

Project Proposal

A comparative analysis of the LAMP (Linux, Apache, MySQL and PHP) and Microsoft .NET (Windows XP, IIS, Microsoft SQL Server and ASP.NET) frameworks within the challenging domain of limited connectivity and internet speeds as presented by African countries.

By: Christo Crampton

Supervisor: Madeleine Wright

1.) Overview of the research

With the recent releases of ASP.NET and PHP 5, there has been much debate over which technology is better. I propose to develop and implement an industry strength online journal management system (AJOL – **African Journals OnLine**) using both frameworks, and use this as the basis for comparative analysis of the two frameworks against each other.

2.) Product Specification

AJOL is an existing website (www.ajol.org) which acts as an aggregation agent for a number of publishers of African journals who wish to publish their journals online and of course for researchers looking for information. The existing system is based on the OJS (Open Journal System) developed by Berkeley University. The system consists of a user frontend – where users can browse and search the contents of the database online – and an administration frontend – where publishers can log in and manage their journals on the database by performing tasks such as adding new abstracts or editing existing abstracts. It is currently developed in PHP with a MySQL backend.

The proposed system will consist of an online user interface, and online administration interface for publishers, as well as an offline administration interface for publishers. The online and offline administration interfaces are complementary and publishers can use either or both according to their preference. (See figure 1: interaction diagram).

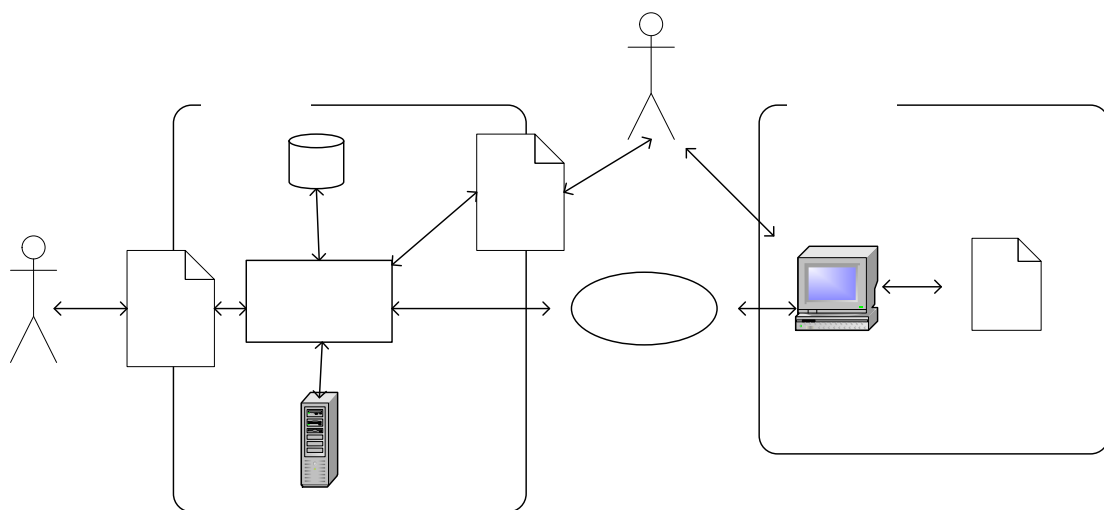


Figure 1 - User and System interaction

I propose a complete redesign of the frontend, and partial restructuring of the database structure. Numerous important extensions are also required, most notably; code optimisation and caching; the inclusion of full-text of articles in searchable PDF format; online payment and automated document delivery.

Furthermore, I propose to produce an offline version of the online publisher administration application. This will be downloaded to the publisher's computer and can operate as a stand-alone application offline, storing indexing data in a lightweight XML file. This information can then be merged with the main online database using a batch mechanism. This is a solution to slow internet connectivity where online management systems are impractical due to time overheads.

The online system above will be developed and implemented using LAMP *and* the .NET framework. The offline system will be developed only once and implemented in Java using XML as a storage medium, as well as a medium for communicating and merging data with the online system (live database) – see figure 1.

It is also important that web standards are met (as specified by the W3C), and that information displayed, especially dynamic data from the database, which is input by the publishers, is compliant to these standards. I also propose to design the pages using CSS (Cascading Style Sheets) instead of tables as these will load more quickly.

3.) Analysis

AJOL will provide the basis for discussion and comparative evaluation of the two frameworks within the problem domain - providing industrial strength enterprise applications to African countries with low bandwidth and poor connectivity.

This problem domain is likely to present interesting challenges, especially for an application of this scope and will provide an excellent tool to test the two frameworks under difficult conditions.

Some areas of interest highlighted by the problem domain:

- Connectivity issues

- Speed
- Error and timeout handling
- Data integrity
- Ability to handle broken connections
- Security (especially with regards to the security / performance trade-off)
- Interoperability with other languages
- Support for XML and web standards

Advanced mechanisms will be used to test these frameworks against each other. Standard testing will include unit testing and security testing.

The major technologies to be used and examined are:

- ASP.NET C#
- Microsoft SQL Server
- IIS (Internet Information Services) web server
- PHP 5
- MySQL Server
- Apache web server

Periphery technologies which will be used:

- Java
- XML
- CSS
- XHTML

4.) System requirements for the project:

I will need to set up the two development environments. This entails:

- 1.) A dual boot PC with Linux and Windows.
- 2.) On the Linux share I will need to install:
 - Apache web server
 - MySQL database server
 - PHP 5 interpreter
 (All the above are open source and available free)
- 3.) On the Windows share I will need to install:
 - IIS (Internet Information Services) web server

- MS SQL Server database server
- .NET framework

In addition, I anticipate that a live production server will be required to make these sites available on the internet. I propose the use of a dedicated server so that performance issues are consistent for testing purposes.

4.1.) Proposed software needs:

LAMP:

- Zend Studio
- Phpmyadmin (Open Source)
- SQLyog

.NET:

- Microsoft Visual Studio .NET

Other software:

- Macromedia Dreamweaver MX 2004
- XMLSpy
- Rational Rose
- Microsoft Visio

With the exception of Zend Studio, all this software is available within the Computer Science department or as Open Source software under the GPL (GNU Public License)

5.) Action Plan:

I have broken the project into five different phases:

1. Study of the literature
2. Design
3. Implementation
 - a. LAMP
 - b. Java
 - c. .NET
4. Analysis
5. Modification and Final Analysis

5.1) Study of the Literature

Books available on the subject include:

Freuks, H. 2003. **The PHP anthology**. Published by Sitepoint.com

This book outlines a best practices approach to PHP programming within the LAMP framework. The book also covers important aspects of advanced programming such as patterns and unit testing. It also recommends a number of good links which can be followed for further information.

Webb, J. with Microsoft Corporation. 2003. **Developing ASP.NET applications with Microsoft Visual Basic .NET and Visual C#**. Published by Microsoft Press. Washington

This book outlines best practice technique for programming within the .NET framework.

Williams, K. 2000. **Professional XML Databases**. Wrox press, Birmingham

With the production of the dual framework online systems it is going to be necessary to import and export data between a number of databases of different kinds (for example we will need to convert the data from the MySQL database to the SQL Server database). Using XML databases as a standard intermediate platform independent means of data storage will be the easiest way to get data between these different databases.

Other useful resources are the forums on Sitepoint.com (www.sitepoint.com)

Sitepoint.com forums are an excellent way to keep up with new developments in the world of web development, and also a good indication of current sentiment within the industry at the time.

Php.net

Microsoft.com

Apache.org

MySQL.com

Sun.com

The homepages for the major technologies will be an important tool for keeping up-to-date with latest features and products offered.

It will also be helpful to examine sites similar to the proposed AJOL project. Such sites may include: Ingenta (www.ingentaconnect.com), citeceer.com and the current AJOL website (www.ajol.org).

A simple study of the Google.com website and services (especially Google scholar) also falls within the province of this study.

5.2) Design

With an application of this scale it is critical that a solid design is produced prior to development. I propose to do the design using Rational Rose and UML. Microsoft Visio will be used for the database design.

The design process will also include communicating with experts in the field of journal publishing, and system users.

5.3.) Implementation

The three different components will be developed in the order:

- 1.) LAMP (online system)
- 2.) Java (offline system)
- 3.) .NET (online system)

I plan to complete all implementation in the BETA phase by the end of the June/July vacation. This will allow good time for the analysis of the system – The really important part of the project.

5.4.) Analysis

During this phase I will test and compare the two systems developed. Testing will be done at varying connectivity speeds (from as low as 14K), using a throttle to manage the speed. The results of these tests will be compared. From these a matrix determining performance at varying connectivity's can be determined. This comparison will test the technologies not only in terms of speed of content delivery, but also in terms of fail safe mechanisms (for example protecting data integrity in the case of partially executed batches of SQL statements).

The applications will also be tested in terms of security. To do this I will use (WebGoat), an application for detecting common security holes in web applications.

I will also perform unit testing on the applications.

The BETA applications will also undergo user testing. This is an infinitely useful tool for determining usability of the system and obtaining a third-party opinion on the project.

During the process of literature review and development it is likely that other means of testing may present themselves, and these testing actions will be performed too.

From the testing and analysis above I will establish a list of possible improvements and modifications to the applications.

5.5.) Modification and final analysis

Time permitting, the modifications and improvements found above will be implemented and analysis in 5.4 above redone, and the results from the two testing attempts compared.

6.) Possible Extensions

6.1) Java and JSP (Java Server Pages)

The project could be extended to compare Java and JSP to the technologies above. Furthermore, since Java is a platform independent language, and not married or tied to any particular framework, a comparative analysis of Java's performance against that of PHP and ASP.NET would provide deeper insight into the performance of these languages against each other, independent of their surrounding framework.

By comparing an application implemented in Linux, Apache, MySQL, PHP with the same application implemented in Linux, Apache, MySQL, **Java**; and similarly a Windows, IIS, MS SQL Server, ASP.NET application with a Windows, IIS, MS SQL Server, **Java** application, we could effectively compare PHP with ASP.NET independently of their platforms by comparing their relative success against Java within their appropriate framework.

6.2) Extension of the problem domain

We could also test the systems at the other end of the spectrum, in an environment of very high connectivity, and their ability to handle advanced features

6.3) Mozilla Firefox plugins

Some of the problems presented in the project could be resolved by writing plugins for the popular open source browser, Firefox.