Performance study of several DBMS connectivity using different server scripting environments

Adel A. El-Zoghabi, Randa M. Youssef and Hesham A. Abdel-Hamid
Civil Eng. Dept., Faculty of Eng., Alexandria University Alexandria, Egypt
E-mail: zoghabi@yahoo.com

This paper aims at testing the performance of some of the HTML-embedded server scripts when querying some of the industry standard databases based on different situations derived from a three-tier, web-based medical information system. Active Server Pages (ASP), Java Server Pages (JSP), PHP Hypertext Processor (PHP) and ASP.NET were used in coding Web pages to test their performance in connecting, retrieving and displaying information from Microsoft Access 2000, Microsoft SQL Server 2000, ORACLE 9i and MySQL. The tests record the response time at the web server as a sign of performance. The study revealed that PHP was the best performing script with Access while ASP.NET was the best with SQL Server. ASP.NET was as good as JSP with ORACLE, better than others. ASP.NET, JSP and PHP were equally good with MySQL under Windows while PHP was better than JSP under Linux.

يهدف هذا البحث إلى إختبار كفاءة الأداء لبعض طرق برمجة صفحات الخادم فى جلب البيانات من بعض النظم القياسية لإدارة قواعد البيانات إعتمادا على حالات برمجية مستمدة من نظام معلومات طبى ثلاثى الطبقات قائم على الإنترنت. وقد تم إستخدام أكواد متشابهة من تقنيات برمجة الخادم الآتية: إيه إس بى(ASP) وجيه إس بى (JSP) وبى إتش بى (PHP) و إيه إس بى دوت نت (ASP.NET) لبناء صفحات ويب لقياس كفاءة أدائها فى الإتصال وجلب وعرض المعلومات من قواعد البيانات أكسس (Access) و خادم سيكويل (SQL Server) و أوراكل (ORACLE) و ماى سيكويل(MySQL) . وقد تم الأعتماد على زمن إستجابة جهة الخادم كعلامة لقياس كفاءة الأداء. وقد أظهرت النتائج تفوق أداء تقنية بى إتش بى فى الإتصال بأكسس فى حين تفوقت تقنية إيه إس بى دوت نت فى الإتصال بخادم سيكويل وتساوت تقنية إيه إس بى دوت نت فى الكفاءة مع تقنية جيه إس بى مع قاعدة بيانات أوركل متفوقة على باقى التقنيات وتساوى أداء كل من جيه إس بى و بى إتش بى و إيه إس بى دوت نت مع قاعدة بيانات ماى سيكويل تحت منصة ويندوز فى حين تفوقت بى إتش بى تحت منصة لينوكس.

Keywords: Dynamic web pages design, Server-side scripting, Three-tier architecture, Html-embedded server scripts

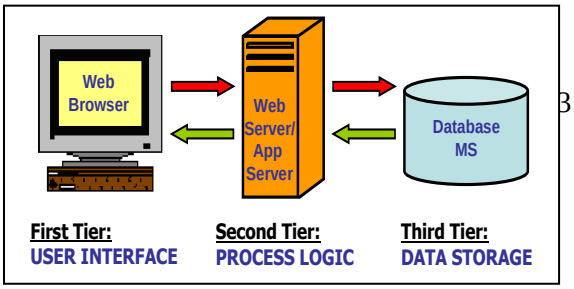
1. Introduction

The strategy of a thin Web client against a thick Web server in building dynamic Web applications is dominant nowadays [1,2]. This strategy depends mostly on the Web server, rather than the client, to access information sources and build customized Web pages on the fly. Little work is left for the browser, mainly formatting the page and responding to user events.

Many technologies were invented to implement Web page dynamism on the server side [3,4]. The basic concept of these technologies is to accept client request, connect to databases and other programs, build an HTML page on the fly, and send it back to the client. The page sent is server-code-free, pure HTML including only client-side scripts. This is often called three-tier architecture, as illustrated by fig. 1, consisting of the Web browser, the Web server and the database [5].

HTML-embedded scripts are the most popular server techniques. They are stateful, work in-process, separate the page logic from its layout, well integrated with the Web server and most important, easy to learn although they also still have disadvantages [6-9].

With Web server HTML-embedded scripts, an HTML writer can script an external component and format the output. The scripts is written inside the HTML page and executed when the page is requested. These technologies separate the layout and design from the page logic. These technologies, therefore, make server applications easier to create and maintain and more powerful.



Alexandria Engineering Journal, Vol. 43 (2004), No. 3, 331-340

Fig. 1. The three-tier architecture.

Some of the HTML-embedded scripting technologies, such as ASP are proprietary, working on Web servers of specific vendors. [9] Others are not proprietary such as PHP and JSP, which are not tight to a specific vendor [7, 8].

When developing a Web-based information system, it is needed to choose the best technologies to implement it. One of the most important factors determining the best technologies is performance [10]. Here, the performance of accessing the industry standard database management systems through the most commonly used Web server scripting technologies, based on different situations derived from this system, was tested.

2. Aim of the study

This study aims at determining and comparing the performance of accessing some of the industry standard DBMS through some of the most commonly used Web server technologies, based on different situations derived from a real already-developed, medical information system. This study focused only on data querying, so other aspects of accessing DBMS were not considered.

The DBMSs used in this study are: Microsoft Access 2000, Microsoft SQL Server 2000, ORACLE 9i and MySQL 3.25. The Web server technologies used are: ASP, JSP, PHP and ASP.NET.

3. Methodology

Eight different situations, numbered from A to H, were defined to cover almost all the situations in the system under study. The first three situations, A, B and C, select a single row from a single table. The next three situations, D, E and F, select multiple rows from a single table. And finally, the last two situations, G and H, select multiple rows from two joined tables. The details of these situations are described in table 1.

A number of server-side dynamic Web pages were written to test the different response time in each of the different situations. The code of these pages is similar although each of them was adapted to match a specific server technology and a specific database.

These Web pages were connected to the DBMS using the technique of highest performance available. ASP pages were connected to MS Access 2000 through an OLE DB and to other DBMS through ODBC. JSP pages were connected to MS Access 2000 through JDBC-ODBC bridge and ODBC driver and to other DBMSs through a JDBC driver type II. PHP were connected to MS Access 2000 through an ODBC driver and to other DBMSs through a native API. Lastly, ASP.NET pages were connected to MS SQL Server 2000 through a native API and to MS Access 2000 through an OLE DB and to the other two DBMSs through ODBC drivers.

Table 1
Criteria of different situations defined to test the performance

Situation	Base table(s)	Selected record(s)	Selection criteria
A	Single, 1000-row table	one record	Indexed field
B	Single, 1000-row table	one record	Non-indexed field
C	Single, 1000-row table	one record	Text matching
D	Single, 1000-row table	100 records	Indexed field
E	Single, 1000-row table	100 records	Non-indexed field
F	Single, 1000-row table	100 records	Text matching
G	Two inner-joined tables	100 records	One indexed field
H	Two inner-joined tables	100 records	Two indexed fields

An SQL 'SELECT' statement was used to access the database and retrieve the query re-

sults. These results were stored in an array and then a simple HTML page was created and pumped out to the Web browser.

The server time was determined at the beginning and at the end of page main script using the suitable functions. An extra piece of code was added to the page to calculate the time difference in milliseconds, which is the page response time, and save it to a text file. The time of running this extra piece of code was not calculated in the page response time.

The response time, which is the sum of time needed by the Web server to access the database and pump out the results, was recorded at the server side ,not the client side, to eliminate the effect of network traffic and the time consumed by the Web browser to parse the HTML. All the databases were loaded on the same PC on which the Web server was installed so the traffic time delay between the middle and third tiers were eliminated.

Each of these pages has been run 41 times. The response time of the first run was ignored to stabilize the system. The other 40 response times of the other 40 runs were calculated in milliseconds and numbered from 1 to 40.

Two operating systems were used: Microsoft Windows 2000 Server and Red Hat Linux 7.2. Under Windows, four DBMSs, Microsoft Access 2000, Microsoft SQL Server 2000, ORACLE 9i and MySQL 3.25, were installed and loaded with the same tables. ASP, ASP.NET, JSP and PHP similar pages were used to access the DBMS and to calculate the response time. ASP and ASP.NET were running within IIS 5.0 where JSP and PHP were running within Apache Web server.

Under Linux, only two DBMS out of the previously mentioned four were installed: ORACLE 9i and MySQL 3.25 because there are no Linux versions of the other two until the time of writing this paper. For the same reason, Only JSP and PHP pages were used and were running within Apache Web server.

The hardware configuration of the server machine used in this study was Intel PIII 600 MHz processor with 512 KB cache memory, 256 MB RAM and a free 20 GB of hard disk space.

Table 2
Statistical results for microsoft access 2000

4. Statistical methodology

The recorded data were fed into an IBM compatible PC loaded with Statistical Package for Social Science version 10 (SPSS 10) by importing the text files carrying the results to SPSS. The data were statistically described using the mean, standard deviation, and the corresponding 95% confidence interval.

For each DBMS, a t-test and/or one-way Analysis Of Variance ANOVA were performed to compare the mean response time of the pages that use different server technologies. The latter was followed by post-hoc comparison between each pair of means based on the least significant difference (LSD) [10]. For each Web server script, a t-test and/or one-way Analysis Of Variance ANOVA were performed to compare the mean response time of the pages connected to different DBMS. The latter was followed by post-hoc comparison between each pair of means based on the least Significant Difference (LSD).

5. Results

5.1. Effect of the web server scripting technology on the mean response time for each DBMS

5.1.1. Microsoft Access 2000

The recorded results, given by table 2 and illustrated by fig. 2, revealed that PHP had the lowest mean response time in all the predefined situations. This was followed by ASP.NET then ASP. JSP had the highest mean response time.

A one-way Analysis Of Variance ANOVA proved significant difference of means between the four scripting technologies. Post-hoc comparisons between each pair of means based on the Least Significant Difference (LSD) proved significant difference between means of all the individual pairs of scripting technologies for all the predefined situations.

5.1.2. Microsoft SQL Server 2000

The recorded results, as abstracted by fig. 3, revealed that ASP.NET had the lowest mean response time in all the predefined situations.

Situation	N	Mean for ASP, JSP, PHP, ASP.NET	Standard deviation	95% Confidence int. (lower bound)	95% Confidence int. (upper bound)
A	40	50.93, 55.95, 44.10, 47.95	1.56, 3.54, 3.32, 3.27	50.43, 54.82, 43.04, 46.91	51.42, 57.08, 45.16, 48.99
B	40	55.25, 57.83, 46.28, 50.85	2.44, 2.11, 2.21, 2.21	54.47, 57.15, 45.57, 50.14	56.03, 58.50, 46.98, 51.56
C	40	69.88, 72.93, 61.18, 65.18	1.18, 3.52, 3.12, 3.00	69.50, 71.80, 60.18, 64.21	70.25, 74.05, 62.17, 66.14
D	40	88.15, 89.25, 76.88, 80.97	4.35, 1.90, 4.14, 4.30	86.76, 88.64, 75.55, 79.60	89.54, 89.86, 78.20, 82.35
E	40	84.68, 87.20, 72.97, 77.00	5.62, 2.14, 5.55, 5.57	82.88, 86.52, 71.20, 75.22	86.47, 87.88, 74.75, 78.78
F	40	99.23, 101.80, 87.65, 91.90	2.63, 1.73, 2.63, 2.55	98.39, 101.25, 86.81, 91.08	100.06, 102.35, 88.49, 92.72
G	40	88.90, 95.23, 77.88, 82.03	3.03, 3.34, 2.54, 2.55	87.93, 94.16, 77.06, 81.21	98.87, 96.29, 78.69, 82.84
H	40	83.40, 89.43, 71.90, 76.00	3.18, 1.97, 3.50, 3.57	82.38, 88.79, 70.78, 74.86	84.42, 90.06, 73.02, 77.14

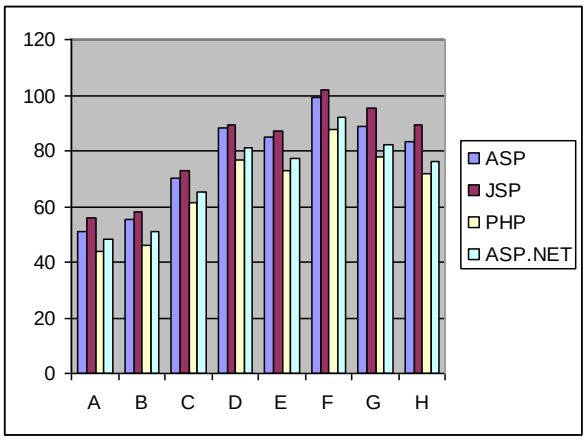


Fig. 2. The mean response time for Microsoft Access 2000.

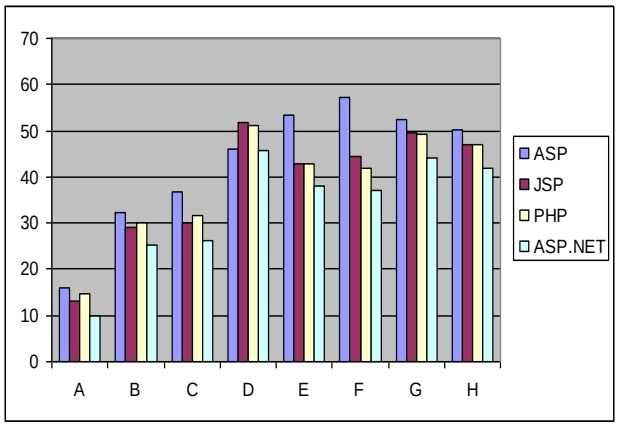


Fig. 3. The mean response time for Microsoft SQL server 2000.

This was followed by JSP then PHP. ASP had the highest mean response time.

A one-way Analysis Of Variance (ANOVA) proved significant difference of means between the four scripting technologies. Post-hoc comparisons between each pair of means based on the Least Significant Difference (LSD) proved significant difference means between ASP.NET and other scripts in all situations. Post-hoc comparisons also proved that the means difference between JSP and PHP was insignificant in all situations except situation A. The mean difference between ASP and other scripts was significant for all the predefined situations except situation A for ASP-PHP pair of means.

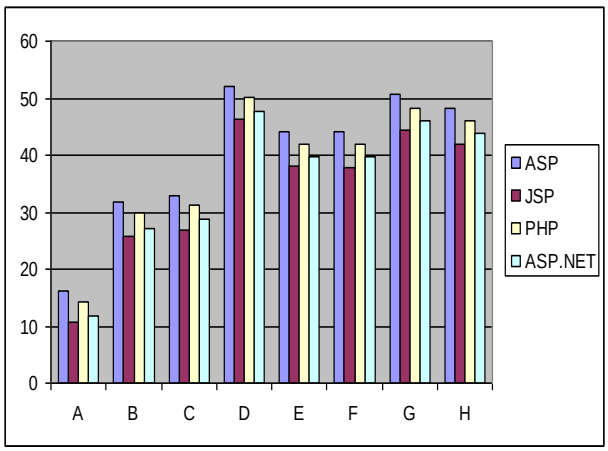


Fig. 4. The mean response time for ORACLE 9i under Windows 2000.

5.1.3. ORACLE 9i

- *Under windows 2000 Server O.S. .*The recorded results, as abstracted by fig. 4, revealed that JSP had the lowest mean response

time in all the predefined situations. This was followed by ASP.NET then PHP. ASP had the highest mean response time.

A one-way ANOVA proved significant difference of means between the four scripting technologies. Post-hoc comparisons between each pair of means based on the LSD proved that the mean difference between JSP and ASP.NET was insignificant in situations A and B, while, it was significant in the other situations. Post-hoc comparisons also proved significant difference between means of all the other individual pairs of scripting technologies for all the predefined situations.

- *Under Red Hat Linux 7.2 O.S.:* The recorded results, as abstracted by fig. 5, revealed that JSP had lower mean response time than PHP for all the predefined situations. This difference was proved to be significant by a t-test.

5.1.4. MySQL 3.25

- *Under Windows 2000 server O.S.:* The recorded results, as abstracted by fig. 6, revealed almost equal mean response time in ASP.NET, JSP, and PHP with mean difference less than or equal to 1 ms. ASP had a higher mean response time.

A one-way ANOVA proved significant difference of means between the four scripting technologies. Post-hoc comparisons between each pair of means based on the LSD proved insignificant mean difference between each pair of ASP.NET, JSP, and PHP. The mean difference between ASP and other scripts was significant in all situations.

- *Under Red Hat Linux 7.2 O.S.:* The recorder results, as abstracted by fig. 7, revealed that PHP had lower mean response time than JSP for all the predefined situations. This difference is proved to be significant by a t-test.

5.2. Effect of the DBMS on the mean response time for each of the web server scripting technologies

5.2.1. Active server pages

The recorded results, given by table 3 and illustrated by fig. 8, revealed that ORACLE 9i had the lowest mean response time followed by MS SQL Server 2000 in all the predefined situations except situation A in which MS SQL Server 2000 had a slightly lower mean response time.. MS Access 2000 had the highest mean response time.

A one-way ANOVA proved significant difference of means between the four DBMS-connected Web pages. Post-hoc comparison between each pair of means based on the LSD proved insignificant difference between means of SQL Server-connected Web pages and MySQL-connected Web pages in situations A, E, F and H. It also proved insignificant difference between means of SQL Server-connected Web pages and ORACLE-connected Web pages in situations A and B, and insignificant difference between means of ORACLE-connected Web pages and MySQL-connected Web pages in situations A and C. Significant difference between means of the other individual pairs for all the predefined situations was proved.

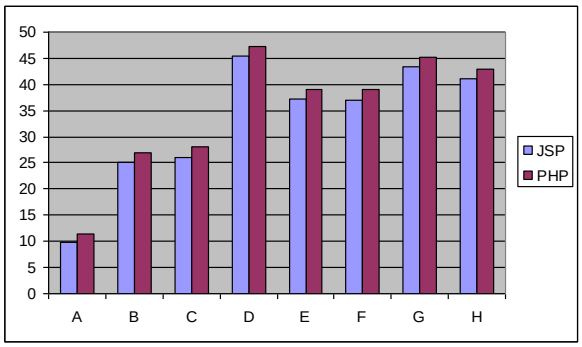


Fig. 5. The mean response time for ORACLE 9i under Read hat Linux 7.2.

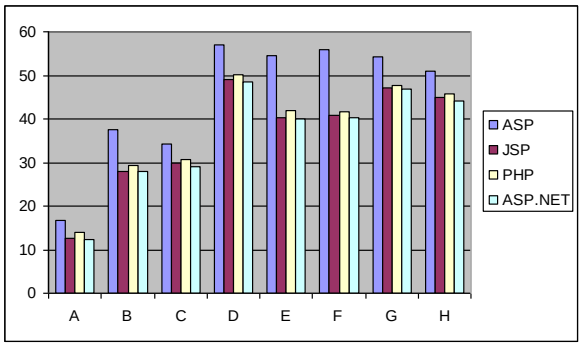


Fig. 6. The mean response time for Mysql 3.25 under Windows 2000

Table 3

Statistical results for asp web pages

Situation	N	Mean for Access, SQL Server, ORACLE, MySQL	Standard deviation	95% Confidence int. (lower bound)	95% Confidence int. (upper bound)
A	40	50.93, 16.10, 16.30, 16.70	1.56, 3.33, 2.46, 3.57	50.43, 15.03, 15.51, 15.56	51.42, 17.17, 17.09, 17.84
B	40	55.25, 32,28, 31, 85, 37, 63	2.44, 2.60, 3.03, 3.39	54.47, 31.44, 30.88, 36.54	56.03, 33.11, 32.82, 38.71
C	40	69.88, 36.78, 32.98, 34.15	1.18, 3.69, 3.12, 3.15	69.50, 35.59, 31.98, 33.14	70.25, 37.96, 33.97, 35.16
D	40	88.15, 54.95, 52.15, 56.85	4.35, 3.94, 2.86, 4.14	86.76, 53.69, 51.24, 55.53	89.54, 56.21, 53.06, 58.17
E	40	84.68, 53.35, 44.10, 54.50	5.62, 2.75, 3.36, 2.75	82.88, 52.47, 43.03, 53.62	86.47, 54.23, 45.17, 55.38
F	40	99.23, 57.15, 44.00, 55.83	2.63, 6.29, 3.39, 2.96	98.39, 55.14, 42.92, 54.88	100.06, 59.16, 45.08, 56.77
G	40	88.90, 52.30, 50.80, 54.33	3.03, 1.95, 1.32, 2.31	87.93, 51.68, 50.38, 53.59	89.87, 52.92, 51.22, 55.06
H	40	83.40, 50.10, 48.15, 50.95	3.18, 2,84, 2.95, 2.97	82.38, 49.19, 47.21, 50.00	84.42, 51.01, 49.09, 51.90

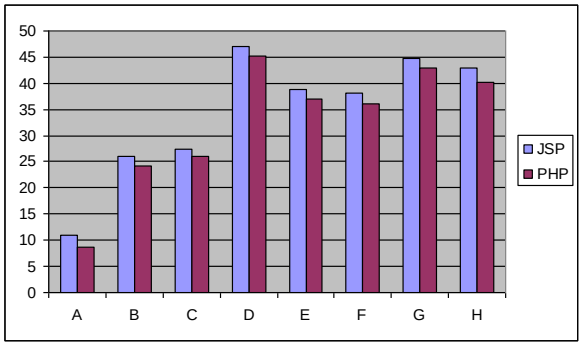


Fig. 7. The mean response time for mysql 3.25 under red hat linux 7.2.

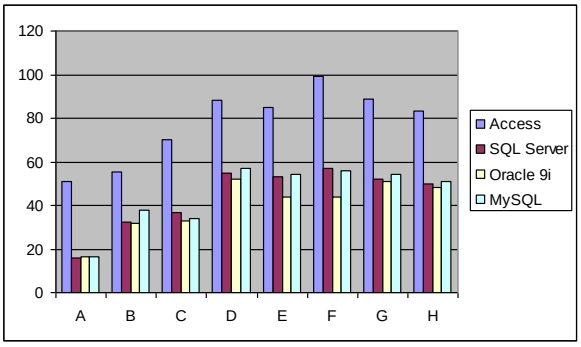


Fig. 8. The mean response time for asp Web pages.

5.2.2. Java server pages

• *Under Windows 2000 server O.S.:* The recorded results, as abstracted by fig. 9, revealed that ORACLE 9i had the lowest mean response time in all the predefined situations. This was followed by MySQL 3.25 then MS SQL Server 2000. MS Access 2000 had the highest mean response time.

A one-way ANOVA proved significant difference of means between the four DBMS-connected Web pages. Post-hoc comparison between each pair of means based on the LSD proved insignificant difference between means of SQL Server-connected Web pages and MySQL-connected Web pages in the all predefined situations except situations D, E, and F. It also proved significant difference between means of the other individual pairs for all the predefined situations.

• *Under Red Hat Linux 7.2 O.S.:* The recorded results, as abstracted by fig. 10, revealed that ORACLE - connected Web pages had lower

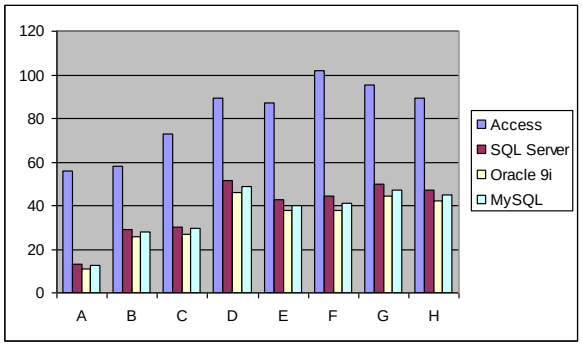


Fig. 9. The mean response time for Jsp under Windows 2000.

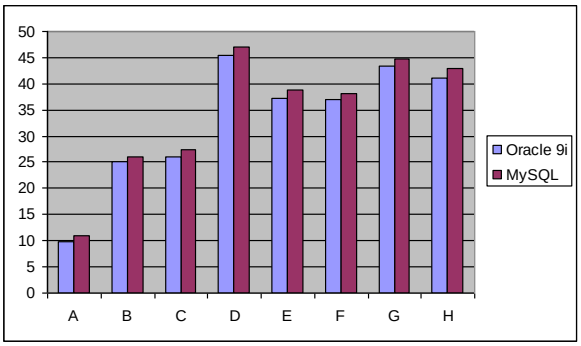


Fig. 10. The mean response time for Jsp under Red Hat Linux 7.2.

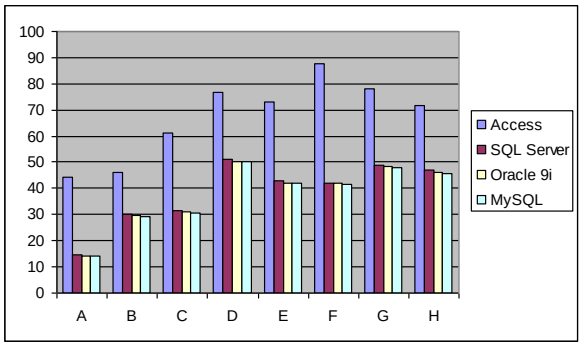


Fig. 11. The mean response time for PHP under Windows 2000.

mean response time than MySQL-connected Web pages for all the predefined situations. This difference is proved to be significant by a t-test in all predefined situations except situations B, C and F.

5.2.3. PHP: hypertext processor

Under Windows 2000 Server O.S.: The recorded results, as abstracted by fig. 11, revealed that MySQL 3.25 had the lowest mean response time in all the predefined situations. This was followed by ORACLE 9i then MS SQL Server 2000 with a slight increase in mean response time. MS Access 2000 had the highest mean response time.

A one-way ANOVA proved significant difference of means between the four DBMS-connected Web pages. Post-hoc comparison between each pair of means based on the least LSD proved insignificant difference between means of individual pairs of MySQL-connected, ORACLE-connected and SQL Server-connected Web pages. On the other hand, it proved significant difference between individual pairs of means of the Access-connected Web pages and other DBMS-connected Web pages all the predefined situations.

- *Under Red Hat Linux 7.2 O.S.:* The recorded results, as abstracted by fig. 12, revealed that MySQL-connected Web pages had lower mean response time than ORACLE-connected Web pages for all the predefined situations. This difference is proved to be significant by a t-test in all predefined situations except situations B, C and F.

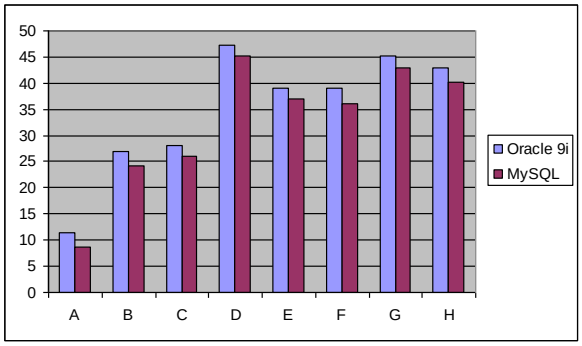


Fig. 12. The mean response time for PHP under Red Hat Linux 7.2.

5.2.4. ASP.NET

The recorded results, as abstracted by fig. 13, revealed that MS SQL Server 2000 had the lowest mean response time in all the predefined situations. This was followed by ORACLE 9i then MySQL 3.25. MS Access 2000 had the highest mean response time.

A one-way (ANOVA) proved significant difference of means between the four DBMS-connected Web pages. Post-hoc comparison between each pair of means based on the LSD proved insignificant difference between means of ORACLE-connected Web pages and MySQL-connected Web pages in the all predefined situations. It also proved significant difference between means of the other individual pairs for all the predefined situations.

6. Discussion

The tests record the response time at the Web server as a sign of performance; lower

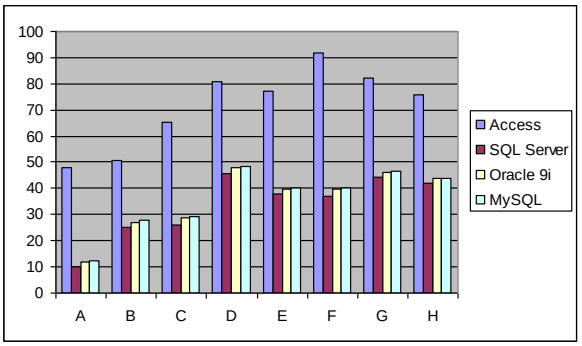


Fig. 13. The mean response time for ASP.NET.

response time indicates better performance. The tests were conducted at the Web server machine not at the client to avoid the effect of traffic time delay and to eliminate time consumed by the Web browser to parse the HTML code. For the same reason, the DBMS was installed on the same server machine to avoid effect of traffic time delay between the Web server and the database server.

The results of the first run of each test were not recorded for two reasons: first, to stabilize the system, and second, to give a chance for JSP and ASP.NET pages to be compiled so the results would not to be affected by the compilation time.

6.1. Effect of the web server script on the mean response time for each database management system

For Microsoft Access 2000, PHP had a significant lower mean response time; i.e. a better performance than the other scripts, while JSP had a significant higher mean response time, i.e. less performance, than the other scripting technologies.

For Microsoft SQL Server 2000, ASP.NET had a significant lower mean response time; i.e. a better performance than the other scripts. JSP and PHP pages followed ASP.NET with almost equal performance. ASP had a significant higher mean response time; i.e. less performance, than other scripting technologies.

For ORACLE 9i, under Windows 2000 Server O.S., JSP and ASP.NET had similar mean response time i.e. similar performance in most situations. This was better than that of PHP and ASP. The latter had significantly higher mean response time; i.e. less performance than the other scripting technologies. The scenario was the same under Red Hat Linux 7.25. As only JSP and PHP were tested, JSP had significant better performance than PHP.

For MySQL 3.25, under Windows 2000 Server O.S., PHP, JSP and ASP.NET had similar mean response time i.e. similar performance in most situations. ASP had significantly higher mean response time; i.e. less performance than the other scripting technologies. The scenario under Red Hat Linux 7.25 was different. PHP had a significant better performance than JSP.

6.2. Effect of the database management system on the mean response time for each web server script

Among the four tested database management systems, Microsoft Access 2000 had the highest mean response time; i.e. the worst performance when used with the four tested Web server scripts.

ASP pages had a lower mean response time i.e. better performance, when connected to ORACLE 9i than when connected to other DBMS. This was followed by MySQL 3.25 and SQL Server 2000 although the difference in mean response time is minimal for the situations in which one single row was retrieved from the database.

JSP pages also performed better when connected to ORACLE 9i, under Windows OS, than SQL Server 2000 and MySQL. The performance of JSP pages when connected to any of the latter two databases is almost the same. The scenario was the same under Linux as JSP had significant better performance when connected to ORACLE 9i than when connected to MySQL 3.25.

In PHP, no significant difference in mean response time were detected among ORACLE 9i, MySQL 3.25 and SQL Server 2000 DBMS under Windows. The performance of pages connected to any of these three DBMS was almost the same. On the other hand, MySQL 3.25 gave lower mean response time; i.e. better performance than ORACLE 9i under Linux.

ASP.NET performed better with SQL Server 2000 than other DBMS. This was followed by ORACLE 9i and MySQL 3.25 which had almost the same performance.

6.3. Explanation of some of the results

JSP had an unexpected bad performance with MS Access due to the long connection time needed to establish connection through the ODBC-JDBC bridge and the ODBC driver [11-13]. This was not the case with other DBMS because JSP connects to them through JDBC type II drivers which have much less overhead [11-13].

ASP.NET performed better with MS SQL Server than other scripts did mainly due to the native class library that Microsoft provided specially for connection with SQL Server [14].

The similarity of performance between ASP.NET and JSP with ORACLE 9i is most probably due to the similar concepts they rely on: pre-compilation then interpretation at run time. They also use similar database client technologies to connect to ORACLE 9i, ODBC and JDBC type II respectively [11-14].

A study, done by Timothy Dyck [6] and another study done by Markus Wirrer [15] tested the performance of different Web server scripts that were not connected to databases. Both studies stated that PHP was faster than ASP. This is in agreement with our results. On the other hand, these studies stated that JSP is much slower than ASP and PHP (about 25% of the speed of PHP). Our results do not agree with that. This big difference may be due to the time needed by JSP to be compiled for the first time. In our study, the first reading of each test was ignored but this was not the case in Wirrer's and Dyck's.

Another study done by Orion Inc. [8] measured the response time of ASP and JSP pages under different stress levels as a sign of performance. JSP gave better performance than ASP in that study which is in agreement with our results.

6.4. Restrictions

1. The study tested only the performance of data querying but not data manipulation. The results of this test are applied only to systems that perform a lot of data retrieval with minimum data manipulation.
2. The study tested only the connection, storage and retrieval functions of the scripts. Other complex functions are not measured.
3. As there is no Web server that can run all the tested scripts efficiently, two different Web servers were used in the study: Internet Information Server and Apache Web server. This tends to make the results affected by the Web server performance not only the script.

7. Conclusions

The performance study, based on different situations derived from this system, revealed that PHP was the best performing script with Microsoft Access and ASP.NET was the best with SQL Server. ASP.NET was as good as JSP with ORACLE, better than other scripts. ASP.NET, JSP and PHP were equally good with MySQL under Windows while PHP was better than JSP under Linux.

References

[1] J. Edwards, 3-Tier Client/Server at Work, John Wiley and Sons, Inc., New York, (1999).

[2] R. Peacock, "Enabling Technologies for E-Business," IT Professional, Vol. 2 (4), (2000).

[3] C. Roe and S. Gonik, "Server-Side Design Principles for Scalable Internet Systems," IEEE Software, Vol. 19 (2), (2002).

[4] E. Altendrof, M. Hohman, and R. Zabicki, "Using J2EE on a Large, Web-Based Project," IEEE Software, Vol. 19 (2), (2002).

[5] R. Peacock, "Distributed Architecture Technologies," IT Professional, Vol. 2 (3), (2000).

[6] T. Dyck, "Four Scripting Languages Speed Development," EWeek, Mar (2000).

[7] D. Orzech, "Advantages of PHP over Java," Zend Technologies, www.zend.com.

[8] Anon, "Benchmarks JSP vs ASP, Orion," www. orionserver. com/ benchmarks/benchmark.htm.

[9] M. Ahmed, C. Garrett, J. Faircloth, and C. Payne, "ASP.NET Web Developer's Guide," Rockland, MA. Syngress (2002).
[10] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurements, Simulation, and Modeling, Wiley Interscience, New York, (1991).
[11] A. Abualsamid, Java Connectivity with JDBC, Network Computing, May (1999).
[12] K. Idehen, Java Database Connectivity without Compromise: a Strategic Technology, White Paper, OpenLink, www.openlinksw.com.
[13] G. Hamilton, R. Cattell, and M. Fisher, JDBC Database Access with Java: a Tutorial and Annotated Reference, Sun Microsystems, java.sun.com.
[14] E. Petroutsos and A. Bilgin, Visual Basic.NET Database Programming, London, San Fransisco: SYBEX. p 227-240 (2002).
[15] M. Wirrer, ASP, JSP, PHP, ColdFusion Interactive Web Systems, in IEX. 2002. Zurich: Namics (2002).

Received May 7, 2003
Accepted September 14, 2003