which can interact with open source tool OpenWebLoad which was developed using Visual C++.

Visual Basic (VB) was designed to be easy to learn and use. The language not only allows programmers to create simple GUI applications, but can also develop fairly complex applications as well. Programming in VB is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code.

Visual Basic can create executables (EXE files), ActiveX controls, DLL files, but is primarily used to develop Windows applications and to interface web database systems.

Visual Basic compiler is also shared with other Visual Studio languages (C, C++). As such it was the most suitable compiler to be used with the components from OpenWebLoad.

5.2 IMPLEMENTATION OF WEB LOAD TEST TOOL

Using the VB 6.0 compiler the test tool developed has been converted into an application file to be saved into the computer of the tester or user of the tool. Once installed the tool is ready for use as long as it is connected to the internet. Please refer to Appendix A for the installation instruction and test procedure.

The main form which is the WebLoad form has a menu called File. Figure 5.1 shows the WebLoad form.

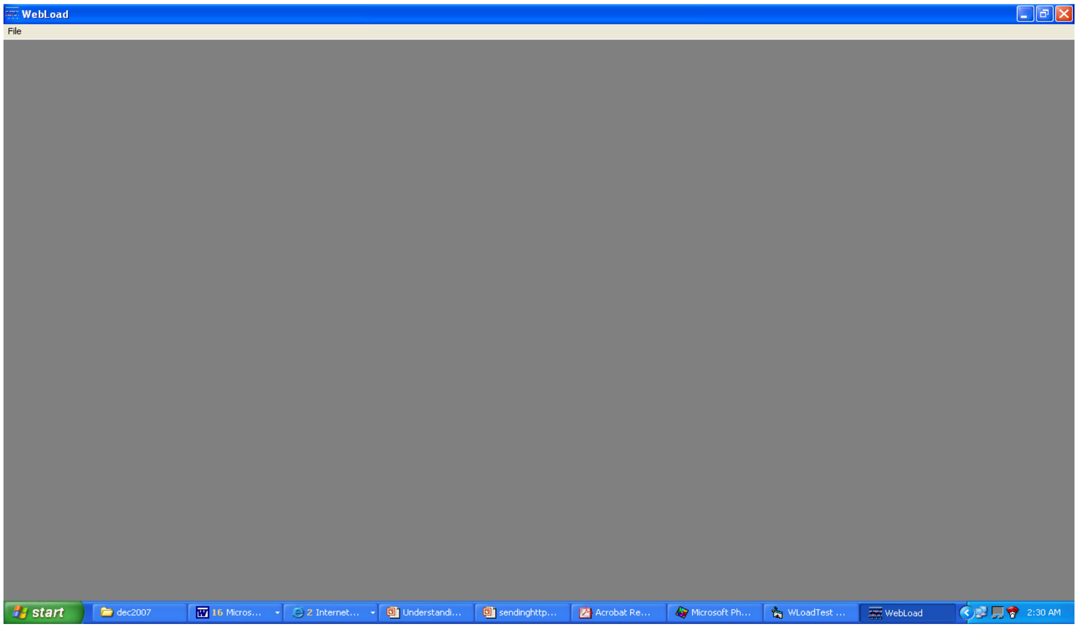


Figure 5.1 : WebLoad form

Once the file is clicked the list of the three main modules of the test tool are displayed. The main modules are:

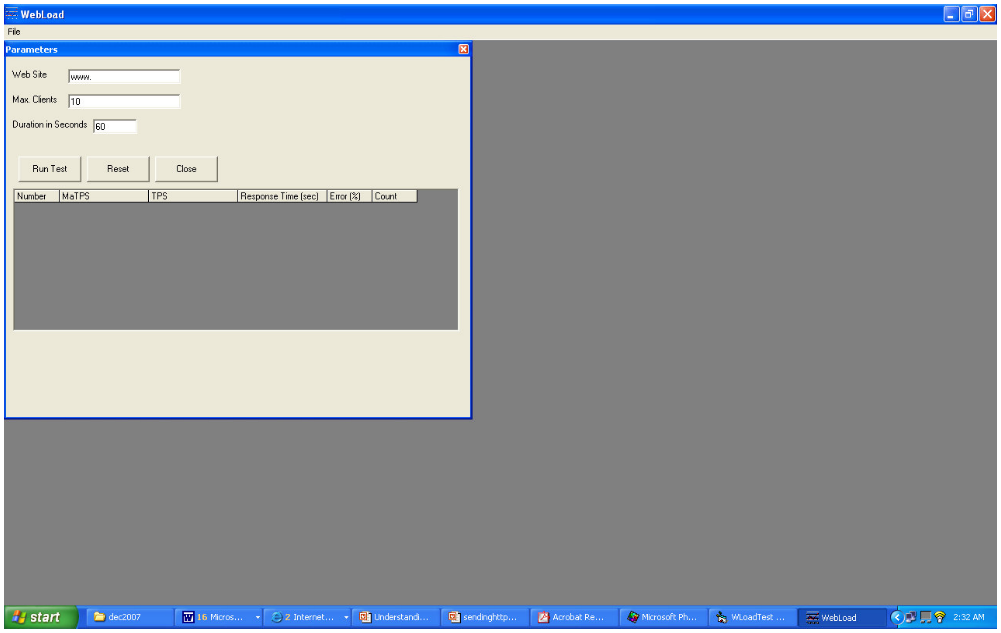
1) Test load

2) Log Info

3) Analysis Graph

5.2.1 TEST LOAD

This is the module where the user enters the test parameters for the test. Figure 5.2 shows the parameter form where the test specifications are entered.



Web Site: www.

Max. Clients: 10

Duration in Seconds: 60

Number	MaTPS	TPS	Response Time (sec)	Error (%)	Count

Figure 5.2 : Parameters form

The parameters entered are:

i) Web site

A valid url must be entered in the text box. Once the url is entered, the specified url is checked using a class called "*ClsResolve*" in the visual basic file before actually passing the parameters to the C++ application. If the url is not valid the following error message would be displayed:

"*Invalid web server specified. Destination cannot be reached. Please enter a valid address*"

However if the url is valid the parameters would be passed to the C++ application.

ii) Max Clients

Maximum amount of virtual users are determined here. The test tool has the capacity to generate up to 5000 virtual users at a time. The actual load which can be achieved is highly dependent on the network infrastructure, the server/client hardware, the file sizes and the e-commerce application itself.

iii) Duration in seconds

The duration has to be adequate to the clients entered for the test. The following table gives guidance to the corresponding duration for the amount of users.

Table 5.1 : Recommended duration for the amount of users

Amount of users	Recommended Duration
1-500	More then 60 seconds
501 – 1000	More then 180 seconds
1001 – 2000	More then 240 seconds
2000 – 3000	More then 300 seconds
3001 – 4000	More then 360 seconds
4001 - 5000	More then 420 seconds

5.2.1.1 INTERGRATION OF THE NEW FEATURES WITH OPENWEBLOAD

The Test Load module is where the integration of the C++ component from OpenWebLoad called "loadtest" and the visual basic(VB) is done. The parameters entered in this module once verified are passed to the C++ codes.

First the user of the tool has to specify the URL of the web server to be tested, the maximum amount of clients for the load test and also the duration of the test. If the web server URL is left empty an error message would be displayed as follows: "*Please enter a valid web site address.*"

The default value for the maximum clients is set at 10 and duration is set at 60 seconds. If the user does not specify the maximum clients and duration, these values are used as the test parameters to run the test.

Once all the three parameters have been specified and the "Run Test" button is clicked, verification is done for the validity of the URL specified before the parameters are passed to the *loadtest* application. *ClsResolve* is the class which is used in the VB codes to verify the URL.

The following pseudo code shows the how the verification is done.

```
First class ClsResolve  is defined

    Dim cResolve As ClsResolve
    Set cResolve = New ClsResolve
    rsv = cResolve.GetIPFromHostName(Trim(tWeb.Text))
```

```
If URL not valid error message is displayed

    If rsv = Empty Then
    MsgBox "Invalid web server specified.  Destination cannot be reached.
    Please enter a valid address.", vbOKOnly + vbCritical, "Error"
    End If

Else pass the parameter to the loadtest application

    Param = App.Path & "\loadtest -l " & Trim(tDuration.Text) & " " &
    trim(tWeb.Text) & " " & Trim(tClients.Text)

End if
```

Once the C++ receives the parameters, the test runs as explained in section 4.1.1 of chapter 4.

While the test is running, a progress reporting tool would pop up to show the remaining time for the load test. Once the result is completed the tester will receive a message stating “test complete” and tester has to close the progress reporting tool. Once the progress reporting tool is closed the “Get Result” button becomes visible in the parameter form.

The result for the test is captured in a text file called loadresult.txt. Once the “GetResult” button is clicked the information from the loadresult.txt is extracted and saved into the table History in the database ltest and the loadresult.txt is cleared. The result of the test is then displayed on the result panel which is a sub form. The result would include the following parameters:

1) Transactions per second (TPS)

   Is the number of completed requests during that second.

2) Maximum Transactions per second (MaTPS)

   Is a 20 second moving average of TPS.

3) Total TPS

   The average TPS for the whole run, that is :

   (Total completed requests) / (Total elapsed time).

4) Response time (Resp Time)

   The average response time in seconds for the elapsed second.

5) Average response time (Avg. Response time)

   The overall average response time in seconds.

6) Maximum Response Time (Max Response time)

   The highest response time during this run.

7) Error (Err)

   The percentage of responses that was erroneous, where the request did not return a HTTP 200 Ok status.

8) Count

   The total number of completed requests.

The following figure shows the VB code that will establish the task mentioned above.

```
        While Not EOF(1)
            Line Input #1, tmpbuf
            liResult.AddItem tmpbuf
        Wend
        Close #1
        'cGraph.Visible = True
        cResult.Visible = False
        'cSave.Visible = True
        Grid1.Rows = liResult.ListCount - 7
        For ictr = 1 To Grid1.Rows - 1
            Grid1.TextMatrix(ictr, 0) = ictr
            tmatps = Mid(liResult.List(ictr + 2), 6, 7)
            ttps1 = Mid(liResult.List(ictr + 2), 18, 7)
            tres = Mid(liResult.List(ictr + 2), 36, 7)
            terr = Mid(liResult.List(ictr + 2), 48, 4)
            tcount = Mid(liResult.List(ictr + 2), 60)
            Grid1.TextMatrix(ictr, 1) = Trim(tmatps)
            Grid1.TextMatrix(ictr, 2) = Trim(ttps1)
            Grid1.TextMatrix(ictr, 3) = Trim(tres)
            Grid1.TextMatrix(ictr, 4) = Trim(terr)
            Grid1.TextMatrix(ictr, 5) = Trim(tcount)
          Next ictr

    Open "loadresult.txt" For Input As #1
LbTTPS.Caption = liResult.List(liResult.ListCount - 5)
```

```
     LbAvgResp.Caption = liResult.List(liResult.ListCount - 4)
     LbMaxResp.Caption = liResult.List(liResult.ListCount - 3)
     LbTotReq.Caption = liResult.List(liResult.ListCount - 2)
     LbTotErr.Caption = liResult.List(liResult.ListCount - 1)
                 LbTTPS.Visible = True
                 LbAvgResp.Visible = True
                 LbMaxResp.Visible = True
                 LbTotReq.Visible = True
                 LbTotErr.Visible = True
                 terfound = 0
For ictr = 1 To Grid1.Rows - 1
                   If Trim(Grid1.TextMatrix(ictr, 4)) = "100" Then
                       terfound = terfound + 1
                   End If
   Next ictr
     If terfound = Grid1.Rows - 1 Then
     MsgBox "Connection was refused on port 80 of server : " &
                tWeb.Text, vbOKOnly + vbCritical, "Error"
                 Else
      cSave_Click
   End If
```

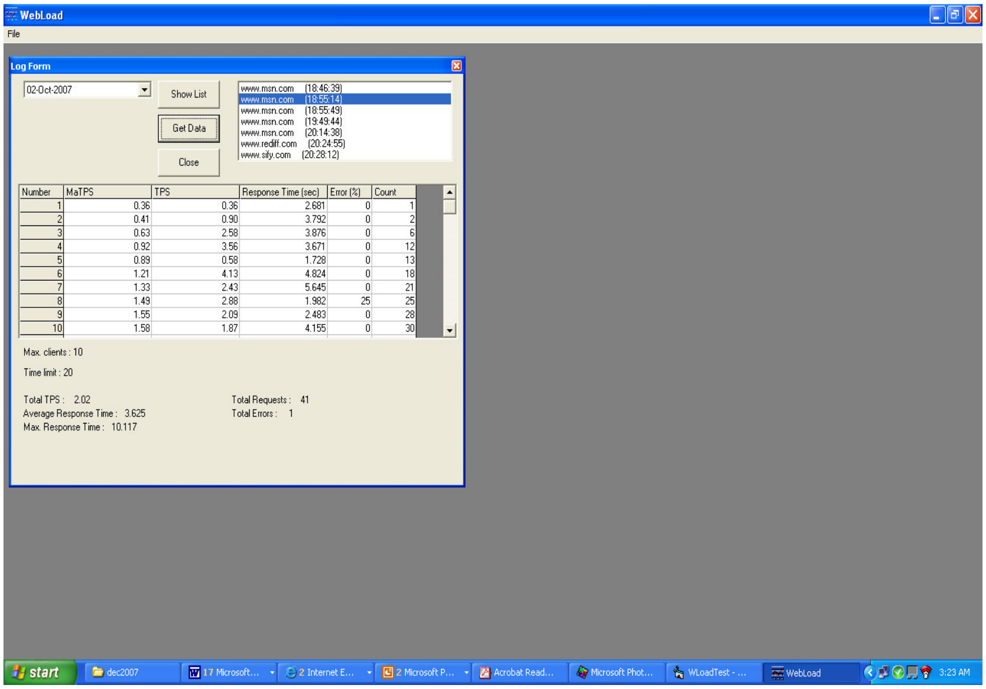
Figure 5.3 : VB code for test load module

However if the connection did not get through the port 80 of the web server specified an error message as follows would be displayed and the details of the test would not be recorded into the database.

Connection was refused on port 80 of server : "the url of web server specified in the parameter form"

5.2.2 LOG INFO

This is module where the information recorded in the database can be retrieved for referral. Once Log Info module is open from the File menu, a combo box with the list of dates tests has been run is shown. The tester then selects the date and clicks on the "Show List" button to retrieve a list of server url along with the test time. Once the web server is selected, the "Get Data" button is clicked. The log information of the test, along with the test parameters is displayed. Figure 5.4 shows the Log form after the "Get Data" button has been clicked.



02-Oct-2007

www.msn.com (18:46:39)
www.msn.com (18:55:14)
www.msn.com (18:55:49)
www.msn.com (19:49:44)
www.msn.com (20:14:38)
www.rediff.com (20:24:55)
www.sify.com (20:28:12)

Number	MaTPS	TPS	Response Time (sec)	Error (%)	Count
1	0.36	0.36	2.681	0	1
2	0.41	0.90	3.792	0	2
3	0.63	2.58	3.876	0	6
4	0.92	3.56	3.671	0	12
5	0.89	0.58	1.728	0	13
6	1.21	4.13	4.824	0	18
7	1.33	2.43	5.645	0	21
8	1.49	2.88	1.982	25	25
9	1.55	2.09	2.483	0	28
10	1.58	1.87	4.155	0	30

Max. clients : 10

Time limit : 20

Total TPS : 2.02
Average Response Time : 3.625
Max. Response Time : 10.117

Total Requests : 41
Total Errors : 1

Figure 5.4 : Log form

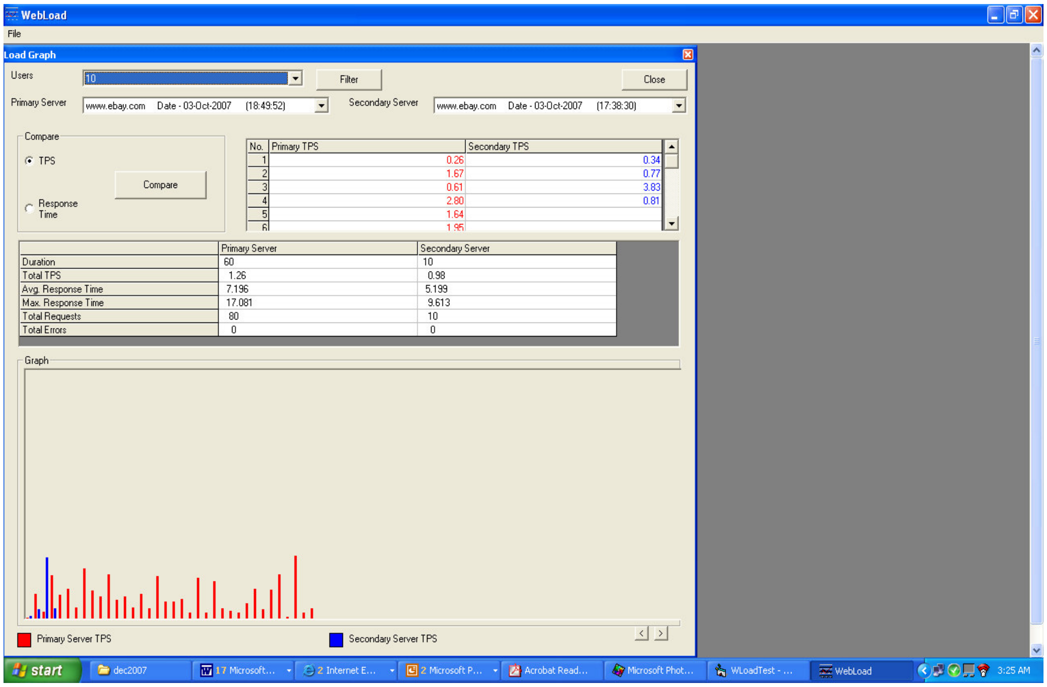
If the “Get Data” button is clicked without selecting the web server then error message as follows is displayed:

“*Please select a server from the list to get the test data*”

The Log Form also has a close button to close the form once used.

5.2.3 ANALYSIS GRAPH

This is the module where comparison of two web servers which were tested is done. The tester has to select the number of maximum virtual users and press the button Filter. Then under primary server and secondary server the list of tests in the database with the specified maximum users are displayed. Then the tester selects the servers to be tested and clicks the Compare button for either TPS(Transactions per second) or response time. The response time for both severs are displayed in a table and also in the form of a graph. Figure 5.5 shows load graph form after clicking the Compare button.



Users: 10

Primary Server: www.ebay.com Date - 03-Oct-2007 (18:49:52)

Secondary Server: www.ebay.com Date - 03-Oct-2007 (17:38:30)

Compare: ● TPS ○ Response Time

No.	Primary TPS	Secondary TPS
1	0.26	0.34
2	1.67	0.77
3	0.61	3.83
4	2.80	0.81
5	1.64	
6	1.95	

	Primary Server	Secondary Server
Duration	60	10
Total TPS	1.26	0.98
Avg. Response Time	7.196	5.199
Max. Response Time	17.081	9.613
Total Requests	80	10
Total Errors	0	0

Figure 5.5 : Load graph form

However if the same test is selected for the primary server and secondary server, an error message as follow is displayed.

"*Cannot compare the same test result. Please select a different test result*"

Once the implementation is complete, the tool is ready for testing. The test procedures and result would be discussed in the following chapter.

CHAPTER 6 : TESTING AND EVALUATION

Testing and evaluation of the load testing tool is an essential part of this dissertation as it allows us to assess whether the solution meets the key objectives as well as the system requirements.

This chapter is to test and validate the test tool which has been developed and implemented. The testing and validation process is to verify and validate the tool and to check if the objectives of the research are met. The testing process also helps to verify if the tool functions according to the specification (Myres, 2004). The researcher adopted a rigorous testing approach in order to verify the functionality of the system, ranging from basic unit testing, followed by the integration testing and finally the system level of testing. The unit testing and integration testing was done through out the development process. Integration testing was very important in the development of the tool as it has to incorporate with an existing system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together. After a rigorous unit and integration testing, the tool was ready for system testing.

Thus, this chapter has two sections which are system testing and validation, and user acceptance test. The system testing results are from the researcher and a few testers who agreed to test the tool before being distributed for further testing and evaluation. Where else the user acceptance test result was collected from fifty E-commerce developers who validated and evaluated the tool.

6.1 SYSTEM TESTING AND VALIDATION

System testing of a software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. According to Myres (2004) system testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system. The system testing was done by the researcher and a few friends.

System testing was done using a live test data. Moreover, system testing must also clearly demonstrate that all stated system performance and operational requirements are satisfied. These include high availability, reliability, security, response time, and scalability. The functionality of the new tool was tested on two different web sites with different load parameters. The following existing E-commerce websites were tested to determine if the tool can be used for load testing and comparison of web servers.

i) www.globat.com

This is a website which offers reliable and affordable Web Hosting to individual and small E-commerce businesses. The reason why this website was chosen was because it is a web hosting company and could be a potential web page to be used by E-commerce developers.

ii) www.maybank2u.com.my

Maybank2u.com is a one-stop financial portal that promises a new, fast, convenient and safe banking environment. The reason why this website was used was simply because it is a popular Malaysian E-commerce web site where lots of business transaction is done.

iii) www.exabytes.com.my

Exabytes is a web hosting company based in Malaysia and provides web hosting packages for E-commerce applications. The reason why this website was chosen was because it could be a potential web host to be used by E-commerce developers.

iv) www.mercumaya.net

Mercumaya.net provides web host service in Malaysia. The reason why this website was chosen was because it could be a potential web host to be used by E-commerce developers.

v) www.hileytech.com

Hileytech.com provides web hosting solutions for E-commerce applications. The reason why this website was chosen was because it is a web hosting company based in Malaysia and could be a potential web host to be used by E-commerce developers.

vi) www.webhosting.com.my

This is the website of EMERGE, a leading web hosting solutions provider. The reason why this website was chosen was because it is a web hosting company based in Malaysia and could be a potential web host to be used by E-commerce developers.

The result of the system testing is displayed in section 6.1.1.

6.1.1 TEST RESULTS

The performance of the load test tool to generate clients, send http request and receive the response, also depends on the capacity of the machine which is hosting the tool and also the configuration of the internet connection. As such it is important to know the configuration of the machine and the internet connection which is used for testing. Due to time constraint, testing was done only using one machine which is the notebook of the researcher with the configuration shown on Table 6.1.

Table 6.1 Configuration of machine used for the testing

Machine Type	Hardware Specification	Software Specification
Notebook Dell Inspiron 6400	Intel(R) Core (TM)2 CPU 1.83 GHz	Microsoft Windows XP Home Edition Version 2002 Service Pack 2
	1.00 GB of RAM , 80GB HD	
	Broadband network with 1Mbps speed	

Test was done with seven different maximum clients value. That is 10, 100, 500, 1000, 1500, 2000 and 2500. The test was done 5 times for each value for 300 seconds. As the new tool can only show one result at a time, the results for the 5 tests has been manually keyed into a spreadsheet file for analysis purpose.

Section 6.1.1.1 to section 6.1.1.6 discusses the result obtained from the test conducted using the parameters mentioned.

6.1.1.1 Globat.com

The test result for total transactions per second (TTPS) and average response time for globat.com is shown in Figure 6.1 and Figure 6.2. However globat.com could only be tested with the maximum of 1000 clients after which there were no more response from the web server of globat.com.

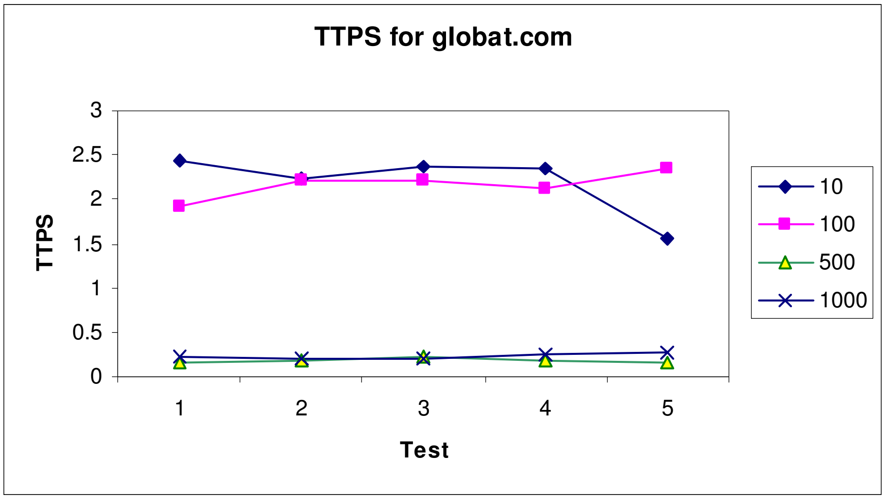


Figure 6.1 : TTPS for globat.com

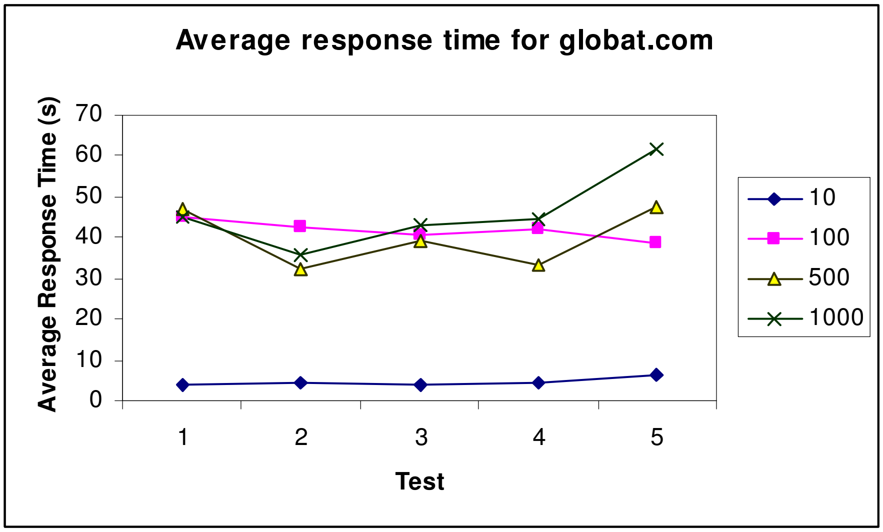


Figure 6.2 : Average response time for globat.com

Table 6.2 shows the average TTPS and Average response time for the 5 tests done on globat.com.

Table 6.2 : Average TTPS and Average response time for globat.com

Max Clients	TTPS	Average response time (Seconds)
10	2.19	4.574
100	2.162	41.7824
500	0.18	39.8572
1000	0.228	46.0344
1500	nil	nil
2000	nil	nil
2500	nil	nil

6.1.1.2 Maybank2u.com.my

The test result for total transactions per second (TTPS) and average response time for maybank2u.com.my is shown in Figure 6.3 and Figure 6.4. However this web site could only be tested with the maximum of 2500 clients after which there were no more response from the web server of maybank2u.com.my

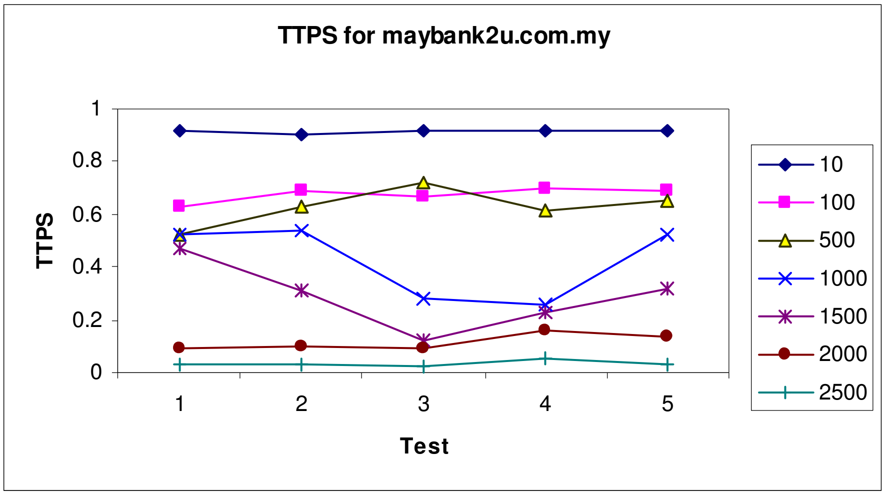


Figure 6.3 : TTPS for maybank2u.com.my

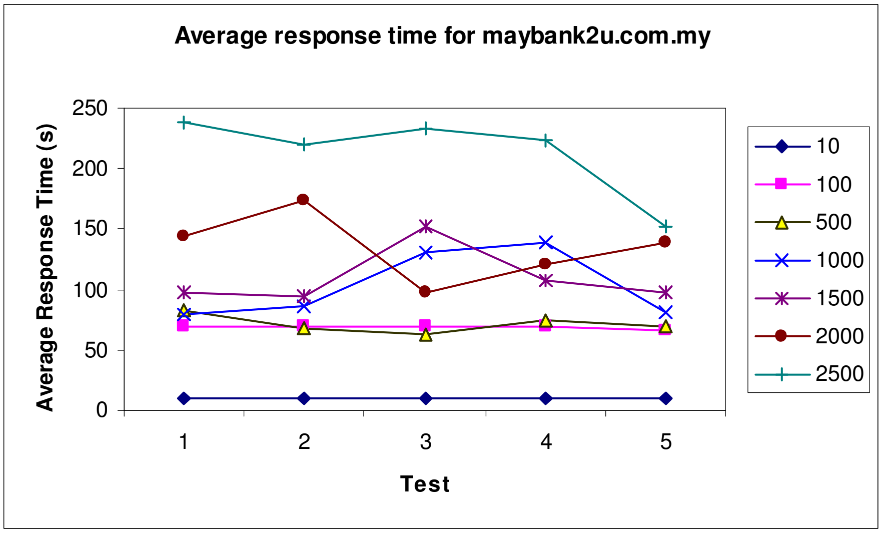


Figure 6.4 : Average response time for maybank2u.com.my

Table 6.3 shows the average TTPS and Average response time for the 5 tests done on maybank2u.com.my.

Table 6.3 : Average TTPS and Average response time for maybank2u.com.my

Max Clients	Total TPS	Average response time (Seconds)
10	0.916	10.191
100	0.676	68.9254
500	0.626	71.6668
1000	0.424	103.2534
1500	0.29	109.6992
2000	0.116	135.1652
2500	0.032	213.7552

6.1.1.3 Exabytes.com.my

The test result for total transactions per second(TTPS) and average response time for exabytes.com.my is shown in Figure 6.5 and Figure 6.6. However this web site could only be tested with the maximum of 1500 clients after which there were no more response from the web server of exabytes.com.my

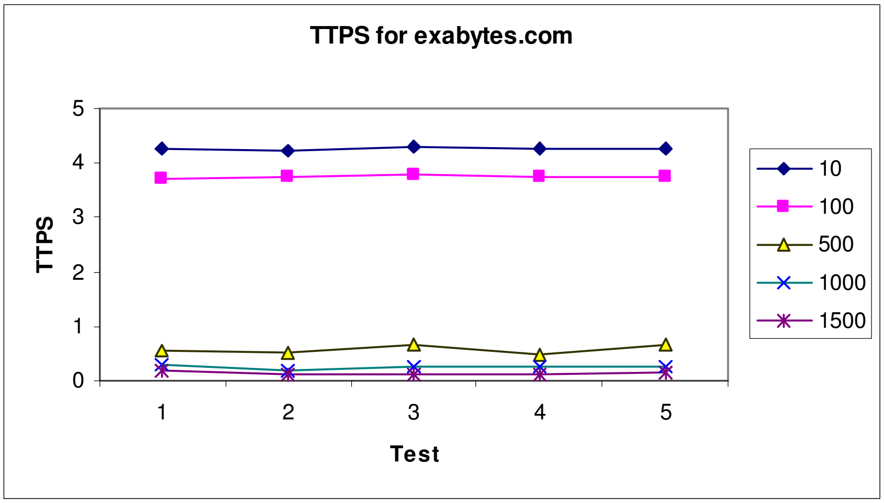


Figure 6.5 : TTPS for exabytes.com.my

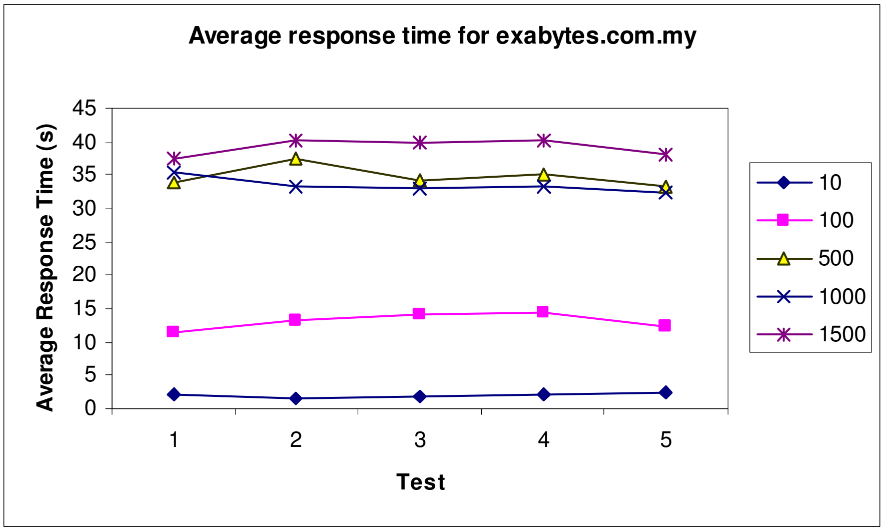


Figure 6.6 : Average Response Time for exabytes.com.my

Table 6.4 shows the average TTPS and Average response time for the 5 tests done on exabytes.com.my.

Table 6.4 : Average TTPS and Average response time for exabytes.com.my

Max Clients	Total TPS	Average response time (Seconds)
10	4.256	1.9514
100	3.756	13.113
500	0.578	34.7728
1000	0.248	33.4576
1500	0.13	39.1774
2000	nil	nil
2500	nil	nil

6.1.1.4 Mercumaya.net

The test result for total transactions per second(TTPS) and average response time for mercumaya.net is shown in Figure 6.5 and Figure 6.6. However this web site could only be tested with the maximum of 1500 clients after which there were no more response from the web server of mercumaya.net

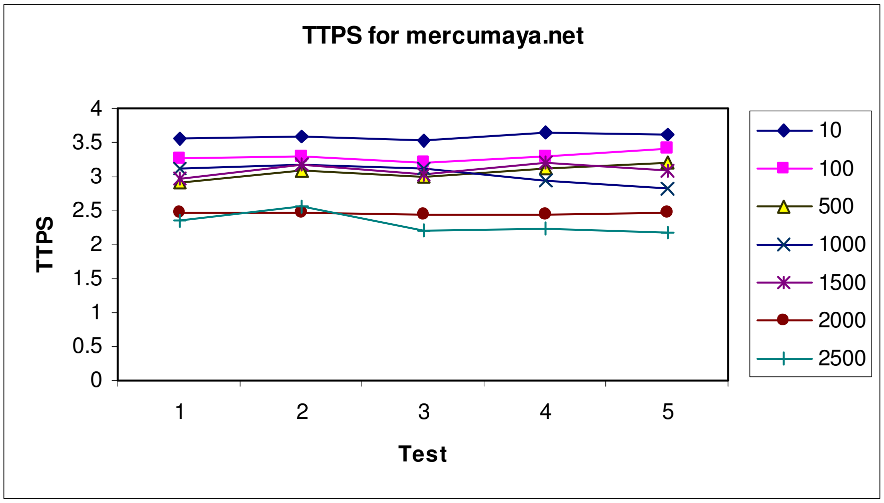


Figure 6.7 : TTPS for mercumaya.net

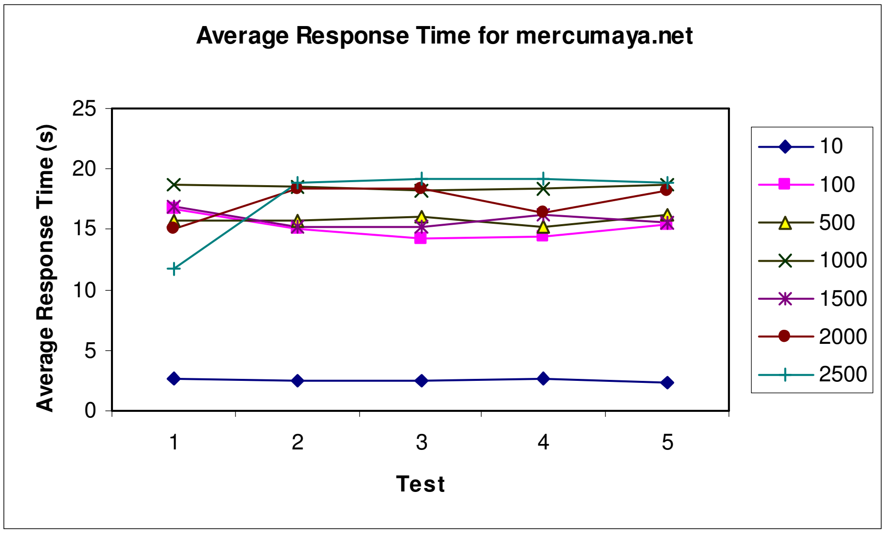


Figure 6.8 : Average Response Time for mercumaya.net

Table 6.5 shows the average TTPS and Average response time for the 5 tests done on mercumaya.net

Table 6.5 : Average TTPS and Average response time for mercumaya.net

Max Clients	Total TPS	Average response time (Seconds)
10	3.59	2.5366
100	3.29	15.118
500	3.066	15.8418
1000	3.038	18.4898
1500	3.094	15.8708
2000	2.456	17.2442
2500	2.306	17.5704

6.1.1.5 Hileytech.com

The test result for total transactions per second(TTPS) and average response time for hileytech.com is shown in Figure 6.9 and Figure 6.10. However this web site could only be tested with the maximum of 500 clients after which there were no more response from the web server of hileytech.com

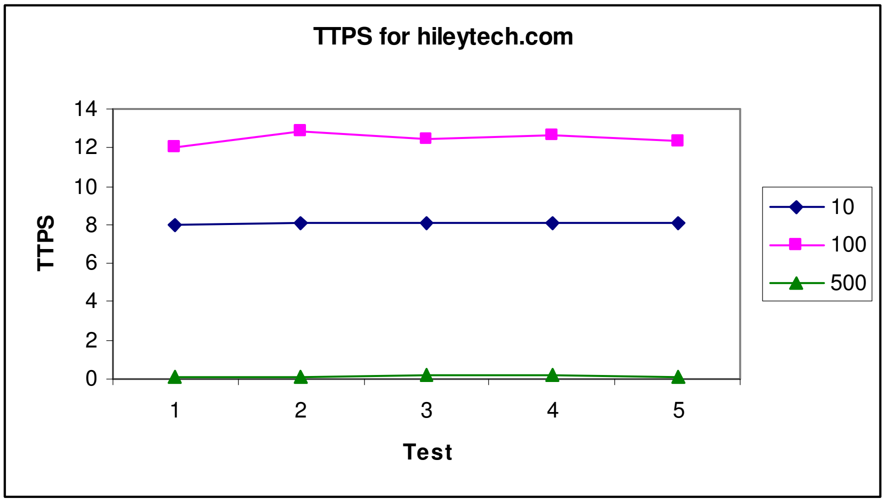


Figure 6.9 : TTPS for hileytech.com

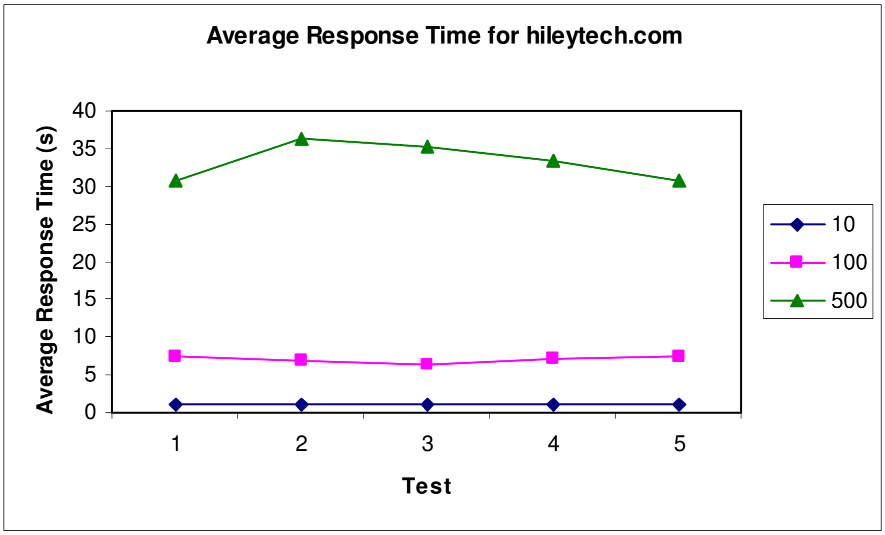


Figure 6.10 : Average Response Time for hileytech.com

Table 6.6 shows the average TTPS and Average response time for the 5 tests done on mercumaya.net

Table 6.6 : Average TTPS and Average response time for hileytech.com

Max Clients	Total TPS	Average response time (Seconds)
10	8.074	1.152
100	12.464	7.0204
500	0.154	33.2564
1000	nil	nil
1500	nil	nil
2000	nil	nil
2500	nil	Nil

6.1.1.6 Webhosting.com.my

The test result for total transactions per second (TTPS) and average response time for webhosting.com.my is shown in Figure 6.11 and Figure 6.12. However this web site could only be tested with the maximum of 2500 clients after which there were no more response from the web server of webhosting.com.my

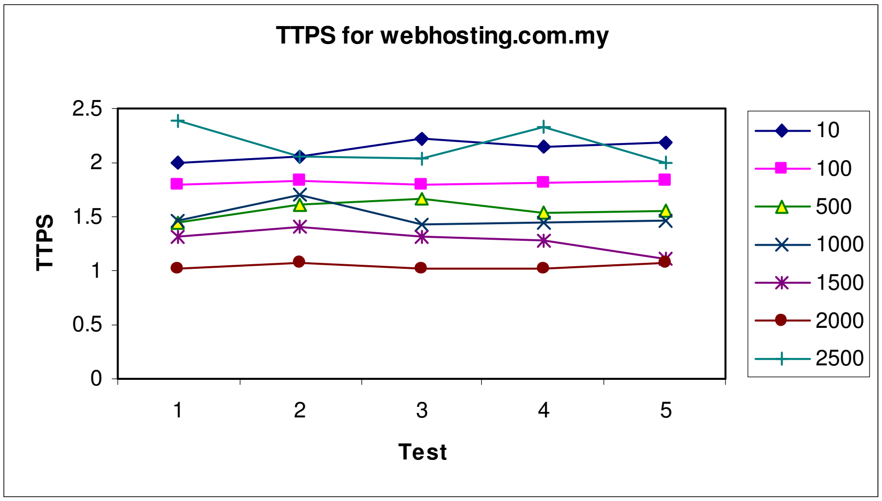


Figure 6.11 : TTPS for webhosting.com.my

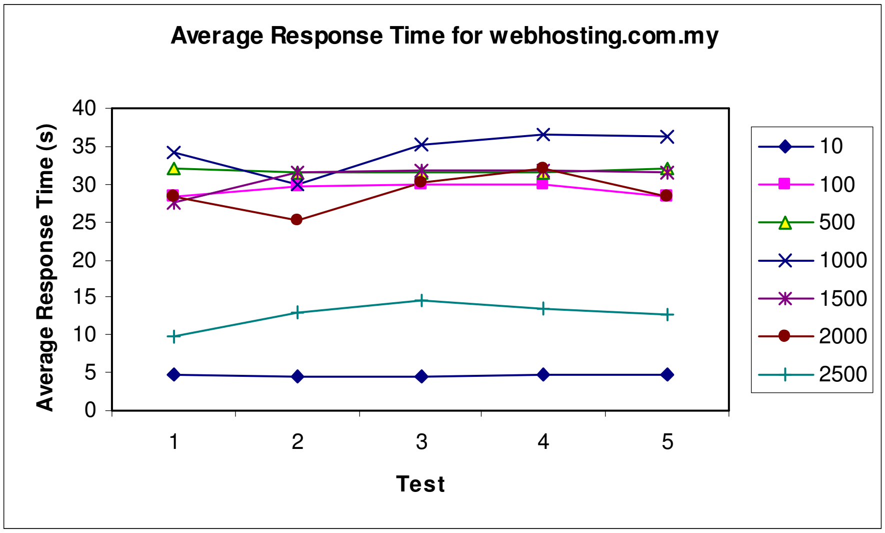


Figure 6.12 : Average Response Time for webhosting.com.my

Table 6.7 shows the average TTPS and Average response time for the 5 tests done on webhosting.com.my

Table 6.7 : Average TTPS and Average response time for webhosting.com.my

Max Clients	Total TPS	Average response time (Seconds)
10	2.12	4.6814
100	1.82	29.2146
500	1.562	31.7398
1000	1.498	34.4486
1500	1.288	30.783
2000	1.04	28.9132
2500	2.16	12.654

6.1.1.7 COMPARISON OF ALL THE WEBSITES TESTED

The total transactions per second (TTPS) and average response time for all the web sites tested can be compared using the results from section 6.1.1.1 to section 6.1.1.6. Figure 6.13 shows the comparison of TTPS for the web sites and Figure 6.14 shows the average response time for the web sites.

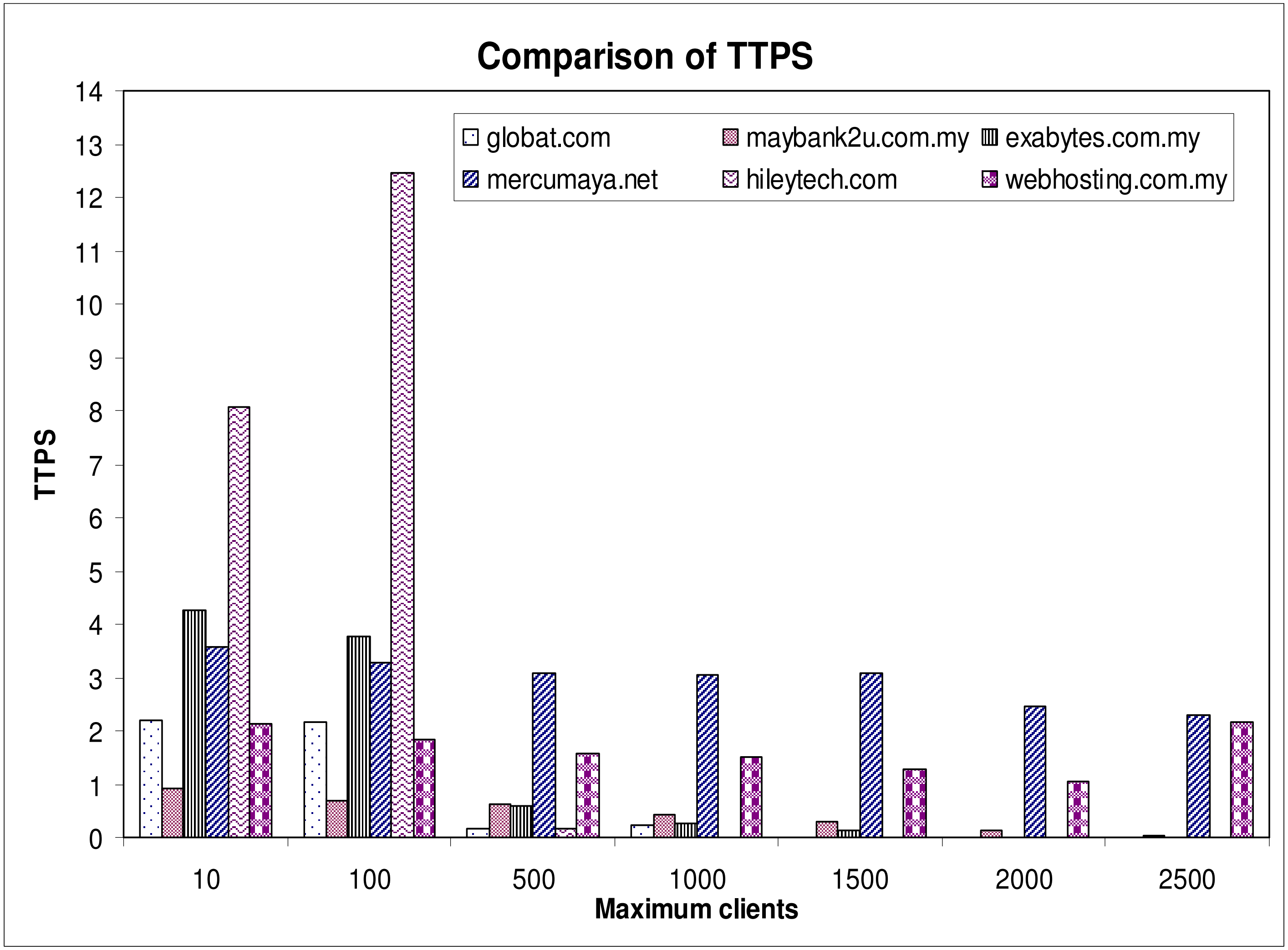


Figure 6.13 : Comparison of TTPS

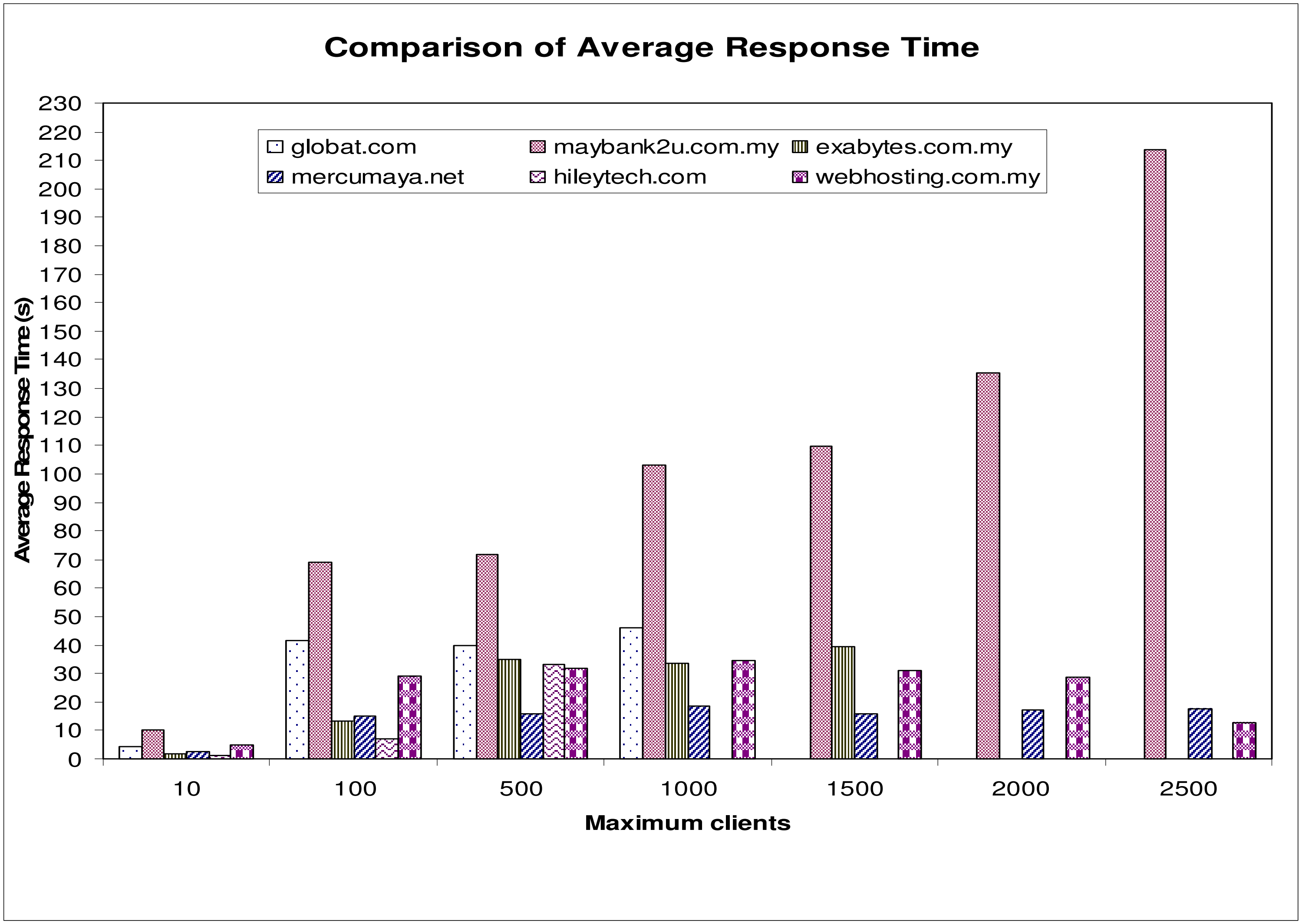


Figure 6.14 : Comparison of Average Response Time

6.1.2 CONCLUSION

After conducting the load test on the web sites using the new load test tool, the performance of the web sites (web server) under various load has been determined. Based on the results of the tests, for the first 100 clients hileytech.com’s web server has the most transactions per second but it can only take a maximum load of 500 clients at a time.

Maybank2u.com.my, mercumaya and webhosting.com.my can handle up to maximum 2500 clients. Among these web sites, mercumaya.net has the best capacity to handle http requests. This is because the response time is very consistent even with the increase of load.

In conclusion it can be said that the best web server which can be chosen is mercumaya.net as it can handle the increase of load and yet response fast to the request.

6.2 USER ACCEPTANCE

The research needs to have real users in validating the application, in term of usefulness, ease of use, and functionality in different and unknown environments.

As mentioned in the research objectives, E-commerce developers are the targeted user for the validation process. The reason E-commerce developers are targeted is because the research is focused on performance testing tool for E-commerce applications. As such the E-commerce developers would be the potential users for this tool.

The tool was sent out to E-commerce application developers to test the web server they were considering to host their application or web servers already hosting their applications. They were also allowed to test other web servers hosting E-commerce applications available on the internet. Along with the tool, installation instruction and test procedure (Appendix A) and a feedback form (Appendix B) were sent out for the developers to give their feedback on the tool. First an email was sent to a few software developing companies and the researcher managed to get fifty E-Commerce application developers responding to the E-mail. As such the following evaluation and feed back is based on the fifty E -Commerce developers.

The information's that were obtained are:

i. platform environment, such as the hardware specification as well as the operating system detail

ii. the internet connectivity type, such as dial up, lease line, etc and the ISP

iii. test duration

iv. web server url

v. the results

vi. comment or suggestion for this tool

6.2.1 USER ACCEPTANCE TEST

The application was distributed to the E-Commerce developers and the users had the opportunity of using the application for two weeks before providing feedback to the research. A total number of 50 developers used the application and they found it to be useful. After completing the tests, the testers sent back the filled feedback form to the researcher. Due to space constrain only two samples of the feedback form has been enclosed in this dissertation. Please refer to the Appendix B for the samples.

6.2.1.1 STATISTIC DESCRIPTION OF THE RESULT

The result from the 50 developers were collected and analyzed. Statistic descriptions of the results are discussed in this chapter.

Following are the statistic description derived from the result obtained from the testers.

i) Application usability

The users were asked to comment yes or no on the user friendliness of the tool. 40 users found it to be user friendly and 10 of them found it to be not user friendly. Figure 6.15 shows a graph representing the users answer on the user friendliness of the tool.

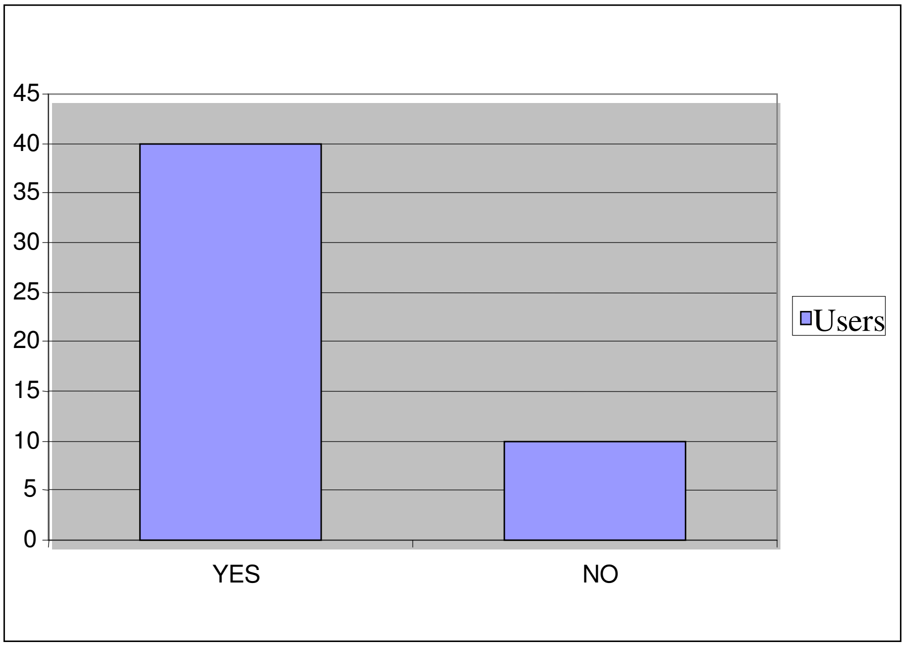


Figure 6.15 : Users answer on the user friendliness of the tool

ii) Effectiveness of the tool

To conclude on the effectiveness of the tool the users were asked to answer yes or no for the following question :

“Is the tool useful in selecting a web server to host your E-commerce application”

Figure 6.16 shows the answer of the users for the above question. 30 out of the 50 users that is, 60 percentage of the users said they will use it to select the web server to host their application. The remaining 20 said they will only use the tool if there were some more features added to it.

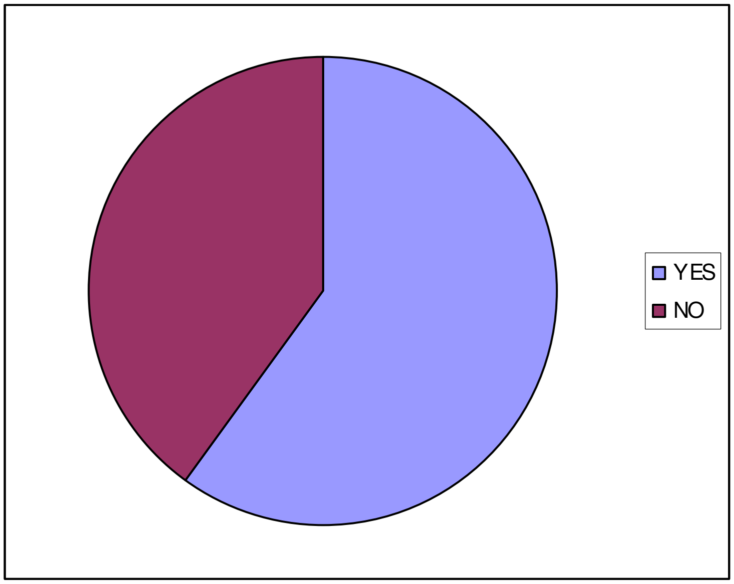


Figure 6.16 : Users answer on the effectiveness of the tool

iii) Suggestions from the users to improve

Apart from the above two questions, the users also gave their suggestion to improve the tool. Below are some of the suggestion which were picked to describe their suggestion which has helped the researcher to plan on the future framework.

Table 6.8 : Users suggestion to improve the tool

Users Name	Suggestion
Veni	The tool is too simple to judge the capacity of the server. Please add more features
Shan	Give more choices for graphs
Latif	A playback tool would be more accurate to test the server with intended scenario

6.3 OBJECTIVES AND RESULT

In this section, we will assess whether the solution has met the key objectives of the research which was determined in the beginning of the research.

1. **Objective :** To identify and present the details of the factors influencing the performance of an E-commerce application as a roadmap to understand the big picture of performance testing for E-commerce applications.

   Result and conclusion : the above objective was achieved in chapter 2 which is the literature review. Factors which determine the performance of E-commerce applications has been recognized based on previous researches.

2. **Objective :** Develop a prototype performance testing tool for E-commerce applications focusing on real time load testing of web servers hosting E-commerce applications. The tool should enable E-Commerce developers to test and compare the performance of web servers before selecting a web host to host their E-commerce applications and also to test the web servers currently hosting their E-commerce applications.

   Result and conclusion : the above objective was done by collecting the tools requirements and specifications and also designing and developing the tool as per the requirements specified. All this information has been documented from chapter 3 to chapter 5. Upon completion of the prototype tool, it was distributed to 50 E-commerce developers and their feedback on the effectiveness of the tool was collected. As sixty percent of the users find this tool would be effective in selecting web server to host their E-commerce application, the research has manage to reach its objective mentioned above. However the comparison technique need to be improvised as currently the comparison is based only on one test result for each web server compared. The prototype tool needs more features to enable a valid comparison of web servers. The features would be discussed in the following chapter.

CHAPTER 7 : CONCLUSION AND FUTURE FRAMEWORK

7.1 CONTRIBUTIONS

This research has contributed to the field of performance testing a tool to load test and compare web servers hosting E-commerce applications.

The tool enables E-commerce developers to do a simple load test on the web servers they are considering to host their application before the deployment of their applications. The load test tool would be able to let them know how their application would perform under the load of expected users at a time.

The importance of web server testing before deployment has been clearly explained in this research, as the scope of performance testing has been narrowed down to load testing of web servers hosting E-Commerce application.

The load test tool is very important in guiding the E-commerce developers to produce an E-commerce application of good quality. The tools ability to compare web servers would enable the developers to choose the most suitable web server to host their application to ensure high performance of their E-commerce applications.

As a conclusion, the tool has contributed in ensuring better performance of E-commerce applications after deployment.

7.2 CRITICAL EVALUATION OF METHODS

In this dissertation we have come up with a prototype tool which can be used by E-commerce developers to select the best web server to suit their application. In the evaluation method, most of the testers did a real time test on existing web server url using the new tool. Even though the sixty percent of the users said they will use the tool, the results collected from the evaluation is not suitable to conclude the efficiency of the tool to select the best web server to host an E-Commerce applications. However the tool is suitable to monitor the web server after the deployment of the E-Commerce applications. Continuous monitoring of the E-commerce applications is very important for businesses to retain its customers.

7.3 FUTURE FRAMEWORK

As this prototype tool to load test the web servers hosting E-Commerce applications for http request response, there are many components which has to be added or improvised on this tool.

In this tool the type of request generated is categorized as one that is http request. However in real life the request for the web server is in coherent with the E-commerce application and its functions. For this purpose a scenario creating tool should be incorporated into this tool to give a more accurate result based on the type of users request. For example, login request or buy request. The response time for the request in the current tool is also not considering the type of request, or the think time involved for a particular task. As mentioned by Sopitkamol (2002) all factors influencing the response time of the web server has to be considered during the calculation of the response time.

Another feature which can be added for this tool is built-in report generator so that reports can be generated as HTML files, word and spreadsheet documents.

Another important feature needed by this tool is to view many test results at a time compared to, only being able to view one log information at a time. This feature is very important to make analysis to compare the performance two or more web server before the highest performing web server is chosen. With all these features incorporated to the current tool, the tool would contribute to the improvement of the performance of E-commerce applications.

APPENDIX A

INSTALLATION AND TEST PROCEDURES

Following are the instruction which is to be followed to install and run the test.

1) INSTALLATION

i) Download the attachment WLoadTest in the email and save it into the computer as WLoadTest.

ii) Open the WLoadTest folder and open the Package folder inside this folder.

iii) Open the setup file and follow the instructions to install the tool into your computer

iv) Once installation is complete the tool is ready to use as long as there is the internet connection. However for the log retrieval and analysis modules there is no need for internet connection.

2) TEST PROCEDURES

First the WebLoad tool is opened. A main form which is the WebLoad form would appear. The WebLoad form has a menu called File. Figure 1 shows the WebLoad form.

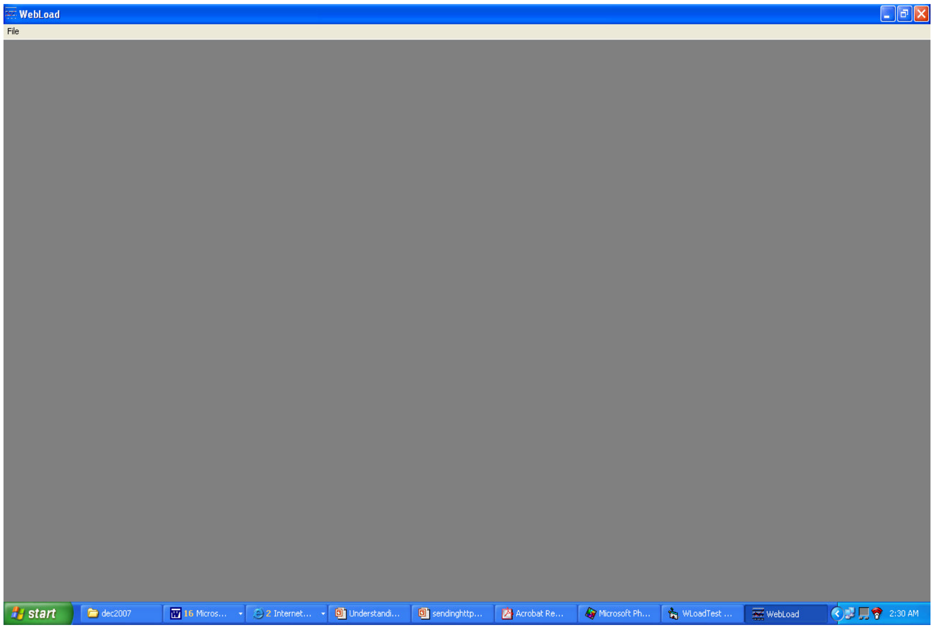


Figure 1 : WebLoad form

Once the file is clicked the list of the three main modules of the test tool are displayed. The main modules are:

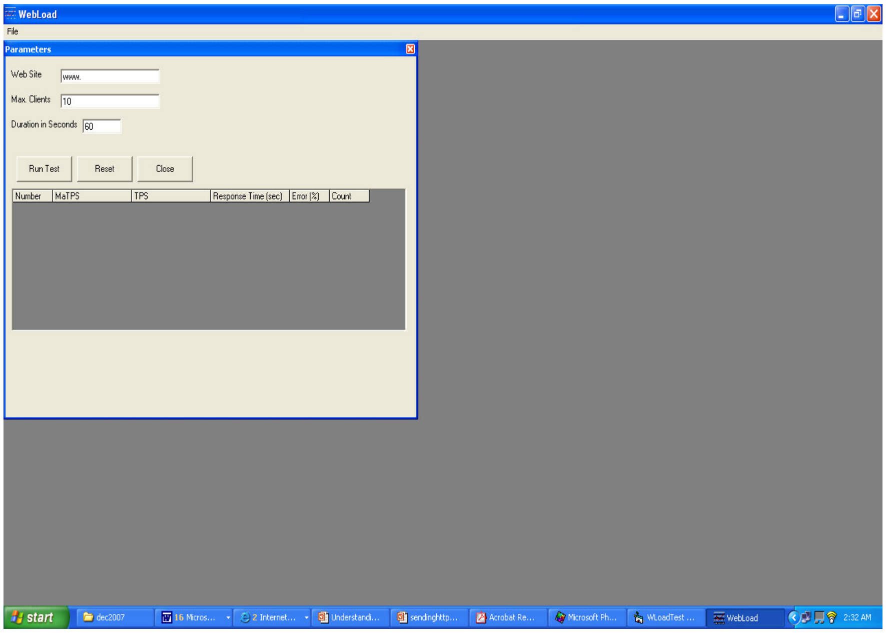
4) Test load

5) Log Info

6) Analysis Graph

1) TEST LOAD

This is the module where the user enters the test parameters for the test. Figure 2 shows the parameter form where the test specifications are entered.



Web Site www.

Max. Clients 10

Duration in Seconds 60

Number	MaTPS	TPS	Response Time (sec)	Error (%)	Count

Figure 2 : Parameters form

The parameters entered are:

iv) Web site

A valid url must be entered in the text box. If the url is not valid the following error message would be displayed:

"*Invalid web server specified. Destination cannot be reached. Please enter a valid address*"

v) Max Clients

Maximum amount of virtual users are determined here. The test tool has the capacity to generate up to 5000 virtual users at a time. The actual load which can be achieved is highly dependent on the network infrastructure, the server/client hardware, the file sizes and the e-commerce application itself.

vi) Duration in seconds

The duration has to be adequate to the clients entered for the test. The following table gives guidance to the corresponding duration for the amount of users.

Table 1 : Recommended duration for the amount of users

Amount of users	**Recommended Duration**
1-500	More then 60 seconds
500 – 1000	More then 180 seconds
1001 – 2000	More then 240 seconds
2000 – 3000	More then 300 seconds
3001 – 4000	More then 360 seconds
4001 - 5000	More then 420 seconds

While the test is running, a progress reporting tool would pop up to show the remaining time for the load test. Once the result is completed the tester will receive a message stating “test complete” and tester has to close the progress reporting tool. Once the progress reporting tool is closed the GetResult button becomes visible in the parameter form.

Once the GetResult button is clicked the result of the test is then displayed on the result panel which is a sub form.

However if the connection did not get through the port 80 of the web server specified an error message as follows would displayed and the details of the test would not be recorded into the database.

“Connection was refused on port 80 of server : “the url of web server specified in the parameter form” ”

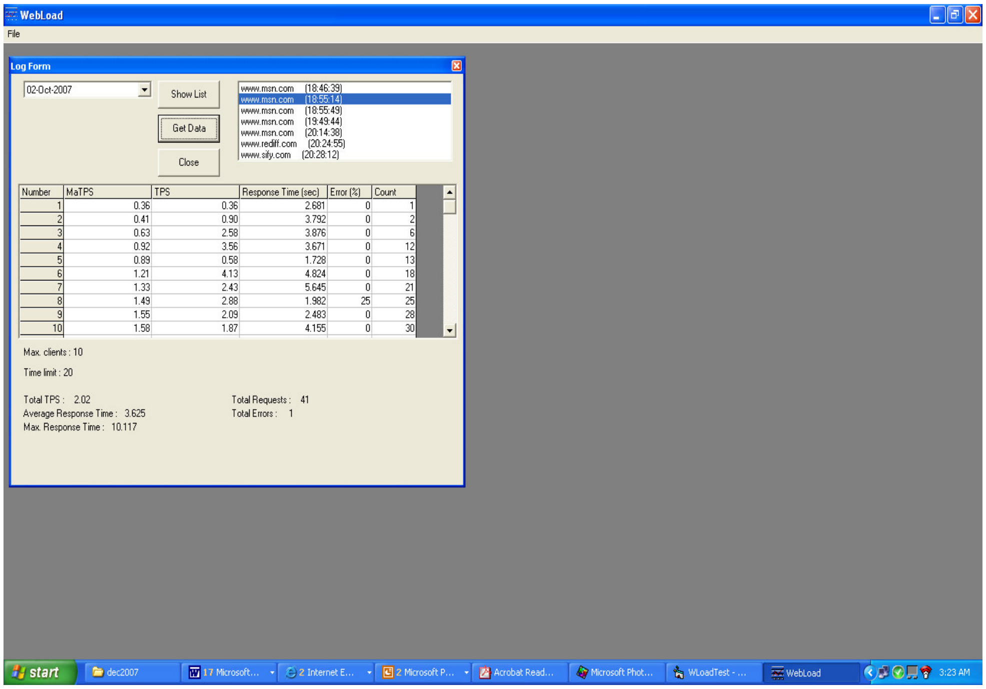
2) LOG INFO

This is module where the information recorded in the database can be retrieved for referral. Once Log Info module is open from the File menu, a combo box with the list of dates tests has been run is shown. The tester then selects the date and clicks on the ShowList button to retrieve a list of server url along with the test time. Once the web server is selected, the GetData button is clicked. The log information of the test, along with the test parameters are displayed. Figure 3 shows the Log form after the GetData button has been clicked.

If the GetData button is clicked without selecting the web server then error message as follows is displayed:

"*Please select a server from the list to get the test data*"

The Log Form also has a close button to close the form once used.



02-Oct-2007

www.msn.com (18:46:39)
www.msn.com (18:55:14)
www.msn.com (18:55:49)
www.msn.com (19:49:44)
www.msn.com (20:14:38)
www.rediff.com (20:24:55)
www.sify.com (20:28:12)

Number	MaTPS	TPS	Response Time (sec)	Error (%)	Count
1	0.36	0.36	2.681	0	1
2	0.41	0.90	3.792	0	2
3	0.63	2.58	3.876	0	6
4	0.92	3.56	3.671	0	12
5	0.89	0.58	1.728	0	13
6	1.21	4.13	4.824	0	18
7	1.33	2.43	5.645	0	21
8	1.49	2.88	1.982	25	25
9	1.55	2.09	2.483	0	28
10	1.58	1.87	4.155	0	30

Max. clients : 10

Time limit : 20

Total TPS : 2.02
Average Response Time : 3.625
Max. Response Time : 10.117

Total Requests : 41
Total Errors : 1

Figure 3 : Log form

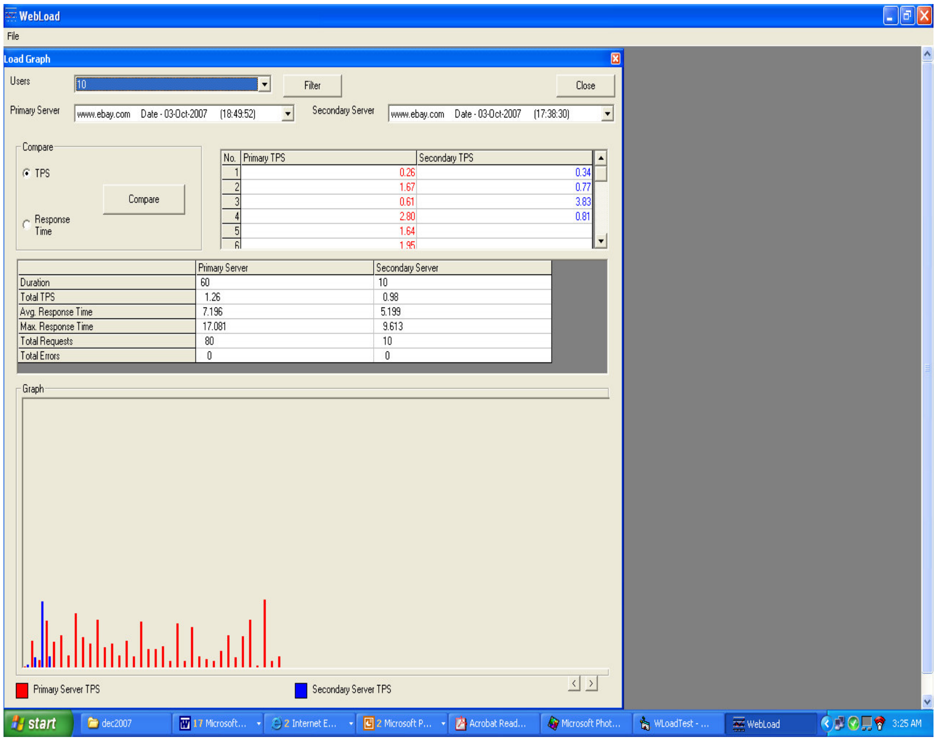
3) ANALYSIS GRAPH

This is the module where comparison of two web servers which were tested is done. The tester has to select the number of maximum virtual users and press the button Filter. Then under primary server and secondary server the list of tests in the database with the specified maximum users are displayed. Then the tester selects the servers to be tested and clicks the Compare button for either TPS(Transactions per second) or response time. The response time for both severs are displayed in a table and also in the form of a graph. Figure 4 shows load graph form after clicking the Compare button.

However if the same server is selected for the test, an error message as follow is displayed.

"*Cannot compare the same test result. Please select a different test result*"

The Log graph form also has a close button to close the form once used.



Users: 10

Primary Server: www.ebay.com Date - 03-Oct-2007 (18:49:52)

Secondary Server: www.ebay.com Date - 03-Oct-2007 (17:38:30)

Compare: ● TPS ○ Response Time

No.	Primary TPS	Secondary TPS
1	0.26	0.34
2	1.67	0.77
3	0.61	3.83
4	2.80	0.81
5	1.64	
6	1.95	

	Primary Server	Secondary Server
Duration	60	10
Total TPS	1.26	0.98
Avg. Response Time	7.196	5.199
Max. Response Time	17.081	9.613
Total Requests	80	10
Total Errors	0	0

Figure 4 : Load graph form

APPENDIX B

FEEDBACK FORM

TESTERS DETAILS

NAME	
DATE	
COUNTRY	
LOCATION	
DESIGNATION	

HARDWARE SPECIFICATION

BRAND	
MEMORY	
PROCESSOR	

SOFTWARE SPECIFICATION

OPERATING SYSTEM	
BROWSER	

INTERNET SPECIFICATION

CONNECTION TYPE	
BANDWITH	
CONDITIONS	

TEST RESULT

PRIMARY SERVER NAME	
PRIMARY SERVER URL	

SECONDARY SERVER NAME	
SECONDARY SERVER URL	

MAXIMUM USERS	

PARAMETER	TEST RESULT	
	PRIMARY SERVER	SECONDARY SERVER
DURATION (SECONDS)		
AVERAGE RESPONSE TIME(SECONDS)		
MAXIMUM RESPONSE TIME(SECONDS)		
TOTAL TRANSACTIONS PER SECOND		
TOTAL REQUEST		
TOTAL ERROR		

OTHER COMMENTS

USER FRIENDLY (YES/NO)	
IS THE TOOL USEFUL IN SELECTING A WEB SERVER TO HOST YOUR E-COMMERCE APPLICATION (YES/NO)	
COMMENTS,PROBLEMS AND SUGGESTION	

FEEDBACK FORM

TESTERS DETAILS

NAME	SHANTINI
DATE	10/10/07
COUNTRY	MALAYSIA
LOCATION	OFFICE
DESIGNATION	SOFTWARE DEVELOPER

HARDWARE SPECIFICATION

BRAND	DELL
MEMORY	128 MB
PROCESSOR	PENTIUM III

SOFTWARE SPECIFICATION

OPERATING SYSTEM	WINDOWS XP
BROWSER	INTERNET EXPLORER 6.0

INTERNET SPECIFICATION

CONNECTION TYPE	BROADBAND
BANDWITH	1.0 MB
CONDITIONS	GOOD

TEST RESULT

PRIMARY SERVER NAME	
PRIMARY SERVER URL	HTTP://WWW.EBAY.COM

SECONDARY SERVER NAME	
SECONDARY SERVER URL	HTTP://WWW.MAYBANK2U.COM.MY

MAXIMUM USERS	1000

PARAMETER	TEST RESULT	
	PRIMARY SERVER	SECONDARY SERVER
DURATION (SECONDS)	300	300
AVERAGE RESPONSE TIME(SECONDS)	82.237	78.432
MAXIMUM RESPONSE TIME(SECONDS)	278.613	262.527
TOTAL TRANSACTIONS PER SECOND	0.48	0.52
TOTAL REQUEST	145	156
TOTAL ERROR	0	0

OTHER COMMENTS

USER FRIENDLY (YES/NO)	YES
IS THE TOOL USEFUL IN SELECTING A WEB SERVER TO HOST YOUR E-COMMERCE APPLICATION (YES/NO)	YES
COMMENTS,PROBLEMS AND SUGGESTION	COULD IMRPOVE ON THE REPORTING AND GRAPGHS.AS IT IS ONLY A PROTOTYPE, THERE MORE FEATURES SUCH SCENARIO CREATING WHICH SHOULD BE ADDED TO THIS TOOL

FEEDBACK FORM

TESTERS DETAILS

NAME	REX
DATE	20/10/07
COUNTRY	MALAYSIA
LOCATION	OFFICE
DESIGNATION	SOFTWARE DEVELOPER

HARDWARE SPECIFICATION

BRAND	HP
MEMORY	128 MB
PROCESSOR	PENTIUM III

SOFTWARE SPECIFICATION

OPERATING SYSTEM	WINDOWS XP
BROWSER	INTERNET EXPLORER 6.0

INTERNET SPECIFICATION

CONNECTION TYPE	BROADBAND
BANDWITH	1.0 MB
CONDITIONS	GOOD

TEST RESULT

PRIMARY SERVER NAME	
PRIMARY SERVER URL	http:// www.globat.com

SECONDARY SERVER NAME	
SECONDARY SERVER URL	http:// www-thehostingchart.com

MAXIMUM USERS	100

PARAMETER	TEST RESULT	
	PRIMARY SERVER	SECONDARY SERVER
DURATION (SECONDS)	300	300
AVERAGE RESPONSE TIME(SECONDS)	46.299	42.998
MAXIMUM RESPONSE TIME(SECONDS)	158.392	125.527
TOTAL TRANSACTIONS PER SECOND	2.55	1.33
TOTAL REQUEST	765	401
TOTAL ERROR	0	0

OTHER COMMENTS

USER FRIENDLY (YES/NO)	YES
IS THE TOOL USEFUL IN SELECTING A WEB SERVER TO HOST YOUR E-COMMERCE APPLICATION (YES/NO)	NO
COMMENTS,PROBLEMS AND SUGGESTION	

REFERENCES

Arlitt, M., Krishnamurthy, D. and Rolia, J., 2001.Characterizing the scalability of a large web-based shopping system, *ACM Transactions on Internet Technology (TOIT),* Volume 1, Issue 1, August 2001. pp. 44 - 69. ACM : New York.

Bacigalupo, A. et al., 2005. An Investigation into the Application of Different Performance Prediction Methods to Distributed Enterprise Applications. *The Journal of Supercomputing,* [online] Issue 2, November. Abstract from ACM Database. Available at: http://portal.acm.org/citation.cfm?id=1084753.1084764&coll=GUIDE&dl=GUIDE [accessed on 23 May 2007].

Bach, J., 1999. Risk-Based Testing. *Software Testing and Quality Engineering Magazine*, Vol. 1, No. 6. STQE Publiching: US.

Barber, S., 2006. *Beyond Performance Testing*. PerfTestPlus. Available at: http://www.perftestplus.com/resources [accessed on 20 May 2007].

Barker, R., 2000. *Optimizing Internet Bandwidth,* Performance Computing Magazine. January 2000. Miller Freeman Inc. : US.

Booch, G., and Rumbaugh, J., 1995. *Unified Method for Object-Oriented Development*, Documentation Set v0.8. Rational Software Corporation: Santa Clara, CA.

British Standards Institute. Available at: http://www.bsi.org.uk. [accessed on 23 September 2007].

Chaudhury, A. et al., 2001. *E-Business and E-Commerce Infrastructure*. 1st ed. United States of America: McGraw-Hill Higher Education.

Crosby, B., 1979. *Quality is free: the art of making quality certain: How to Manage Quality-So That It Becomes a Source of Profit for Your Business.* United States of America: McGraw-Hill Higher Education.

Davison, B., 2001. *HTTP Simulator Validation Using Real Measurements: A Case Study.* Department of Computer Science Rutgers, The State University of New Jersey.

Dibachi, R., 1999. Testing Electronic Commerce. *Software Testing & Quality Engineering Magazine*. STQE Publiching: US.

Di Lucca, A., et al., 2002. Testing Web Applications, *Software Maintenance 2002 Proceedings, International Conference.* Oct. pp. 310 – 319. Abstract from IEEE Database. Available at: http://ieeexplore.ieee.org/servlet/opac?punumber=8357 [accessed on 26 May 2007].

E-Commerce Website Development. Available at: http://www.arnima.com/webdesign/Web-ECommerce-Design.asp [accessed on 19 April 2007].

Elnikety, S., et al., 2004. A Method for Transparent Admission Control and Request Scheduling in E-Commerce Web Sites, *Proceedings of the 13th international conference on World Wide Web*. ACM : New York.

Garvin, D., 1987. Competing on the eight dimensions of quality, *Harvard Business Review November,* pp 101-109.

Hayes, R. and Savoia, A., 2000. How to Load Test e-Commerce Applications. *26th International Computer Measurement Group Conference.* Orlando, Florida, USA December 10-15 2000. Computer Measurement Group : New Jersey.

Hetzel, B. 1993. The *Complete Guide to Software Testing*. 2nd Edition. New York: John Wiley & Son.

Hines, B., Alcott, T., Barcia, R., Botzum, K., 2004. *IBM WebSphere session management technical article*. Prentice Hall. Available at: http://www.phptr.com/articles/article.asp?p=332851&seqNum=4&rl=1 [accessed 10 June 2006]

James W. J. Xue, Bacigalupo, A., Jarvis, A., and Nudd, R.,2006. *Performance Prediction of Distributed Enterprise Applications with Session Persistence*. High Performance Systems Group, Department of Computer Science, University of Warwick, Coventry, UK Available at: http://www.dcs.warwick.ac.uk/~daveb/research/ukpew06.pdf [accessed 11 July 2007]

Krushten, P., 1998.*The Rational Unified Process*. Addison-Wesley: Reading, MA.

Lee, E., 2007. *Malaysia Handheld 2007–2011 Forecast and Analysis*. IDC: Malaysia. Available at: http://www.idc.com.my/ [accessed 11 December 2007]

Marshall, B. 2006. *How e-commerce works.* Available at: http://communication.howstuffworks.com/ecommerce3.htm [accessed on 23 May 2007].

McCall, J., Richards, P., and Walters, G., 1977. Factors in Software Quality. *NTIS AD-A049 014, 015, 055*. November 1977. ACM : New York.

Meier, J., Vasireddy, S., Babbar, A. and Mackman, A., 2004. *Improving .NET Application Performance and Scalability*.United States: Microsoft Corporation. Available at: http://www.microsoft.com/downloads/details.aspx?FamilyId=8A2E454D-F30E-4E72-B531-75384A0F1C47&displaylang=en [accessed 11 July 2007]

Menasec, D., et al., 1999. A methodology for workload characterization of E-commerce sites, *Proceedings of the 1st ACM conference on Electronic Commerce,* [online] Nov, Denver, Colorado, United States of America, pp. 119-128. Abstract from ACM Database. Available at: http://portal.acm.org [accessed on 23 May 2007].

Menasec, D., et al., 2000. Resource management policies for E-commerce servers, *ACM SIGMETRICS Performance Evaluation Review*, Volume 27 Issue 4, March 2000, pp. 27 – 35. ACM : New York.

Menasec, D., 2002. *Load Testing of WebSites, Scaling the web*. Available at: http://computer.org/internet [accessed 11 December 2007]

Myres, G. J., 1979. *The Art of Software Testing*, John Wiley: New York.

Myres, G. J., 2004. *The Art of Software Testing*. 2nd Adition. John Wiley: New York.

Nguyen, H., Johnson, B. and Hackett, M. 2003. *Testing Applications on the Web: Test Planning for Mobile and Internet-based System.* 2nd Edition. Indiana: Wiley Publishing Inc., pp.479-520.

Open System Testing Architecture, OpenSTA. Available at: http://www.opensta.org/docs/ [accessed 20 August 2006]

OpenLoad Components. Available at: http://openwebload.sourceforge.net/developer/components.html [accessed 21 October 2006]

OpenWebLoad, 2005. *What is OpenWebLoad.* Available at: http://openwebload.sourceforge.net [accessed 12 October 2007]

Pressman and Herron, 1991. *Software Shock*. McGraw-Hill: Ohio

Rosen, A., 2000. *The e-Commerce Question and Answer Book: A Survival Guide for Business Managers*. New York: American Management Association.

Sopitkamol Monchai, 2004. Ranking Configuration Parameters in Multi-Tiered E-Commerce Sites. *ACM SIGMETRICS Performance Evaluation Review.* Available at: http://portal.acm.org/citation.cfm?id=1052305.1052310 [accessed 25 May 2007]

Subraya, B. and Subrahmanya, S., 2000. Object driven performance testing of Web Applications, *Quality Software, 2000. 30-31 Oct. 2000,* pp. 17 – 26. First Asia-Pacific Conference : US. Available at: http://doi.ieeecomputersociety.org/10.1109/APAQ.2000.883774 [accessed 20 May 2007]

Subraya, B. and Subrahmanya, S., 2000. Performance Testing: A Methodical Approach to E-commerce Application, *13th International Quality Week 2000,* May, Hyatt Regency, Embarcadero, San Francisco, California. Available at: http://www.soft.com/QualWeek/QW2K/qw2k.program.html [accessed on 19 April 2007].

Swama, D. and Elkadri, N., 2001. Website Quality Testing and Performance Analysis, *University of Ottawa*, March 31, 2001. Available at: http://www.netdesignplus.net/publications/sawma_kadri_report_2001.pdf [accessed 19 March 2006]

The Concise Oxford Dictionary of Current English, 7th Edition .1982. Clarendon Press: UK

Thiru, 2003. *Understanding Factors That Influence Performance of a Web Server Presentation.* Available at: http://web.cs.wpi.edu/~cs535/s03/thiru.ppt. [accessed 25 May 2007]

Viega, J. and McManus, J., 2000. The Importance of Software Testing. *The Cutter Edge Resource Center.* Available at: http://www.cutter.com/research.html [accessed 12 August 2006].

Web Application testing, WAPT 4.0. Available at: http://www.loadtestingtool.com/features.shtml [accessed 12 October 2007].

Xu, L., et al., 2005. Testing web applications focusing on their specialties. *ACM SIGSOFT Software Engineering Notes*, Volume 30 Issue 1, January. ACM : New York. Available at: http://portal.acm.org/citation.cfm?id=1052305.1052310 [accessed 25 May 2007]