A performance and installation research in web server solutions for small e-commerce systems.

Christian Abels (iis01)  
chac01@student.bth.se

Erfan Shirazi (is01)  
er sb01@student.bth.se

Mattias Håkansson (iis01)  
mahg01@student.bth.se

“Computer Science is no more about computers than astronomy is about telescopes”
E. W. Dijkstra

Supervised by:  
Miroslaw Staron  
School of Engineering
Preface

We would like to thank the following persons that have contributed in one way or another to this thesis.

Miroslaw Staron

-------------------------
All the companies participating in the survey

-------------------------
Carl Grundberg
Lars Widmark

-------------------------
Andreas Öberg
Mikael Granberg
Mattias Henriksson
ABSTRACT

This thesis investigates two different web server solutions. One is a commercial, proprietary solution known as the Windows solution that consists of Windows Server 2003, IIS and ASP. The other is a free, open source solution consisting of FreeBSD, Apache and PHP. The both solutions had the database MySQL as a common component.

The hypothesis that was used in this investigation is as follows: \textit{IIS on Windows Server is not better than Apache on FreeBSD for e-commerce systems.} To answer the hypothesis two empirical comparisons were conducted. One was a response time experiment testing two symmetrical web shops developed for the both solutions. For this response time test a stress test application was developed. The second comparison was a case study in the ease of installation of the two different solutions.

The third empirical research method was a survey that was conducted among Swedish web hotel administrators. The survey identifies various factors that play a part when choosing one of the solutions. Open source users prefer performance, security and costs of software while Windows users prefer required knowledge, usability and compatibility.

By analysing our result it is shown that the hypothesis is verified proving that an open source solution reports better performance because it has lower response times than the Windows solution. The results from the case study show that Windows is the easiest solution to install.

Keywords:

Web servers, stress-test, response time, e-commerce, server side scripting languages, installation process, Windows, FreeBSD.
Content

1. INTRODUCTION ................................................................................................................................. 5

1.1 PROBLEM FORMULATION .................................................................................................................. 6
1.2 HYPOTHESIS ...................................................................................................................................... 6
1.3 LIMITATIONS OF THE THESIS .......................................................................................................... 6
1.4 MOTIVATION ...................................................................................................................................... 7
1.5 RESEARCH METHODOLOGY ............................................................................................................ 7
1.6 THE AUDIENCE ............................................................................................................................. 8
1.7 RELATED WORKS ............................................................................................................................ 8
1.8 OUTLINE .......................................................................................................................................... 8

2. E-COMMERCE SERVER SOLUTIONS .................................................................................................. 9

2.1 E-COMMERCE .................................................................................................................................. 10
2.2 THE WINDOWS SOLUTION .............................................................................................................. 10
2.3 THE UNIX SOLUTION ..................................................................................................................... 12

3. EMPIRICAL COMPARISON ............................................................................................................... 13

3.1 THE TEST OBJECT – THE WEB SHOP ............................................................................................ 13
3.2 THE TEST TOOL ............................................................................................................................ 15
3.3 THE RESPONSE TIME EXPERIMENT ............................................................................................. 18
3.4 DATA ANALYSIS FOR THE RESPONSE TIME EXPERIMENT ...................................................... 19
  3.4.1 Index page test ............................................................................................................................. 20
  3.4.1.1 T-test ....................................................................................................................................... 28
  3.4.1.2 Results ................................................................................................................................... 28
  3.4.2 Database access test .................................................................................................................... 29
  3.4.2.1 T-test ..................................................................................................................................... 35
  3.4.2.2 Results .................................................................................................................................. 35
  3.4.3 Static HTML test ......................................................................................................................... 36
  3.4.3.1 T-test ..................................................................................................................................... 43
  3.4.3.2 Results .................................................................................................................................. 43
  3.4.4 Loop test .................................................................................................................................... 44
  3.4.4.1 T-test .................................................................................................................................... 48
  3.4.4.2 Results .................................................................................................................................. 48
  3.4.5 Final test results .......................................................................................................................... 48

3.5 INSTALLATION PROCESS CASE STUDY ......................................................................................... 49
  3.5.1 Analysis of sub categories .......................................................................................................... 50
  3.5.2 Case study results ....................................................................................................................... 54
  3.5.2 Case study conclusion .................................................................................................................. 56

3.6 THE SURVEY ................................................................................................................................... 57
  3.6.1 Survey questions .......................................................................................................................... 59
  3.6.2 Survey analysis ............................................................................................................................ 61
  3.6.3 Survey conclusion ....................................................................................................................... 64

4 DISCUSSION ........................................................................................................................................ 66

5 CONCLUSION ....................................................................................................................................... 67

6 FURTHER WORK .................................................................................................................................. 68

7 REFERENCES ....................................................................................................................................... 69

APPENDIX A – THE TEST TOOL ........................................................................................................... 71

APPENDIX B – THE TEST PAGES ....................................................................................................... 72

APPENDIX C – DATABASE CODE ....................................................................................................... 81
1. Introduction

On Internet today there exists two major market leading web servers. On the one hand we have Microsoft’s Internet Information Services (IIS, [14]). On the other hand we have the open source web server Apache that is currently the market leader according to netcraft’s [1] and securityspace’s [2] web server survey.

On the contrary to the surveys above, according to another survey that investigates what kind of web server that Americas Fortune 1000 companies are using for hosting their web sites, IIS was the market leader [3].

It is a fact that the choice of web server goes hand in hand with the choice of underlying operating system and server side scripting language. When using IIS as web server the underlying operating system is Microsoft’s Windows, since IIS is made for Windows and will not run on any other operating system. When using Apache the choice of operating system almost always falls into a UNIX based system such as various distributions of Linux and *BSD (OpenBSD, NetBSD, FreeBSD, [15]). Apache is also available for Microsoft’s Windows but according to a survey done by netcraft about Apache on Windows, the total number of sites running Apache on Windows is only a small fraction of all the sites that are running Apache [12].

Two of the existing web server solutions are known as LAMP (Linux, Apache, MySQL, PHP) and WISA (Windows, IIS, SQL Server, ASP). These are two different competing solutions. In these solutions it is easy to identify the proprietary, commercial solution and the open source solution.

These two solutions differ in their components, but solve the same problem. LAMP is using the server side scripting language PHP (PHP Hypertext Preprocessor, [16]) and WISA is using the server side scripting language ASP (Active Server Pages, [17]) All of the components within the WISA solution are Microsoft’s own in contrast to the LAMP solution where the components come from different vendors.

The investigation in this thesis will dig deep and test the Windows solution against an open source solution and see if they are equal or if one is better than the other. It will investigate the factors that play a part in choosing the Windows solution or the open source solution.

For the experiment conducted in this thesis a stress test application was developed that measures response time. Two symmetrical e-commerce systems for the solutions were developed to serve as the test object for the stress test application. A survey with questions was answered by appropriate professionals for identifying factors on choosing the solutions. An installation process case study was done to find out which one of the two solutions is the easiest to install.
1.1 Problem Formulation

The problem is that it is unclear which one of the two solutions is the better to run for small e-commerce systems. Is proprietary, commercial software better for hosting e-commerce or is a free open-source solution as good as it or even better? Three different problem aspects are focused upon in this thesis. They are performance, easiest installation process and identification of state of the art opinions from professionals working with the solutions.

This thesis will show which one is the better for cases within the limitation of the thesis and help uncertain system administrators to pick the right solution for their purposes.

1.2 Hypothesis

The hypothesis that has been worked upon throughout this thesis and that we have as a goal to either verify or falsify is:

*IIS on Windows Server is not better than Apache on FreeBSD for e-commerce systems.*

With the word better the following meaning is implied.

- Lesser response time reply from web server solution
- The easiest installation process.

With easiest it is meant least required knowledge from the user’s perspective, which is least number of options and steps in order to install the solution.

1.3 Limitations of the thesis

The limitation of this thesis is that only testing of performance is made in form of response time measurements. There is no research conducted in security, stability or other parts of the solutions. The installation case study looks only on which one requires least amount of knowledge from the user’s perspective in order to install a fully working solution.

With e-commerce systems we mean a small sized application. The e-commerce system is a web shop selling their products to customers. The web shop has approximately 200 products stored in their database. The e-commerce system must have server side scripting language capability. It must also be able to communicate with an underlying database. In our tests we have chosen to use the free open-source database MySQL [29] as a common component for the database.

The motivation behind choosing MySQL is that the experiment needed a database that would work on both solutions but also would not favor any solution. Another part of the
motivation behind choosing MySQL is that with a common component there is no need to consider any affect the database may have on the results in the experiment.

The e-commerce system will be considered in mind for a site with a maximum number of simultaneous connections up to 1200. The system will be considered for a site that has a maximum of a 10000 visitors per day. The e-commerce system will use sessions for storing the shopping cart.

Instead of choosing the default LAMP solution, where the operating system component consists of Linux, the authors choose to use FreeBSD [8] as the operating system component. The motivation for this choice is that the authors wanted to try something different then the ordinary paradigm.

1.4 Motivation

The motivation of this thesis is to find out which one of the two solutions is better for running small e-commerce systems. Another part of the motivation is about comparison between a commercial solution and a freeware solution to see if it is worth investing money in a commercial solution. The choice of web server solutions has been not just to compare the default web servers for each solution but also to compare the two market leading solutions. The goal is to test the two solutions against each other in an experiment.

1.5 Research methodology

The research methodologies used in this thesis are purely empirical. In the first research method an experiment that measured response time was conducted with an application developed by the authors. This choice was made in order to test performance for both solutions. Choosing to measure response time gives a relative good picture on how the solutions perform because response time is made up from the many parts that make the solution. This is the most important and influential part of the researches and that will weigh most importance when deciding the outcome of the hypothesis.

The second research method is a case study in the ease of installation for the both solutions. The installation process case study data is analyzed using a quantitative approach where different categories are used to identify ease of installation. Choosing this research method gives an indication on the required knowledge from the user’s perspective. This is important because the author’s common belief is that open source is much harder to install.

The third empirical research method was a survey among system administrators at Swedish web hotels. This was made in order to get the opinion on different aspects from people working with these solutions on daily bases.
As for analyzing the result data a quantitative approach is used where the solution with the best result wins. T-test measurement [4, page 304] is used for statistical analysis on the response time data to see if the gathered data either verifies or rejects the hypothesis from a statistical perspective. Within the survey there is a qualitative approach for some of the questions where the questions had open answers.

1.6 The audience

The primary audience for this paper consists of system administrators trying to choose between one of the two different solutions and wants to see a scientific evaluation of the two solutions that can help them in their choice. Administrators that are looking to migrate from one solution to the other are a focused audience. It is of interest to people that are about to invest in a new web server solution. Also it is in the interest of regular computer hobbyists that are interested in web server solutions performances.

1.7 Related works

Three different companies/organizations do large scale and serious web server usage statistics surveys. They are netcraft [21] that has a monthly web server survey involving 50 million active websites, port80software [22] conducts monthly surveys of the 1000 leading corporations Web sites in America to determine their Web server software. Securityspace [23] also does a web server survey but to a smaller extent than netcraft. To the author’s best knowledge, there is no scientifically research conducted on web server solutions for small e-commerce systems.

1.8 Outline

The second section of this paper starts with a brief background fact presentation that will make the reader comfortable about the two solutions. In section 3 the test object the web shop, the tool for testing, the experiments, the installation process case study and the survey is presented and a description of the results together with an analysis of the data.
2. E-commerce server solutions

In order to become comfortable with the two different solutions a short presentation about the different components the solutions consist of is presented in this section. A basic e-commerce solution consists, beside the hardware, of operating system, web server, database and server side script language. These parts together build a solution for hosting an e-commerce.

Because the thesis compares and tests this kind specific solution with these software components the written theory about this topic is virtually none existing when it comes to the solutions as a whole. The same goes for tweaking the whole solution. On the other side there is a lot of theory for each component individually but that does not fit our purpose for this thesis. Also it was in the author’s interest to stay objective by not fishing for results too see what was already better.

From now on the two solutions will be referred to as the Windows solution and the UNIX solution.

**The Windows solution consists of:**
Windows Server 2003 Web Server
IIS 6.0
MySQL 4.0.18
ASP 1.1.4322

**The UNIX solution consists of:**
FreeBSD 4.9
Apache 1.3.29.3
MySQL 4.0.18
PHP 4.3.5.7.1

For both solutions a default installation was conducted. 2000 was chosen as maximum number of connections for the web server in both solutions.
2.1 E-commerce

E-commerce is best explained as an alternative channel that companies can use in order to offer their products and services. It is more then just a platform for marketing the company, simple marketing of products and services do not count as e-commerce. For it being an e-commerce it has to contribute more then just simple advertise. It can be a railway company selling tickets and presenting timetable schedule or a local video rental store offering booking of movies for its customers. If the same video rental store would only have a presentation of the store and opening hours it would not be an e-commerce site according to the authors below.

N.Bandyo-padhyay [31, page 6] has the following statement when it comes to defining e-commerce; “In its most basic form e-commerce represents transactions which are handled electronically rather than on paper…. It includes, but is not limited to buying on the Internet”. Another definition by the same author is; ”doing business electronically”.

Fredholm’s [32, page 11] definition of e-commerce is, translated from Swedish; “E-commerce are all activities that takes place to exchange and supply information electronically to support and simplify companies business processes with the world around one”.

2.2 The Windows solution

The components that make up the Windows solution are all proprietary software that costs money. They all come from the same vendor, the multi billion software company Microsoft.

Operating System, Microsoft Windows
A very popular and dominating operating system for desktop computers, it is fully graphical and easy to use for beginners. The first version 1.0 was released back in 1985 and today the latest version released is Windows XP for the desktop market and Windows Server 2003 for the server market. [5]

Web Server, IIS
According to Simmons [6, page 251] IIS has been around since the early 1990s. IIS 1.0 was first introduced as an add-on product for Windows NT 3.51 and included basic support for Hypertext Transfer Protocol, static Web pages, and Common Gateway Interface (CGI) Web applications. When Windows NT 4.0 was released it introduced IIS 2.0. The next release, IIS 3.0, is best known for introducing ASP. Microsoft distributed IIS 4.0 as part of the Windows NT Option Pack. With the release of Windows 2000 came IIS 5.0. IIS 5.1 was released together with Windows XP Professional. The latest version out today is 6.0.
Server side script language, ASP
Microsoft Developer Network (msdn, [19]) says that Active Server Pages is a server-side scripting environment that you can use to create and run dynamic, interactive Web server applications. With ASP you can combine HTML pages, script commands, and COM components to create interactive Web pages or powerful Web-based applications, which are easy to develop and modify.

The motivation behind choosing ASP 1.1.4322 instead of ASP.NET [29] is that ASP.NET is precompiled code in opposite of ASP that is a basic script language like PHP that is compiled during request from client.
2.3 The UNIX solution

The components that make up the UNIX solution are all open source and free of charge under the GNU general public license [18].

**Operating System, FreeBSD**
According to the official website for FreeBSD [8], FreeBSD is an advanced operating system for x86 compatible, AMD64, Alpha, IA-64, PC-98 and UltraSPARC® architectures. It is derived from BSD, the version of UNIX developed at the University of California, Berkeley. It is developed and maintained by a large team of individuals.

FreeBSD comes default with no GUI (Graphical User Interface). All the commands are entered in a shell, a text based interface.

**Web Server, Apache**
According to Kabir [9, page 3] the first public version of Apache was released in April 1995. The 1.0 version was released on December 1, 1995. Since the beginning, the Apache Group has expanded and incorporated as a non-profit group. The group operates entirely via Internet. However, the development of the Apache server is not limited in any way by the group. Anyone who has the know-how to participate in the development of the server or its component modules is welcome to do so, although the group is the final authority on what gets included in the standard distribution of what is known as the Apache Web server.

**Server side script language, PHP**
Zandstra [10, page 8] says that PHP is officially known as PHP: HyperText Preprocessor. It is a server side scripting language written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers. The first version of PHP was created in 1994 as a set of web publishing macros. These were released as the Personal Home Page Tools and later rewritten and extended to include a package called the Form Interpreter (PHP/FI). From a user’s perspective, PHP/FI was already an attractive proposition, and its popularity grew steadily. It also began to attract interest from the developer community. By 1997, a team of programmers was working on the project. PHP is the most used module for the Apache web server according to a survey done by securitiespace [13].
3. Empirical Comparison

This section presents the test object, the tool for carrying out the experiment, the experiment itself, the installation process case study and the survey. All the result data is analyzed upon.

3.1 The test object – The web shop

For the response time test experiment something was needed to test for both solutions so a decision was made to develop a web shop. Two symmetrical web shops were developed that were identical except for the server side scripting language used. The Windows solution was coded with ASP and the UNIX solution was coded with PHP.

This background information was written for the requirement specification:

The e-commerce site sells computer games. It has 4 different genres (action, strategy, sport, adventure and other). All the titles are stored in a database table.

The site is built up with a static header and a left frame. The various genres are chosen from the left frame. The site logo will be shown in the header.

When viewing the different titles the user can add a game to his shopping cart. This calls a function that stores the product in a session that was started when the user entered the web shop. When the client wants to proceed with the purchase he will have to click on the “Your Cart” link that takes him to a page where information about what is currently in the shopping cart is displayed. Here the user can remove products. When the user is pleased with the content in his shopping cart he clicks on the proceed link which takes him to a page where information about the customer must be entered.

Also preferred payment method will be selected in this page, credit card or postal payment.

When the user is done with this information he will click on the proceed link and the customer information together with the ordered products will be saved in the database.
From the developed web shops a decision was made to test four different components.

This is what was tested for the both solutions.

- **Index page test**
  
The welcome page of the web shop, here a session is initialized.

- **Database access test**
  
Displaying a page with 200 computer games fetched from the underlying database.

- **Static html test**
  
The top frame was accessed alone, displaying the web shop logo.

- **Loop test**
  
Contains a ‘for loop’ doing 100 loops with an ‘if else’ clause inside checking if modulus is applicable on the current loop number.
Motivation behind the pages tested

The index page was used because it is a common part of a web shop. It also contains sessions that is used for identifying individuals and storing their shopping cart. The database access page was used because an e-commerce solution has to have some sort of database in order to handle the commerce, which is storing and presenting information. The static html page was used because the basic html capacity of the solution needed to be tested and the loop test page was used because the script language capacity also needed to be tested.

These tests were conducted various times with the stress test application that was developed for these tests and that will be presented in the next section.

3.2 The test tool

There exist a lot of client-side stress test applications but most of them perform more than just the response time test. Those kinds of tests were unnecessary; therefore we programmed our own stress test application “IVIK Web Server Test” (now referred to as IVIK) in the programming language C#. IVIK is using an HTTP-GET request. This means, according to fielding [11], the client that sends out the request, retrieve information from the URI (Uniform Resource Identifiers) specified by the source of the request. This information can be any kind of data; plain text as well as various kinds of coding.

The response time, in IVIK, is measured from the exact time the HTTP-GET request is sent from the client until the time when the data has been sent back from the server and received by the client. When running the response time test in IVIK the information of the URI is presented in the application along with the response time in seconds.

When explaining this on a higher level, what the application does is that it simulates a number of clients accessing a specific http address on the internet. You can say that it does the same thing as a normal internet user does when he access a web page from his internet browser.

The figure below illustrates four connections to a server both by normal users and the IVIK application. In the tests there were much more simultaneous connections than only 4. For example when doing the index test 6 six computers, running the IVIK application, were used. Each one of them simulated 100 clients. The figure only illustrates the basics of how the application works.
IVIK has four options which can be set to perform different tests.

The first option (1), ‘ClientNr’, helps the user to maintain a structure of the log files created by the application. This option field demands a number. Depending on which number entered, the log-file is named ‘client’ followed by the number.

The second option (2), ‘Nr of loops’, demands a number which decides how many times the application is going to send HTTP-GET requests to an URI. The reason for doing this is to get the responding server to get a workload over a specific time.

The third option (3), ‘Nr of threads’, demands a number and is the option which simulates a number of clients depending on what is entered in the option field. This option is the most important part of the application. One thread is equivalent to a user accessing an URI from a web browser, i.e. Internet Explorer.

The fourth option (4), ‘Address’, is the actual URI to the server page which the response time test is performed on.
When running this test the application creates a log-file called ‘client1’ (1 in Appendix A) in the same directory as the IVIK executable. The second action is that it creates 10 threads(2). When all threads are ready(3) the timer starts(4) and 10 simultaneous HTTP-GET requests are sent(5) to the URI specified.

The corresponding server processes the request and sends back the requested information to the client-side, in our case the IVIK-client. When the information reaches the IVIK-client the timer stops(6). The information is presented in the application along with the average response time(7) for each loop which is also stored in the log-file(8) mentioned earlier. The ‘Nr Of Loops’-option indicates that the scenario presented above will run 10 times.
3.3 The response time experiment

Below is a table that presents the various tests that was conducted for the response time experiment.

**Test protocol for Windows and UNIX solution with IVIK Application**

<table>
<thead>
<tr>
<th>Test object</th>
<th>Test nr</th>
<th>Nr of loops</th>
<th>Nr of threads</th>
<th>Nr of computers used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index page</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Index page</td>
<td>2</td>
<td>100</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>Index page</td>
<td>3</td>
<td>100</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>Index page</td>
<td>4</td>
<td>100</td>
<td>200</td>
<td>4</td>
</tr>
<tr>
<td>Index page</td>
<td>5</td>
<td>100</td>
<td>400</td>
<td>8</td>
</tr>
<tr>
<td>Index page</td>
<td>6</td>
<td>100</td>
<td>600</td>
<td>6</td>
</tr>
<tr>
<td>Index page</td>
<td>7</td>
<td>100</td>
<td>800</td>
<td>8</td>
</tr>
<tr>
<td>Database acc.</td>
<td>8</td>
<td>50</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Database acc.</td>
<td>9</td>
<td>50</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td>Database acc.</td>
<td>10</td>
<td>50</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>Database acc.</td>
<td>11</td>
<td>50</td>
<td>200</td>
<td>4</td>
</tr>
<tr>
<td>Database acc.</td>
<td>12</td>
<td>50</td>
<td>300</td>
<td>6</td>
</tr>
<tr>
<td>Static html</td>
<td>13</td>
<td>50</td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>Static html</td>
<td>14</td>
<td>50</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>Static html</td>
<td>15</td>
<td>50</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>Static html</td>
<td>16</td>
<td>50</td>
<td>600</td>
<td>6</td>
</tr>
<tr>
<td>Static html</td>
<td>17</td>
<td>50</td>
<td>800</td>
<td>4</td>
</tr>
<tr>
<td>Static html</td>
<td>18</td>
<td>50</td>
<td>1200</td>
<td>6</td>
</tr>
<tr>
<td>Loop</td>
<td>19</td>
<td>50</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>Loop</td>
<td>20</td>
<td>50</td>
<td>400</td>
<td>4</td>
</tr>
<tr>
<td>Loop</td>
<td>21</td>
<td>50</td>
<td>600</td>
<td>6</td>
</tr>
</tbody>
</table>

Figure 3.3.1 Test protocol

Totally 21 tests were conducted for the response time experiment for both solutions. The tests took place during a couple of days at a student computer room at Blekinge Institute of Technology with the computers running the IVIK application. The server that hosted the e-commerce system was placed at a student dorm.

The test server was a Pentium III 500 MHz with 320Mb RAM, 100 Mbit Network Interface Card, 30Gb Disk UTA100.

The experiment took place on the same network named Ronneby Student Network [24] that is located on the Swedish University Computer Network [25]. The author’s
motivation behind choosing to do the tests on the same network was to avoid unnecessary internet traffic and by narrowing down the distance the packets need to travel thereby avoiding that the response times were made up of times of long distance to travel instead of just the solutions performances.

3.4 Data analysis for the response time experiment

The following section presents all the response time tests that were done on the two solutions. It starts of with showing overall statistics for the specific test and then breaks down into the different number of simultaneous connections showing descriptive diagrams, statistics, and frequency tables on the response time data for the specific test. At the end of each tested component there is a t-test on the result data together with a conclusion. As for the t-test, it is included to give an insight of the tests in a statistical point of view. In computer science test results like the results from the response time test are unpredictable and can result in, some cases, extreme values. Though the different tests were conducted in the same environment during a specific time frame there are factors like network load that can affect the results in a minimal fraction of the tests. Since the t-test are looking into extreme values of the data sets it is presented that the resulting mean values are assumed to be made out of extreme values in some cases and those cases will not be taken into consideration when evaluating the two solutions.

T-test

According to Körner [4] the t-test is a statistical test for comparing mean values of a data set, in this case the response time of two different web server solutions.

<table>
<thead>
<tr>
<th>Test</th>
<th>Sig.</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
<th>Mean Diff.</th>
</tr>
</thead>
</table>

Figure 3.4.1 T-test headers

According to SPSS [20] the columns in Figure 3.4.1 are explained in the following way. The sig. value in the table above decides if the data points in the data set are assumed to have equal variation or not. If the sig. value (1) is greater than 0.1 it can be assumed that there is an equal variation otherwise it can not be assumed. If the sig. value is greater, the second test (row) can be taken out of the result and the first row is the final result. If the sig. value is lesser, the second test (row) is the final result.

The t column (2) displays the observed t statistics for each sample, calculated as the ratio of the difference between sample means divided by the standard error of the difference. The df column (3) displays degrees of freedom. For the independent samples t test, this equals the total number of cases in both samples minus 2.

The column labelled Sig (2-tailed) (4) displays a probability from the t distribution with (df) degrees of freedom. The value listed is the probability of obtaining an absolute value greater than or equal to the observed t statistic, if the difference between the sample means is purely random.

The mean difference (5) is obtained by subtracting the sample mean for group 2 from the sample mean for group 1.
3.4.1 Index page test

The above figure shows the mean response times for the index page test. A total of seven tests were done on each solution from 10 simultaneous connections up to 800. What this figure shows is that up to 500 connections the two solutions have quite the same response time but over 500, the response time for the Windows solution becomes higher than the UNIX solution, implying that the UNIX solution wins over Windows in the Index page test when comparing overall means.
Windows vs. UNIX 10 simultaneous connections (test1)

The above Figure displays 10 simultaneous connections on the both solutions. As shown there is some jitter in the beginning and in the end of the Windows solution. The UNIX solution follows a more stable line.

The UNIX solution has a lower mean than the Windows solution. The Windows solution has the lowest minimum response time and also the highest maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Valid</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.0247187</td>
<td>0.014203</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.0617274</td>
<td>0.011887</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.00313</td>
<td>0.00781</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.52188</td>
<td>0.12500</td>
</tr>
</tbody>
</table>

Table 3.4.1.1 showing statistics for 10 simultaneous connections

These frequency figures show that the both solutions have almost identical piles of frequency.
Windows vs. UNIX 50 simultaneous connections (test2)

![Graph showing response time for 50 simultaneous connections](image)

Figure 3.4.1.4 showing response time for 50 simultaneous connections

The above Figure displays 50 simultaneous connections on the both solutions. As shown the response times are more spread out on the Windows solution then for the corresponding test on the UNIX solution. The both solutions have some jitter in the beginning. The UNIX solution has a lower mean then the Windows solution. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>.1122969</td>
<td>.0848682</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>.0467916</td>
<td>.0439054</td>
</tr>
<tr>
<td>Minimum</td>
<td>.04094</td>
<td>.06125</td>
</tr>
<tr>
<td>Maximum</td>
<td>.40844</td>
<td>.44218</td>
</tr>
</tbody>
</table>

Table 3.4.1.2 showing statistics for 50 simultaneous connections

![Graph showing frequencies for 50 simultaneous connections](image)

Figure 3.4.1.5 showing frequencies for 50 simultaneous connections

The Windows solution has two piles of each around 40 in frequency and the UNIX solution has one big pile around 80 in frequency.
Windows vs. UNIX 100 simultaneous connections (test3)

The above Figure displays 100 simultaneous connections on the both solutions. As shown the response times are more spread out on the UNIX solution then for the corresponding test on the Windows solution. The both solutions have some jitter in the beginning.

The UNIX solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th>N Valid</th>
<th>N Missing</th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>.1126516</td>
<td>.1109551</td>
<td></td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>.04180361</td>
<td>.07187674</td>
<td></td>
</tr>
<tr>
<td>Minimum</td>
<td>.06047</td>
<td>.06563</td>
<td></td>
</tr>
<tr>
<td>Maximum</td>
<td>.41281</td>
<td>.46469</td>
<td></td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Figure 3.4.1.6 showing response time for 100 simultaneous connections

Table 3.4.1.3 showing statistics for 100 simultaneous connections

Figure 3.4.1.7 showing frequencies for 100 simultaneous connections
Windows vs. UNIX 200 simultaneous connections (test4)

The above Figure displays 200 simultaneous connections on the both solutions. As shown the response times are more spread out on the UNIX solution then for the corresponding test on the Windows solution. The both solutions have some jitter in the beginning.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Valid Missing</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1216873</td>
<td>0.1759847</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.04517552</td>
<td>0.19031367</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.07914</td>
<td>0.07320</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.51383</td>
<td>1.79553</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.
Windows vs. UNIX 400 simultaneous connections (test5)

The above Figure displays 400 simultaneous connections on the both solutions. As shown the response times are more spread out on the UNIX solution then for the corresponding test on the Windows solution. The both solutions have some jitter in the beginning but especially the UNIX solution.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N Valid</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>N Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.2168904</td>
<td>0.3672054</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.2215526</td>
<td>0.28012101</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.08813</td>
<td>0.10906</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.59895</td>
<td>2.61713</td>
</tr>
</tbody>
</table>

Table 3.4.1.5 showing statistics for 400 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.
Windows vs. UNIX 600 simultaneous connections (test6)

The above Figure displays 600 simultaneous connections on the both solutions. In this test the two solutions are pretty much the same except for the Windows solution during iteration 13 up to 17. The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Valid</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Missing</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>.8324136</td>
<td>.6954252</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.6139212</td>
<td>.46731271</td>
</tr>
<tr>
<td>Minimum</td>
<td>.21970</td>
<td>.18573</td>
</tr>
<tr>
<td>Maximum</td>
<td>11.95178</td>
<td>2.95074</td>
</tr>
</tbody>
</table>

Table 3.4.1.6 showing statistics for 600 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.

Figure 3.4.1.12 showing response time for 600 simultaneous connections

Figure 3.4.1.13 showing frequencies for 600 simultaneous connections
Windows vs. UNIX 800 simultaneous connections (test7)

The above Figure displays 800 simultaneous connections on the both solutions. In this test there is a big noticeable difference in the end of the diagram where the Windows solution goes up and down. It is also interesting that the UNIX solution goes up and down in the beginning.

The UNIX solution has the lowest mean for this test. The Windows solution has the minimum response time and the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>2.1013433</td>
<td>1.2718056</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.4513736</td>
<td>1.1127309</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.30591</td>
<td>0.33651</td>
</tr>
<tr>
<td>Maximum</td>
<td>10.81709</td>
<td>7.17153</td>
</tr>
</tbody>
</table>

Table 3.4.1.7 showing statistics for 800 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.

Figure 3.4.1.14 showing response time for 800 simultaneous connections

Figure 3.4.1.15 showing frequencies for 800 simultaneous connections
3.4.1.1 T-test

### T-test Response Time (INDEX)

<table>
<thead>
<tr>
<th>Test</th>
<th>Sig.</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
<th>Mean Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0.000</td>
<td>1.672622</td>
<td>198</td>
<td>0.096</td>
<td>0.01056157</td>
</tr>
<tr>
<td>test2</td>
<td>0.163</td>
<td>4.274713</td>
<td>197</td>
<td>0.000</td>
<td>0.0274297</td>
</tr>
<tr>
<td>test3</td>
<td>0.001</td>
<td>0.044031</td>
<td>198</td>
<td>0.833</td>
<td>0.00168966</td>
</tr>
<tr>
<td>test4</td>
<td>0.000</td>
<td>-2.77592</td>
<td>198</td>
<td>0.006</td>
<td>-0.054297</td>
</tr>
<tr>
<td>test5</td>
<td>0.107</td>
<td>-4.20673</td>
<td>188</td>
<td>0.000</td>
<td>-0.150316</td>
</tr>
<tr>
<td>test6</td>
<td>0.008</td>
<td>0.815301</td>
<td>198</td>
<td>0.416</td>
<td>0.1389953</td>
</tr>
<tr>
<td>test7</td>
<td>0.000</td>
<td>3.081379</td>
<td>198</td>
<td>0.002</td>
<td>0.6296577</td>
</tr>
</tbody>
</table>

**Table 3.4.1.1.1** showing t-test results for the index page test

When looking at the t-test results, some of the test results can not be taken into consideration. All the tests with a significance (2-tailed) level lesser than 0.05 will not be included when grading the different solutions. The reason for this is that the mean values of those tests are most likely made up out of extreme values. Test2, test4, test5 and test7 will not be taken into consideration.

### 3.4.1.2 Results

**Windows vs. UNIX (INDEX)**

<table>
<thead>
<tr>
<th>Test</th>
<th>Windows(points)</th>
<th>UNIX(points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test7</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

| SUM   | 0               | 3            |

**Table 3.4.1.2.1** Result points for the solutions
3.4.2 Database access test

Figure 3.4.2.1 showing mean response times for simultaneous connections of the database access test

The above figure shows the mean response times for the Database access test. A total of five tests were done on each solution from 10 simultaneous connections up to 300. What this figure shows is that up to 50 connections the two solutions have quite the same response time but over 50 the response time for the Windows solution becomes higher than the UNIX solution, implying that the UNIX solution wins over Windows in the Database access test when comparing overall means.
Windows vs. UNIX 10 simultaneous connections (test1)

The above Figure displays 10 simultaneous connections on the both solutions. The UNIX solution follows a pretty stable line all the way in contrary to the Windows solution that goes up and down and has a big peak at around the 40th iteration.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solution the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.3615289</td>
<td>0.1957175</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.6472654</td>
<td>0.08811895</td>
</tr>
<tr>
<td>Minimum</td>
<td>4.27031</td>
<td>1.16875</td>
</tr>
<tr>
<td>Maximum</td>
<td>4.74841</td>
<td>0.80468</td>
</tr>
</tbody>
</table>

Table 3.4.2.1 showing statistics for 10 simultaneous connections

The frequency diagrams to the left show that both the solution has about the same height of their highest piles.
Windows vs. UNIX 50 simultaneous connections (test2)

The above Figure displays 50 simultaneous connections on the both solutions. The UNIX solution follows a pretty stable line all the way in contrary to the Windows solution that goes up and down in identifiable intervals.

The UNIX solution has the lowest mean for this test. The Windows solution has the minimum response time and the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>1.1646738</td>
<td>0.976950</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.7629834</td>
<td>0.07678096</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.57250</td>
<td>0.91562</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.55217</td>
<td>1.30905</td>
</tr>
</tbody>
</table>

Table 3.4.2.2 showing statistics for 50 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times but the response times for the UNIX solution is more centralised in the beginning.

Figure 3.4.2.5 showing frequencies for 50 simultaneous connections
Windows vs. UNIX 100 simultaneous connections (test3)

The above Figure displays 100 simultaneous connections on the both solutions. The UNIX solution follows a pretty stable line all the way in contrary to the Windows solution that is more spread out.

The UNIX solution has the lowest mean for this test. The Windows solution has the minimum response time and the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>2.1991328</td>
<td>1.1390945</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.9481024</td>
<td>0.30973113</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.74468</td>
<td>0.89922</td>
</tr>
<tr>
<td>Maximum</td>
<td>13.70085</td>
<td>2.15095</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times but the response times for the Windows solution is more centralised in the beginning but showing a much higher interval in seconds then the UNIX solution.

Figure 3.4.2.6 showing response time for 100 simultaneous connections

Figure 3.4.2.7 showing frequencies for 100 simultaneous connections
Windows vs. UNIX 200 simultaneous connections (test4)

The above Figure displays 200 simultaneous connections on the both solutions. The UNIX solution is more stable then the Windows solution that is more spread out and shows higher peaks.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solutions has the maximum response time.

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>3.7953403</td>
<td>1.9745335</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.1263872</td>
<td>0.59933315</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.12629</td>
<td>.95455</td>
</tr>
<tr>
<td>Maximum</td>
<td>10.54774</td>
<td>3.96259</td>
</tr>
</tbody>
</table>

Table 3.4.2.4 showing statistics for 200 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times. The intervals are again much higher for the Windows solution.

Figure 3.4.2.8 showing response time for 200 simultaneous connections

Figure 3.4.2.9 showing frequencies for 200 simultaneous connections
The above Figure displays 300 simultaneous connections on the both solutions. The UNIX solution is more stable than the Windows solution that is more spread out and shows higher peaks. The response times are lower for the UNIX solution.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solutions has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>N Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>6.7050596</td>
<td>2.4981449</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>2.0627924</td>
<td>0.93211066</td>
</tr>
<tr>
<td>Minimum</td>
<td>3.14998</td>
<td>1.26718</td>
</tr>
<tr>
<td>Maximum</td>
<td>11.41332</td>
<td>4.67294</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times. The intervals are again much higher for the Windows solution.

Figure 3.4.2.10 showing response time for 300 simultaneous connections

Table 3.4.2.5 showing statistics for 300 simultaneous connections

Figure 3.4.2.11 showing frequencies for 300 simultaneous connections
3.4.2.1 T-test

<table>
<thead>
<tr>
<th>Test</th>
<th>Sig.</th>
<th>Equal variances assumed</th>
<th>t</th>
<th>df</th>
<th>Sig. (2 tailed)</th>
<th>Mean Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0.013</td>
<td>Equal variances assumed</td>
<td>1.794555</td>
<td>96</td>
<td>0.076</td>
<td>0.1656114</td>
</tr>
<tr>
<td>test2</td>
<td>0.000</td>
<td>Equal variances assumed</td>
<td>1.731017</td>
<td>96</td>
<td>0.087</td>
<td>0.1877238</td>
</tr>
<tr>
<td>test3</td>
<td>0.001</td>
<td>Equal variances assumed</td>
<td>3.799915</td>
<td>96</td>
<td>0.000</td>
<td>1.0803363</td>
</tr>
<tr>
<td>test4</td>
<td>0.000</td>
<td>Equal variances assumed</td>
<td>5.827829</td>
<td>96</td>
<td>0.000</td>
<td>1.6208686</td>
</tr>
<tr>
<td>test5</td>
<td>0.000</td>
<td>Equal variances assumed</td>
<td>13.14155</td>
<td>96</td>
<td>0.000</td>
<td>4.2069147</td>
</tr>
</tbody>
</table>

Table 3.4.2.1.1 showing t-test results for the database access test.

When looking at the t-test results, some of the test results can not be taken into consideration. All the tests with a significance (2-tailed) level lesser than 0.05 will not be included when grading the different solutions. The reason for this is that the mean values of those tests are most likely made up out of extreme values. The only valid test results in the DBACCESS test are test1 and test2.

3.4.2.2 Results

Windows vs. UNIX (DBACCESS) The table to the left shows that the UNIX solution wins over the Windows solution when measuring response time for database access. Both of the solutions have a spread out response time during the tests but the Windows solution had the higher mean by comparing overall. The test results printed in *italic* are the ones that did not pass the t-test and therefore can not be included in the results.

<table>
<thead>
<tr>
<th>Test</th>
<th>Windows(points)</th>
<th>UNIX(points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SUM</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3.4.2.2.1 Result points for the solutions
3.4.3 Static HTML test

![Graph showing mean response times for simultaneous connections of the static HTML test](image)

The above figure shows the mean response times for the Static HTML test. A total of 6 tests were done on each solution from 100 simultaneous connections up to 1200. What this figure shows is that up to 600 connections the two solutions have quite the same response time but over 600 the response time for the UNIX solution becomes higher than the Windows solution, implying that the Windows solution wins over UNIX in the Static HTML test when comparing overall means.
Windows vs. UNIX 100 simultaneous connections (test1)

The above Figure displays 100 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. The both solutions show some jitter in the beginning.

The Windows solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solutions has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.125867</td>
<td>0.1585896</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.10958232</td>
<td>0.08850017</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.04395</td>
<td>0.11656</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.74671</td>
<td>0.73906</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Figure 3.4.3.2 showing response time for 100 simultaneous connections

Table 3.4.3.1 showing statistics for 100 simultaneous connections

Figure 3.4.3.3 showing frequencies for 100 simultaneous connections
Windows vs. UNIX 200 simultaneous connections (test2)

The above Figure displays 200 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. The both solutions show some jitter in the beginning.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1258867</td>
<td>0.1585896</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.10958232</td>
<td>0.08850017</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.04359</td>
<td>0.11656</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.74671</td>
<td>0.73906</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Figure 3.4.3.5 showing frequencies for 200 simultaneous connections
Windows vs. UNIX 400 simultaneous connections (test3)

![Figure 3.4.3.6 showing response time for 400 simultaneous connections](image)

The above Figure displays 400 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. The both solutions show some jitter in the beginning.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1446133</td>
<td>0.2556287</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.09291287</td>
<td>0.15902290</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.07211</td>
<td>0.12110</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.73574</td>
<td>0.98698</td>
</tr>
</tbody>
</table>

Table 3.4.3.3 showing statistics for 400 simultaneous connections

![Figure 3.4.3.7 showing frequencies for 400 simultaneous connections](image)
Windows vs. UNIX 600 simultaneous connections (test4)

The above Figure displays 600 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. The both solutions show some jitter in the beginning.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.1615067</td>
<td>0.3777779</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.11110647</td>
<td>0.21576841</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.07568</td>
<td>0.17172</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.88042</td>
<td>1.20737</td>
</tr>
</tbody>
</table>

Table 3.4.3.4 showing statistics for 600 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.

Figure 3.4.3.9 showing frequencies for 600 simultaneous connections
Windows vs. UNIX 800 simultaneous connections (test5)

The above Figure displays 800 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. Here the differences are very noticeable and the Windows solution has way better response times.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.

Table 3.4.3.5 showing statistics for 800 simultaneous connections

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>N Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.22789</td>
<td>1.15093</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.30252</td>
<td>0.62691</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.11971</td>
<td>0.33899</td>
</tr>
<tr>
<td>Maximum</td>
<td>2.31124</td>
<td>3.24609</td>
</tr>
</tbody>
</table>
Windows vs. UNIX 1200 simultaneous connections (test6)

The above Figure displays 1200 simultaneous connections on the both solutions. The Windows solution has lower response times then the UNIX solution. Here the differences are very noticeable and the Windows solution has way better response times.

The Windows solution has the lowest mean for this test. The Windows solution has the minimum response time and the UNIX solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>0.2490125</td>
<td>1.5714488</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0.21456792</td>
<td>1.5397866</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.13744</td>
<td>0.42954</td>
</tr>
<tr>
<td>Maximum</td>
<td>1.68656</td>
<td>10.60699</td>
</tr>
</tbody>
</table>

Table 3.4.3.6 showing statistics for 1200 simultaneous connections

The frequency diagrams to the left show that the Windows solution has the highest pile with the same response times.

Figure 3.4.3.13 showing statistics for 1200 simultaneous connections
3.4.3.1 T-test

### T-test Response Time (STATIC HTML)

<table>
<thead>
<tr>
<th>Test</th>
<th>Sig.</th>
<th>Equal variances</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
<th>Mean Diff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0.213</td>
<td>not assumed</td>
<td>-1.6417</td>
<td>93,64267</td>
<td>0.104</td>
<td>-0.032703</td>
</tr>
<tr>
<td>test2</td>
<td>0.725</td>
<td>not assumed</td>
<td>-1.97045</td>
<td>86,6284</td>
<td>0.052</td>
<td>-0.046957</td>
</tr>
<tr>
<td>test3</td>
<td>0.002</td>
<td>assumed</td>
<td>-4.2622</td>
<td>98</td>
<td>0.000</td>
<td>-0.111016</td>
</tr>
<tr>
<td>test4</td>
<td>0.000</td>
<td>assumed</td>
<td>-5.3012</td>
<td>98</td>
<td>0.000</td>
<td>-0.216271</td>
</tr>
<tr>
<td>test5</td>
<td>0.000</td>
<td>assumed</td>
<td>-9.37653</td>
<td>98</td>
<td>0.000</td>
<td>-0.925034</td>
</tr>
<tr>
<td>test6</td>
<td>0.000</td>
<td>assumed</td>
<td>-6.01483</td>
<td>98</td>
<td>0.000</td>
<td>-1.322436</td>
</tr>
</tbody>
</table>

Table 3.4.3.1.1 showing t-test results for the static HTML test

When looking at the t-test results, some of the test results cannot be taken into consideration. All the tests with a significance (2-tailed) level lesser than 0.05 will not be included when grading the different solutions. The reason for this is that the mean values of those tests are most likely made up out of extreme values. The only valid test results in the STATIC HTML test are test1 and test2.

3.4.3.2 Results

### Windows vs. UNIX (STATIC HTML)

<table>
<thead>
<tr>
<th>Test</th>
<th>Windows(points)</th>
<th>UNIX(points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>test6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SUM</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.4.3.2.1 Result points for the solutions

It is shown to the left that the Windows solution wins over the UNIX solution when measuring response time for handling static HTML. The two solutions are pretty even but the Windows solution goes out of this test as the winner. The test results printed in *italic* are the ones that did not pass the t-test and therefore cannot be included in the results.
3.4.4 Loop test

![Graph showing mean response times for simultaneous connections of the loop test](image)

The above figure shows the mean response times for the Loop test. A total of 3 tests were done on each solution from 200 simultaneous connections up to 600. What this figure shows is that there is a major difference already at 200 connections. The UNIX solution wins this when comparing overall means.
Windows vs. UNIX 200 simultaneous connections (test1)

The above Figure displays 200 simultaneous connections on the both solutions. The UNIX solution has way better response times then the Windows solutions. Also the UNIX solution has a more stable line.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>1,8958723</td>
<td>0,1874866</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0,34464245</td>
<td>0,11190464</td>
</tr>
<tr>
<td>Minimum</td>
<td>1,31883</td>
<td>0,13484</td>
</tr>
<tr>
<td>Maximum</td>
<td>3,16534</td>
<td>0,92524</td>
</tr>
</tbody>
</table>

Table 3.4.4.1 showing statistics for 200 simultaneous connections

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Figure 3.4.4.2 showing response time for 200 simultaneous connections

Figure 3.4.4.3 showing frequencies for 200 simultaneous connections
Windows vs. UNIX 400 simultaneous connections (test2)

The above Figure displays 400 simultaneous connections on the both solutions. The UNIX solution has way better response times then the Windows solutions. The both solutions show pretty stable lines.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solution has the maximum response time.

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>1,9563727</td>
<td>0,2149417</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>0,42504727</td>
<td>0,23760981</td>
</tr>
<tr>
<td>Minimum</td>
<td>1,46907</td>
<td>0,14211</td>
</tr>
<tr>
<td>Maximum</td>
<td>4,65398</td>
<td>1,85447</td>
</tr>
</tbody>
</table>

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Figure 3.4.4.4 showing response time for 400 simultaneous connections

Table 3.4.4.2 showing statistics for 400 simultaneous connections

Figure 3.4.4.5 showing frequencies for 400 simultaneous connections
A performance and installation research in web server solutions for small e-commerce systems  June 2004

Windows vs. UNIX 600 simultaneous connections (test3)

The above Figure displays 600 simultaneous connections on the both solutions. The UNIX solution has way better response times then the Windows solutions. The both solutions show pretty stable lines.

The UNIX solution has the lowest mean for this test. The UNIX solution has the minimum response time and the Windows solution has the maximum response time.

The frequency diagrams to the left show that the UNIX solution has the highest pile with the same response times.

Table 3.4.4.3 showing statistics for 600 simultaneous connections

<table>
<thead>
<tr>
<th></th>
<th>Windows</th>
<th>UNIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Valid</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Missing</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mean</td>
<td>1.9067662</td>
<td>0.2429377</td>
</tr>
<tr>
<td>Std. Deviation</td>
<td>1.4621577</td>
<td>0.28356313</td>
</tr>
<tr>
<td>Minimum</td>
<td>1.44708</td>
<td>0.14458</td>
</tr>
<tr>
<td>Maximum</td>
<td>4.88888</td>
<td>2.19441</td>
</tr>
</tbody>
</table>

Figure 3.4.4.6 showing response time for 600 simultaneous connections

Figure 3.4.4.7 showing frequencies for 600 simultaneous connections
3.4.4.1 T-test

When looking at the t-test results none of the test results can be taken into consideration. All the tests with a significance (2-tailed) level lesser than 0.05 will not be included when grading the different solutions. The reason for this is that the mean values of those tests are most likely made up out of extreme values. There are no valid tests, from a statistical point of view, in the LOOP test.

3.4.4.2 Results

Windows vs. UNIX (LOOP)

It is shown that no winner can be pronounced in the LOOP test because the mean values of the LOOP test most likely were made up out of extreme values, theoretically. The practical results can be viewed at the graphs in the beginning of this test section. The test results printed in italic are the ones that did not pass the t-test and therefore can not be included in the results.

<table>
<thead>
<tr>
<th>Test</th>
<th>Windows(points)</th>
<th>UNIX(points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>test1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>test3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>SUM</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.4.4.2.1 Result points for the solutions

3.4.5 Final test results

Windows vs. UNIX (Final Result)

As an overall result for the response time tests the figure above shows that the UNIX solution wins over the Windows solution with 2 against 1.
3.5 Installation Process case study

Before starting the installation of the two different solutions, a consensus was obtained among the authors on the subjective feelings on how difficult it has been to install a UNIX system in the past compared to installing Windows. Those of you working with UNIX maybe remember difficulty encountered when installing UNIX for the first time? Especially for those of you that started working with UNIX before the mid 90’s but then that was a couple of years ago, when UNIX still was not for the average user.

In the beginning there were only text based installation programs, while today it is superseded by the graphic installation programs. They approximately work like the installation process that you would find in a newer version of Windows, like Windows XP. Those of us working with the newest Windows have gotten use to the ease of installing Windows. The knowledge base required is at its minimum. Microsoft has made this process easy to conduct, even for those not working with computers on daily bases.

But how is it to install FreeBSD for somebody used to working with Windows without any knowledge in FreeBSD? In this chapter a case study is conducted on the installation process of the two different solutions. The goals of the study are to:

- Install the operating system, web server and server side script language.
- Divide the installation process into different sub categories.
- Compare the sub categories and also the entire installation process.

In order to reach our goals and find answer to our question, we have divided both installation processes into these six sub categories:

- Installation setup screen.
- Partition and formatting.
- Regional and language options.
- Network installation.
- Web server and server side script language installation.
- Other options and installations.

Analysis in this case study will be conducted on the above categories by the following observations and measurements:

- Number of options and screens per category.
- Time taken for installation of goal.
- Given information/help to the user during these categories.
- Total number of steps required for reaching our goal.

Installation has been done on the same computer that was used for server in our tests. This research is conducted in the same way for both solutions; we will do the basic
installation on both systems using the default values presented by the installation process. We will also install network on both system, which will be basic network for accessing the Internet. The installation will be done from a CD image for both solutions. This is the installations that were made:

- FreeBSD 4.9 Standard installation with User distribution.

### 3.5.1 Analysis of sub categories

**Installation setup screen**

At the beginning of both installations the user will be presented with a setup screen that gives the user several choices that can be conducted. In Windows case there were four choices and FreeBSD had up to thirteen different options. Windows setup screen was only shown at the beginning, FreeBSD on the other hand has the setup screen through the whole installation, mainly because all the options and configurations that can be made from the setup screen as shown in Figure 3.5.1.1.

![Visual sysinstall](image)

**Figure 3.5.1.1 Visual sysinstall**

FreeBSD has installation instructions and glossary of functions that can be accessed by the user for help and information as shown in the above figure. These two options were missing in Windows but the small information given is sufficient for the task.

**Partition and formatting**
A couple of steps further into the installations, the user has to make necessary partitions and formatting the partitions with a chosen file system. Once again Windows has chosen a simple approach with only three options as shown in figure 3.5.1.2.

Five steps are required in order to make and format the required partition, total of eight steps are required if you have to delete a partition first.

In FreeBSD the numbers of options for partitioning are eleven as shown in figure 3.5.1.3. For default partitioning, that is an option, the user has to go through 5 screens in order to create and format the partitions. Once again there is not any help for this process in Windows besides basic information about what can be done. FreeBSD on the other hand has a help section explaining all the different options.
For Windows using NTFS “New Technology File System” quick format it takes only a few seconds to format the partition needed for installation. For FreeBSD using default options it takes about 5 minutes to format the necessary partitions.

**Regional and language options**
Both systems have the capability to change regional and language preferences. In Windows this is done by choosing your location in the world. By doing that Windows automatically changes all configuration so that they match user’s chosen region. It takes two steps to make this change. And no help section is supplied, only basic information. Four options are encountered during this category.

This category works in the same way in FreeBSD; you choose your country and FreeBSD configures the rest, which is keyboard layout. It takes only one step and the only option is to choose your location.

**Network installation**
For this case study customized network installation was used because insertion of the TCP/IP properties had to be done in order to access Internet, this applies for both systems. For Windows it required four steps, involving ten options. There is not any help section supplied until the fourth step where the user can configure the TCP/IP properties, shown in figure 3.5.1.4. As always basic information is show.

![Figure 3.5.1.4 TCP/IP properties for Windows.](image)

In FreeBSD the user has to go through five steps involving six options. During this category the user has access to a help section and also basic information about the process given to the user as shown in figure 3.5.1.5.
Web server and server side script language installation
Windows automatically installs both the Web server and the server side script language. FreeBSD on the other hand does not install these when installing the system; instead you have to choose to install them both. After network configuration you can download them both from the different FreeBSD FTP sites nearest your location. It takes seven steps, involving seventeen options. During this process the user has as always access to a help section.

Other options and installations
In Windows there are not much more options and configurations remaining. There is still couple of configuration remaining that takes four steps, involving seven options. No help was supplied but basic information was shown for the user. In FreeBSD there is also couple of configuration remaining, all of them being simple options that took three steps with two options per step. A help section was supplied for the user. There are a couple of options that was not addressed because they were not in the scope of this case study, and these options are the following;

- If this machine will function as a network gateway.
- Configure inetd and network services.
- Anonymous FTP access to this machine.
- Configure this machine as an NFS server.
- Configure this machine as an NFS client.
- Customize system console settings.
- If a non-USB mouse is attached.
- Would like to insert new users.
3.5.2 Case study results

This case study has looked at the number of steps and average number of options required at each category but also through the entire installation process. Time has also been a measured entity. Table 3.5.2.1 illustrates summary of category measurements made in this case study.

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of steps</th>
<th>Number of options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation setup screen</td>
<td>Windows 1</td>
<td>FreeBSD 1</td>
</tr>
<tr>
<td>Partition and formatting</td>
<td>Windows 5</td>
<td>FreeBSD 3</td>
</tr>
<tr>
<td>Regional and language settings</td>
<td>Windows 2</td>
<td>FreeBSD 4</td>
</tr>
<tr>
<td>Network installation</td>
<td>Windows 4</td>
<td>FreeBSD 5</td>
</tr>
<tr>
<td>Web server and script language installation</td>
<td>Windows 0</td>
<td>FreeBSD 7</td>
</tr>
<tr>
<td>Other options and installations</td>
<td>Windows 4</td>
<td>FreeBSD 3</td>
</tr>
</tbody>
</table>

Table 3.5.2.1 Summary of category measurements

Figure 3.5.2.1 Number of steps per category

Figure 3.5.2.1 illustrates number of steps needed for each category; note no need of steps for installation of web server and server side script language for Windows.
Figure 3.5.2.2 Number of options per category

Figure 3.5.2.2 illustrates number of options/category for each installation process.

<table>
<thead>
<tr>
<th>Measurements on the whole process</th>
<th>Windows</th>
<th>FreeBSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation time</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>Total number of screens</td>
<td>20</td>
<td>27</td>
</tr>
<tr>
<td>Total number of options</td>
<td>30</td>
<td>52</td>
</tr>
</tbody>
</table>

Table 3.5.2.2 Measurements for the both installation processes

Figure 3.5.2.3 Measurements on installation process

Table 3.5.2.2 and Figure 3.5.2.3 illustrates measurements on both installation processes.
3.5.2 Case study conclusion

As we talked about in the beginning of this chapter, the installation of UNIX systems has gotten much easier to conduct compared to the older versions. Today the user do not have to know anything about the hardware his computer has in order to install FreeBSD, compared to older versions where the user had to know everything about each hardware component in his computer; in that case the user do not have to be a guru of any kind for being able to install FreeBSD. Today the user has access to a graphical installation process and also a graphical environment to work in.

Installing Windows does not require much knowledge, the user does not have to know much about computers in order to install Windows, there is not many options and only one that had a help section, all necessary information is presented for the user and the number of steps required is at its minimum. When installing Windows you will start with almost a text based installation but it quickly changes into a more graphical process that always updates the user on what is being done and amount of time left.

FreeBSD on the other hand has a lot of options, many difficult to understand for somebody new on FreeBSD. But then you also have a system with more functions, like: FTP Server, NFS Server, Firewall capabilities and so on. All if these functions generate more options and time required to finish the installation. The FreeBSD installation never used a full end graphical installation environment that Windows switched to further into the installation. Instead FreeBSD used a graphical text based installation through the whole process. And that left us just being able to work with our keyboard and not using our mouse. That is not bad in any way, some prefers graphical environment and some do not, just matter of opinion.

They both had good information about the different things happening; FreeBSD also had a good help section giving the user information about different entities and installation options. FreeBSD has more information in the installation process and it also has a large help section but then it also has much more options.

Looking at these values; the thesis can draw the conclusion that Windows still is the easier one to install. But that does not make FreeBSD an operating system that is hard to install, it is just that Windows is easier.
3.6 The survey

In order to find out what professionals with experience in e-commerce thinks of these solutions a survey was sent out to 18 companies dealing with e-commerce and hosting web server solutions. In order to get a broad opinion for the survey, the following criteria had to be met:

- Operating system: Windows, Open Source or Both
- Web server: IIS, Apache or both.

Another important aspect is the number of visitors the solutions had to deal with, therefore the survey was sent to companies not only meeting our criteria but also being a target group that had visitors from 100 visitors/week up to 100 000 visitors/week. In order to also get the opinion from companies using a different solution, there is one company in the survey using Mac OS X [26] as operating system. Down below there are three figures illustrating our chosen target group, figure 3.6.1 illustrates the variety in operating systems.

As the percentage shows in Figure 3.6.1, the majority of web hotels use both solutions at the same time. It indicated that there is no overall best solution, but depending on the usage, one of the solutions is chosen.
As shown in figure 3.6.2, the majority of companies again use both web server solutions. It indicates that there is no generally better web server, but rather there are different solutions for different purposes. It could also be that Windows users automatically use IIS because it is integrated with Windows.

As it shows in figure 3.6.3, there is a big variety of chosen companies in the target group. The average number of hits/week in the target group is around 180 000 hits/week.
3.6.1 Survey questions

Before starting the analysis and drawing any conclusions, the questions and the answers will be presented.

What factors were most important when choosing operating system? (Max 3)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>77.8%</td>
<td>14</td>
</tr>
<tr>
<td>Cost for software</td>
<td>38.8%</td>
<td>7</td>
</tr>
<tr>
<td>Operating cost</td>
<td>44.4%</td>
<td>8</td>
</tr>
<tr>
<td>Security</td>
<td>72.2%</td>
<td>13</td>
</tr>
<tr>
<td>Maintenance</td>
<td>5.55%</td>
<td>1</td>
</tr>
<tr>
<td>Required knowledge</td>
<td>16.6%</td>
<td>3</td>
</tr>
<tr>
<td>Compatibility</td>
<td>22.2%</td>
<td>4</td>
</tr>
<tr>
<td>Usability</td>
<td>22.2%</td>
<td>4</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3.6.1.1

What factors were most important when choosing web server? (Max 3)

<table>
<thead>
<tr>
<th>Factor</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>72.2%</td>
<td>13</td>
</tr>
<tr>
<td>Cost for software</td>
<td>27.7%</td>
<td>5</td>
</tr>
<tr>
<td>Operating cost</td>
<td>38.8%</td>
<td>7</td>
</tr>
<tr>
<td>Security</td>
<td>61.1%</td>
<td>11</td>
</tr>
<tr>
<td>Maintenance</td>
<td>22.2%</td>
<td>4</td>
</tr>
<tr>
<td>Required knowledge</td>
<td>11.1%</td>
<td>2</td>
</tr>
<tr>
<td>Compatibility</td>
<td>27.7%</td>
<td>5</td>
</tr>
<tr>
<td>Usability</td>
<td>38.8%</td>
<td>7</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3.6.1.2

What factor was most important for your purpose?

<table>
<thead>
<tr>
<th>Factor</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>16.7%</td>
<td>3</td>
</tr>
<tr>
<td>Cost for software</td>
<td>5.6%</td>
<td>1</td>
</tr>
<tr>
<td>Operating cost</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Security</td>
<td>44.4%</td>
<td>8</td>
</tr>
<tr>
<td>Maintenance</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Required knowledge</td>
<td>11.1%</td>
<td>2</td>
</tr>
<tr>
<td>Compatibility</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Usability</td>
<td>11.1%</td>
<td>2</td>
</tr>
<tr>
<td>Other</td>
<td>11.1%</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 3.6.1.3
How often does some sort of problem occur with your web server solution?

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/day</td>
<td>5.6%</td>
<td>(0)</td>
</tr>
<tr>
<td>1/week</td>
<td>11.1%</td>
<td>(1)</td>
</tr>
<tr>
<td>2/month</td>
<td>22.2%</td>
<td>(4)</td>
</tr>
<tr>
<td>1/month</td>
<td>27.8%</td>
<td>(5)</td>
</tr>
<tr>
<td>4/year</td>
<td>22.2%</td>
<td>(4)</td>
</tr>
<tr>
<td>2/year</td>
<td>11.1%</td>
<td>(2)</td>
</tr>
<tr>
<td>1/year</td>
<td>11.1%</td>
<td>(1)</td>
</tr>
<tr>
<td>Never</td>
<td>16.7%</td>
<td>(3)</td>
</tr>
</tbody>
</table>

Figure 3.6.1.4

What kinds of problems occur?

<table>
<thead>
<tr>
<th>Problem</th>
<th>Percentage</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virus</td>
<td>5.6%</td>
<td>(0)</td>
</tr>
<tr>
<td>Trespassing</td>
<td>27.8%</td>
<td>(1)</td>
</tr>
<tr>
<td>Overload</td>
<td>50.0%</td>
<td>(5)</td>
</tr>
<tr>
<td>Hardware</td>
<td>55.6%</td>
<td>(9)</td>
</tr>
<tr>
<td>Software</td>
<td>55.6%</td>
<td>(10)</td>
</tr>
</tbody>
</table>

Figure 3.6.1.5

Figure 3.6.1.6 illustrates how they have ranked their own software solution.

Grade your software solution in terms of...

<table>
<thead>
<tr>
<th>Feature</th>
<th>Average rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>(4.2)</td>
</tr>
<tr>
<td>Security</td>
<td>(4.2)</td>
</tr>
<tr>
<td>Price</td>
<td>(3.8)</td>
</tr>
<tr>
<td>Usability</td>
<td>(4.1)</td>
</tr>
</tbody>
</table>

Figure 3.6.1.6
3.6.2 Survey analysis

When it comes to the opinions of the participants, the opinion is different depending on if they are using Windows or an open source solution. It is the same with what kind of web server they are using. And then we have participants using both solutions, both Windows and open source but also participants using both IIS and Apache. There are three participants using only Windows as operating system. In these three, two of them are using IIS and the third one is using Apache. Six participants are using only open source systems; all of those are using Apache. Eight participants are using both Windows and open source systems; all of those are using both IIS and Apache.

Figure 3.6.2.1 presents participants opinion on the three most important factors when choosing operating system. As shown in figure 3.6.2.1 Windows users have usability, compatibility and required knowledge for their three most important factors when choosing operating system. All three of them have usability and compatibility as their choice. In fact two of them that use IIS as web server had exactly the same three factors. The third one that had Apache as web server chooses security instead of required knowledge. Once again when asked what factor was most important for their purpose, IIS users answered required knowledge and the Apache user answered security.

The same pattern is presented when asked about the three most important factors when choosing web server. The two IIS users had required knowledge, compatibility and usability. While the Apache user had Performance, Operating cost and maintenance as his most three important factors as shown in figure 3.6.2.2.

Open source user’s opinion of the three most important factors when choosing operating system is rather different than Windows users as shown in figure 3.6.2.1. The major concern is about performance and security, but also costs for software. There is a big
difference in opinion between open source users and Window users when it comes to usability, compatibility and required knowledge. In the case of Windows users these three are the most important factors, but for open source users these factors do not concern them as much.

Participants using both solutions have fairly similar opinion as those using open source when it comes to operating systems. They both have performance and security as their two most important factors. The difference lies in open source user having costs for software as their third choice of factor while participants using both solutions have operating costs as their third factor as shown in figure 3.6.2.1.

As mentioned in the beginning of this chapter, there is a participant using Mac OS X as operating system and Webstar [27] as web server. His opinion on the three most important factors when choosing an operating system are; usability, maintenance and performance as shown in figure 3.6.2.1.

This participant is in fact the only one having maintenance as an important factor. When it comes to choice of web server, the participants three most important factors differs quite a bit compared to participants using Windows, open source or even both. Their choice is; usability, security and operating costs, as shown in figure 3.6.2.2.
All of the participants using an open source solution for operating system also use Apache. Their most three important factors in choosing web server are shown in figure 3.6.2.2. Once again there is a big difference in the opinion of the Apache users. While IIS users have knowledge, compatibility and usability as their most important factors, Apache users have performance, security and costs of software as their most important factors.

Participants having both solutions stated that the three most important factors when choosing a web server are at first place performance, second most important factor is security and at third place it became operating costs and usability as shown in figure 3.6.2.2. When participants were asked about the most important factor for their purpose, security came at first place for open source users and participants using both solutions as shown in figure 3.6.2.3.

For participants using Windows, required knowledge was the most important factor and the participant using Mac OS X answered required knowledge as his most important factor as shown in figure 3.6.2.3.

When asked how often there is a problem with their solution, open source users were the only ones that had three participants that never had any problems with the solution while participants using both solutions had almost as little problems as open source users. Windows users seem to have most problems of all participants as shown in figure 3.6.2.4.

**Figure 3.6.2.3** Participants opinion on the most important factor for their purpose.
For Windows participants most problems were about software problems and overload. But on the contrary the Windows participant using Apache did not have any problems with overload. The participant using Mac OS X stated problems with software. Open source participants that had any problems, stated them being about either software or hardware problems and participants using both solutions had problems mainly in software and hardware but also some problems with overloading and one of them also had problems with trespassing.

### 3.6.3 Survey conclusion

Of all data extracted from the survey the following conclusions can be drawn. When it comes to operating system, participants using open source, choose open source mainly because in their opinion that is the right choice if you value security and performance and costs for software. Participants using Windows on the other hand, has chosen Windows mainly for usability, compatibility and required knowledge. Then there is participants using both solutions and their opinion for their choice has been; performance, security and operating costs.

The first patterns presented are that open source participants and participants using both solutions both value the performance and security given by the open source solution. The third reason stated is about cost. Costs of the software and costs operating the solution have also been an important factor. Same pattern is found when participants were choosing web server solution. IIS users stated same reasons for choosing IIS as they did for choosing Windows and the same with Apache users and participants using them both.

The participant using Mac OS X stated Usability, same as Windows participants and performance, same as open source participants, but also maintenance. Although Mac OS X is not an open source software, it looks much like Windows but it is built on UNIX and therefore it could be said that it is a piece from both worlds, and maybe that is why this participant choose one of both important factors from both Windows and open source.
Security once again came at first place when participants using open source had to choose the most important factor and Windows participants choose required knowledge. When asked about problems that occurred and the number of times they occurred, open source participants were the only participants that had no problems. Looking at the other participants, open source users were still the ones with fewer problems. Another observation is that Apache users did not have any problems with overload.

The main conclusion for this chapter is that participants that values performance, security and have costs as a big factor has chosen open source solution in both operating system and web server, that is a complete open source solution. Meanwhile, participants valuing the required knowledge, usability and compatibility have chosen Windows.
4 Discussion

When we started working with the thesis we thought that performance would be the most important factor when choosing an e-commerce solution. In order to measure the performance of the solutions we decided that a response time test would be the best approach. When the survey was done we realized that security is the most important factor and performance the second most important according to the people maintaining and working with e-commerce. Due to lack of time, we had to conduct the survey simultaneous as we did the response time test and therefore we did not have the information that security was the most important factor and therefore no testing on security was made.

The response time experiment lasted for three days and it was very interesting. We had no idea of what the outcome could be so therefore we expected nothing in favour for any of the solutions. Surely some of the result data for the experiment could be made up of a value that are unexplainable, that is why we made the t-test. A lot of factors can occur in computers like disk cache, early initialization, a sudden heavy network load or the TCP/IP packets travelling some other route. That is what you get when dealing with computers and all of these cannot be taken into consideration.

For the solutions we used default configurations except for the maximum number of connections. IIS has a default of unlimited number of connections and Apache has a default of 150 maximum numbers of connections. So we decided to set 2000 maximum number of connections for both IIS and Apache. Later after the tests were done we found out that Apache by default is compiled to not take on more than 512 simultaneous connections. Perhaps this could explain why the UNIX solution lost the static html test, or maybe another factor played a part that day when we conducted the static html test on the UNIX solution.

Anyway it is quite interesting that despite this fact the UNIX solution still beats the Windows solution in some tests that had more than 512 simultaneous connections. That is something to think about. When looking at the response times for all the tests we are pretty quite surprised about that the Windows solution won the static html test with such margins and we have no reasonable explanation behind this, It would be interesting to redo this test and see if the outcome is the same.
5 Conclusion

Two conclusions can be drawn depending on the user’s requirements. If the user wants performance in the form of lesser response time, the results show that the UNIX solution is the right choice. On the other hand if the user wants an easy to install solution then the Windows solution is the right choice to pick according to the result from the use case study.

From a user of the e-commerce systems perspective he surely prefers lesser response times than an easier to install system since he is only interested in fast loading web pages.

When it comes to the survey, security was the first of the three most important factors. Open source users has chosen open source because of their opinion that the UNIX solution is more secure. If this is correct or not, is left unsaid.

Performance is the second most important factor for those that choose an open source solution and as our response time test showed, the UNIX solution was better in performance when it comes to lesser response times. Required knowledge is an important factor for those who choose a Windows solution. According to our case study, this was Windows strength because it was the easiest to install.

The choice of web server solution depends on the user’s demands and knowledge. For administrators with experience within Windows system a user friendly and easy to install system like Windows suits them better than then complex UNIX solution. For administrators with knowledge about UNIX or that are willing to spend extra time gaining knowledge about the UNIX environment, a UNIX solution is the choice when it comes to performance and custom made systems that can be obtained through the complex installation that is rich in installation choices.

With all of the above in consideration and the stress test being the most influential part, we can verify our hypothesis, especially if knowledge about the two solutions is available. If knowledge about the UNIX solution is lacking then Windows could be an easier choice, but if you are willing to learn UNIX then it is the right choice for better performance.
6 Further work

More response time tests could be conducted, especially tests that have higher number of simultaneous connections to see if the two solutions differ even more or if they at some point both reach a stable line. The database in the Windows solution could be swapped to Microsoft SQL [28] to see if that would make the response times for the Windows solution better. It would also be interesting to test with a server with better hardware and see if the response times become lower. The tests could be redone with Apache running on Windows to see if there is any difference. Other test aspects could be introduced that compares security, maintenance and economics. Finally the test server and the test clients could be placed on the same subnet thereby isolating the test environment from traffic not related to the tests to see if perhaps some of the extreme response time values could be removed.
7 References


http://www.port80software.com/surveys/top1000webservers/


http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html


[20] SPSS Version 12.0.0 / 4 Sep 2003


Appendix A – The test tool

StreamWriter sw = File.AppendText("client"+clientNr);  (1)
for(int i=0;i<nrOfIterations;i++)
{
    ............
    for(int j=0;j<nrOfThreads;j++)
    {
        ............
        Thread aThread = new Thread(new ThreadStart(request));  (2)
        ............
    }
    ............
    foreach (Thread aThread in ary)
    {
        aThread.Start();  (2, 3)
        Thread.Sleep(10);  (2, 3)
    }
    foreach (Thread aThread in ary)
    {
        aThread.Join();  (2, 3)
    }
    averageTime = averageTime/nrOfThreads;  (7)
    ............
    sw.WriteLine(s);  (8)
    ............
}
sw.Close();

public void request()
{
    ............
    ............
    try
    {
        ............
        DateTime startTime = DateTime.Now;  (4)
        httpResp = (HttpWebResponse)httpReq.GetResponse();  (5)
        TimeSpan time = DateTime.Now-startTime;  (6)
        ............
    }
    ............
}
Appendix B – The test pages

Below are the files presented that were used in test 1 (index page), test 2 (database access), test 3 (static html) and test 4 (loop).

For test 1 index.php and index.asp were requested.
For test 2 mainFrame.php?access=Other and mainframe.asp?access=Other were requested.
For test 3 the topFrame.asp and topFrame.php were requested alone.

For test 4 loop.php and loop.asp were requested.

PHP

index.php

```php
<?
session_start();
$session = session_id();
$cart = array();
session_register("session");
session_register("cart");
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>IVIK Game Webshop</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<frameset rows="130,*" cols="*" frameborder="YES" border="1" framespacing="0">
<frame src="topFrame.php" name="topFrame" scrolling="NO" noresize >
<frameset rows="*" cols="130,*" framespacing="1" frameborder="YES" border="1">
<frame src="leftFrame.php" name="leftFrame" scrolling="NO" noresize>
<frame src="mainFrame.php?access=Home" name="mainFrame">
</frameset>
</frameset>
<noframes><body>
</body></noframes>
</html>
```

leftFrame.php

```php
<?
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IVIK Game Webshop</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```
A performance and installation research in web server solutions for small e-commerce systems  June 2004

<topFrame.php>

<?
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IVIK Game Webshop</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
/*
.style1 {font-size: 14px}
*/
</style>
</head>
<body>
<center><img src="img/logo.jpeg" width="303" height="85"><img src="img/logo3.jpeg" width="303" height="85"><img src="img/logo2.jpeg" width="303" height="85"></center>
</body>
</html>

<mainFrame.php>

<?
session_start();
include_once "include/dbconnect.php";
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>IVIK Game Webshop</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
/*
.style5 {
  font-size: 36px;
  font-weight: bold;
}
.style7 {font-size: 14px}
*/
</style>
</head>
<body>
</body>
</html>
<? if($access == "Home") {
    ?><div align="center"><span class="style5">Welcome to our webshop...</span></div><?
} else {
    $query = "SELECT * FROM products WHERE genre = '$access';";
    $result = mysql_query($query) or die(mysql_error());
    $num = mysql_numrows($result);
    if($tmpCart != "")
        $cart[] = $tmpCart;
    ?></td>
    <table width="60%" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td width="90%">
            <table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
                <tr>
                    <td width="6" valign="top"></td>
                    <td>&nbsp;</td>
                    <td width="6" valign="top"></td>
                    <tr>
                        <td rowspan="3">&nbsp;</td>
                        <td>
                            <div align="center"> <font color="#000000" size="3" face="Verdana, Arial, Helvetica, sans-serif">$access Games</font> <br>
                        </div>
                        </td>
                        <td rowspan="3">&nbsp;</td>
                        </tr>
                        <tr>
                            <td>&nbsp;</td>
                            <td></td>
                            </tr>
                            <tr>
                                <td><table width="100%" border="0" align="center" cellpadding="4" cellspacing="1">
                                    <tr>
                                        <td width="35%"> <div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">Name</font></div></td>
                                        <td width="10%"> <div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">Price</font></div></td>
                                        <td width="20%"> <div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">Image</font></div></td>
                                        <td width="25%"> <div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">Buy</font></div></td>
                                        </tr>
                                    <? if ($num>0) {
                                        for ($i=0; $i<$num; $i++)
                                        {
                                            $productID= mysql_result($result, $i, "productID");
                                            $name   = mysql_result($result, $i, "name");
                                            $genre   = mysql_result($result, $i, "genre");
                                            $price   = mysql_result($result, $i, "price");
                                        }
                                    }
A performance and installation research in web server solutions for small e-commerce systems  June 2004

```php
$quantity = mysql_result($result, $i, "quantity");
$image = mysql_result($result, $i, "image");

<?
<tr bgcolor="#999999">
<td width="35%"><div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">$name</font></div></td>
<td width="10%"><div align="center"><font color="#000000" size="1" face="Arial, Helvetica, sans-serif">$price kr</font></div></td>
<td width="20%"><div align="center"><img src="$image"></div></td>
<td width="25%"><div align="center"><a href="mainFrame.php?access=$genre&tmpCart=$productID" target="mainFrame"><img src="img/add-to-cart.gif"></a></div></td>
</tr>
<?
} }
</table>
</td>
</tr>
<tr>
<td valign="bottom"></td>
<td>&nbsp;</td>
<td valign="bottom"></td>
</tr>
</table>
<table width="25%" border="0" align="right">
<tr>
<td width="57%"><div align="center"><span class="style7">Number of Items: </span></div></td>
<td width="43%">Number of Items: $count</td>
</tr>
<tr>
<td><div align="center"><span class="style7">Total Price: </span></div></td>
<td>Total Price: $advanced feature</td>
</tr>
</table>

<?
} }
</html>
```

dbconnect.php

```php
<?
$HostDB           = "localhost";
$UserDB           = "mewebshop";
$PasswordDB       = "changeme";
$DatabaseName     = "mattias";
$Connessione = mysql_pconnect($HostDB, $UserDB, $PasswordDB);
@mysql_select_db($DatabaseName) or die("Unable to select database ");
?>
```
loop.php

```php
<?for($i = 0; $i < 100; $i++){
    echo"<TABLE BORDER=1 CELLPACING=1 CELLPADDING=1 WIDTH='100%'>";
    echo "<TR>";
    if($i%2){
        echo "<TD>This line shows when modulus is available</TD>";
    }
    else{
        echo "<TD>This line shows when modulus is not available</TD>";
    }
    echo "</TR>";
    echo "</TABLE>";
}?>
</BODY>
</HTML>
```

ASP

index.asp

```asp
<% Dim cart(0) Session("session") Session("cart") = cart %>
<DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head>
<title>IVIK Game Webshop</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
</head>

<frameset rows="130,*" cols="*" frameborder="NO" noresize >
<frame src="topFrame.asp" name="topFrame" scrolling="NO"> <frameset rows="*" cols="130,*" frameborder="YES" border="1" framespacing="2">
<frame src="leftFrame.asp" name="leftFrame" scrolling="NO"> <frame src="mainFrame.asp?access=Home" name="mainFrame">
</frameset>
</frameset>
</noframes>
</body>
```

76
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

<%}
.style5 {  
    font-size: 36px;
    font-weight: bold;
}
.style7 {font-size: 14px

</ifth(Request("tmpCart")="") then
'do nothing'
else
Dim tempArray
tempArray = Session("cart")
Dim CurRecords
CurRecords = uBound(tempArray)
REDIM PRESERVE tempArray(CurRecords+1)
tempArray(CurRecords) = Request("tmpCart")
Session("cart") = tempArray
end if

if(Request("access")="Home") then
  %><div align="center"><span class="style5">Welcome to our webshop...</span></div>
  %>
else
  rs.Source = "SELECT * FROM products WHERE genre =" & Request("access") & ""
  rs.Open()
%
<%>
<table width="60%" border="0" align="center" cellspacing="0" cellpadding="0">
<tr>
  <td width="90%">
    <table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
      <tr>
        <td width="6" valign="top"></td>
        <td>&nbsp;</td>
        <td width="6" valign="top"></td>
      </tr>
      <tr>
        <td rowspan="3"></td>
        <td><div align="center">
            <font color="#000000" size="2" face="Verdana, Arial, Helvetica, sans-serif">Games</font>
        </div></td>
        <td rowspan="3"></td>
      </tr>
      <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
      </tr>
      <tr>
        <table width="100%" border="0" align="center" cellpadding="4" cellspacing="1">
          <tr>
            <td width="35%"><div align="center">Name</div></td>
            <td width="35%">
              <div align="center">Description</div>
            </td>
            <td width="35%">
              <div align="center">Price</div>
            </td>
          </tr>
          <tr>
            <td width="35%">
              <div align="center">Product 1</div>
            </td>
            <td width="35%">
              <div align="center">Description 1</div>
            </td>
            <td width="35%">
              <div align="center">Price 1</div>
            </td>
          </tr>
          <!-- Add more rows as needed -->
        </table>
      </td>
      <td rowspan="3"></td>
    </tr>
  </table>
</td>
</tr>
</table>
</div>
</%>
<table>
<thead>
<tr>
<th>Price</th>
<th>Image</th>
<th>Buy</th>
</tr>
</thead>
</table>

```html
<% Do while not rs.eof

<TR bgcolor="#999999">
<% rs.movenext Loop %>
</TR>
```
**OpenDB.asp**

```vbscript
<%  
set rs = Server.CreateObject("ADODB.Recordset")  
rs.ActiveConnection = "DRIVER={MySQL ODBC 3.51 Driver};SERVER=192.168.0.2;DATABASE=mattias;UID=root;PWD=;"  
rs.CursorType = 3  
rs.CursorLocation = 3  
rs.LockType = 3  
%>
```

**CloseDB.asp**

```vbscript
<%  
set rs = nothing  
%>
```

**itemsInCart.asp**

```vbscript
<%  
Dim tempArray2  
tempArray2 = Session("cart")  
Dim CurRecords2  
CurRecords2 = uBound(tempArray2)  
response.write CurRecords2  
%>
```

**loop.asp**

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
title="Loop"  
</head>  
<body>  
<%  
for i = 0 to 100  
response.Write "<table border=1 cellspacing=1 cellpadding=1 width='100%'>"  
response.Write "<tr>"  
if (i Mod 2) then  
    response.Write "<td>This line shows when modulus is available</td>"  
else  
    response.Write "<td>This line shows when modulus is not available</td>"  
end if  
response.Write "</tr>"  
next  
%>  
</body>  
</html>
```
Appendix C – Database code

CREATE TABLE `customerOrder` (  
    `customerOrderID` int(5) NOT NULL auto_increment,  
    `customerID` int(5) NOT NULL default '0',  
    `orderID` varchar(100) NOT NULL default '',  
    `shipping` varchar(30) NOT NULL default '',  
    `totalPrice` int(6) NOT NULL default '0',  
    `IPnumber` varchar(15) NOT NULL default '',  
    `date` datetime NOT NULL default '0000-00-00 00:00:00',  
    PRIMARY KEY (`customerOrderID`)  
) TYPE=MyISAM AUTO_INCREMENT=12 ;

CREATE TABLE `customers` (  
    `customerID` int(5) NOT NULL auto_increment,  
    `name` varchar(50) NOT NULL default '',  
    `adress` varchar(30) NOT NULL default '',  
    `zipcode` varchar(10) NOT NULL default '',  
    `city` varchar(30) NOT NULL default '',  
    `country` varchar(15) NOT NULL default '',  
    `phonenr` varchar(15) NOT NULL default '',  
    `email` varchar(50) NOT NULL default '',  
    `pass` varchar(15) NOT NULL default '',  
    PRIMARY KEY (`customerID`),  
    FULLTEXT KEY `city` (`city`),  
    FULLTEXT KEY `pass` (`pass`)  
) TYPE=MyISAM AUTO_INCREMENT=21 ;

CREATE TABLE `orders` (  
    `orderID` int(5) NOT NULL auto_increment,  
    `customerID` int(5) NOT NULL default '0',  
    `productID` int(5) NOT NULL default '0',  
    `quantity` int(4) NOT NULL default '0',  
    `totalPrice` int(6) NOT NULL default '0',  
    PRIMARY KEY (`orderID`)  
) TYPE=MyISAM AUTO_INCREMENT=1 ;

CREATE TABLE `products` (  
    `productID` int(5) NOT NULL auto_increment,  
    `name` varchar(40) NOT NULL default '',  
    `genre` varchar(20) NOT NULL default '',  
    `price` int(4) NOT NULL default '0',  
    `quantity` int(6) NOT NULL default '0',  
    `image` varchar(40) default NULL,  
    PRIMARY KEY (`productID`),  
    FULLTEXT KEY `image` (`image`)  
) TYPE=MyISAM AUTO_INCREMENT=17 ;