Possibilities for Using NoSQL Databases in Information Systems in Transport

Andrzej Czerepicki
Warsaw University of Technology, Poland

The article presents directions for development of end-user-oriented information systems in transport. Characteristics of databases belonging to the trend of NoSQL have been enumerated as well as the represented data models. The example of a relational model transformation to a graph form and the concept of application of a graph database for connections search in public transport system have been presented. The general concept of a distributed information system has been presented which uses NoSQL databases in particular modules.

Keywords: databases, transport, information systems.

1. INTRODUCTION

The development of modern information systems in transport takes place in directions which may be characterized as decentralization and globalization. The main aspects can be enumerated as migration of centralized systems in the direction of distributed applications, growing integration of data originating from heterogeneous sources and increasing activity of end users in the process of data exchange with the system. The requirements are respectively growing for information systems within the scope of efficient service of a large number of users, stable rendering of a vast list of services and processing larger and larger sets of data. These requirements translate into databases as a key element of most information systems in transport.

In order to store and process data, information systems in transport currently use relational databases. Based on formal mathematical apparatus and verified by years of intensive use, the relational database model has some constraints connected with practical realization. This refers to scalability of the system, capability of efficient service of a very large number of parallel transactions and possibility of operation on weakly structured data.

NoSQL databases [1] constitute a dynamically developing trend of modern computer technology. Most of them were created within the last few years and since the beginning they have been designed with the consideration of high demands as for the volume of data storage and efficiency of its processing. In view of the abovementioned requirements, application of this type of databases can constitute a prospective development direction for modern information systems in transport.

2. FEATURES OF MODERN INFORMATION SYSTEMS IN TRANSPORT

Centralized information systems used in transport, sooner or later face the challenge of integration. This results from growing expectations of users regarding functionality of such systems and integration processes of transport systems on an international scale. Migration of information systems in the direction of distributed systems results from the necessity of ensuring comprehensive and safe access to data [2]. From technological perspective, it is supported by the development of network services, data processing in a cloud and modern distributed database systems.

Technological development in the range of mobile devices contributed to creation of end user oriented applications, such as embedded navigation systems, public transport connections search

engines, etc. In many information systems, the user starts to play active role not only as a consumer of data, but also as a supplier of feedback. For example, the car navigation application, while calculating optimal travel route, can use information on current traffic situation made available by other users of the system [3]. The bigger the number of users simultaneously using the application, the more reliable data will be gathered by the system. However, at the same time this means a significant increase of demands in relation to the information system, among others, in the range of the number of customers served and the speed of data processing.

Majority of existing information systems in transport process data in different formats. Before they are transferred to the central database, initial processing is required in order to validate and convert data into target format. For this purpose, the software of ETL (*Extract, Transform, Loading)* class is often used, which allows for the read of source data which is saved in various formats, initial filtering, transformation to the unified form and loading to the data warehouse [4]. Inclusion of end users in the process of delivering data, forces initial verification of transferred data and its structuring. The problem arises of effective storage of weakly structured data and its initial processing.

Effective solution of problems connected with intensive development of information systems in transport requires from modern database systems high scalability and efficiency of processing large sets of data. Efficiency of relational databases significantly depends on the structure of stored data and the degree of its connection, as well as the number of simultaneously realized transactions in the system. In situations when the system deals with a large number of weakly connected data stored in the form of various structures without a defined rigid schema, application of a database implementing other data model than relational can be considered.

3. CHARACTERISTICS AND POSSIBILITIES FOR USING NoSQL DATABASES IN TRANSPORT

NoSQL databases constitute a group of modern, dynamically developing systems of data processing. They were created for the purpose of processing large data sets provided, for example, by users of social networks or for storing semantic connections in WWW network. The common feature of databases belonging to the trend of NoSQL is partial resignation from ACID properties (atomicity, consistency, isolation and durability) characteristic for relational databases. In accordance with the CAP theorem [5], any realization of a distributed system is capable of ensuring only two properties out of the three enumerated: *consistency*, *availability*, *partition tolerance*. In such realization, the ability is obtained of storing, effective processing and partition of vary large data sets.

Databases realizing a "key-value" data model can be interpreted as a hash table which stores data of any structure addressed by means of a key [6]. The main advantage of the model is the speed of data recording and reading, as well as lack of impositions on the structure of data storage. This allows for building flexible applications, particularly in the field of weakly associated data processing. Among NoSQL databases, "key-value" databases are also characterized by capability of serving the largest number of users. The disadvantage of the model is high additional workload in the case of realization of systems of strong data connections, which restricts a potential range of their use.

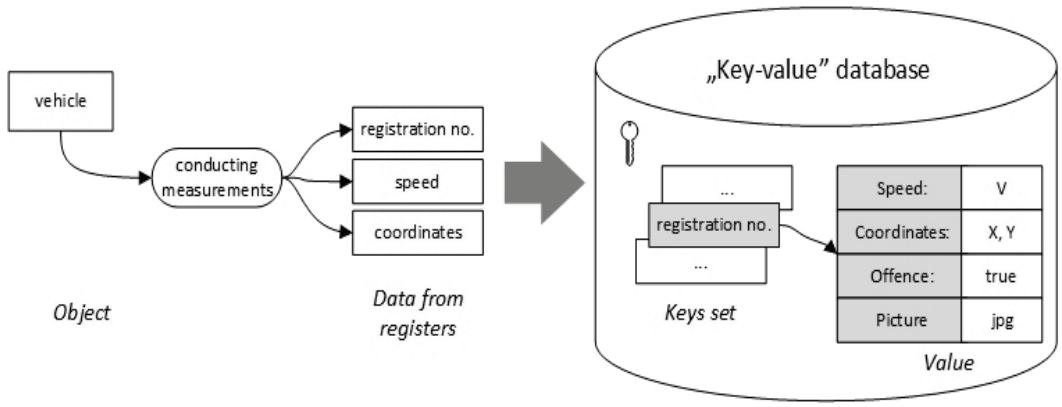

Fig. 1. Example of use of "key-value" database.

The proposed application of "key-value" database at the entry of information system will allow for initial filtering and aggregation of data obtained from users or automated measuring systems. In the case of doubling data a hash table is a proper structure for data aggregation of identical key values, which solves the problem of efficient update of data.

Column-oriented databases implement a data model similar to the relational one, differing, however, in the physical way of storing data. In the classic relational database, the table records are stored in a data file one after another. Column-oriented databases offer the possibility of storing data grouped in columns. It facilitates placing the table on several servers, accelerates executing queries selecting data from a small number of columns and allows for applying simple methods of compression (e.g. RLE) for data stored in one column (fig. 2).

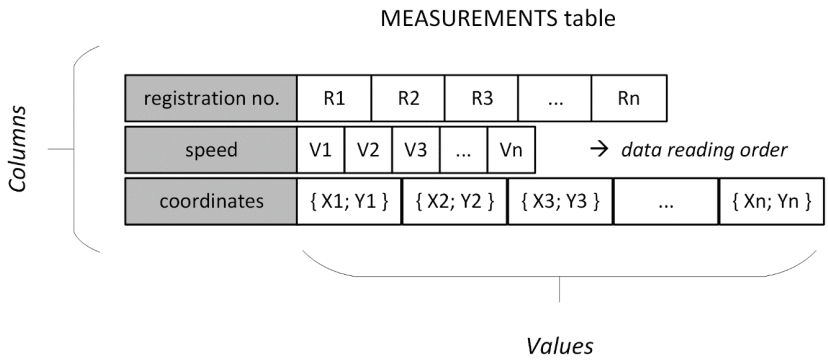

Fig. 2. Structure of column-oriented database table.

This solution brings a significant increase of efficiency when downloading data especially in the case of a small number of returned columns [7]. Thus, the SQL command in the form `SELECT AVG(SPEED) FROM MEASUREMENTS` calculating the average speed from the table `MEASUREMENTS` of the structure presented in fig. 2, will be executed in time $T_c = t_{seek} + N \times t_{read}$, where N – number of records in the table, t_{seek} – hard disc head positioning time in order to establish the current position of the read, t_{read} – time of read of one record from the disc (for simplification, it can be assumed that one read operation reads in exactly one record, data compression is not used). For a relational database, the respective time of operations will be higher and will equal $T_r = N \times (t_{seek} + t_{read})$ because each read operation of a consecutive value from a speed column is preceded by the operation of head positioning.

Fast data read from selected columns constitutes a premise to the use of column-oriented databases in order to gather and store data intended for further analytical processing OLAP. In this case low efficiency of data record in data warehouse does not affect the whole system, whereas, data compression allows for saving space on a device. Among the disadvantages of column-oriented databases there is slower data input and lower efficiency of queries operating on many columns, therefore, their use for operational data processing may be not very efficient.

Graph databases use directed graphs as a data model. Data is stored in graph nodes, wherein, the database does not impose a target structure, allowing for its flexible modification during system operation. Data connections are realized in the form of a graph edge. Each edge connecting nodes can have any number of attributes, which allows for realization of complex semantic dependencies between data. The model realized by graph databases can be used in those transport information systems in which one of basic operations is indication of optimal travel route. Among them are navigation systems, passenger information systems, etc. The concept of structure organization of such a system will be presented in the context of algorithm of designating public transport connections. The simplified data model consists of the entity *line* (number and description of public transport line), *stop* (identifier, name of the stop) and *connection* (direct connection between two stops belonging to the same line with indication of their sequence, distance and travel time). Connections between database entities of the system are presented in Fig. 3.

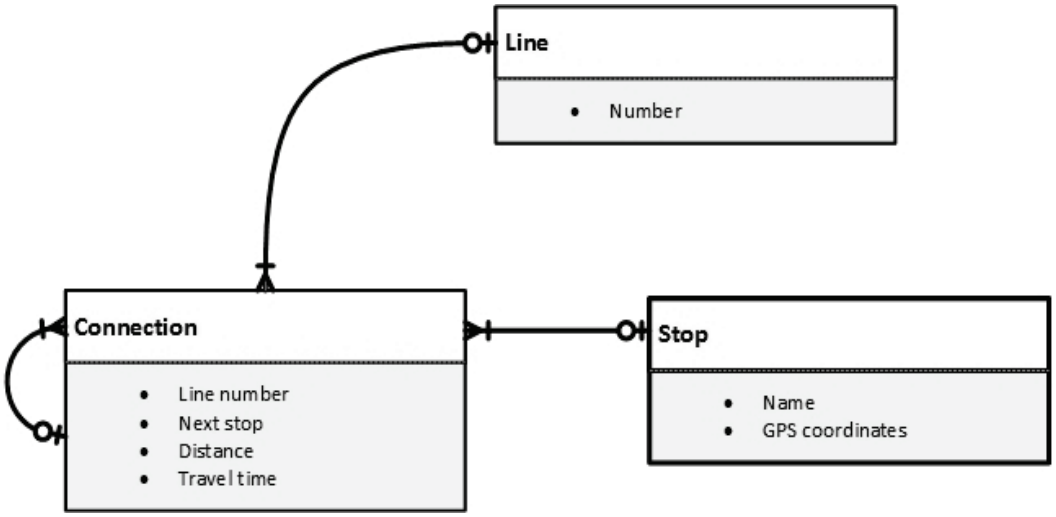

Fig. 3. Entity relationship diagram of database model of public transport connections.

The equivalent of this structure in a graph data model is the structure presented in fig 4. The graph node is the *stop*. The set of attributes of the stop is constituted by its *name* and e.g. *GPS coordinates*. Graph edges correspond to *connections* between the consecutive two stops of a chosen line. Each edge connecting two stops has the attributes: *line number*, *distance* and *travel time* (the simplified model does not account for timetables for each of the stops).

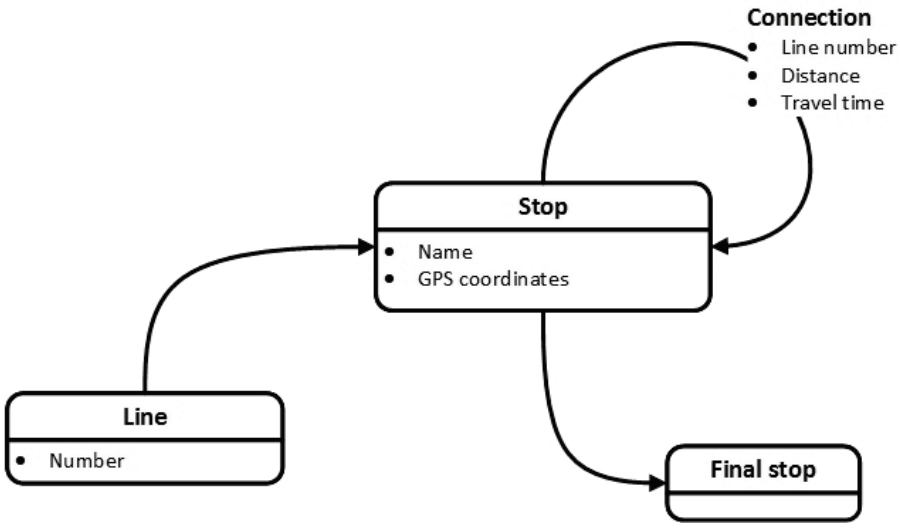

Fig. 4. Data structure of a graph model.

Indication of a *travel route* from the stop P_a to the stop P_b ($P_a, P_b \in P$ where P is a set of all stops in the system) can be defined as a task of path-finding $S = \{ S_1, S_2, \dots S_k \}$ consisting of direct *connections* $S_i = P_x \rightarrow P_y$between the stops P_x i P_y, wherein, the final stop P_y of the connection S_i is the initial stop P_x of the connection $S_{i+1}$for $i = [1, k-1]$. Fig. 5 presents a fragment of Neo4j graph database created according to the above-described scheme. Example includes five stops P1 ... P5 served by two lines A and B.

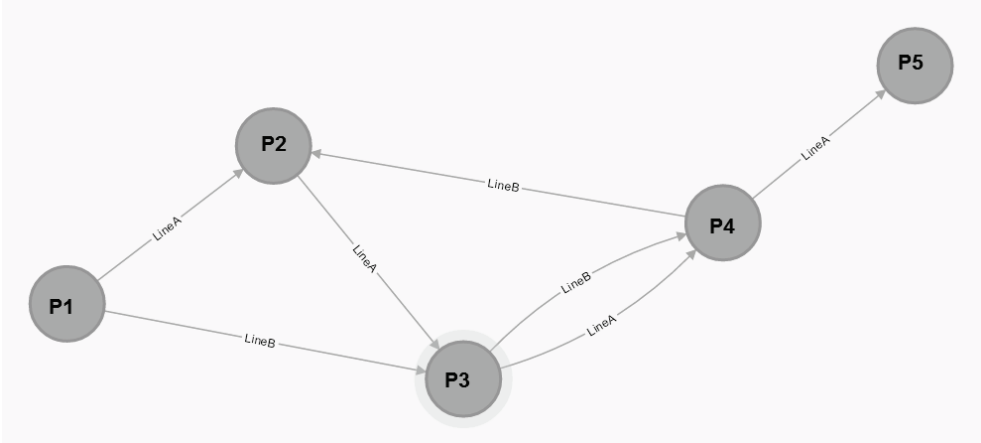

Fig. 5. Example nodes and relations created using Neo4j graph database.

In the system using a relational database, indication of a travel route requires programmatic implementation of search algorithm in a graph (e.g. Dijkstra's algorithm) with the use of recurrent queries in SQL language. In a graph database, the same task can be formulated by means of a single query in *Cypher* language [8] presented in fig. 6.

```
match
    p=(start: Stop)-[*]->(finish: Stop)
where
    start.name='P1' and finish.name='P4'
return
    p,
    start.name as Start, finish.name as Finish,
    extract( n in nodes(p) | n.name ) as Stops,
    extract( r in relationships(p) | r.name ) as Connections
```

Fig. 6. All connections search between two stops in a graph database.

The main asset of graph databases is the speed of path-finding in a graph: analogous structure realized in a relational database requires application of complex SQL queries and may be less effective [9]. The strength of the model is the use of a graph theory providing solutions of many problems in the form of ready algorithms. Among the disadvantages of graph databases, a limited range of effective application should be included. Although theoretically, each data set can be presented in the form of a graph, from the practical point of view it is not always cost effective. Relational databases can much faster deal with, for example, calculating the arithmetic mean of values stored in a given column.

Summing up, the advantages of each type of NoSQL database can be used in transport information systems. However, it should be noted that none of the NoSQL database system used individually is able to ensure comprehensive functionality in the whole system perspective. The reason is mainly lack of certainty that data processed in such a system meets the criteria of integrity. Therefore, NoSQL databases should be treated, first of all, as a tool extending the possibilities of relational databases in selected areas of application.

The concept of possible cooperation between databases within a distributed information system is presented in fig. 7.

4. SUMMARY

NoSQL databases constitute a dynamically developing segment of data processing systems market. Due to the ability to process large sets of data, they have grown in popularity mainly in the use distributed information systems oriented for a large number of users and storing large sets of data.

The article has presented the directions for development of modern information systems, enumerated basic characteristics of NoSQL databases and presented potential areas of their application. On the example of a relational model transformation to a graph form, the concept of application of a graph database to searching connections in public transport has been demonstrated. The general concept of a distributed information system has been presented, which uses cooperation of relational databases with NoSQL databases.

Development of information systems in transport in the direction of distributed systems oriented for active participation of users, allows for forecasting a gradual increase of application of the NoSQL databases in such systems. However, it should be noticed that specific faults of NoSQL databases, practically exclude their use in critical areas such as crisis management systems or toll

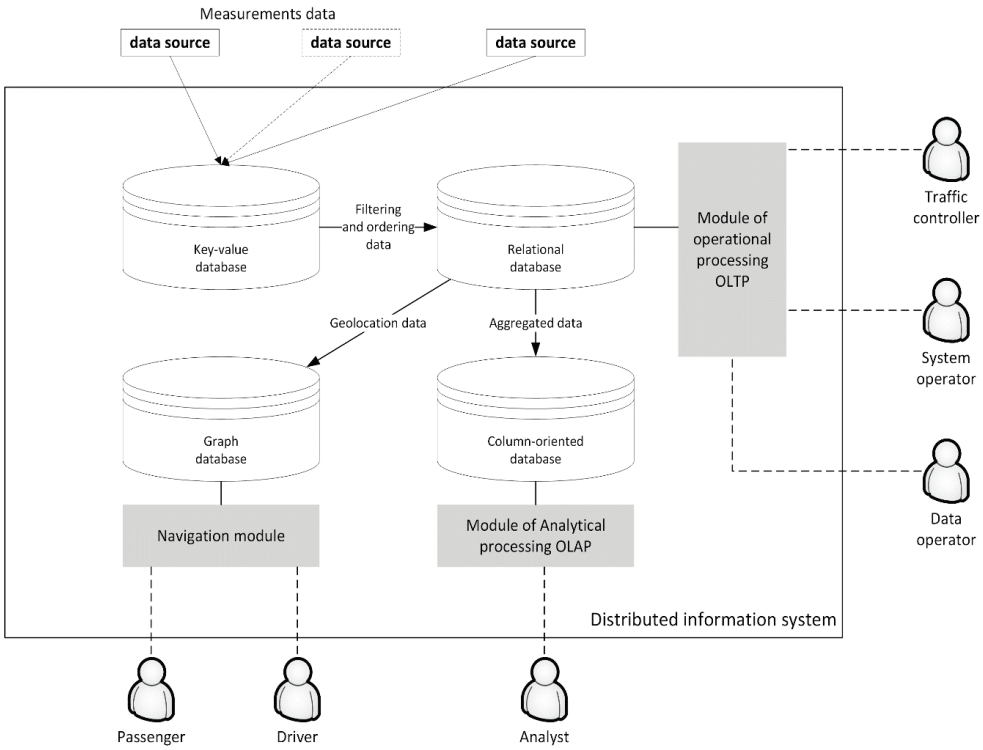

Fig. 7. Concept of distributed information system in transport with the use of NoSQL databases.

collection systems. Therefore, application of NoSQL databases should be considered, first of all in the context of complementing the functionality of classic relational systems.

REFERENCES

[1] Jing H. Survey on NoSQL database. 6th International Conference of Pervasive Computing and Applications (ICPCA), 2011, pp. 363-366.
[2] Grochowski L.: Distributed information systems, Publishing House Elipsa, Warsaw 2003.
[3] http://yanosik.pl/android (electronic source, checked 2016.05.28).
[4] Czerepicki A., Góralski A.: Transformation of heterogeneous data in Data Warehouse Systems. "Computer science studies" No 1(3)/2012, Warsaw Management Academy, Warsaw 2012.
[5] Brewer E. Certain Freedom: Thoughts on the CAP Theorem. Proceeding of the XXIX ACM SIGACT-SIGOPS symposium on Principles of distributed computing. ACM, New York, 2010.
[6] Skalski D.: NoSQL – non-relational database systems. Software Developer 08/2011, Warsaw, 2012.
[7] Stonebraker M., Bear C., Çetintemel U., Cherniack M. One Size Fits All? – Part 2: Benchmarking Results. 3rd Biennial Conference on Innovative Data Systems Research, January 7-10, 2007, Asilomar, California, USA.
[8] Cypher Language [http://neo4j.com/docs/developer-manual/current/#cypher-query-lang, checked 2016.05.28]
[9] Renzo Angles, Claudio Gutierrez. Survey of Graph Database Models. ACM Computing Surveys, vol. 40 Issue 1, February 2008. ACM New York, USA.

Date submitted: 2013-11-03
Date accepted for publishing: 2016-07-01

Andrzej Czerepicki
Warsaw University of Technology, Poland
aczerepicki@wp.pl