EVALUATION OF SERVER-SIDE TECHNOLOGY FOR WEB DEPLOYMENT

Dr. Alexander Pons, University of Miami, apons@miami.edu

ABSTRACT

The deployment of Web applications consisting of dynamic content requires the selection of implementing technologies. These technology options form the bases for selecting Web server execution methods and data sources in the form of database technology. In this paper, we address the issues concerning the development of dynamic Web page rendering and which technology (execution model and database) offers the best approach regarding ease of installation and configuration compared to their performance. Our study focuses on the leading technology of ASP/JSP and SQL Server/Oracle to determine which combination renders greater execution speeds, while addressing the complexities in establishing a functional Web application.

Keywords: Web Technology, Database, Oracle, SQL Server, ASP, JSP

INTRODUCTION

Server-side programming via the Web has traditionally been done using CGI (Common Gateway Interface) scripts. These scripts, usually written in C or Perl, would run on the server each time they were called by a client. A key drawback with CGI is that when multiple clients requested the CGI script, it would run multiple times, leading to slowdowns and increased memory usage on the server. Two separate technologies have emerged as a replacement to CGI scripts, Active Server Pages (ASP) and Java Server Pages (JSP).

Microsoft introduced the first technology, ASP (5), in 1996. The server uses ASP code to generate HTML code dynamically. The ASP code is written in either Visual Basic script or Java Script. The script runs without the need for any compilation, allowing for rapid Web-application development. The typical ASP transaction operates in the following manner: A client browser requests a Web page from a server containing ASP code. The server executes the ASP script, which may obtain information from a database in generating a dynamic HTML page, which is then returned to the client's browser. Although, these interpretive code scripts are compliable to COM (Common Object Model) and integrated into some Web servers, we address in our study the uses of interpretive code script, since not all Web servers support this model.

The second technology, JSP (3, 6), is Sun Microsystems's answer to Active Server Pages. JSP was introduced in 1999. A JSP page will contain standard HTML and scriptlets that will interact with either a Java servlet or bean. When a client makes a JSP request, the server will execute the scriptlet and generate dynamic HTML to be returned to the client. The use of servlets or beans allows for greatly increased functionality for the server-side script. Also, since JSP is part of the overall Java platform, this technology is platform independent.

The use of these two technologies has increased dramatically in recent years. The argument over which approach is better has its roots in the ongoing battle between Microsoft supporters and Java supporters. Each side claims to have the best solution for a variety of reasons. This study attempts to answer the following questions. Which technology ASP or JSP performs best under client loads? Which database SQL Server2000 or Oracle8i combined with both of the previous technologies actual has overall higher performance?

WEB TECHNOLOGY

Web technologies at various levels are required in the development and deployment of Web–applications that support dynamic Web content, while adhering to an extendible architecture. The decomposition of Web applications into three layers of functionality increases its flexibility and scalability; these layers are extremely significant in evolving the application in various directions. The framework for such layering is the Client/Server paradigm, which is one of the most dominant concepts in Information Technology. These application layers and the corresponding tiers are shown in Figure 1. The middle and backend tiers can contain more than one Web Server and Data Server respectively, expanding to an n-tier design.

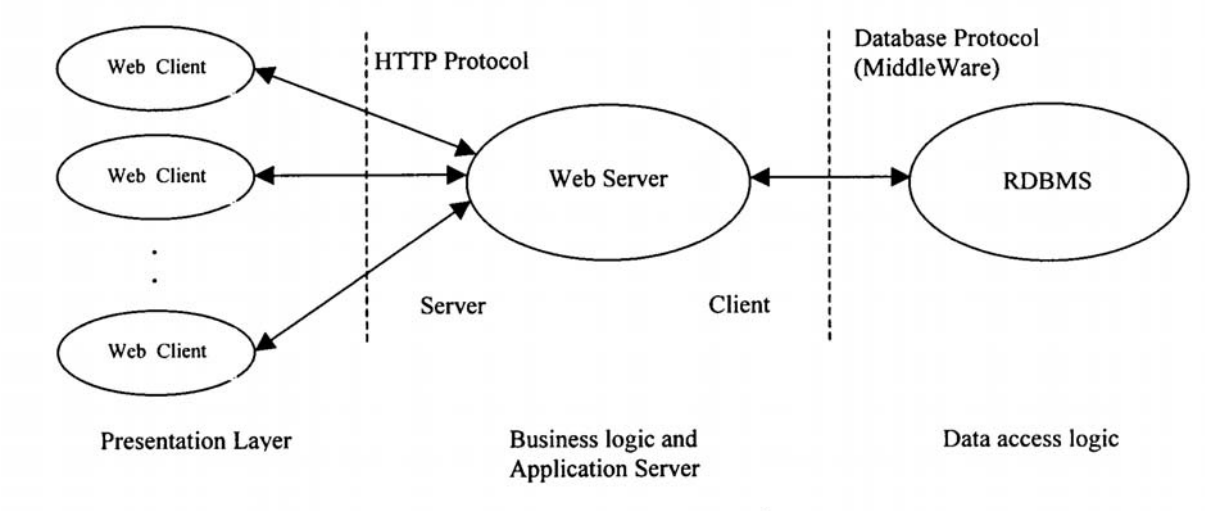

Figure 1. Three-Tier Web Design

- Presentation layer - handles how the user interacts with the web-application; usually implemented by providing an easy to use graphical user interface (GUI) with appropriate front-end data validation.

- Business logic - the mechanism or business rules of the web-application referred to as the application server. This is the back-end Web Server that provides the HTML pages and contains the logic corresponding to the application. Its data requirements are satisfied when it acts as a client to the database server who provides the data access logic.

- Data access logic - handles the storage, retrieval, and modification of data, while verifying data integrity. This is most often accomplished with a relational database management system (RDBMS) that becomes a server to the Web Server.

The exchange of data between the tiers (front-middle-back) is conducted from the Web Client to the Web Server to the RDBMS. The client communicates with the middle tier using standard communications protocols such as TCP/IP. The middle tier interfaces with the backend RDBMS using standard database protocols or by means of database middleware (thus making the solution independent of the RDBMS). The middle tier provides basic message switching and contains the business rules of the application. It acts upon the clients' request, applying business logic while invoking database request and handling the database response, then applying further business logic and generating a client page.

The Internet back-end technologies normally consist of the Web Server and Data Server (RDBMS). The Web Server provides the HTML pages and contains the logic or business rules corresponding to the web-application. Server-side coding extensions implement the logic portion of the application, which processes client requests, interacts with the database server, and renders dynamic web page content. These server-side extensions normal constituted writing CGI (Common Gateway Interface) programs in a language like Java, C, C++, Perl, and others, which are either compiled or interpreted. These extensions respond to form submissions, realize the application logic and processes database data. Alternative mechanisms for developing server extensions have been created which address the performance pitfalls of using traditional CGI processes. The primary concern with CGI is the resource cost associated with process activation and startup. Therefore, several Web Servers support the Internet Server Application Programming Interface (ISAPI) that enables the programmable logic to reside within the Web Server's process space-such logic would be loaded, once and for all, on first demand. It considerably increases the Web-application’s response-time and the number of clients that can be simultaneously serviced.

The Web Server is located on the middle tier and is used to distribute the client logic and integrate client sessions with the business logic using CGI/ISAPI. The business and data access logic should be modularized enabling it to be distributed over multiple machines. This has given raise to the n-tier concept of multiple Web Servers and Data Servers. The Web Server and Data Server can be hosted on the same physical system, but for performance advantages they are often placed on different systems and communicate to exchange data.

EXPERIMENT IMPLEMENTATION

The experiment setup required two Web servers that would host ASP and JSP scripts. During the installation of Windows 2000 Server, an option allows the inclusion of the Web server Internet Information Server 5.0 (IIS) (11) that offers the processing of ASP pages. The processing of JSP pages requires a compatible Web server; the Apache-Tomcat (1) Web server was selected and installed on another machine. These setups support the execution of ASP pages under IIS and Windows 2000 server and JSP pages under Apache-Tomcat and Windows 2000 server. The

installation and setup of these servers was accomplished in a relatively short amount of time and did not involve any major problems. In addition, two other machines were setup running the SQL Server (2, 4, 12) and Oracle8i (7, 8, 9, 10) DBMSs. These databases were setup for access to a user's schema, which contained two linked tables.

STUDY SETUP AND REULTS

Two tables were created in respective schemas under each database. The first table named *Person* contains the columns first_name, last_name, SSN, and address, with SSN being the primary key. The second table *Phone* contains the columns phone_number, and SSN with SSN as a foreign key constraint and their combination as the primary key. A total of 1000 records were placed in the Person table and 2500 records in the Phone table.

Next, we developed two separate ASP pages, which are required to access these two databases, using ADO technology. Both pages execute the same query on the respective databases, "SELECT * FROM Person, Phone WHERE Person.SSN=Phone.SSN ORDER BY Person.last_name." The results of this query are then displayed on the client browser completing the ASP transaction. To exercise the Java interface, two JSP pages were created with each calling a respective Java Bean to connect to each database and display the query results on the client browser. The same query performed in the ASP pages was used for the JSP pages.

We also added page timers to each of the four pages; in order to record and display the time required to fully render the Web pages. Additionally, redirect pages are used which display the time in milliseconds and initiate a pop-up window to the ASP and JSP pages. By comparing the initial time on the redirect page to the time after the ASP/JSP page is loaded, we can compute the time to load the individual pages and compare the performance of the Web server and database technologies.

In order to test the connection results from a client browser, a local area network connection was established directly between the client and server machines. This allows for a more accurate test, focusing on the amount of time necessary in loading the page, while minimizing any interference associated with the connection as well as any external DNS lookups.

A final component was required for our study, a stress tool that would perform in parallel several client requests of the Web servers, which in turn would exercise the databases. We developed a C++ based tool that would launch a user specified number of threads, each requesting the same Web page from the Web server under study. Table 1, contains the average time in milliseconds when rendering one, ten, and one hundred dynamic Web pages utilizing the various technologies. The result values consist of the average time to display a single Web page requested one hundred times.

Connections	ASP - SQL Server	ASP –Oracle	JSP - SQL Server	JSP – Oracle
One Page	192	185	231	258
10 Pages	845	812	1067	1189
100 Pages	7967	7508	9045	10781

Table 1. Page rendering time for different technology combinations

CONCLUSION

This simple test has shown that with one, ten, and one hundred simultaneous connections, the ASP pages perform slightly higher than the JSP pages. The results also show that there is no significant difference between the two databases. In addition, it is our opinion the ASP platform is easier to setup and install. However, by using this solution we are locking ourselves into Microsoft's Internet Information Server, which has many highly publicized security flaws and vulnerabilities. For the number of connections tested in this study, a configuration that employs SQL Server 2000 compared to Oracle is superior. From a purely performance standpoint, the ASP-Oracle combination was slightly higher, but the ASP-SQL Server combination is a viable selection for rapid Web-application deployment.

REFERENCES

1. Apache Web Server. (2003). http://jakarta.apache.org/tomcat.
2. Blackburn, I., Conard, J., and et. al. (2000). Professional ADO 2.5 Programming. Wrox Press Inc.
3. Brown, K., Craig, G., and et. al. (2001). Enterprise Java Programming with IBM WebSphere. Addison-Wesley.
4. Conte, P. and Otey, M. (2000), SQL Server 2000 Developer's Guide. Osborne McGraw-Hill.
5. Homer, A. (2003). Professional ASP Techniques for Web Masters. Wrox Press Inc.
6. Java.Sun.com online Documentation 2003 Oracle Corporation
7. Kyte, T. (2001). Expert one-on-one Oracle. Wrox Press.
8. Loney, K. and Koch, G., (2000). Oracle8i The Complete Reference. McGraw-Hill Inc.
9. Microsoft MSDN (2003). http://msdn.microsoft.com/
10. Sanders, R. (1998). ODBC 3.5 Developer's Guide. Osborne McGraw-Hill.
11. Trent, R. (2003). Windows 2000 IIS 5.0: A Beginner's Guide. Osborne McGraw-Hill.
12. Vieria, R. (1999). Professional SQL Server 7.0 Programming. Wrox Press.