An evaluation of Free and Open Source e-commerce web application technology with regard to SMMEs

1Mpumelelo Msimanga  2Jim Chadwick

1 Primary Author, ELCB Information Services, EAST LONDON, TEL: 0822003602, email: mmsimanga@elcb.co.za
2 Department of Computer Science, University of Fort Hare, ALICE, TEL: 0824423009, email: j.chadwick@ru.ac.za

Abstract - One of the challenges faced by Small to Medium Micro Enterprises (SMME) when looking for means of Internet-enabling their businesses is the high cost of software used in the building and running of the web application. In this paper, an e-commerce application to sell art and craft online is built and deployed on four Java 2 Enterprise Edition (J2EE) web application containers and four freely available database servers. All tools used in the building of the application were open source or freeware. Each stage of building the application is described along with the tools used. A number of benchmarking tests were then run to ascertain the performance of each server under various loads. At the end of the paper, recommendations on software to use are given based on the test results acquired and practical experience gained while building the application using the various tools is described.

I. INTRODUCTION

The SMME sector has been identified as critical to achieving the goal of empowering communities historically oppressed by apartheid.[1]. The idea for the application created by this project was inspired by a group of 10 previously unemployed women who took matters into their own hands and formed a sewing group. They produce beautiful beaded goods, which include bags, jewelry, bedding and clothes. The group is known as Illingelihle Women’s Group. They are based on campus at the University of Fort Hare, in an unused dining hall. They sell their wares to visitors to the University of Fort Hare and at festivals such as the Grahamstown Arts Festival. This group of women and other SMMEs were the primary motivation for this project.

SMMEs in general do not make a great deal of money, as most are survivalist in nature. Set up and running costs are to be kept at a minimum. Purchasing of expensive software normally priced in US dollars is out of their financial reach. Hence an exploration into Free and Open Source Software (FOSS) to implement and run the e-commerce website was undertaken. This is in line with the Department of Communication (DoC) policy of making digital opportunities universally affordable and accessible.[2]

To carry out the evaluation of the software, a sample e-commerce application was built and during each stage of the development and deployment of the application, the software used was evaluated. The ecommerce application that was implemented was for selling goods online but in theory the technology can be used to build any web application that will meet the needs of SMMEs. It is envisaged that in the near future the site will then be extended into an e-mail and other SMME’s in the surrounding community will be able to offer their services or goods online. This will mean that costs will be spread over more SMMEs, thus reducing those costs even further.

II. SYSTEM SPECIFICATION

The e-commerce system was designed to meet requirements which are typical of most Business to Customer (B2C) e-commerce sites on the Internet. Listed below are three of the requirements typical of e-commerce sites included in the application. These requirements are listed as they are the most common actions carried out on an e-commerce site by Internet surfers [7]:

- Internet surfers can search or browse through list of assorted goods
- Frequent customers can register and login every time they visit the site; this saves them the necessity to fill in their details after making a purchase
- A shopping cart will keep track of a customer’s purchases

The site was built based on code from Clinton Begin’s JPetstore [8] which in turn was based on the Sun Microsystems® Pet Store [9] reference application. The payment system was not included in this project.

III. THE ARCHITECTURE

The site was deployed on the Java three-tier model as illustrated in Figure 1 [10].

Figure 1: E-commerce site architecture [3]

Adapted from Sun Microsystems®

The ecommerce site was deployed on a web container. The front-end of the application which, in this context, refers to the actual pages that were served to the customer, was built using the following technologies:

- HTML – Hypertext Markup Language
- CSS – Cascading style sheets
- JavaScript,
• DHTML – Dynamic HTML
• JSP – Java Servlet Pages

The business logic for the site was programmed using JSP and servlets. Both JSP and Servlets are server side technologies and are thus processed by the web container [11][12].

The application was built entirely on the Windows Operating System. The reason for using Windows with an open sources and free solution is that most new computers in South Africa and the world at large are sold with the Windows OS as the default operating system and thus Windows is the most likely OS that most SMMEs are using.

This project was implemented using Java, but it must be noted that Java is not the only free or open source technology that can be used to implement web applications. There are several technologies and the most commonly used technologies are PHP and Java. Java was chosen over PHP for the following main reasons:

• Scalability – the J2EE framework was designed for distributed applications and is thus very scalable as various components of the application can be run on different computers.
• Performance – various benchmarks have shown Java to have a better performance in benchmark tests [13]

Another major factor that played a role in choosing Java was the fact that Java is platform independent and thus once the application was created on one J2EE compliant server, it was then possible to deploy the application on the other application servers with no customization needed. In fact the application could then have been deployed on any web container on any operating system that had the Java Virtual Machine installed.

IV. THE IMPLEMENTATION

The system was implemented using the Model View Controller (MVC) architecture and built using the Struts framework from the Apache Software Foundation [5]. Struts is modeled on the MVC architecture and uses various technologies including Java Servlets, JSPs, and Java Tag Libraries. Figure 2 above summarizes the MVC architecture.

Figure 2: The MVC architecture flow diagram [4]

The model consists of the Java classes that handle the site’s business logic and is responsible of storing information about the state of the application. This included information such as the products in each customer’s shopping basket. The view is the interface between the customer and the rest of the system. The controller in the implementation is a Servlet that forwards requests made by the customer or other components to the appropriate resource that will processes them, such as a Java class, Servlet or JSP. MVC is a well thought out and tried and trusted architecture and using frameworks such as Struts it should be possible to build complex web applications to meet the business needs of any SMME.

V. EVALUATION CRITERIA

Figure 1 shows the components that made up the entire site. The Enterprise Information System (EIS) depicted in Figure 1 was a database in the case of this project. There were three main types of software involved in building and deploying the site and in broad terms these were:

• The development tools used to create the site
• The web container
• The database

The evaluation of software is an entire field in the Information Systems profession that includes broad fields such as usability testing, functionality testing and benchmarking. This project was not intended to conduct a complete evaluation of the software but rather to look at the needs of a typical SMME and then see how well the software used in the implementation and deployment of the e-commerce application met these needs. Bearing in mind that the entire web system was targeted at the SMME sector the software had to have the following qualities:

• Required functionality to perform the required task
• Relatively straightforward to install, setup, use and maintain
• Stable, well documented and constantly updated to include developments in industry standards

Based on these desired qualities the following criteria were used for the evaluation of the all the different types of software evaluated:

• Functionality – a list of the essential functionalities was drawn up for each software group and each piece of software within that group was checked for the listed functionality
• Ease of use – this was split into three distinct sections. The first section looked at the complexity of installing the software, the second section evaluated the amount of documentation and support available for the software and the third section evaluated the user interface for the software.
• Performance – looked at how well the software performed under different workloads.
• Stability and continued development – looked at how often the software was upgraded to fix bugs and to keep abreast of evolving industry standards.

Only software that was found to be stable i.e. showed signs of continuous development and had substantial user documentation was used in this project.
VI. THE DEVELOPMENT SOFTWARE

Developing a Java web application requires the developer to perform several different tasks and these include:

- Create the application directory structure – this process requires the developer to create several folders and files according to the servlet specification[11][12].
- Create the application – this will involve working with .java, .jsp, .html, .xml, .tld, .js1, .css and .property files.
- Compile the necessary files – this process involved compiling the Java files that were part of the application.
- Preview the application – this involved the following steps:
  - the application had to be first deployed on the server then
  - the server had to be started/restarted
  - then the application is then viewed through a browser
- Work with the database – quite often the developer has to be able to access the database to make changes to the database, to ensure that the application is actually working or view the structure of the database to ensure that correct calls are made to aptly named fields.
- When making changes to the application, the server had to be stopped, the application then had to be compiled and redeployed. The server then had to be started and the application viewed through a browser. This process had to be done every time a change was made to the application, even the minutest of changes.
- Creating WARS – when the application was complete a WAR file had to be created to deploy the application on the many servers being evaluated.

Using different programs to work with the different file types and to perform the different tasks listed above would be tedious and not very productive. To develop the application an Integrated Development Environment (IDE) was used. Two IDEs were evaluated for use when developing the web application. Both were able to accomplish all the points listed in the paragraph above. The two were NetBeans[14] and Eclipse[15]. Both IDEs proved capable of performing all the tasks listed and it was found that it is a matter of personal preference and system hardware specification when choosing which IDE to work with. If the system has at least 256MB RAM and a processor speed greater than 700MHz then NetBeans was considered to be the best choice for creating JSP and servlet technology based web applications. This was because NetBeans was created for and has all the functionality required for creating Java web applications on installation. Eclipse on the other hand needed to be extended through downloaded plugins in order to perform the necessary functions. Eclipse however had more powerful general editing tools.

VII. BENCHMARKING CONDITIONS

Some factors were common in the evaluation of both the databases and the web container. The first common factor was the tools used to conduct the benchmarks and this was JMeter [16]. Another tool called Badboy[17] was used in benchmarking the web containers as a form of verification of the results obtained from using JMeter.

Apache JMeter is a 100% pure Java desktop application designed to load test functional behaviour and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions [16]. Using the appropriate JDBC driver JMeter allows the user to connect to a database and run queries on the database. JMeter will then run the queries for a specified number of times simulating a specified number of users and then give feedback which can be in the form of graphs, aggregates or individual data on each query execution. Using JMeter the following data was obtained for each individual request sent to a database or web container.

- $N_C$ = number of concurrent requests a Web site or database is processing.
- $Z$ = average think time2, in seconds.
- $R$ = average response time for a request, in seconds.
- $X_0$ = average throughput, in requests per second

Using the figures obtained from JMeter and then using Little’s Response Time Law [18] which has the following equation:

\[ R = \frac{N_{VU}}{X_0} - Z \]

it was then possible to work out the realistic figure of how many concurrent requests the software being tested could handle. This is known as $N_{VU}$ which is number of virtual users. When a number of users access a web page, the likelihood of all of the users performing the exact same action at the same time is very slim. $N_C$ is the number of concurrent users simulated by JMeter performing the exact same action on the database or webserver and $N_{VU}$ is a calculation of the possible number of users that could be accessing the system. The equations below was used to calculate $N_{VU}$:

\[ N_{VU} = (R + Z) X_0 \]

This equation was used to calculate the number of users for both the database set of results and the web container set of results. The second common factor between the database and web container benchmarking was the hardware and network platform which was used to run the tests shown in Figure 3.

Figure 3: The benchmarking configuration

---

1 JavaScript file

2 The amount of time between two consecutive requests
The network island was isolated from the University LAN. All the hard drives were first formatted using the NTFS and then a fresh copy of the Windows 2000 OS was installed. The only software installed in the computers was the Java 2 Standard Development Toolkit and the software used in the evaluation. This was all done to ensure that no other unknown factors would affect the outcome of the benchmarks [19].

Drivers, memory tuning and database design issues were the three factors that had the most impact on performance in our tests [20]. The views on factors affecting the performance are also echoed by the Middleware Company recommendations for testing a database backed website [21]. For the benchmarks performed in this project all settings for all the software was left at the default settings.

VIII. THE DATABASES

The following databases were evaluated in the course of this project.

- MySQL – [www.mysql.com](http://www.mysql.com)
- MaxDB – [www.mysql.com/maxdb](http://www.mysql.com/maxdb)

The five database servers evaluated in this project were all found sufficient for the needs of an SMME. MaxDB had by far the most complete set of GUI tools for manipulating just about all aspects of the database and the server as a whole. However for all the databases there were plenty of third party free GUIs to work on the databases and achieve all required tasks. Of particular note was the OpenOffice.org 1.1[22] user interface that provided an MS Access³ like interface to each database. This made data entry through forms and the creation of reports quite user friendly, even for novice users. A choice had to be made and MySQL was chosen based on its overwhelming support and documentation available on the Internet and also for its performance figures shown in the graphs below.

Using Jmeter, performance tests were conducted on all the databases simulating different numbers of users. The performance data obtained from JMeter is shown in Figure 4 and Figure 5 below.

Figure 4: Database performance results

![Database performance results](image)

Figure 5: Virtual Number of Users for MySQL Database

![MySQL:Virtual Number of Users](image)

Figure 4 shows that all databases performed well below the six-second [23] guideline with up to 25 concurrent users. The best databases went as high as 150 concurrent users. Figure 5 shows the calculated virtual number of users that the database chosen for the final implementation MySQL, should according to the Response Time Law[18]. This means that on the platform used MySQL should be able to support as much as 300 users.

IX. THE WEB CONTAINERS

There are plenty of free web containers available free on the Internet. Four were evaluated and the results were found to be extremely satisfactory and the performance suitable for commercial deployment. Each one of the following containers was evaluated for use as the platform to deploy the website created.

- Jo

With regard to usability, Sun ONE had the best GUI and was easiest to use. Starting, stopping and deploying applications was fairly straightforward using the start menu options of the GUI supplied with the server. Deploying applications on all the servers was fairly simple as they conformed to the J2EE specification.

³ Desktop database which is part of the MS Suit of programs
Performance wise Figure 6 shows the performance of the containers when serving out static web content. All the servers were able to handle at least 70 users and still give sub-second responses.

**Figure 6: Web Container Performance results**

![Graph showing Web Container Performance results](image)

Figure 7 combines three types of performance data:
- Normal page - shows the response times of a static page
- MySQL – represents response results obtained when sending a search for product request to the database
- The rest of the lines show the response times of the top three performing web containers when a search for a product request is made.

The graph shows the differences in performance for these three types of requests. Clearly when web pages have to access a data source the response time of the web container suffers. [11][12]

**Figure 7: Comparison of response times for data accessing page, static page and direct database request**

![Graph showing Database Access Page Response Times](image)

Figure 8 below shows the virtual user calculation for the web containers. From this diagram it is possible to see that the top three performing web containers should be able to serve sub six second databases accessing pages responses for at least 250 users using the MySQL database.

**Figure 8: Calculation of virtual users for data intensive pages**

![Graph showing Virtual Users Accessing Data Intensive Pages](image)

JMeter was also able to give the percentage of the requests that have errors made by the server. Up until over a hundred concurrent users all servers displayed almost no errors at all. Most errors occurred when 160 simultaneous users were connected. Tomcat was chosen as the web container of choice mainly because of its performance figures were best. Wide support was available on the Internet and also Tomcat was the most supported of the two major IDEs evaluated in this project.

**X. SUMMARY**

First and foremost, with the correct knowledge, it was found that it is possible to build highly attractive and functional websites using only FOSS with technologies such as CSS, DHTML and using frameworks such as Struts. During the course of this project it was found that there were two main areas of discussion when it comes to working with free and open source software:
- Usability and documentation of the software
- Performance and functionality of the software

Usability in this case refers to how easy it is to use the software. Is there adequate documentation and are there support groups or forums that will assist should one experience a glitch? These are important areas of concern because though the software may be good enough, it may prove impossible to set up and get running. With regard to this issue, while working with the various free and open source software, it was found that for quite a number of the programs evaluated the documentation appeared to have been written by individuals with extensive knowledge of the program. As a result of this, one common mistake made with the documentation was that the documentation did not include basic information on the product but just the highly technical information. There were problems experienced getting started with some of the software. The good thing about the OSS community is that there are very helpful faceless community. With all problems that were experienced, when a request was posted for help on a forum, positive replies were always received and all problems were thus all solved. In conclusion one needs to have a sound knowledge of the technology they are using when working with OSS.

With regard to performance and functionality, the software evaluated was found to be just about as good as any commercial software that the author has worked with. It was shown that the web containers can give sub six second responses to as much as 250 concurrently connected users. This figure should suffice for the purposes of an SMME. All in all OSS may have adequate functionality but most of the software is realistically usable by highly skilled personnel and until the software becomes user friendly this shall remain so.

**REFERENCES**

[2]“OPEN SOURCE FRAMEWORK FOR THE DEPARTMENT OF COMMUNICATION”, Dept. of Communication, South Africa.


[23] BUTCH BLASINGAME, Building Your Web Site, 2001 http://www.hal-pc.org/journal/jan01/Column/wensite/wensite.html

Biographical Note: The principal author recently left the University of Fort Hare where he was studying for a Master of Science degree and working as a lecturer in the Computer Science Department. He is now currently working as a software developer with ELCB Information Services in East London developing web applications for various organizations in the Eastern Cape. He lists his hobbies as squash, Java, Open Source and the All Blacks.