

Selection of Appropriate Data Storage for Wavelet Transform Coefficients

I. Žouželková, R. Vala and M. Juřík
Department of Informatics and Artificial Intelligence
Department of Computer and Communication Systems
Faculty of Applied Informatics
Tomas Bata University in Zlín
Nad Stráněmi 4511
760 05 Zlín, Czech Republic
zouzelkova@fai.utb.cz

Abstract- The study of transform coefficients brought a question of a useful tool for signal evaluation and comparison. The application should use a database storage system for fast inserting and selecting data. This paper deals with performance tests of selected database management systems: relational MySQL and object-relational PostgreSQL. The tests were focused on different storage engines and used data types. Besides the relational database models, a multidimensional approach was projected and analyzed.

I. INTRODUCTION

Transform coefficients, the outputs of the discrete wavelet transform, are widely employed in time-series processing and related fields to describe the frequencies contained in a sampled signal. The wavelet transform is further used for signal coding, to represent a discrete signal in a less redundant form, and often as a preconditioning for data compression.

The study of transform coefficient has brought a question of a useful tool for signal evaluation. The application should use a database system for storage of a large volume of real numbers with very short response time. Moreover, each signal could be described by an array of coefficients which should be properly saved and should enable efficient search through signal records. These requirements have led to selection of an appropriate database management system (DBMS).

Increasing demand for data storage has heightened the number of offered DBMS and database servers. Furthermore, different database models such as relational, object-oriented, object-relational and other schemes were projected. Developers have been confused by the wide choice and, consequently, the need for performance tests, which could bring an objective comparison, has become a favorite topic for different analysis [1].

Several researchers have designed time response tests of database systems, such as MySQL, MS SQL, PostgreSQL and other DBMS, in order to investigate their suitability for various applications. The testing methods and conditions differed and, consequently, the comparison of findings was inconclusive.

This study is designed to evaluate performance of the chosen database engines using the same hardware configuration and software tool. Moreover, in order to gain the best solution for array storage, the paper compares the performance of two main approaches: a traditional relational

database (MySQL and PostgreSQL) and a multidimensional model (MS SQL).

II. RELATIONAL DATABASE CONCEPT

A. Data Types and Testing Methods

In order to investigate the suitability of the three storage engines (MySQL-MyISAM and InnoDB, PostgreSQL) a time response test using ADOdb tool was designed. ADOdb is a database abstraction library for PHP which allows developer to write applications in a consistent way and change the database without rewriting every call to it in the application [2]. In other words, the engines were tested by the similar script on the same hardware configuration under the equal conditions.

The structure of the tested databases was designed in the next step. The most attention was paid to the choice of the proper data type for the coefficients records. As previously mentioned, each signal could be described by an array of 64 coefficients. The first concept, saving the coefficients into a data type *array* seemed to be logical. However, the only DBMS offering this solution was PostgreSQL. Inserting the array into 64 created attributes (for test purposes 16 attributes were sufficient) with a numeric data type appeared to be the next possibility. This approach was feasible to work across different platforms. The last option offered by all systems was the usage of an attribute defined as *text*, saving the values into a string. This solution could bring problems with comparing the particular coefficients and searching for data in the defined range of values. Table I shows a brief summary of the tested data types on the mentioned platforms.

According to Table I, three concepts of table structure were created. The first model can be seen in Fig. 1. The database included four tables and the table signal contained 16 attributes for the coefficients. The attributes were replaced by only one column with the *text* data type in the second model and with the *array* in the last model (for PostgreSQL only).

Furthermore, the SQL *INSERT* and *SELECT* statements were chosen for the performance tests. The *INSERT* query was used to enter 100, 1000 and 10,000 records. The *SELECT* statement with 16 restricted coefficients was applied on 100,000 rows. Each test was ten times repeated and the results were averaged out.

TABLE I
TESTED DATA TYPES ON CHOSEN PLATFORMS

Data type	Platform		
	MySQL		PostgreSQL
	MyISAM	InnoDB	
Decimal (16 attributes)	X	X	X
Text (1 attribute)	X	X	-
Array (1 attribute)	-	-	X

B. MySQL Tests

MySQL offers several data storage engines. This work deals with the two of them - MyISAM and InnoDB. Reference [3] mentions the main differences between the engines. MyISAM was developed as a default MySQL storage system and contains many functions programmed during years of usage. The engine is very simple, suitable for many systems and widely known for very short response time. On the other hand, InnoDB was designed for transactions providing a data consistency check, which can be more demanding. Long response times of InnoDB can be caused by clustering primary keys, automatic check summing and other techniques preventing data corruptions [4]. Thanks to its properties, InnoDB is expected to be slower than MyISAM. However, InnoDB supports multi thread processing and can perform better in specific situations.

The test of each engine (MyISAM and InnoDB) was divided into two parts. The first part dealt with the INSERT query and the second part tested the SELECT query. Fig. 2. shows the time response of the INSERT depending on the storage engine. According to this preliminary study, MyISAM tended to be more than three times faster than InnoDB. This expected difference in the engines may be due to the previously mentioned data storage process. The exact time values are not very important, because they depend on the used hardware configuration.

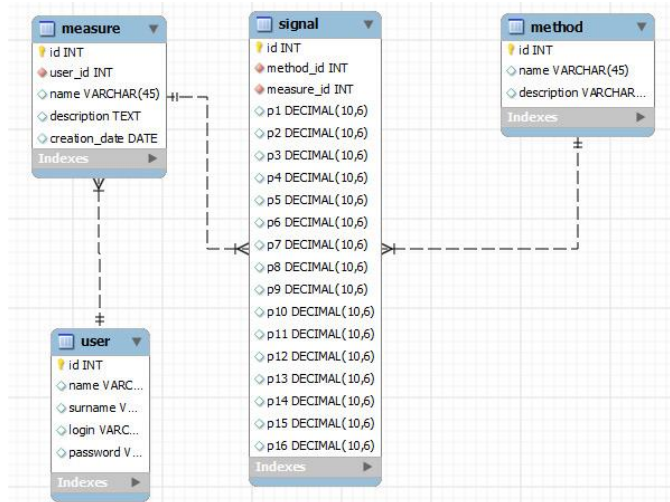


Fig. 1. MySQL and PostgreSQL database structure

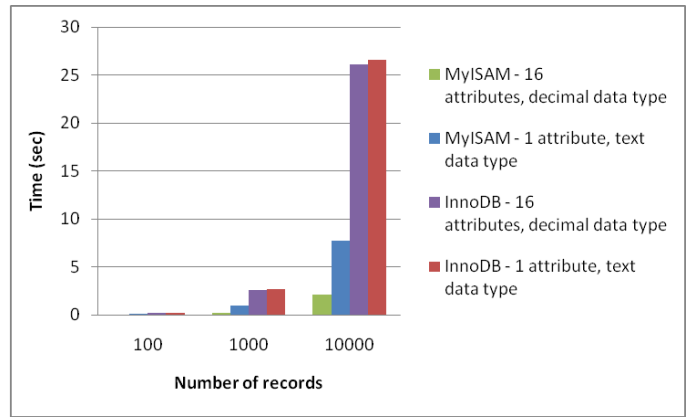


Fig. 2. MyISAM vs InnoDB: Speed of the INSERT query

As can be seen, saving numeral values into the 16 separate attributes was faster than inserting the coefficients as a string into one attribute.

Fig. 3 compares the results of the SELECT query test. Searching the MyISAM decimal attributes seemed to be considerably faster than selecting the InnoDB decimal attributes. Also faster text search is offered by the MyISAM engine. These results could lead to a misinterpreted conclusion that InnoDB responds mostly worse than MyISAM. In real life workload results are likely to be very different, because the time response can be influenced by server stress. Reference [5] focused on InnoDB and MySQL benchmark running the SELECT queries on several co-current threads.

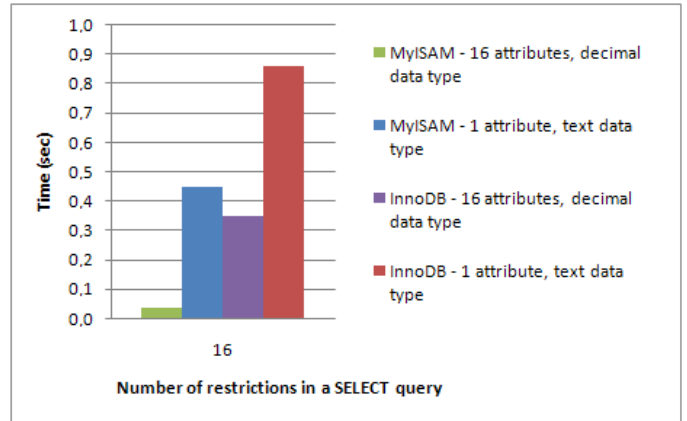


Fig. 3. MyISAM vs InnoDB: Speed of the SELECT query (100,000 rows, 16 restrictions)

C. PostgreSQL Tests

PostgreSQL is an object-relational DBMS allowing columns of a table to be defined as variable-length multidimensional arrays of any built-in or user-defined base type [6]. In this case one array attribute of integer was created for the first part of the test. The second part dealt with 16 numeric attributes as it was done with MySQL DBMS.

As revealed by the Fig. 4, inserting the array type attribute is almost 10% faster than inserting the 16 numeric coefficients. In contrast, selecting from 100,000 of rows is faster in case of the numeric coefficients (Fig. 5). This dissimilarity between

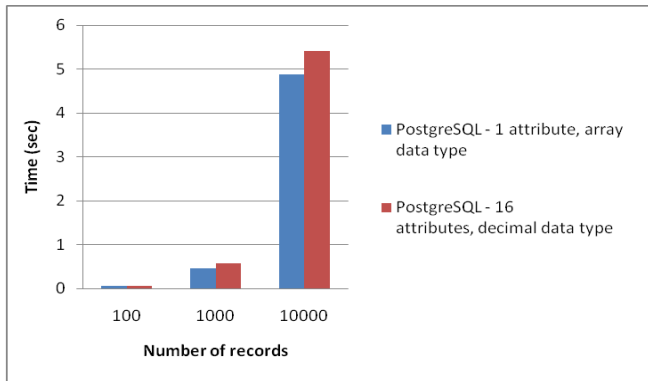


Fig. 4. PostgreSQL: Speed of the *INSERT* query

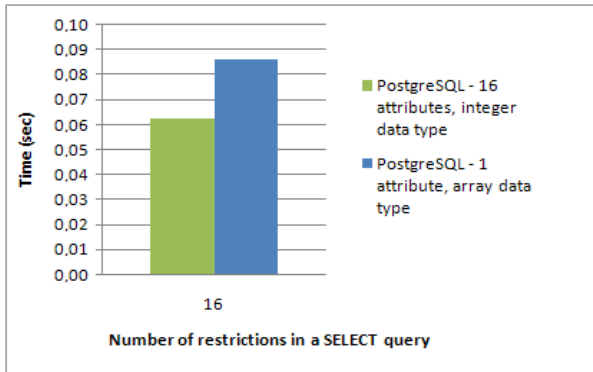


Fig. 5. PostgreSQL: Speed of the *SELECT* query

the queries is logical. Inserting the 16 different attributes is more demanding than saving values into the only one column. On the other hand, searching for appropriate records, selecting according to the 16 restrictions, can be faster with more attributes than searching inside an array.

D. MySQL vs. PostgreSQL – result comparison

The total comparison of the tested DBMS is given in Fig. 6 and Fig. 7. The first graph provides complex results of

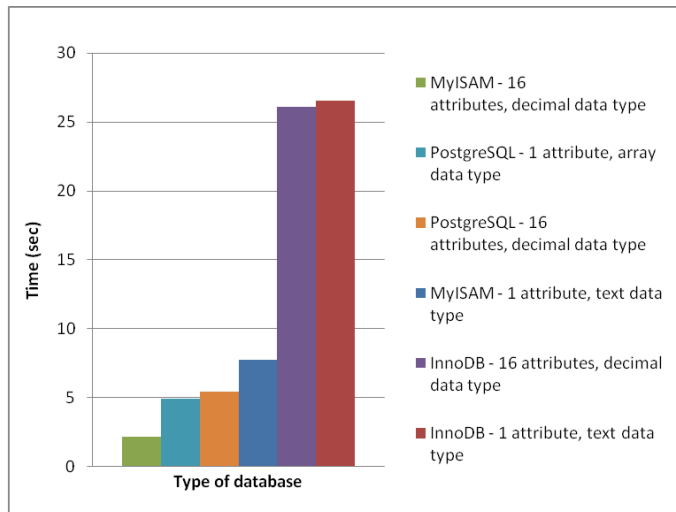


Fig. 6. PostgreSQL vs MySQL: Speed of the *INSERT* query (10,000 rows)

the *INSERT* query. On the limited data available, MyISAM (MySQL engine) tended to perform better than other tested engines. It was almost three times faster in inserting 16 numeral attributes than PostgreSQL and even thirteen times faster than InnoDB. According to this simulation study, inserting the transform coefficients into the 16 fields appears to be the best solution.

As revealed by the second graph, the results of the *SELECT* statement are similar to the previous one. MyISAM reaches the best results again.

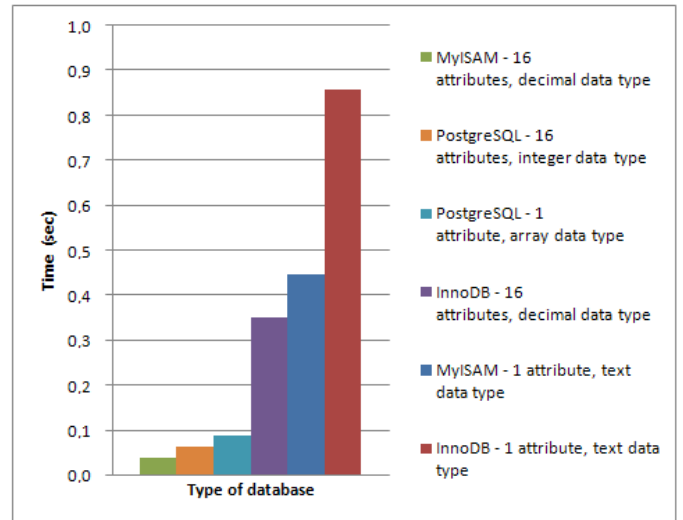


Fig. 7. PostgreSQL vs MyISAM vs InnoDB: Speed of the *SELECT* query (100,000 rows, 16 restrictions)

III. MULTIDIMENSIONAL DATABASE CONCEPT

A multidimensional database concept is closely connected with online analytical processing (OLAP). Traditionally, OLAP is a technique for aggregating data to solve business problems. On the other hand, it can be used in scientific analysis and research as well [7].

The multidimensional approach requires different view of the data. The fundamental parts of any OLAP solution are cubes, dimensions and measures. A cube collects numeric data organized by arrays of discrete identifiers. Essentially, a cube is defined by its measures and dimensions [7]. Dimensions can be interpreted as categories used to analyze the data. Each signal can be defined by three variables: a date, a method of transform and transform coefficients, which form the dimensions of the cube. The measures are the numeric data inside the cube. Details of the cube structure are provided in Fig. 8. Tables *dim_date*, *dim_coeff*, *dim_method* represent the dimensions, while table *fact_values* saves the measures.

The cube was created in MS SQL Server 2008 which offers Analytical Services for OLAP analysis. Because of the different structure, the comparison with standard relational databases would be quite complicated. The first outputs from this system were not satisfactory considering the defined requirements for the application. In fact, OLAP analysis deals with aggregated data and offers summarized facts depending

on the depth of selected dimensions. The application for signal comparison would rather employ short response time of inserting and selecting data. According to these specified requirements, the multidimensional structure tends to be a less convenient method than the classical relational approach.

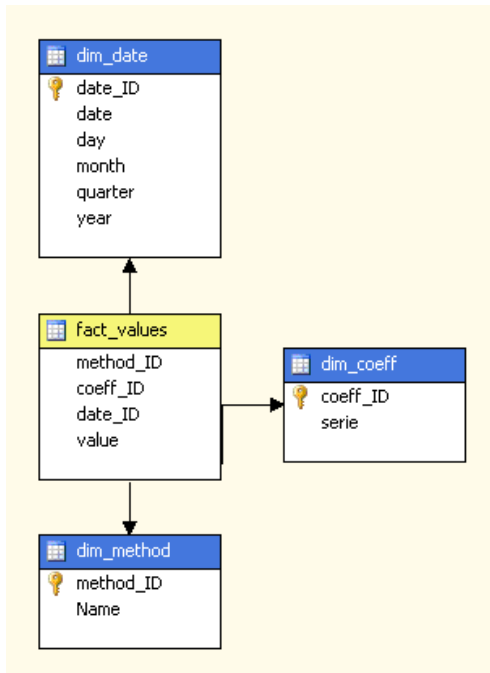


Fig. 8. The multidimensional structure of the cube

IV. CONCLUSION

The main purpose of the survey was to analyse the performances of different database approaches. The multidimensional model appeared to be less acceptable for the real application. On the other hand, relational DBMS seemed to be more suitable. Especially the MySQL database engine - MyISAM acquired the best results in all tests. As can be seen from the data, the testing queries also revealed that implementation of the 16 attributes of a numeral data type would perform better than using arrays or strings.

It should be noted that the findings of this study are restricted to the *INSERT* and *SELECT* query usage and, consequently, embedded procedures can perform differently.

REFERENCES

- [1] VAUGHN, William. Choosing the "Right" DBMS Engine. *Internet.com* [online]. October 22, 2007 <<http://www.developer.com/db/article.php/3706476/Choosing-the-Right-DBMS-Engine.htm>>.
- [2] ADOdb. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 21 March 2005, last modified on 26 August 2010 [cit. 2010-10-20]. <<http://en.wikipedia.org/wiki/ADOdb>>.
- [3] NEWTON, Narayan. Home Tag1 Consulting, Inc. [online]. 2008 MySQL Engines: MyISAM vs. InnoDB. <http://tag1consulting.com/MySQL_Engines_MyISAM_vs_InnoDB>.
- [4] MySQL [online]. 2010 [cit. 2010-10-22]. The InnoDB Storage Engine. <<http://dev.mysql.com/doc/refman/5.5/en/innodb.html>>.
- [5] *MySQL Performance Blog* [online]. 2007 InnoDB vs MyISAM vs Falcon benchmarks. <<http://www.mysqlperformanceblog.com/2007/01/08/innodb-vs-mysam-vs-falcon-benchmarks-part-1/>>.
- [6] PostgreSQL 8.0.26 Documentation [online]. 2005 PostgreSQL. <<http://www.postgresql.org/docs/8.0/interactive/arrays.html>>.
- [7] JANUS, Philo; FOUCHE, Guy. *Pro SQL Server 2008 Analysis Services*. New York, USA : Apress, 2010. 480 s. ISBN 978-1-4302-1995-8.