Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 361–370.

Comparison of the Efficiency of Combination of Database Servers, Application Servers and Operating Systems with the TPC-W Benchmark

Nóra Sterbinszky, Gábor Fazekas

University of Debrecen, Faculty of Informatics, Department of Information Technology
e-mail: snorav3@gmail.com, fazekas.gabor@inf.unideb.hu

Abstract

E-commerce websites are only prosperous if they are well designed, implemented, and adequate hardware and software configurations are applied. Such a software configuration includes selecting the appropriate combination of database management system(s), application server(s) and operating system(s).

Five, nowadays frequently used database management systems (Apache Derby, Sun MySQL, Oracle Database, PostgreSQL, IBM DB2), four application servers (Apache Tomcat, Oracle WebLogic, JBoss, Sun GlassFish) and two operating systems (Microsoft Windows and a Linux distribution) were tested with TPC-W (Transaction Processing Performance Council Web) benchmark that's performance metric is the number of web interactions processed per second (WIPS).

*Keywords:*relational database management system, application server, benchmarking, TPC-W, efficiency

MSC: H.3.4

1. Introduction

To choose the appropriate combination of database management system, application server and operating system for an e-commerce web application, we wanted to make a comparison with the help of the TPC-W benchmark.

First, we would like to show up the testing environment – the examined database management systems (DBMSs) and application servers, so the TPC's web benchmark.

Then, with evaluation of the results, we hope to find an answer to the problem above.

2. The Testing Environment

The hardware configuration of the testing environment is a desktop computer that has an Intel Core 2 Duo CPU E8400 3.0GHz processor and 4 GB RAM.

Two operating systems are installed on this computer: Windows 7 Professional and Ubuntu Linux 9.10. Both operating systems run five database servers (Apache Derby 10.5.3.0, Sun MySQL 5.5.1, Oracle Database 11.1.0.6.0, PostgreSQL 8.4.2-1 and IBM DB2 9.7) and four application servers (Apache Tomcat 6.0.20, Oracle WebLogic Server R1 10.3.2, JBoss Application Server 5.1.0 and Sun GlassFish v3 Server).

That means $2 \times 5 \times 4 = 40$ different configurations.

2.1. The Examined Database Management Systems

Derby (Apache) is a relational database management system. It has an Embedded Database Engine and a Network Server for online transaction processing and is small and easy to install. The embedded database engine has IBM DB2 SQL syntax. Derby has only a few tools (ij that allows SQL scripts to be executed, dblook that is a Schema extraction tool, and sysinfo as a Utility to display version numbers and class path).

It is developed in Java, as an open source project under the Apache 2.0 licence, for cross-platform. [1, 11]

MySQL (Sun/Oracle) is a relational database management system that runs as a server providing multi-user access. It has many graphical administrative and development tools (for editing SQL, modeling data and handling database administration).

MySQL is written in C/C++ and developed as an open source project under the terms of the GNU General Public License (or Proprietary EULA), for multiplatform.

It is used in many large-scale WWW products including Wikipedia, Google (though not for searches) Facebook, Nokia, YouTube. [2, 11]

PostgreSQL (or simply Postgres) is an object-relational database management system. It has a global community of developers and companies for developing. Among many tools and features we can find a built-in language (PL/pgSQL), support for the statistical language R, binary and textual large-object storage, online backup, regular expressions, many add-ons and native programming interfaces for C/C++, Java, .NET, Perl, Python, Ruby.

It is written in C, released under PostgreSQL licence (free and open source) and developed for cross-platform. [3, 11]

The Oracle Database (Oracle Corporation) is an object-relational database management system. As Database Options, Oracle DB has many extensions to

the core functionality (Data Recovery, Data Replication, RAC - Real Application Clusters, PL/SQL, built-in Java, Database Vault, Data Mining, Oracle OLAP, Oracle Programmer, Partitioning, Oracle Spatial, Total Recall, Oracle Warehouse Builder, Oracle Enterprise Manager).

It is written in C/C++ and developed under Proprietary License, for multi-platform.

Among partners of Oracle DB we can find ABN AMRO, AEGON, British Airways plc, Dell, E.ON UK, Hungarian Trade Bank (MKB), Hyundai Securities, Japan Airlines, K&H Bank, LG Electronics, Motorola, Nokia Corp., Toshiba TEC, Yahoo. [4, 5, 11]

The IBM DB2 is a relational model database server. Its relevant features are supporting additional data types and concurrency models, OLTP, OLAP, Xquery, XML data storage, APIs for many programming languages such as COBOL, C++, C, Delphi, .NET CLI, Java, Python, Perl, PHP, Ruby, supporting integration into the Eclipse and Visual Studio .NET IDEs.

It is developed in C/C++, under Proprietary EULA License, for cross-platform.

DB2 is used by AMD, Intel, Red Hat, Mandriva, Novell, Symantec. [6, 7, 11]

2.2. The Examined Application Servers

Apache Tomcat (or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation. It implements the Java Servlet and the JavaServer Pages (JSP) specifications. Catalina is Tomcat's servlet container, Coyote is the HTTP Connector component. Jasper is Tomcat's JSP Engine.

Among its features, there are JSP Tag library pooling, background JSP compilation, recompile JSP when included page changes, reduced garbage collection, improved performance and scalability, faster JSP parsing. Depending on the usage purpose, Tomcat may either be deployed as a standalone server or as a component in a complex configuration.

It is written in Java, developed under Apache License 2.0 and for cross-platform. [11]

Oracle WebLogic, owned by Oracle Corp., includes a Java EE application server (WebLogic Application Server), an enterprise portal (WebLogic Portal), an Enterprise Application Integration platform, an HTTP web server and support for the Oracle Fusion Middleware portfolio with Oracle, DB2, Microsoft SQL Server, MySQL Enterprise and other JDBC-compliant databases connectivity. It has a comprehensive feature-palette: WebLogic Workshop (an Eclipse IDE for Java, SOA and Rich Internet applications), JRockit (a custom JVM), .NET interoperability, JMS messaging, Java EE Connector Architecture, CORBA connectivity, IBM WebSphere MQ connectivity, Business Process Management, Data Mapping functionality, JPA, JAAS, XSLT and XQuery, BPEL, JMX, SOAP, WSDL, UDDI, WS-Security, WSRP. [11]

Oracle WebLogic is developed under a Proprietary Software License, for multi-platform.

JBoss Application Server (or JBoss) is a Java EE-based application server. JBoss is now being developed by Red Hat. Its product features are clustering, failover, load balancing, distributed deployment (farming), Aspect-Oriented Programming support, JSP/Servlet (Tomcat), JSF (Mojarra), Enterprise Java Beans, JNDI, Hibernate-integration, JPA, JAX-WS, JMS integration, JAAS.

JBoss Application Server is developed in Java, under a GNU Lesser General Public License. It operates on cross-platform. [11]

GlassFish is an open source application server project developed by Sun Microsystems for the Java EE platform. Some of its features are support for JPA, JAXB, JAX-WS, Enterprise Java Beans, Java NIO for scalability and speed (Grizzly).

It is written Java, and dual-licensed under two free software licences: the Common Development and Distribution License (CDDL) and the GNU General Public License (GPL) with the classpath exception. The proprietary version is called Sun GlassFish Enterprise Server. It can be used on multi-platform. [8, 11]

2.3. About the Benchmark

TPC-W is a transactional web benchmark. The workload is performed in a controlled internet commerce environment that simulates the activities of a business-oriented transactional web server. Among numerous TPC-benchmarks, TPC-W is appropriate for benchmarking the co-operation of database and application servers. Its performance metric is the number of web interactions processed per second (WIPS). It offers three different profiles, those cover the behaviors of users, by varying the ratio of browsing (reading processes) to buying (writing processes): browsing (WIPSb, means 95% reading - 5% writing), shopping (WIPS, stands for 80% reading - 20% writing) and ordering (WIPSo, contains 50% reading - 50% writing).

The TPC-W standard specifies an implementation of an online book store of fourteen web pages. The users' activities are realized by Remote Browsing Emulators (RBEs). Their behavior matches the load of the real network traffic. The size of the database can be set between 1000 and 10000000 items. [9]

Three database sizes and four RBE numbers cover the different workloads. Small database can be characterized by 1000 items and 30 EBs, medium-sized one by 10000 items and 150 EBs, large one by 10000 items and 300 EBs. At testing, varying the number of the RBEs realizes 30, 150, 300 and 600 ones. [10]

That means $3 \times 4 \times 3 = 36$ runs on each configuration.

3. Evaluation of the Results

The combination of five database management systems (DBMSs), four application servers and two operating systems supplied the testing environment. In addition, these softwares mean the three dimensions, according to which the results can be

analysed. The primary aspect is the examination of the efficiency of the database management systems.

First, we group the presentation of the results according to operating systems, including the performance of various DBMSs by application servers. Later, we sum up our experiences based on these three different dimensions.

Each configuration was tested with different mixes containing reading and writing operations, and with different browser numbers (30, 150, 300, 600) respectively. The running of each test with given settings was repeated three times. The following figures show the average of the three mixes.

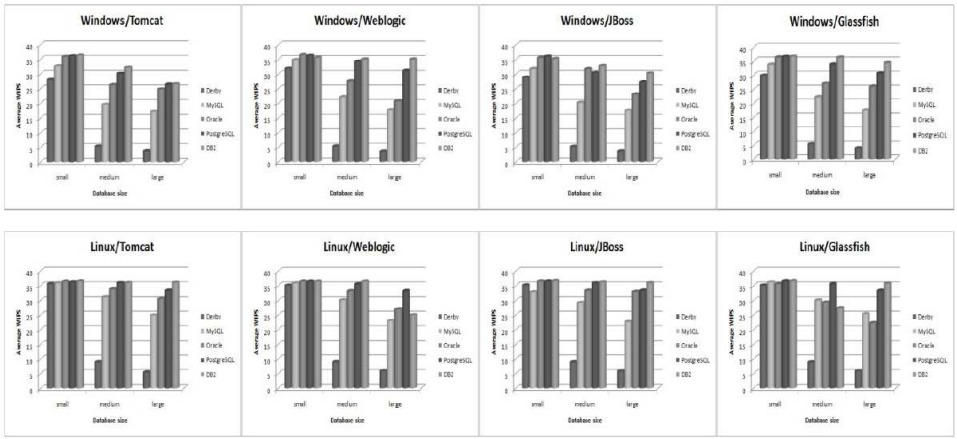

Figure 1: Average WIPS rate of the DBMSs by database size and platform

The different DBMSs testing with web server Tomcat running on Microsoft's operating system don't show significant difference under small database size. Their performance is increasing in proportion by the increase of the number of emulated browsers (the average WIPS number is increasing). Some differences are already noticeable in case of a medium database size. Here, the values are increasing proportionally also with lifting the number of the browsers, but this tendency is smaller than in case of a little database size. The performance doesn't double up entirely with doubling the number of the browsers. The performance of Derby stands on a very low level with each number of browsers. The indices of PostgreSQL and of DB2 stay beneath the values measured with smaller database size. The average WIPS with MySQL and Oracle shows minimal rise despite the doubling of the number of the browsers. The tendency is very similar in large databases. We may take an exciting observation considering the distribution of reading and writing operations: PostgreSQL behaves differently than the other DBMSs if the number of the browsers is lifted; it performs better beside a bigger proportion of reading operations than the others. By their default settings the database management systems are more efficient in transactional operations than in reading (OLTP databases). This difference is especially conspicuous in case of a large database where the rate

of the proportion is decreasing even more.

If the application server is changed from Tomcat to WebLogic, we can receive similar results. However, there are some differences: by the increase of the size of the database, the performance of PostgreSQL and of DB2 does not fall off, and Oracle performs much better in writing operations than combined with Tomcat.

The cooperation of the database and of the application server of the same vendor Oracle doesn't provide the highest performance, contrary to the expectations. This can be brought into contact with some more factors. One of the possible explanations is the big memory claim: the DBMS and the application server from Oracle have fairly large resource requirement separately, which doubles at their cooperation. According to another possible explanation, these systems have not been configured completely optimally to each other yet. Now, the vendor of both products is Oracle, which has already been developing the DBMS for a long time but the WebLogic server has just been bought from BEA System Inc. recently, that's why it is probably that the suitable optimisation for the cooperation has not done yet.

JBoss is the application server that is used for the next testing. Here, the picture is more uniform than previously. Each DBMS responds to rising the size of the database nearly equivalent. Derby, as the only exception, shows low performance, just like earlier.

The results measured with GlassFish application server are near equivalent to the values measured beside the Oracle's product.

Let's change the operating system to complete the same experiments.

The DBMSs combined with Tomcat under Ubuntu Linux show similar rates than those under Windows. Except for Derby, the principal difference disappears here. Increasing the database size takes only an irrelevant effect to the indices. Their degrees are similar with small, medium and also large databases, in the opposite of testing under Windows where the performance is decreasing significantly when rising the database size. We can also see here the high performance of Oracle in writing and of PostgreSQL in reading. IBM's DBMS is the only that hits totally same results beside different database sizes.

The performance of the DBMSs is likewise better in case of testing with WebLogic application server under Linux operating system than under Windows. It is independent of the database size and on the number of browsers. The only exception is DB2 that doesn't represent a proportionally better performance in case of a big database size and browser number. IBM's DBMS gains balanced results in reading and writing operations in most cases of configurations but here the difference is significant because of the slowness of transactional processes. The indices achieved with Oracle's application server under Linux show higher values than under Windows, though Tomcat running on Ubuntu serves requests more efficiently.

JBoss provides a similar basis than Tomcat. MySQL and Oracle DB look somewhat weaker but the difference is not significant. On the other hand, a considerable performance increasing can be observed in contrast with the results of the same

configuration under Windows.

The cooperation efficiency of GlassFish web server and of MySQL from the same vendor is exemplary but not enough to overtake PostgreSQL or DB2 which has difficulties in case of a medium sized database and of a raised browser number. Oracle's performance is surprisingly low on a large database. MySQL's and Oracle's indices stand contrary than those under Windows where MySQL denotes much lower level than Oracle which looks better mainly in transactional operations.

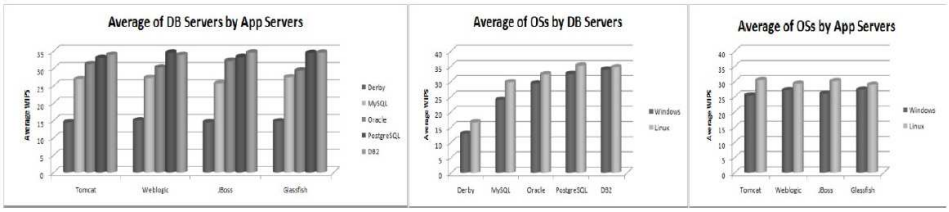

Figure 2: Average results in different aspects

To sum up, it is verifiable that all examined DBMSs achieve better under Linux than under the Microsoft's operating system. The measure of the discrepancy between reading and writing is smaller, so the accessing processes signal bigger numbers than under Windows. The difference is especially conspicuous when Tomcat is applied. The only combination is the cooperation of GlassFish and of Oracle Database which is for the benefit of Windows. Now, the Sun got to Oracle as an acquisition, that's why perhaps we don't have to wait long for a suitable, platform independent interoperability of these vendors' software products.

Since each examined database management system has an OLTP character, it is an expectation to execute transactional processes faster than queries. They live up to this expectation with all database sizes and browser numbers. The only exception is PostgreSQL that executes the reading operations more quickly while the writing ones are done much slower with larger input data. After all, this proves to be the most reliable database management system. It performs stable good combined with any examined operating system or application server.

The results justify that none of the combination of DBMSs and of application servers can be chosen that would be much more efficient than the others. We can verify that the performance of the database management systems is quasi independent from the selected application server if the operating system is out of consideration.

The results show which DBMS is the most efficient one besides different database sizes: by small databases, Oracle gets to top. Rising the database size, DB2 conquers increasingly bigger proportion (2/3 of the tests). PostgreSQL has a quota of 20-30% relatively steadily. MySQL is on the top only once or twice while Derby is quasi not at all.

In the aspect of mixes containing the combination of queries and transactions in different rates Oracle's strength become obvious in writing processes: here, the WIPS with ordering mix attains almost tenfold value compared to the browsing

one. The PostgreSQL is rather expeditious in querying while the DB2 is somewhat more in updating where on the other hand, Oracle forces back its participation significantly. MySQL proves to be faster at executing of transactions with each configuration but however, it precedes the other DBMSs only in some cases. Derby shows an interesting result considering the kind of operations: its efficiency double up running with mixes including more transactional processes. Such a great proportion of difference can be observed at no other DBMS.

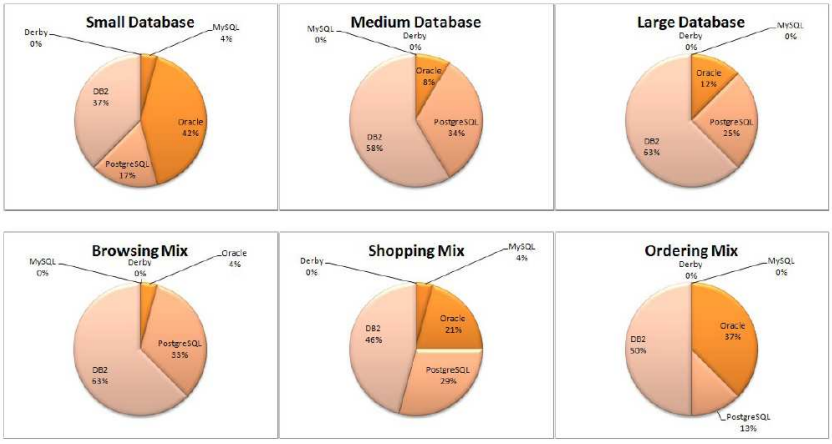

Figure 3: Partition of highest results of the DBMSs by database size and mixes

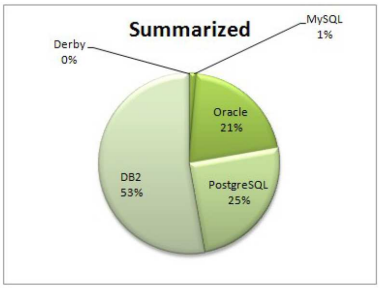

Figure 4: Partition of highest results of the DBMSs by summarized average WIPS

The IBM's database management system is the most efficient one in the half of cases while Oracle and PostgreSQL share the other half in an equal rate approximately as it turned out of the aggregated indices (those of mixes, database sizes, browser numbers, different operating systems and application servers). MySQL and Derby do not have a considerable slice, although there has been realized a very sharp deviance between the two systems. MySQL overtakes the other DBMSs only in a little percentage of the cases though it does not get with a lot behind them.

Average WIPS of MySQL, Oracle, PostgreSQL and DB2 show similar performance compared to one another and it is twice as big as the Derby's index numbers.

Figure 2 shows well that the performance of DBMSs is independent from the chosen application server: the deviance is minimal.

The operating system slightly influences the results: either the database servers or the application servers come uniformly up to a higher level on Linux than on Windows.

4. Summary

According to our aims you can decide which database management system is the most useful. There is a wide offer to choose. If we would like to try our application on relatively few data, Derby is a suitable choice: it is integrated in the NetBeans IDE. While using another Integrated Development Environment (e.g. Eclipse), Derby is the best decision again since after downloading and unpacking, it is ready to use. If we would like to handle a big quantity of data efficiently and/or our application should serve many users, it is expedient to choose among the other four examined DBMSs. MySQL can be favoured because it is free to download and it has an open source. However installing the software with default settings, its indices seems to be somewhat lower than those of the other three DBMSs. IBM's system wins the gold medal among Oracle, PostgreSQL and DB2. On the other hand, it may report a serious disadvantage that it is not freely accessible. The product is valid only 90 days then it has to be reinstalled or bought. PostgreSQL's and the Oracle's DBMS software are both free to download. If we want to obtain additional products from the same vendor, PostgreSQL proves better choice to save costs.

Although, the customers have to pay for most of the Oracle's products, the firm is in regard to serve more and more comprehensive demand, to support comfort functions and it also possesses an appropriate background to produce the most efficient implementations. Its database management system executes writing operations faster while that of the PostgreSQL is more efficient in case of reading processes. Nowadays, OLAP functions are accessible to nearly every DBMSs that's why we agree with the traditional views, according to which these softwares should work faster in processing transactions rather than in queries.

Generally, a database management system has to gain as high efficiency as possible (executing operations in the shortest time). Consequently, your choice depends on the amount of your investment or on the goal of our application (need for more querying or transactional processes) then it is worthy to choose DB2, PostgreSQL or Oracle if you are planning to install your system promptly without any additional settings.

5. Future work

We would like to extend the testing onto those quite widely-used database management systems (Sybase, Teradata) and application servers (IBM WebSphere) that ones are not examined yet. Furthermore, beside the Java technology, our aim consists to test Microsoft's products (C# and .NET, MS SQL Server, Windows Server) and to examine the configurations including these different technologies. After finishing the extended measurements, we hope to be able to evaluate the results also in the aspect of the cooperation of softwares from the same vendor.

References

[1] Apache Derby, http://db.apache.org/derby/index.html (2010)

[2] MySQL :: The world's most popular open source database, http://www.mysql.com/ (2010)

[3] PostgreSQL: The world's most advanced open source database, http://www.postgresql.org/ (2010)

[4] Database 11g | Oracle Database 11g | Oracle, http://www.oracle.com/us/products/database/index.html (2010)

[5] Oracle Database Customers | Oracle Customer Successes, http://www.oracle.com/customers/products/database.html (2010)

[6] IBM - DB2 - Data server - database software - databse management - open source, http://www-01.ibm.com/software/data/db2/ (2010)

[7] IBM developerWorks: Wikis - Information Management - DB2 Business Partner Directory, http://www.ibm.com/developerworks/wikis/display/im/DB2+Business+Partner+Directory (2010)

[8] Sun SDN Article - The Basics of GlassFish, http://java.sun.com/developer/technicalArticles/glassfish/GFBasics.html (2010)

[9] Transaction Processing Performance Council (TPC) BenchmarkTM W (Web Commerce) Specification Version 2.0r, http://www.tpc.org/tpcw/default.asp (2010)

[10] ANDERSON SUPRIANO, GUSTAVO M. D. VIEIRA, LUIZ E. BUZATO, Evaluation of a read-optimized database for dynamic web applications, http://www.ic.unicamp.br/~gdvieira/publications/supriano08.pdf (2010)

[11] http://en.wikipedia.org/wiki/Apache_Derby|MySQL|MySQL_Workbench#GUI_Tools|PostgreSQL|Oracle_Database|IBM_DB2|Apache_Tomcat|JBoss_application_server|GlassFish|Oracle_WebLogic_Server (2010)