Confronting the Performance Issues Related to Dynamic Web Technology

John A. Hines

Arkansas State University P.O. Box 239 State University Arkansas 72467
1-870-972-3416 * 1-870-236-4730
John.Hines@smail.astate.edu

ABSTRACT

Many who develop dynamic web applications do so with information from biased sources, such as the open source community or those who favor the more familiar Microsoft brand technologies. Many assume that there is no difference in overall performance or suitability among the various dynamic web technologies. Different dynamic web applications were evaluated through benchmarking on different machine configurations to determine the best configuration performance-wise. The results make it clear that dynamic web technologies that were developed for specific platforms tend to perform better within their native environments. While these technologies may be capable of use under non-native environments, this would not be ideal. The decision concerning what dynamic web technology to use is ultimately dependant of the needs of the web application and the cost involved.

INTRODUCTION

We have moved from a virtual military environment in the days of ARPAnet to today's huge world of retail sales sites, information portals, educational resource repositories, and personal web spaces. When the Internet transitioned from the government to the public domain the opportunity for technologies to develop also evolved. Throughout the mid to late 1990s and on into the 21st century, the World Wide Web has evolved into a mass marketplace for information and for retail businesses. Deciding which technologies to use when building a web presence has thus become a more difficult decision as more of these technologies have become available.

Many web technologies, meaning the hardware, software (including the operating system, web server, scripting engine, database etc.), are available at a wide variety of costs. Available hardware consists of many preconfigured machines, such as the Dell PowerEdge, HP Pavilion, and the IBM Blade servers. As far as operating systems are concerned, SUN Solaris 10, UNIX, Linux and it's many distributions (for example, RedHat Enterprise Linux, Novell SuSE Enterprise Linux, Mandrake, and CentOS 4.0) and the different flavors of Microsoft Windows (such as Windows 2000 Advanced Server and Windows 2003 Enterprise Server) are the most popular and are available at costs ranging from nothing to many thousands of dollars. To compliment these operating systems, a number of web server technologies are available, with Oracle Application Server, Apache, and Microsoft Internet Information Server comprising nearly 100% of the market. These web servers will work in conjunction with the vast array of scripting engines available such as Cold Fusion, the common gateway interface (CGI or FastCGI), Perl, Python, Java Server Pages (JSP), Java Servlets, ASP.NET, and classic ASP. These scripting engines are in turn able to interact with database technologies such as Oracle, MySQL, PostgreSQL, Microsoft SQL Server, and even Microsoft Access.

Previous research into the vast collection of web technologies has been thus far narrow in its focus. Some has targeted web server technologies such as Microsoft Internet Information Server and the Apache Web Server (Apache). Other research has been aimed toward specific programming languages (ASPMaker) or database performance (Huang). Furthermore, some of the prior research into this topic has been biased by corporate sponsors of that research. In one article called "Lies, Damn Lies and Benchmarks" the author tells the story about the Mindcraft benchmarking incident (Whittman). Mindcraft reported that a Microsoft Windows NT system running Microsoft IIS was 3.7 times faster than an Apache Web Server running under the Linux operating system (Welcome to Mindcraft). This research was funded by Microsoft and the reported results have come under scrutiny.

The continuing developments in web technologies dictate the need for a broad approach to comparative studies within this field. While, on the surface, some of this research may seem to be common sense to the Information Technology professional or educator, the results should be useful to many small to midsize firms. They need to know which, if any, web technology platform will perform best, in spite of various contradictory claims by industry participants. Thus it will be assumed initially that all web server and web language technologies are in essence the same. That is, there are no significant differences in overall performance or suitability among the various web technologies. The goal of this research is to prove otherwise through performance benchmarking and comparison of those benchmarks to find which technologies perform best.

METHODS AND MATERIALS

For this research it was decided to concentrate upon Active Server Pages (ASP) (Microsoft Office Developer Center) and Personal Homepage (PHP) (PHP Hypertext Preprocessor). Each of those web languages were processed upon Microsoft IIS version 6 as well as Apache Web Server versions 1.3.29 and 2.0.54. Since the schedule for this research was limited the technologies used were chosen on the basis of popularity and functionality (ServerWatch). According to Security

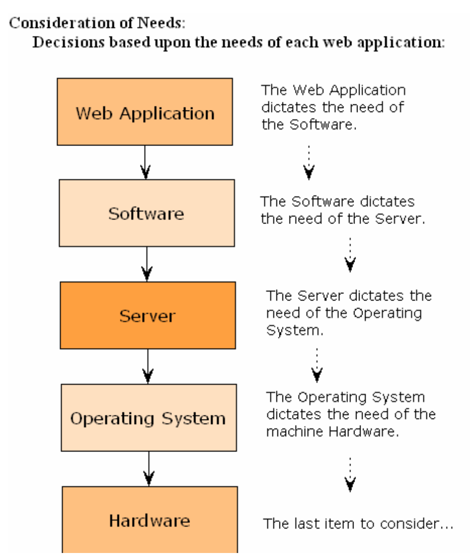

Figure 1: Assessing needs of web applications.

Space (Security Space), a web based organization provided by a company called E-Soft Inc.; the Apache Web Server enjoys a dominant 72.1% share of the web server market while Microsoft Internet Information Server has a share of 22.26%; other web server technologies have a 5.63% share of the current web server market (Web Server Survey). Figure 1 outlines the process for considering the dependent technologies addressed in developing web applications.

This format was followed to list what is required to perform the data collection. For the processing of ASP and PHP pages, PHP 5.0.4 was installed as well as Sun Microsystems ASP ONE server (Sun Java System Active Server Pages 4.0). ASP ONE is a server application used to process Active Server Pages in the non-native Apache environment. The content of the PHP pages were generated with a trial version of a program called PHPMaker (PHPMaker). The content of the ASP pages were generated with a trial version of a program called ASPMaker (ASPMaker). Both of these programs generate dynamic pages that have the ability to query and edit a database table.

The benchmarking of static web technologies was bypassed due to vast amounts of past research (Shiloh Consulting) on those technologies. Concentrating instead on dynamic web content, as is used in many ecommerce and information sites today, database software was acquired. For this purpose, MySQL (MySQL) and Microsoft Access 2003 (Microsoft Office Online) were used to store data. MySQL is a database server that runs separately in a process as opposed to Microsoft Access databases which reside in files with the .mdb extension. The construction of the databases was simple one line entries into a table with generic information. A simple first name, last name, email address, and identification number were stored. A simple database can be made for Microsoft Access 2003 by entering the data by hand in MS Access. Otherwise, a converter program such as MySQL to Access (Huang) can be purchased.

A Gateway brand server was acquired equipped with a two 1GHz processors and 512MB Random Access Memory (RAM). This server was configured to perform in a dual boot capacity running Windows Server 2003 Enterprise (Microsoft Windows Server System) as well as SuSE Linux Enterprise Server 9 (SLES for the remainder of this document) (SuSE Linux Enterprise Server 9). JBlitz Professional 4.2 (JBlitz Professional) was used to perform the actual benchmarking of the web applications. Because JBlitz Professional is a Java application, the Java Runtime Environment (JRE) (Java Runtime Environment) was installed on both operating systems used. Both operating systems were configured to run the web server software and database server automatically upon system start up.

JBlitz Professional was configured to access each of the web applications generated by PHPMaker and ASPMaker. This configuration consisted of a test case of 7 virtual users, each representing a person requesting the application to load in their favorite web browser. Every 200 milliseconds one of the virtual users would send a request to the web application for a dynamically generated page. JBlitz Professional was configured to stop processing the page requests after 10,000 successful responses. If an error occurred, the benchmarking utility would add an extra request to the program queue. This same benchmark configuration was used to test the following web site or a variety of configurations consisting of PHP 5.0.4, ASP, MySQL, Microsoft Access 2003, Apache 2.0.54, or Microsoft IIS 6.0 running under either the Microsoft Windows Server 2003 or the Suse Enterprise Linux Server operating systems. Each configuration was tested with JBlitz Professional separately and the resulting data was then entered into a spread sheet program for comparison later on.

Codes:
- **Languages:**
  - **PHP:** PHP 5.0.4
  - **ASP:** Active Server Pages (ASP)
- **Web Server:**
  - **Ap:** Apache
  - **IIS:** MS Internet Information Services
- **Operating System:**
  - **Win:** Windows 2003 Enterprise Server
  - **SUSE:** SuSE Linux Enterprise Server
- **Database:**
  - **Acc:** Microsoft Access 2003
  - **My:** MySQL

Figure 3: The above code listing refers to the configuration types used within Figures 4-12 below.

RESULTS

During the course of these tests, a number of errors were encountered and these errors constitute a measure of performance. The common error amongst the entire set of different configuration

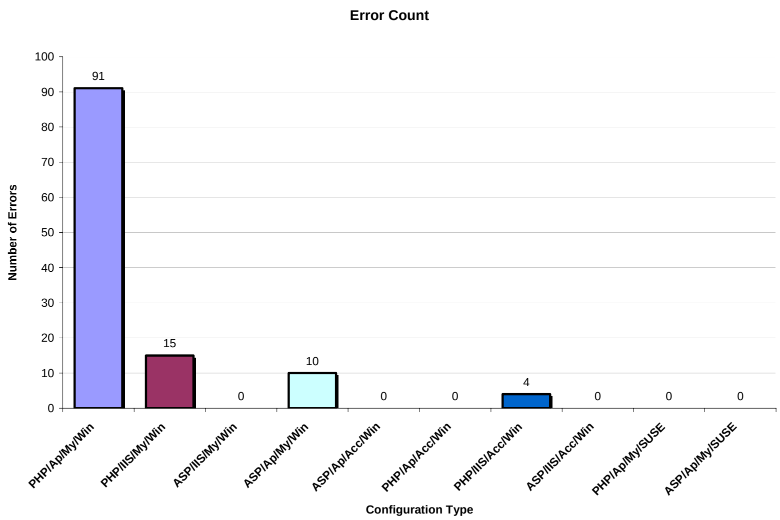

Figure 4: This graph shows the number of errors encountered during the course of testing each configuration type. Each configuration type processed until 10,000 successful page hits.

types was address related. The error message produced by JBlitz Professional was “The connection could not be established – Address already in use”. Ninety one errors were observed when using PHP with the Apache Web Server under Windows 2003 using the MySQL database. Fifteen more errors were observed when using PHP with the Microsoft IIS6 under Windows 2003 using the MySQL database. Ten more errors were encountered when using ASP with the Apache Web Server under Windows 2003 using the MySQL database. Only four errors occurred when using PHP with Microsoft IIS6 under Windows 2003 using Microsoft Access 2003 to retrieve dynamic data. The other configurations tested did not produce error. These errors are summarized in Figure 4.

Each of the benchmarked configuration types performed at different speeds. For each of the configurations 10,000 completed transactions were processed (net of the number of errors plus 10,000 successful hits). As is illustrated below in Figure 5, benchmark test times varied from 10 minutes and 10 seconds for PHP running on the Apache Web Server in a Linux environment to 15 minutes and 57 seconds for PHP running under Microsoft IIS on Windows 2003 Enterprise Server.

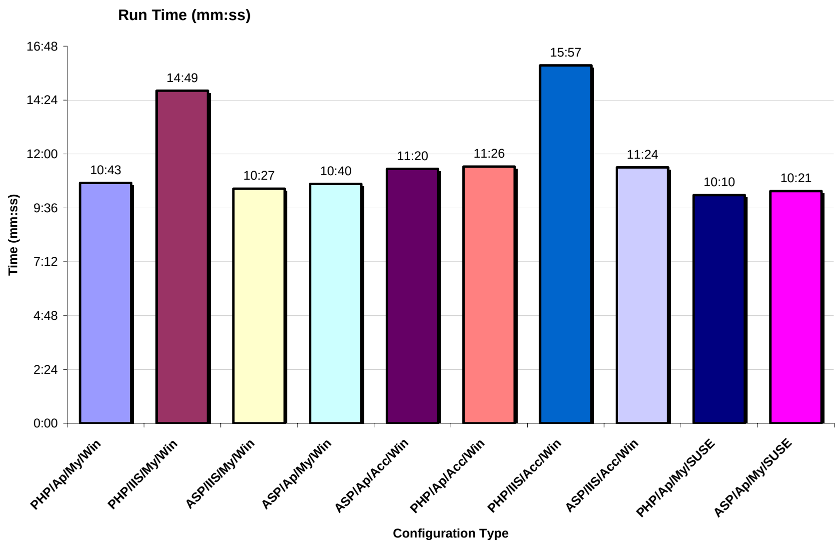

Figure 5: This graph outlines the length of time each configuration type took to process 10,000 hits.

The various configuration types returned different download (i.e., response) sizes. That is, the headers for each of the different scripting environments returned varied in size. As Figure 6 illustrates, PHP 5.0.4 pages that ran under both the Apache Web Server and Microsoft IIS6 and

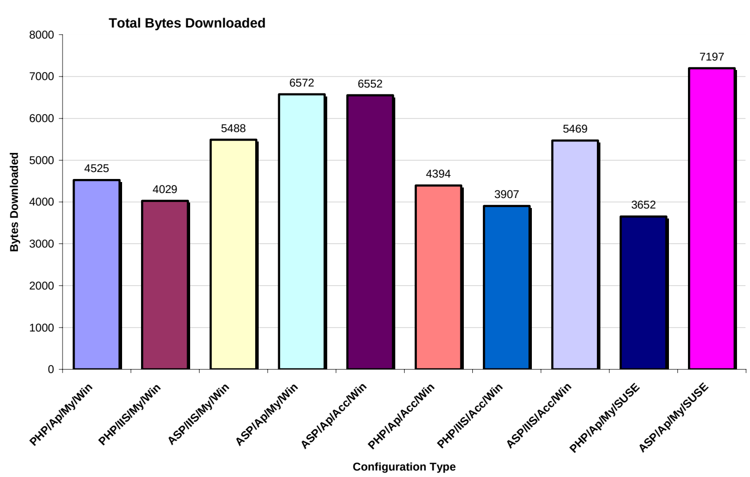

Figure 6: This graph illustrates a difference in download size among the different configuration types benchmarked.

using Microsoft Access 2003 files for data storage downloaded significantly smaller amounts of data compared to the other configuration types. The other configurations performed consistently suggesting the header information returned doesn't deviate much in size. The abnormity in the data set suggests an error in processing, however, JBlitz Professional didn't produce any related error messages.

Figure 8) shows average response time in seconds, which reflects how long, on average, it takes to receive each response. The time was measured from when a connection had been made to when the entire response was received. This is also a reflection of all of the download events that occurred whether or not they produced error. Longer connection times mirror resulting averages for response time, which in turn reflects overall response time (Figure 5).

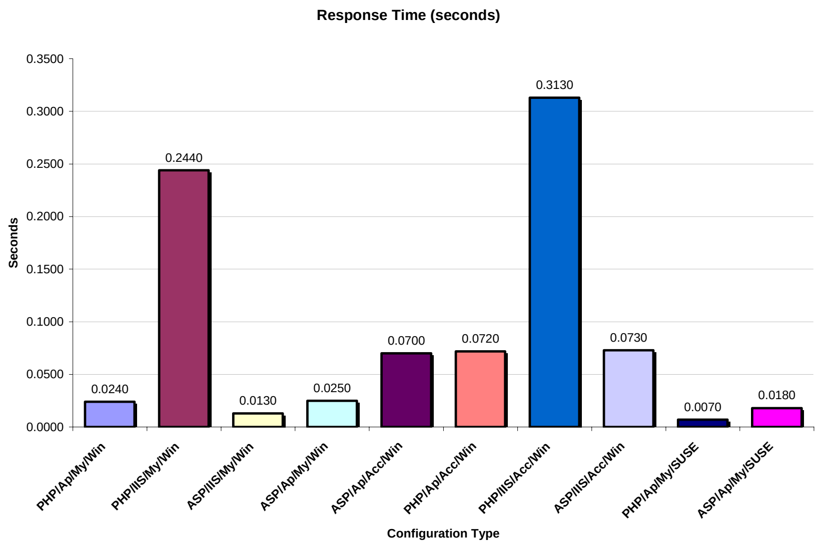

Figure 8: This graph shows the average response time from connection to the completion of downloading data.

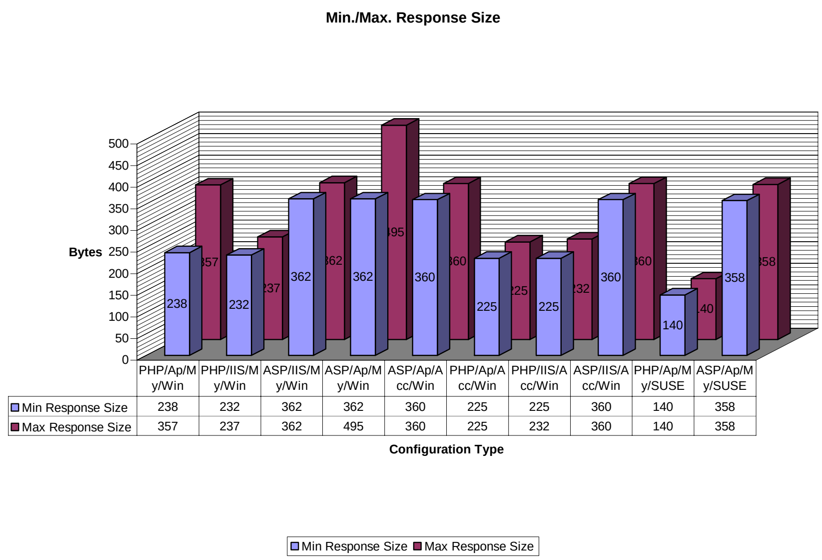

Figure 9: The data shown in this graph represents a comparison between minimum and maximum response data in bytes.

During the benchmarking, data downloaded varied in size. Each response delivered to each virtual user was calculated and then averaged. The data summarized in Figure 9 shows differences in minimum and maximum response sizes. While most of the responses were consistent, some response data collected did show signs of variation within a few configuration types. The data does not reflect any correlation to errors in data consistency or network overload.

In Figure 10, the average length of a response is shown. This data includes successful hits as well as any errors encountered. The mean response size shows direct correspondence with the total of downloaded data depicted in Figure 6.

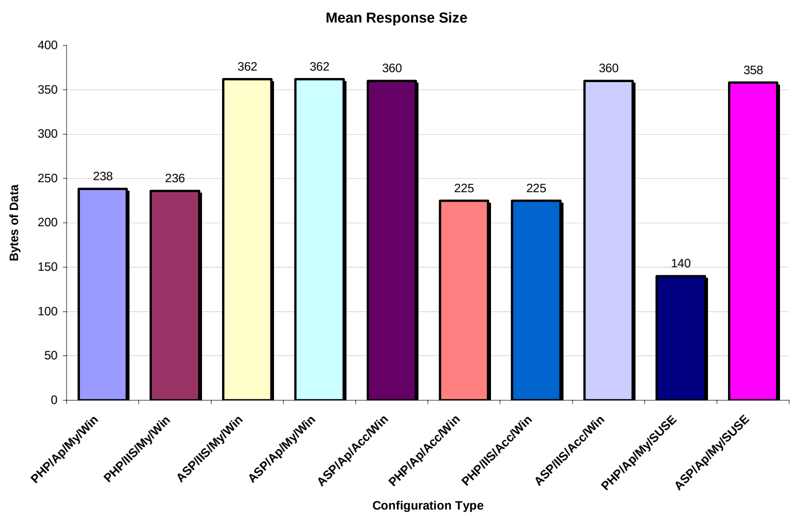

Figure 10: This graph shows the average length of a response for both errors and successful hits.

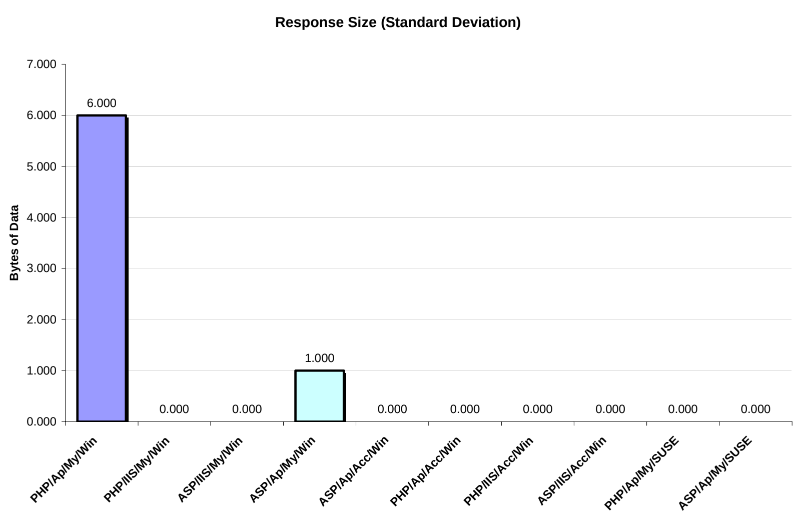

Figure 11: This graph shows the standard deviation in data length for all responses received

Figure 11 shows standard deviation in response length for all response headers received by each virtual user. This data also reflects directly upon how many errors were received during the course of benchmarking each configuration type. These errors reflect traffic errors within the local network and IP address translation errors encountered and reported in Figure 4.

CONCLUDING REMARKS

From the data gathered during the benchmarking of the two outlined web applications some assumptions can be made. First, from the transport of data size and speed of the transaction process it can be assumed that there is a direct benefit to accessing data from an outside source through a direct file transfer rather than a connection to a database server. The connection to a database server through whichever means, whether it is a Data Source Name (DSN) or direct TCP/IP connection to the database server (DSN-less), takes extra steps to process. That is, it creates more network traffic and overhead to connect to a database server than to access data directly from a file. To solidify this assumption, more benchmarking should be performed related to data access. There are many ways to import data into a web application. Data can be accessed through text files, spread sheets, Microsoft Access files, and many different flavors of database server such as Oracle 10g, PostgreSQL, Microsoft SQL Server, MySQL, and Sybase to name a few.

It can also be assumed, secondly, that web application run times will vary according to error rates. An error prone web application will take longer to render than one that is precisely developed and used within the confines of well established technologies suited for the web application. That is, and thirdly, a web application is better suited to the environment it was developed for. For example, it would not be in a developer's best interest to develop a web application to run under a Linux environment and to have that web application depend upon a Microsoft Access data file for its content. This type of configuration is currently not supported by the environment. It is possible to develop such connectivity between Microsoft Access, the Apache Web Server, and whichever scripting language is used. Why go through the trouble and expense to develop that sort of connectivity when the alternative technology exists to make the development easier in that scenario?

In this research, very basic web applications were generated using Active Server Pages (ASP) and Personal Homepage (PHP). In order to further support these assumptions, the opportunity to further investigate the issues related to dynamic web technologies presents itself. Further study using more scripting languages and comparing more directly to each of the languages tested. Investigation of the performance issues surrounding dynamic web technologies related to different hardware configurations must be examined. This research was performed on a dual 1 GHz processor with 512MB RAM. Hardware that conforms to configurations used in today's web hosting provider organizations should be examined. The hardware used should be up to date and current with the industry.

In closing, a final assumption can be made in a general sense. The needs of the web application used will dictate the hardware, software, operating system, and web server technologies used. In order to achieve the best performance possible native technologies need to be utilized. Using native web technologies for precisely developed web applications will perform optimally as well as incur the least expense in development and web hosting costs.

ACKNOWLEDGEMENTS

Special thanks and appreciation is extended to Dr. John Seydel for his toleration of many interruptions and annoyances. Thanks are also extended to Dr. Robyn Hannigan, Gail McDonald, Betty Pulford, and Ruth Greenfield of the McNair Achievement Program for their continued support throughout the course of this research. Last, but certainly not least, appreciation is extended to Kris Williams and Todd Reed for their help in attaining the equipment needed to perform these benchmarks.

This research was funded, in part, by the ASU McNair Achievement Program and by a grant from the US Department of Higher Education (P217A0300001 to Hannigan and Sustich) and by the Department of Computer and Information Technology, Arkansas State University.

REFERENCES

Apache HTTP Server Project. (n.d.). Retrieved Jun. 9, 2005, from HTTPD Web site: http://httpd.apache.org/.

ASPMaker (n.d.). Retrieved Jun. 30, 2005, from ASPMaker Web site: http://www.hkvstore.com/aspmaker/.

Huang, Z. (n.d.). MySQL to Access / Access to MySQL Converter. Retrieved Jun. 30, 2005, from MySQL to Access / Access to MySQL Converter Web site: http://www.fonlow.com/zijianhuang/dbconverter/index.html.

Java Runtime Environment. (n.d.). Retrieved Jun. 30, 2005, from Download Java 2 Platform, Standard Edition, v 1.4.2 (J2SE) Web site: http://java.sun.com/j2se/1.4.2/download.html.

JBlitz Professional 4.2. (n.d.). Retrieved Jun. 30, 2005, from JBlitz Professional Home Web site: http://www.clanproductions.com/jblitz/pro/.

Microsoft Office Developer Center. (n.d.). Retrieved Jun. 30, 2005, from Active Server Pages Web site: http://msdn.microsoft.com/office/understanding/frontpage/infocenter/asp/default.aspx.

Microsoft Office Online. (n.d.). Retrieved Jun. 30, 2005, from Access 2003 Home Page Web site: http://office.microsoft.com/en-us/FX010857911033.aspx.

Microsoft Windows Server System. (n.d.). Retrieved Jun. 30, 2005, from Windows Server 2003 Home Web site: http://www.microsoft.com/windowsserver2003/default.mspx.

MySQL. (n.d.). Retrieved Jun. 30, 2005, from MySQL Web site: http://www.mysql.com.

PHP Hypertext Preprocessor. (n.d.). Retrieved Jun. 29, 2005, from the PHP Web site: http://www.php.net/.

PHPMaker (n.d.). Retrieved Jun. 30, 2005, from PHPMaker Web site: http://www.hkvstore.com/phpmaker/.

SuSE Linux Enterprise Server 9. (n.d.). Retrieved Jun. 30, 2005, from SUSE LINUX Enterprise Server Web site: http://www.novell.com/products/linuxenterpriseserver/index.html?sourceidint=productsmenu_sles.

Security Space. (n.d.). Retrieved Jul. 6, 2005, from Security Space Web site: http://www.securityspace.com/sspace/index.html.

ServerWatch, (n.d.). Server Compare. Retrieved Jun. 26, 2005, from Server Compare: Comparison of Apache and Microsoft IIS6 Web site: http://www.serverwatch.com/stypes/compare/index.php/compare2_17755,18097.

Shiloh Consulting (1995, Nov). Performance Benchmark Comparison of UNIX Web Servers Using API and CGI External Gateways: Netscape Communications Server 1.12 (with NSAPI and with CGI), Open Market Non-Secure Webserver 1.1-eval, and NCSA httpd version 1.4.2. Retrieved Aug 04, 2005, from http://wp.netscape.com/comprod/server_central/performance_benchmarks.html.

Sun Java System Active Server Pages 4.0. (n.d.). Retrieved Jun. 16, 2005, from Sun Java System Active Server Pages 4.0 Web site: http://www.sun.com/software/chilisoft/.

Web Server Survey. (n.d.). Retrieved Jul. 11, 2005, from Web Server Survey Web site: http://www.securityspace.com/s_survey/data/200507/index.html.

Welcome to Mindcraft. (n.d.). Retrieved Jul. 24, 2005, from Mindcraft Inc. Web site: http://www.mindcraft.com/.

Whittman, A. (1999). Lies, Damn Lies, and Benchmarks. *Network Computing*, 10 Issue 11. Retrieved Jul 16, 2005, from http://ejournals.ebsco.com/Home.asp.

Windows Server System. (n.d.). Retrieved Jul. 11, 2005, from Internet Information Services Web site: http://www.microsoft.com/WindowsServer2003/iis/default.mspx.