

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/237337386>

A Performance Comparison of Web Development Technologies to Distribute Multimedia across an Intranet

Article · January 2003

CITATIONS

3

READS

1,730

3 authors, including:



Alfredo Terzoli

Rhodes University

131 PUBLICATIONS 500 CITATIONS

SEE PROFILE

A Performance Comparison of Web Development Technologies to Distribute Multimedia across an Intranet

D. Swales, D. Sewry, A. Terzoli
Computer Science Department

Rhodes University
Grahamstown, 6140

Email: g97s5130@campus.ru.ac.za

Tel: (046) 6038640

Fax: (046) 6361915

Topic: Broadband Services

Sub-Topic: Multimedia

ABSTRACT - In recent years the World Wide Web has transformed into a dynamic, interactive medium, exposing a proliferation of on-line services that distribute large quantities of multimedia. This has increased awareness of the need to select an appropriate Web programming technology when creating Web-based services. This paper compares three dynamic Web programming technologies from the point of view of performance in multimedia distribution. In particular, this paper examines Sun Microsystem's Java Server Pages (JSP), Microsoft's Active Server Page's (ASP) and the more recent ASP.NET. The comparison is based on testing applications that distribute multiple images from an Oracle 9i database to Web-enabled clients. Previous research conducted by Oracle indicates that JSP and the underlying Java platform outperform ASP. The results in this paper, however, do not agree with Oracle's statement as JSP tends to be marginally slower than ASP.

I. INTRODUCTION

In the past, rich media, such as images, audio and video was stored in directory structures as traditional databases were not equipped to deal easily with these complex data types. Currently, few databases have evolved to meet the challenge of storing multimedia and have provided developers and database administrators with the ability to manage that content within a relational scheme [1].

At the same time, the World Wide Web (WWW) has transformed from a static content-distribution medium to an interactive, dynamic medium [2]. Content on the Web is now dynamically generated and often used to distribute multimedia from back-end databases to Web-enabled clients. This has resulted both in an appreciation for the value of multimedia and a realization of the challenges in distributing that multimedia to Web-enabled clients. Emerging Web and e-commerce applications found in diverse areas need effective, efficient, and reliable management of multimedia content [3].

From the Web developers' perspective, dynamic Web content requires a decision regarding the Web programming

technology to be employed in creating a Web-based service or product. This decision is based on several factors. Among the factors are ease of programming, richness of features, maintainability, reliability, and performance. Since the Web is inherently interactive, performance is a key requirement, and often demands careful analysis of the system [2].

In this paper, the performance of three Web programming technologies, namely JSP, ASP, and ASP.NET are compared and contrasted based upon an application that distributes images from an Oracle 9i database to a client's Web browser. The study was motivated by the need to make a choice for developing software that would support a Web-based application for distributing multimedia over networks. The comparison is accomplished by developing a prototype application, testing it, and eliciting the foremost factors contributing to the final performance and suitability of the different technologies.

The rest of this paper is organized as follows. In Section II a motivation for conducting such a comparative study is provided, Section III gives a brief description of the various dynamic Web platforms to be compared. Section IV describes the testing environment and methodology, Section V provides the testing results and comparisons in detail. Finally, Section VI provides a summary and conclusion.

II. MOTIVATION

The work described in this paper focuses on media management and delivery from an Oracle9i database. The programming technologies used to provide these services centre on the Java 2 platform Enterprise Edition (J2EE), ASP and newly deployed ASP.NET architecture.

Oracle offers strong support for developing in Java and provides developers with an end-to-end Java solution for creating, deploying, and managing Java applications [1]. Furthermore, the Java APIs are designed to meet the increasing demand for multimedia, supporting audio, video, animations and telephony [4].

By contrast, until now, Oracle has found Microsoft's ASP architecture and the underlying Windows platform unsuitable to deliver the required levels of performance, scalability, reliability and flexibility of an enterprise development platform such as J2EE [5].

However, little research has been done on evaluating each Web technology against the other, or on assessing in an unbiased way their viability as a platform to distribute media rich Web applications. The intention of this paper and

This work is under the auspices of the Centre of Excellence in Distributed Multimedia at Rhodes University, which is funded by Telkom, Comparex Africa and Letlapa Mobile Solutions

ongoing research is to do an evaluation of these technologies with regard to media management and distribution from an Oracle 9i database. This paper reports on a specific aspect of the overall research initiative, namely image distribution. Other areas of research include audio and video management and distribution as well as insight into special purpose Integrated Development Environments (IDEs) developed by Oracle and third party vendors.

III. DYNAMIC WEB PLATFORMS

There are a multitude of different technologies employed to produce dynamic Web-based applications. CGI, Java Servlets, JSP, ASP, ASP.NET, PHP, and Java Beans are among the many technologies. This paper focuses on three technologies: JSP, ASP and ASP.NET.

A. Java Server Pages (JSP)

JSP is a Java-based technology that provides Web developers with a framework to create dynamic content using HTML and Java code, which is secure, fast, and independent of server platforms [6]. A JSP file contains standard HTML, interspersed with Java code used to generate dynamic content. JSP technology adheres to the "Write Once, Run Anywhere" philosophy of the Java architecture. Pages built using JSP technology are typically implemented using a translation phase that is performed once, the first time the page is called. The page is compiled into a Java Servlet class and remains in server memory, so subsequent calls to the page results in a shorter response time [7].

B. Active Server Pages (ASP)

Microsoft introduced ASP, along with its Internet Information Server 3.0, to allow developers to create Web pages that can interact with databases and other applications. Microsoft defines ASP as "an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions." [8]. Most ASP pages are written using VBScript (a subset of Microsoft's Visual Basic programming language) or JScript, but support for languages such as Perl and Python are available through third-party vendors [9].

ASP is similar to JSP in that they both render dynamic content and separate HTML from server-side code. ASP has however, three important limitations. Firstly, ASP is currently available only with Microsoft's Internet Information Server (IIS), running under the Windows operating system. Secondly, JSP code is written in Java, which is better suited to complex applications that require reusable components [6]. Lastly, ASP pages need to be recompiled on every request and do not store cached, compiled pages within server memory.

C. ASP.NET

ASP and Windows DNA, being based on the Component Object Model (COM) and Win32 API technologies, has not provided a very coherent architecture for modern distributed applications, whereas with Java 2 Enterprise Edition (J2EE), Sun had a suite of technologies that developers could

employ, beginning with Standard Edition projects and scaling up to full Enterprise JavaBeans [10].

With the advent of Microsoft's .NET framework, ASP.NET has shifted from traditional scripting to a fully-fledged programming architecture. Programmers can employ Visual Basic (VB), C# and a variety of other languages found within ASP.NET. ASP.NET is now an object-oriented Web application development platform, and has seen many improvements with regard to error handling, memory management, scalability, and flexibility to name a few [10].

IV. TESTING ENVIRONMENT AND METHODOLOGY

As previously mentioned, this paper provides a performance comparison of JSP, ASP, and ASP.NET with regard to image distribution over a network. Performance comparisons are based upon the amount of time taken to retrieve 1000 images from an Oracle 9i database and display them within a client's Web browser.

A. Testing Environment

Identical applications were developed and deployed in JSP, ASP and ASP.NET. Figure 1 depicts the three-tier architecture used during the performance comparison. The performance tests were conducted in a closed environment: the database, Web server, and client machines constituted an isolated LAN. The JSP application was implemented using the Apache Tomcat server whereas ASP and ASP.NET applications were implemented using Internet Information Server (IIS). The database and client machine remained identical within the developed prototype.

The process of image retrieval is initiated from a client's request for a Web page to the Web server. Images are retrieved from the database and passed to the Web server upon which they are finally distributed to the client. Times are captured on the Web Server and taken from the moment a client's request is received until all the images have been distributed from the Web server to that client.

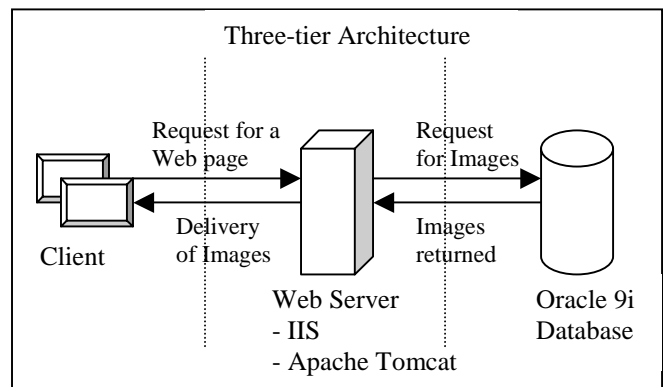


Figure 1. Three-tier architecture of the testing environment.

Each image stored within the Oracle 9i database is approximately 55KB in size and stored as an Oracle interMedia ORDImage data type.

B. Implementation Methodologies

The process of image retrieval is essentially the same for the various Web technologies.

Firstly a connection is established to the database using the appropriate database drivers. The JSP application uses the Oracle thin database driver whereas both ASP and ASP.NET applications use the Oracle OLE DB driver. It should however be noted that various other database drivers can be used to access the Oracle 9i database and will be tested in future performance comparisons.

Secondly, an in-line SQL query is used to populate server-side objects with images from the database. The JSP application uses the Oracle ResultSet object whereas the ASP application uses the ADO (ActiveX Data Object) RecordSet object. ASP.NET applications have two objects that developers can utilize to retrieve images from an Oracle 9i database. The first uses a connection oriented DataReader object which is similar to the ASP RecordSet object. The second uses the DataSet object which provides a disconnected model for database retrieval.

Lastly, the images are outputted to the client's browser by iterating through the server-side object and the database connection is closed.

C. Timing Methodologies

Times are expressed in milliseconds and time measurements are repeated 30 times to reach statistical stability.

Timing methodologies vary according to the architecture tested. The timing methodology used within the ASP.NET uses a class file that captures results in measurements of a tick (1/10 of a millisecond). JSP timing results were captured using the system's current time in milliseconds. ASP, however, cannot use the system's time as it is taken in seconds. Therefore the timing methodology used within ASP applications was implemented using an external component that captures times to within 5 decimal places of a second.

V. RESULTS AND COMPARISON

The main metric used in the comparison was the amount of time taken to display 1000 images, from the issue of the request to the moment in which all images have been displayed in the client's browser. Figure 2 depicts the measured times taken to display all the images in a series of 30 experiments using ASP.

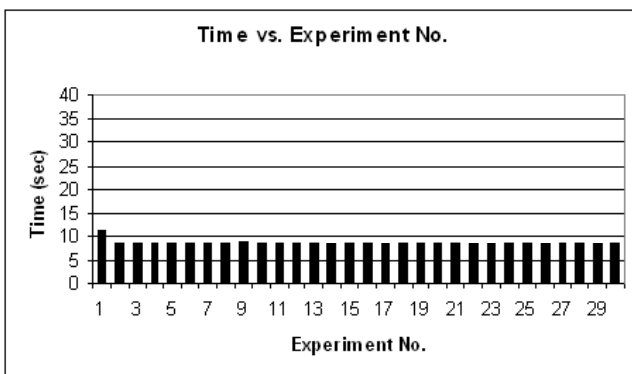


Figure 2. Time vs. Experiment No. – ASP (RecordSet Object)

As shown in Table 1, the amount of time taken to complete a single experiment using ASP (RecordSet Object) ranged from 11.3 seconds (the first request) to 8.4 seconds, with a mean time of 8.6 seconds. The first request remains the highest, where caching and initial server processing could be the determining factor for a larger elapsed time.

Table 1. Timing Results – ASP (RecordSet Object)

	Time (sec)
Maximum Time (sec)	11.3
Minimum Time (sec)	8.4
Average Time (sec)	8.6

Figure 3 depicts the same test conducted using the ASP.NET DataSet object.

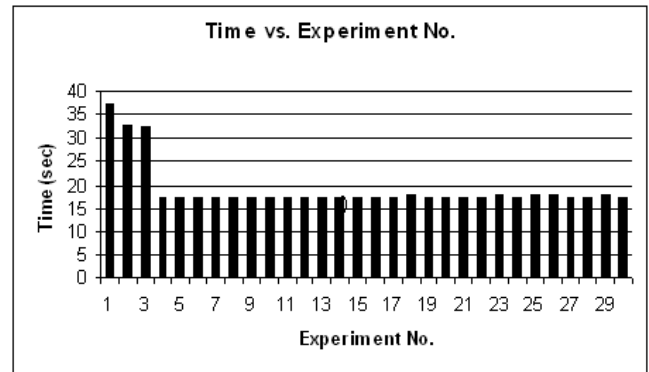


Figure 3. Time vs. Experiment No. – ASP.NET (DataSet Object)

Table 2 shows the mean time of 19.1 seconds which was approximately 10 seconds slower than the same test conducted within ASP. The longest periods of time taken to distribute all the images focused on the first three experiments. These larger time values could be caused by caching as well as server parsing and compiling on the first request.

Table 2. Timing Results – ASP.NET (DataSet Object)

	Time (sec)
Maximum Time (sec)	37.3
Minimum Time (sec)	17.3
Average Time (sec)	19.1

Figure 4 depicts the same test conducted within ASP.NET using the DataReader object in place of the DataSet object.

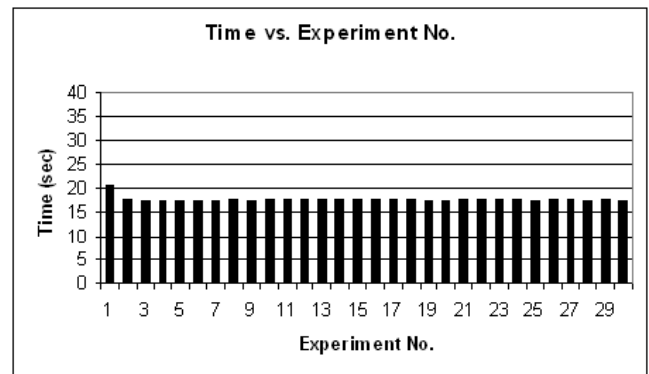


Figure 4. Time vs. Experiment No. – ASP.NET (DataReader Object)

Table 3 shows a mean time of 17.6 seconds which remains significantly slower than ASP but has a marginal increase on elapsed time when contrasted to the DataSet object

Table 3. Timing results – ASP.NET (DataReader Object)

	Time (sec)
Maximum Time (sec)	20.5
Minimum Time (sec)	17.1
Average Time (sec)	17.6

Figure 5 depicts the time taken to serve 1000 images, per 30 experiments, using JSP.

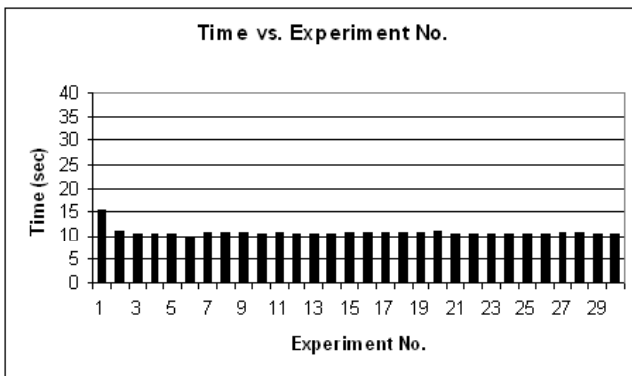


Figure 5. Time vs. Experiment No. – JSP

Table 4 shows the maximum time (the first experimental request) of 15.3 seconds and the minimum time of 10.1 seconds respectively. The mean across all 30 experiments was 10.5 seconds. Once again, Figure 4 shows a longer time to serve the first request, followed by relatively constant times. The translation phase that occurs at the first request in JSP may be the determining factor for the larger time value.

Table 4 Timing Results – JSP

	Time (sec)
Maximum Time (sec)	15.3
Minimum Time (sec)	10.1
Average Time (sec)	10.5

A. ASP vs. ASP.NET

The major difference between ASP and ASP.NET is the mean time taken to complete the 30 experiments. ASP had a mean of 8.6 seconds whereas the DataSet and DataReader objects used within ASP.NET had a mean of 19.1 and 17.6 seconds respectively. In order to standardize the results to ensure maximum reliability, the Web Server, database and OLE DB driver remained constant with the time taken to service the requests being the only independent factor. Therefore, the determining factor for the varied times elicited in the tests can only be due to the underlying architecture found within ASP and ASP.NET. As a result, it is clear that ASP outperforms ASP.NET with regard to media distribution (images) from an Oracle 9i database. As mentioned previously, these results may vary when alternative database drivers are used (specific for ASP and ASP.NET).

B. ASP.NET: DataReader object vs DataSet object

As mentioned, ASP.NET provides developers with the ability to utilize two ADO.NET objects when retrieving data, the DataReader and DataSet. Microsoft states that the DataReader object outperforms the DataSet object when numerous records are retrieved from a database and sequentially accessed [10]. From the results found in Table 2 and 3 the mean time of the ASP.NET DataSet object and the DataReader object was 19.1 and 17.6 seconds respectively. These results agree with Microsoft's statement.

C. JSP vs. ASP and ASP.NET

As illustrated by the results from Tables 1 to 4, ASP was found to outperform both JSP and ASP.NET in all areas of image distribution. The first request to the server, using JSP, was 4 seconds slower than ASP, 22 seconds faster than ASP.NET using the DataSet object and 5.2 seconds faster than ASP.NET using the DataReader object. The most important observation, however, was the mean time within JSP which was significantly faster than ASP.NET's DataSet and DataReader objects by 8.7 and 7.1 seconds respectively and was marginally slower than ASP by 1.9 seconds. As a result, ASP seems to be the most efficient Web technology to use when distributing multimedia (images) from an Oracle 9i database to Web-enabled clients.

It should however be noted that the performance measurement conducted within this paper focuses on a single aspect of image delivery. In the future more tests will be conducted using different database drivers and associated architectural and implementation changes in each technological environment.

VI. CONCLUSION

The work described in this paper focuses on media management and delivery from an Oracle9i database to Web-enabled clients. Previous research conducted by Oracle indicates that JSP and the underlying Java platform outperform Microsoft's ASP architecture. The performance tests conducted in this paper targeted multimedia (image) distribution in three Web development technologies, namely JSP, ASP and ASP.NET. The results do not agree with Oracle's statement as JSP tends to be marginally slower than ASP. However, both ASP and JSP significantly outperform Microsoft's newly deployed ASP.NET architecture.

REFERENCES

- [1] R. Ward. "Oracle® *intermediate*: Users Guide and Reference. Release 9.0.1." Part No. A88786-01, 2001.
- [2] V. Apte, T. Hansen, and P. Reeser. (2002, August). *Performance comparison of dynamic we platforms*. Volume 26, 888-898. Available: <http://www.computerscienceweb.com>.
- [3] M. Chittister. "Oracle® *intermediate* Java Classes: Users Guide and Reference." Release 9.0.1 Part No. A88785, 2001.
- [4] M. Claypool, T. Coates, S. Hooley, E. Shea, C. Spellacy. "The Performance of Distance Education Video in Java." Computer Science Technical Report

Series Journal, Worcester Polytechnic Institute, pp. 22-23, June 2000.

- [5] Unknown. "Overview: Oracle9iAS Migration Kit for ASP." Unpublished. Jan 2002. Available: <http://otn.oracle.com/tech/migration/asp.html>
- [6] M. Hall, *Core Servlets and JavaServer Pages*. Upper Saddle River, NJ: Prentice Hall, 2000, pp. 9-11.
- [7] Wassef, M. *ASP vs JSP. Overview of the technology*. Available: <http://www.palestinewebdevelopers.com>, 2000
- [8] A. Michalas *et al.* (2000, January). *A comparison of multimedia application development platforms towards an object Web*. Computer Standards and Interfaces, Volume 22, 13-26. Available: <http://www.elsevier.com>.
- [9] Gladwin, L.C. *Background: ASP and JSP*. Available: <http://www.itworld.com>, 1999.
- [10] M. Ahmed *et al.* *ASP.NET Web Developer's Guide*. Rockland, MA: Syngress publications, 2002, pp. 46-49.

Dylan Swales is a post-graduate student who is currently reading for his Masters Degree in Computer Science at Rhodes University. He has been working in the area of Multimedia and Database driven Web sites for the past three years.