SCIENTIFIC PUBLICATIONS OF THE STATE UNIVERSITY OF NOVI PAZAR
SER. A: APPL. MATH. INFORM. AND MECH. vol. 15, 2, 2023, 121–129

Comparative Analysis of Performance of Open Source Databases in Web Application Development

D. Savić, S. Vučetić, P. Milić

Abstract: One of the dilemmas faced by developers of Web application is whether to use SQL or NoSQL database with the unavoidable fact that application must have a quick response. Often these applications work with large amounts of data. This paper presents a comparative analysis of performance of open source SQL and NoSQL databases used in web application development. We tested behavior of different open source SQL and NoSQL databases by different parameters and give some conclusions about suitability of their application in different cases.

Keywords: database, SQL, NoSQL, performance, web application

1 Introduction

Database systems can be organized on a basis of different database models which logically structures the data that are managed. These models are the first step and the biggest determiner of how a database application will work and handle the information it deals with [1]. One of the most popular database model is Relational Model, which clearly and strictly provides the mean of structuring the data. In Relational Model data-tables, contains the information in a structured way (e.g. a Person's name and address), relating all the input by assigning values to attributes (e.g. a Person's ID number). Popular models of relational databases used in web application development are MySQL and PostgreSQL [2], [3].

Recently, a series of different systems and applications called NoSQL databases started to gain popularity, with their promise of offering some very interesting additional functionalities which are not covered in relational (SQL) databases. By eradicating the strictly structured data keeping style defined within the relational model, these database systems work by offering a much more freely shaped way of working with information providing in that manner flexibility and ease, despite the fact that they come with their own problems, considering the important and indispensable nature of data. NoSQL represents a

Manuscript received September 3, 2023; accepted November 30, 2023.

D. Savić is with the Faculty of Computer Science University Singidunum, Belgrade, Serbia; S. Vučetić and P. Milić are with Department of Computer Science and Informatics, Faculty of Technical Sciences, University of Priština, Priština – Kosovska Mitrovica, Serbia

notion of a whole one class of databases which do not have characteristics of traditional relational databases and that are developed in response to a rising volume of data stored about users, objects and products, triggered by the needs of Web 2.0 companies such as Facebook, Google and Amazon [4]. Examples of popular NoSQL databases are MongoDB and CauchDB [5][6]. It is important to men- tion that NoSQL database models have four types, document based, key-value oriented, column oriented and graph databases. In our experiments we used only document based NoSQL databases such as previously mentioned MongoDB and CouchDB.

This paper analyzes both database models from the aspects of their performance in reading and writing of numerical and textual data and their applicability in web application development. We tested different database models via Web with different sizes of data. Also we analyzed their performance and present the observations that have been made in this experiment in terms of complexity, stability, suitable application areas, efficiency and scalability. On the basis of them are presented the adequate suggestions for web application developers.

2 Aspects of performance

Web applications that use databases have the need for fast access to data stored in them in order for their proper processing, before presenting to the end user. In that way we come to the fact that databases are essential for performance of web applications. Response time of databases is one of the key characteristics when developers choose which database to use, in accordance with purpose and need of application. Several things affect response time of the databases, from which the most important are:

- data type (string, number,) and
- volume of data.

Database response time is growing as grows volume of data requested from her [7]. Also, depending on type of data stored inside, the time of processing will vary. Obviously, this will have an impact on performance of web application.

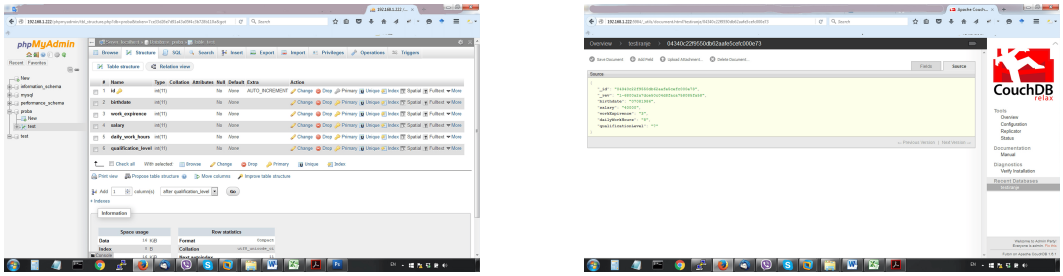
In the following sections we will describe some aspects of performance of databases that are used in development of web applications. We will measure how different database models act during data insertion and read for different volumes of data and different types, especially numbers and strings. On the basis of that, we will compare results for same data types and volumes of data for different database models and give some conclusions about suitability of use of these models in web applications with specific purpose.

3 Defining a framework for testing

For the purposes of the tests, we used server with Intel Core2Duo processor at 2GHz with 2GB of RAM memory where were located both database and web server. Operating system was CentOS6 with Apache web server and application for the measurements

written in PHP. In order to access databases from PHP, it was necessary to install some packages/drivers in CentOS operating system such as php-mysql for MySQL, php-pgsql for PostgreSQL, mongo for MongoDB and PHP-on-Couch library for CouchDB.

In the test, a part of database of employees in company was used. This database contains among other tables, table Persons (Name, Surname, Address, City, Country, Birthdate, Work Expirence, Salary, Daily Work Hours and Qualification Level), The first five attributes (fields) in the table have String data type, while the remain have numeric data type. We have split this table in two parts, and formed two new tables, one with attributes (fields) which all are strings, and another one which all are numeric, to test our work hypothesis. The first table is called Persons (Name, Surname, Address, City, Country) and second one Details (Birthdate, Work expirence, Salary, Daily Work Hours). We will use volume sizes of 100, 1000, 10000, 100000 and 1000000 number of records, for read and write operations. In NoSQL terminology, there is slight difference in naming of components of database structure. For example, SQL notion "table" is equal to NoSQL notion "collection" and SQL notion "row" is equal to NoSQL notion "document". Database structure for both database models was shown on Figure 1 (a) and Figure 1 (b).



a) SQL b) NoSQL

Fig. 1: Database model structure

To make our tests more reliable, for each volume size we conducted a number of measurements, and then we took the average of the total number of measurements as the final result. Determining number of measurements for each volume size we base on statistical approach as a reliable method with some restrictions, such as confidence level and margin of error [8].

$$nm = \frac{Z^2 * p * [1-p]}{C^2} \tag{1}$$

$$nm = \frac{nm}{1+\frac{nm-1}{vol}}, \tag{2}$$

Equation (1) explains the process of calculating the measurement size based on confidence level, margin of error and expected accuracy. The margin of error (also called

confidence interval) is a plus/minus value that indicates the precision of a chosen measurement and allows deviation of the expected results. The margin of error is denoted as c in the equation, and our research it was valued with 10%. Z represents the chosen confidence level, which is expressed as a percentage and represents how often the true percentage of the measurements satisfies the required characteristic and lies within the confidence interval. Usually Z is chosen to be 90% or 95%. We have chosen a 95% confidence for which Z takes on the value 1.65 in the calculation, according to the table of the standard normal curve area values. Finally, p represents accuracy, which is expressed as a percentage of the sampled data that would truly satisfy the required characteristics. Because there is not a trustworthy way for reliable prediction of such percentage, we have used the 50% value. Equation (2) represents a correction of the calculated sample size according to the true volume size of data records (documents) which is denoted as *vol*.

4 Results

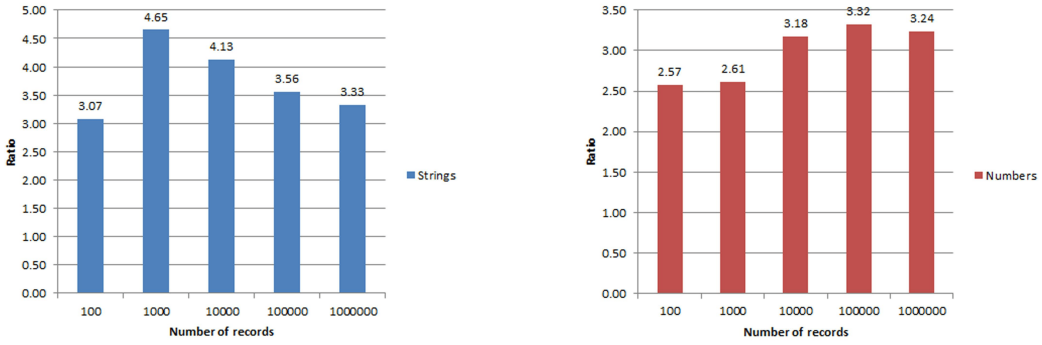
Firstly, we conducted measurements for SQL database models from PHP application, which for operation of insertion of data records in defined table use ***for*** loop. Before beginning the loop we start the measurement of time and after closing the loop we end with measurement of time via ***microtime()*** PHP embedded function.

Table 1: Results of measurement for MySQL and PostgreSQL database model

Number of documents	Strings		Numbers	
	Time in seconds for read operation	Time in seconds for write operation	Time in seconds for read operation	Time in seconds for write operation
MySQL				
100	0.0010671616	0.2517450613	0.0013379025	0.2118950081
1000	0.0101490021	2.6572120190	0.0065619946	1.8477721214
10000	0.2536389828	26.082078933	0.0776991367	20.995476961
100000	3.2280069828	226.39012598	0.7731300926	218.38372707
1000000	8.729529857	2390.6246559	7.982856035	2241.1519470
PostgreSQL				
100	0.0056259632	0.0819129944	0.0025367737	0.0822930241
1000	0.0406560173	0.5710120678	0.0068559647	0.7076749611
10000	0.3232379532	6.3157559586	0.0552437935	6.6095761490
100000	0.4702560902	63.612433195	0.4659690666	65.820328950
1000000	5.028455972	717.56896901	4.727311039	691.97046303

Data from 1 shows that for same number of records there is difference in time necessary for read and write operation for both strings and numbers. Also, both database models needed more time for read/write of strings relative to numbers. If we compare results for this database models, we can notice difference in time required for write operation. For example, for write operation and different volume of records used in measurement, PostgreSQL has better performances than MySQL as shown on Figure 2(a) and 2(b).

In order to better explain Figure 2(a) and 2(b), for example look at columns for 1000 number of records. These columns says that PostgreSQL has 4.65 time better performances in time for write operation for string data type and 2.61 time better performances in time for write operation of numbers. Regarding read operation, MySQL shows better performances for 100, 1000 and 10000 number of records, while for 100000 and 1000000 PostgreSQL is better.



a) SQL b) NoSQL

Fig. 2: Performance ratio between PostgreSQL and MySQL database models

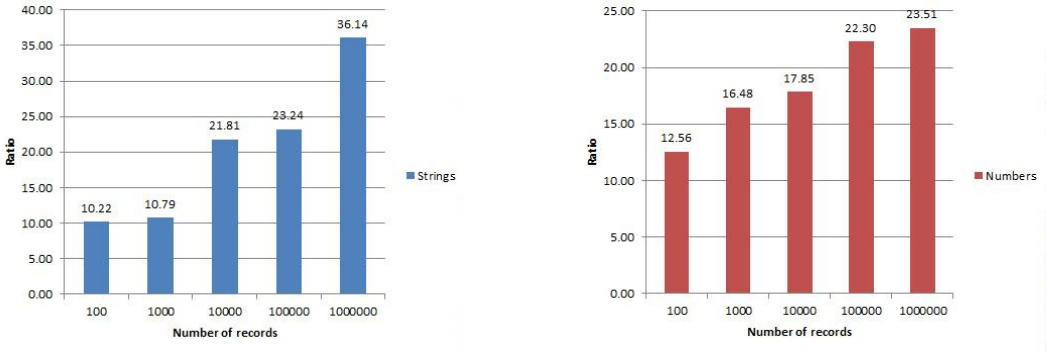
Measurements for NoSQL database models was also done via PHP application using same approach as for SQL database models. Results are shown on Table 2.

Table 2: Results of measurement for MongoDB and CouchDB database model

Number of documents	Strings		Numbers	
	Time in seconds for read operation	Time in seconds for write operation	Time in seconds for read operation	Time in seconds for write operation
MongoDB				
100	0.0013380051	0.0238151550	0.0012719631	0.0200681686
1000	0.0461540222	0.2932071686	0.0066139698	0.1877689362
10000	0.1923961639	1.757361888	0.1501100063	2.0995390415
100000	1.0762059689	19.40317797	0.6817579269	20.553742885
1000000	7.180605173	186.88542294	6.5281951427	211.40931606
CouchDB				
100	0.0149431229	0.2432780266	0.0082700253	0.2520020008
1000	0.0853118896	3.1650080681	0.0899269581	3.0943131447
10000	0.9365057945	38.320665121	0.9261910915	37.470549821
100000	9.7576241493	450.85154700	9.7311799526	458.40157198
1000000	/	6754.6811602	/	4970.5689010

Observing Table 2, we notice similar situation as for SQL database models, i.e. differences in time necessary for read and write operation for both strings and numbers. In this case, for write operation, the difference in time lies between 0 and 36 times for different

number of documents, indicating that MongoDB has dramatically better performances than CouchDB as shown on Figures 3(a) and 3(b). In read operation, MongoDB also shows better performances. During measurements we were not able to test behavior of CouchDB database for read of 1000000 of documents in current setup of hardware environment, and results are not shown on Table 2. This is because of fact that system failed in reading of that number of documents.



a) SQL b) NoSQL

Fig. 3: Performance ratio between MongoDB and CouchDB database models

After we conducted measurements for both database models separately, now we can compare them. Firstly, we give comparison for best rated representatives from both database models for numeric data type and then for text data type. Then we give graphical representation of this case.

As we can see from Table 3, MongoDB (NoSQL) database is quicker in relation to PostgreSQL (SQL) database for both read and write operation. NoSQL simpler denormalized store allows retrieving all information about a specific item in a single request [9]. Theres no need for related JOINs or complex SQL queries, which speeds up execution of queries. But, we must keep in mind that project design and data requirements will have most impact on performance of database model. A well-designed SQL database will almost certainly perform better than a badly designed NoSQL equivalent and vice versa. Due to the fact that NoSQL database models are disk-based and retain in a buffer pool as well as multi-threaded architecture [10] and that NoSQL have sacrified ACID (Accessibility, Consistency, Isolation, Durability) compliance, it becomes clear why NoSQL database models have better performances.

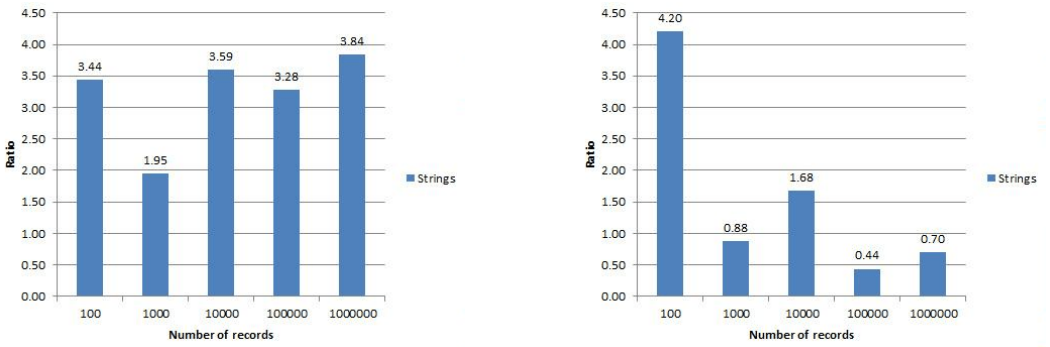
In previous paragraph we have measured the time required for processing of different volumes of data in different open source database models that are used in couple with web applications in order to analyze how application of different database model will influence on performance of web application in a whole and to suggest application of appropriate database model depending on the purpose of web application.

Obtained results tell us that best performance among examined database models give NoSQL. This database models are suitable to use for processing of Big Data and thousands

or millions of users doing updates as well as reads and also for dealing with huge quantities of unstructured data which conventional RDBMS solutions could not cope. NoSQL databases are stable in dealing with exponential growth of the volume of data generated by users, systems and sensors. Also, using of NoSQL databases for the purpose of data analytics is

Table 3: Comparison of results for PostgreSQL and MongoDB

Number of documents	PostgreSQL		MongoDB	
	Time in seconds for read operation	Time in seconds for write operation	Time in seconds for read operation	Time in seconds for write operation
Numeric data type				
100	0.0025367737	0.0822930241	0.0012719631	0.0200681686
1000	0.0068559647	0.7076749611	0.0066139698	0.1877689362
10000	0.0552437935	6.609576149	0.1501100063	2.0995390415
100000	0.4659690666	65.820328950	0.6817579269	20.553742885
1000000	4.727311039	691.97046303	6.5281951427	211.40931606
String data type				
100	0.0056259632	0.0819129944	0.0013380051	0.0238151550
1000	0.0406560173	0.5710120678	0.0461540222	0.2932071686
10000	0.3232379532	6.3157559586	0.1923961639	1.7573618889
100000	0.4702560902	63.612433195	1.0762059689	19.403177976
1000000	5.0284559727	717.56896901	7.1806051731	186.88542294



a) SQL b) NoSQL

Fig. 4: Performance ratio between PostgreSQL and MongoDB database models

recommended because they reduce the web latency that appears while web application processes data. Low time consumption exhibit their ability to store and index arbitrarily big data sets while enabling a large amount of concurrent user requests.

Unlike the NoSQL database models, SQL databases have shown worse per- formance because they follow ACID transactions, have complete predefined schema and must provide clear and strict meaning of structure of the data. Measurements carried out on the SQL

database models indicates that PostgreSQL database shows the best performances and that's the reason why this database is implemented in many web solutions such as CKAN (Comprehensive Knowledge Archive Network), Moodle, Skype and etc. In web applications where is important to maintain relation between data, to execute join queries to obtain data from multiple tables and to handle data redundancy, application of SQL database models is recommended.

5 Conclusion

In this paper we have analyzed performance of open source SQL and NoSQL databases that is using in web application development from the point view of reading and writing numerical and textual data from and to database. On the basis of conducted measurements and aforementioned analysis it can be concluded that NoSQL databases have an advantage over SQL databases in the view of dealing with of large quantities of data that are generated on the Web. Such sources of data are surely social networks, which imposes NoSQL databases as suitable solution. The results we obtained also indicate that NoSQL databases are suitable for data analytics which in case of the web is crucial, bearing in mind that web may have limited bandwidth between client and server. Development of web applications for which is not expected generation of large volumes of data and where is important to have consistency between data suggests SQL databases as excellent solution, and our analysis are going in favor to this claim, indicating that this databases offers acceptable performance. Generally, depending on the need and the purpose for which web application is built, a trade-off between performance and data integrity, consistency and availability must be created. This paper offers a view only from one aspect, but answers to the questions such as what is the level of security of both database models, how they behave in centralized and distributed environment, what about commercial sollutions and many others will give deep insight in this area and extend the choise of web developers.

References

[1] W. KHAN, T. KUMAR, C. ZHANG, K. RAJ, A. M. ROY, B. LUO, *SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review*, Big Data and Cognitive Computing. Vol. 7, 2 (2023), 97–102.

[2] A. D. PRABA, M. SAFITRI, *Studi perbandingan performansi antara mysql dan postgresql*, Jurnal Khatulistiwa Informatika. Vol. 8, 2 (2020), 88–93.

[3] R. POLJAK, P. POŠČIĆ, D. JAKŠIĆ, *Comparative analysis of the selected relational database management systems*, In: Proceedings of a Conference on Information and Communication Technology, Electronics and Microelectronics (P. Biljanovic, eds.), Opatia, Croatia, 2017, pp. 1496–1500.

[4] A. MEIER, M. KAUFMANN, *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*, Springer, New York, 2019.

[5] N. D. BHARDWAJ, *Comparative study of couchdb and mongodb–nosql document oriented databases*, International Journal of Computer Applications. Vol. 136, 3 (2016), 24–26.

[6] R. KUMBHARE, S. NIMBALKAR, S. CHOPADE, V. K. PACHGHARE, *Tamper detection in MongoDB and CouchDB database*, In: Proceedings of a Conference on Computational Science and Applications: ICCSA 2019 (O. Gervasi, eds.), Thuy Loi University, Hanoi, 2019, pp. 109–117.

[7] B. MILOŠEVIĆ, S. JOVKOVIĆ, L. PETKOVIĆ, D. KRSTIĆ, S. OBRADOVIĆ, *Results of comparison between open source and commercial methods of organizing databases*, In: Proceedings of a INFOTEH conference (S. Lubura, eds.), Jahorina, Pale, 2012, pp. 643–648.

[8] N. ALAN HECKERT, J. J. FILLIBEN, C. M. CROARKIN, B. HEMBREE, W. F. GUTHRIE, P. TOBIAS, J. PRINZ, *e-Handbook of statistical methods*, NIST, Washington, 2002.

[9] L. YISH, S. MANOHARAN, *Performance comparison of SQL and NoSQL databases*, In: Proceedings of Pacific Rim Conference on Communications (M. McGuire, eds.), Victoria BC, Canada, 2013, pp. 15–19.

[10] L. BONNET, A. LAURENT, M. SALA, B. LAURENT, N. SICARD, *Reduce, you say: What NoSQL can do for data aggregation and BI in large repositories.*, In: Proceedings on Database and Expert Systems Applications Conference (M. A. Hady, eds.), Toulouse, 2011, pp. 483–488.