DOI 10.20544/TTS2018.P02
UDK 656.07:004.658.2

MONGODB DATABASES IN BIG DATA APPLICATIONS IN TRANSPORTATION INDUSTRY[1]

Slađana Janković, Snežana Mladenović, Stefan Zdravković, Slavko Vesković, Ana Uzelac
University of Belgrade
Faculty of Transport and Traffic Engineering, Vojvode Stepe 305
Belgrade, Serbia
s.jankovic@sf.bg.ac.rs

Abstract

M2M (Machine-to-Machine) technologies provide a number of solutions in logistics and transport, such as fleet management solutions, asset tracking systems, parking space management and payment, road tolls, traffic volume monitoring, connected road signs, traffic lights and cameras, connected navigation, passenger services, etc. The data obtained is huge and need high performance to be usable. NoSQL data systems are the answer to it. When compared to relational databases, many NoSQL systems share several key characteristics including a more flexible data model, higher scalability, and a superior performance. Although there are dozens of NoSQL databases, they primarily fall into one of the following three categories: document databases, graph databases and databases based on key-value and wide column model. MongoDB is an open-source document database. In this paper the possibilities of using MongoDB databases in innovative applications of Big Data in the transportation industry will be explored.

Keywords - NoSQL database; document database; Big Data technology

[1] professional paper

INTRODUCTION

Smart M2M (Machine-to-Machine) solutions can be applied to manage fleets of commercial vehicles and to passenger vehicles to improve safety, quality of service, comfort and convenience for drivers [1]. Intelligent transport systems based on M2M technologies can help to optimize and control the traffic flow of vehicles, reducing costs and carbon footprint. Its applications are countless: parking space management and payment, road tolls, traffic volume monitoring, connected road signs, traffic lights and cameras, etc. Modern urban traffic data is obtained from hundreds traffic sensor loops, thousands GPS (Global Positioning System) car routes from traffic and navigation applications, millions of traffic-related short messages in social networks. As a result, traffic data volume and dimensions grow rapidly and data becomes more heterogeneous, which make application of conventional data management methods difficult and inefficient [2]. Today, the traffic data management is a field which requires the application of Big Data technologies [3]. Big Data refers to technologies and initiatives that involve data that is too diverse, fast-changing or massive for conventional technologies, skills and infrastructure to address efficiently. Said differently, the volume, velocity or variety of data is too great [4].

In this paper, we will consider using one of the most popular NoSQL databases today – MongoDB, in Big Data applications in transportation. In the second section of the paper, an overview of MongoDB is given. Some of the successful stories of using MongoDB databases in transportation are presented in the third section of the paper. In order to discover the possibilities of MongoDB databases we have implemented our case study. The fourth section presents our MongoDB in transportation case study. In the conclusion section, the main advantages of using MongoDB databases in transport, will be highlighted.

MONGODB OVERVIEW

MongoDB is an open source, document oriented database that provides high performance, high availability, and easy scalability. MongoDB is based on concept of collection and document. Database is a physical container for collections [5]. Each database gets its own set of files on the file system. A single MongoDB server typically has multiple databases. Collection is a group of MongoDB documents. It is the equivalent of an RDBMS (Relational Database Management System) table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection are of similar or related purpose. A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documents in the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data [6]. Any relational database has a typical schema design that shows number of tables and the relationships between these tables, while the concept of relationship does not exist in MongoDB. Advantages of MongoDB compared to RDBMS:

- Schema less.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that is almost as powerful as SQL.
- MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects is not needed.
- Uses internal memory for storing the working set, enabling faster access to data.

MONGODB IN TRANSPORTATION USE CASES

The United States Department of Transportation is setting up central service to monitor traffic conditions nationwide, deploying sensors throughout the interstate system that monitor traffic conditions including car speeds, pavement and weather conditions, as well as accidents, construction, and other sources of traffic tie ups. MongoDB has been selected as the database for this application [7].

In order to meet challenges that come with Big Data and IoT (Internet of Things), Bosch started to use MongoDB. This led to the following three solutions: Community-based parking [8], Monitor Freight cars solution [9], and Connected Vehicle Fleet Management system [10].

Their Community-based parking solution simplifies parking in cities using cloud-based solutions and can be used to rapidly locate available parking spaces on the street. The “community” consists of two groups: drivers that are actively seeking for a parking spaces and drivers who are taking their way around the city. Transmitter vehicles drive through the city identifying available spaces making this data available to receiver vehicles. As the number of participant grows, the parking maps become more precise and up to date and can be used to guide drivers to the available parking spaces [8].

The second Bosch solution represents condition monitoring system for rail freight transport and enables asset monitoring. A box with different sensors is attached to every freight car and collects various data such as the precise position of the railcar, the temperature in the storage area, etc. Collected information is transferred to the server where fleet operators, customers or the goods recipients can access the data on the server any time to get a snap shot of the data about car and the freight [9].

Bosch’s Connected Vehicle Fleet Management system is based on custom-built series of sensors and monitoring systems and enables real-time journey analysis, tracks vehicle consumption and geo-fences vehicles supporting complete transparency in every moment. A minute-by-minute information on each vehicle is enabled giving the key insights into best routes, fuel, maintenance and the level each driver contributes to the overall performance. It can be used to optimize the safety, security and energy capabilities of the fleet [10].

MONGODB IN TRANSPORTATION CASE STUDY

In order to investigate the possibilities of MongoDB databases, we have built a case study in the domain of the road transport. The case study includes storing and processing of data generated by several automatic traffic counters (ATCs) located in the Republic of Serbia in 2015. 14 automatic traffic counters of type QLTC-10C were located on 10 different locations in the area of town Novi Sad and its surroundings. ATCs are used to track traffic and are located on roads and city boulevards: a single counter is placed on roads and is responsible for tracking traffic in both directions, while two counters are placed on city boulevards – each tracking traffic for one direction. Every day, each ATC generates a single txt file. This file contains all the data related to all the vehicles that were registered by the counter that day. This means that each ATC during one year generates 365 txt files. Each record of the txt files consists of the following fields: index, date and time, channel, lane, vehicle class, vehicle speed, and vehicle length. Based on this data, volume of traffic, as well as the vehicles speed, can be tracked. In our study, we have calculated basic indicators of traffic volume, such as: Annual Average Daily Traffic (AADT), Monthly Average Daily Traffic (MADT), AADT by directions, AADT by vehicle categories, AADT by directions and vehicle categories. The case study is realized through the following phases:

1. Invalid records are removed from txt files,
2. The content of 365 txt files generated by single ATC are merged in a single large txt file. That way, 14 large txt files are generated, each sized up to 1.5 GB.
3. Using MongoDB Community Edition 3.6, MongoDB database traffic_counting is created,
4. Using Pentaho Data Integration software and its graphical transformation and job designer – Spoon, data received from 14 large txt files formed a collection named annual_traffic in MangoDB database traffic_counting. Every registered vehicle is related to the one document in MongoDB collection annual_traffic. In the Fig 1, one document from annual_traffic collection in the graphical user interface for MongoDB – MongoDB Compass Community is presented. In Fig 2, it can be seen that collection annual_traffic consists of 47 778 518 documents.
5. Using query shown in Fig 3, the collection annual_number_of_vehicles_by_counter from MongoDB database traffic_counting is created and filled with data. One

document that belongs to the collection is presented in Fig 4. Using similar queries the following collections were have created: annual_number_of_vehicles_by_counter_and_by_direction, annual_number_of_vehicles_by_counter_and_by_vehicle_category, annual_number_of_vehicles_by_counter_and_by_direction_and_by_vehicle_category.

```
_id: ObjectId("5a5e55a82bde852520d8900b")
counter_id: 7
counter_name: "Sancevi"
counter_latitude: 45.318827
counter_longitude: 19.834347
vehicle_ordinal_number: 325
date: "02.01.15"
time: "08:23:25"
loop_number: 3
direction: 1
vehicle_category: "B1"
vehicle_speed: 51
vehicle_length: 556
```

Fig.1. One document from annual_traffic MongoDB collection

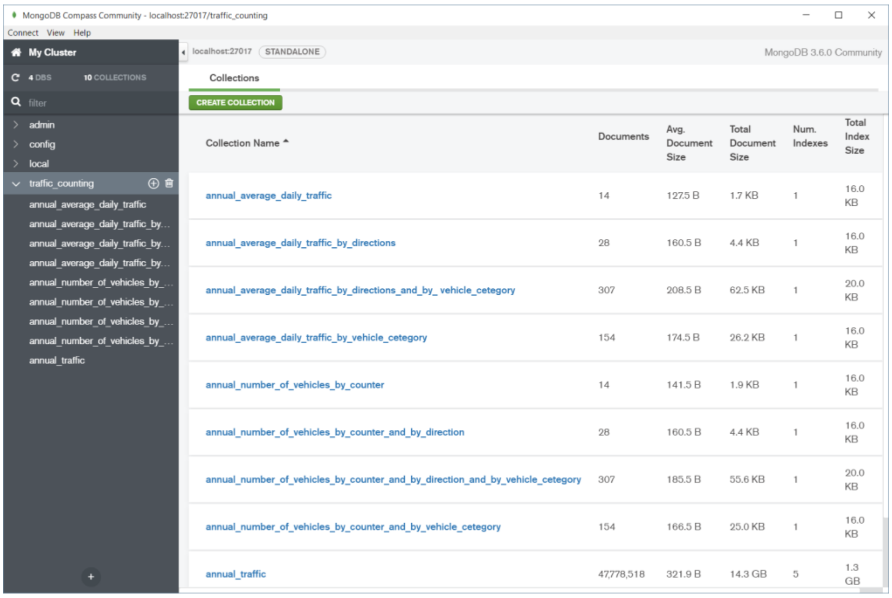

Fig.2. The MongoDB database traffic_counting

```
db.annual_traffic.aggregate([
      {$group:
            _id: {counter_name: "$counter_name",
            counter_latitude: "$counter_latitude",
            counter_longitude: "$counter_longitude"},
            number_of_vehicles: {$sum:1}}},
            $sort: {_id : 1}},
      {$out:"annual_number_of_vehicles_by_counter"}])
```

Fig.3. The MongoDB Query which generates annual number of vehicles_by_counter MongoDB collection

```
⌄ _id: Object
    counter_name: "Sancevi"
    counter_latitude: 45.318827
    counter_longitude: 19.834347
  number_of_vehicles: 4240576
```

Fig.4. One document from annual_number_of_vehicles_by_counter MongoDB collection

```
⌄ _id: Object
    counter_name: "Sancevi"
    counter_latitude: 45.318827
    counter_longitude: 19.834347
  AADT: 11618
```

Fig.5. One document from annual_average_daily_traffic MongoDB collection

One document that belongs to the collection annual_average_ daily_traffic is presented on Fig 5. Using query presented in Fig 6, the collection annual_average_daily_traffic in MongoDB database traffic_counting is created and filled. The data from the collection annual_average_daily_traffic are presented on the Table 1.

```
db. annual_number_of_vehicles_by_counter.aggregate([
      {$project:
            {_id:1,
            AADT:{$floor:
            {$divide:["$number_of_vehicles", 365]}}}},
      {$sort: {_id: 1}},
      {$out: "annual_average_daily_traffic"}])
```

Fig.6. The MongoDB Query which generates annual_average_daily_traffic MongoDB collection

Table 1. Data from the annual_average_daily_traffic MongoDB collection

Automatic Traffic Counter Location	**Annual Average Daily Traffic**
Alibegovac	8011
Bocke	12230
Bulevar Evrope 3 - K Stankovica - Rumenacki put	5581
Bulevar Evrope 3 - Rumenacki put - K Stankovica	4400
Klisa	9352
Paragovo	10067
Sancevi	11618
Somborski bul (Patrijarha Pavla) - Bul Evrope - Vrsacka	11435
Somborski bul (Patrijarha Pavla) - Vrsacka - Bul Evrope	10298
Tekije	13701
Tunel - Irig - Novi Sad	5775
Tunel - Novi Sad - Irig	7976
Vojvode Stepe - Rumenacka - S Jovanovica	13108
Vojvode Stepe - S Jovanovica - Rumenacka	12342

Using similar queries, the following collections were created: annual_average_daily_traffic_by_directions, annual_average_daily_traffic_by_vehicle_category, annual_average_daily_traffic_by_directions_and_by_vehicle_category. The Fig 2 shows MongoDB database traffic_counting and its all 9 collections.

6. Using Pentaho Data Integration software and its graphical transformation and job designer – Spoon, tabular and graphical visualization of collections from MongoDB database traffic_counting is made. A window from the Spoon tool that contains data from the annual_average_daily_traffic collection presented on geographic map is shown in Fig 7.

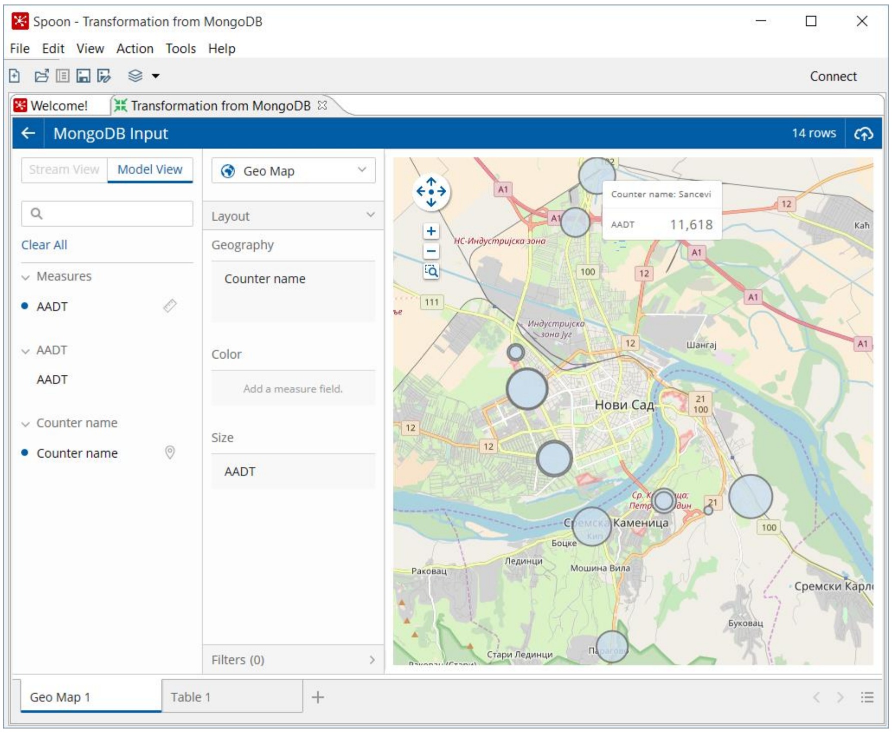

Fig.7. Geo map view of selected data from MongoDB database in Spoon

In Fig 8 a window from Spoon tool that represents graphical presentation of the percentage distribution of AADT for cargo vehicles classified by categories for the traffic counter Paragovo is shown. Spoon enables creating and executing transformations and jobs. Transformations are used to describe the data flows for ETL (Extract, Transform, Load) such as reading from a source, transforming data and loading it into a target location. We have created two types of transformations: one used to transfer data from txt files to MongoDB database, while the other is used to grafically present and geolocate the results received from MongoDB database data processing.

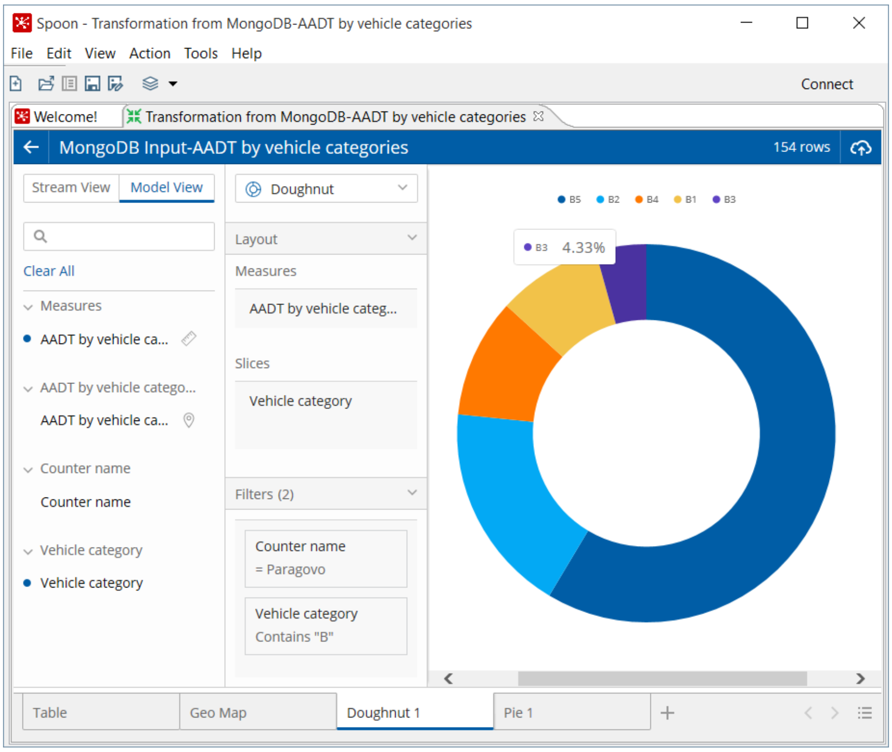

Fig.8. Doughnut view of selected data from MongoDB database in Spoon

CONCLUSION

Unlike relational databases, creating schema of MongoDB database is simpler and takes less time. In order to optimize queries, it is more efficient to create different non-relational schemas for the same raw data, then to design one relational schema together with complicated and demanding queries. The number of data sources in the traffic industry grows every day and there is a constant need to connect data received from various and heterogeneous sources. MongosDB's flexibility and dynamics represents the key advantage comparing to the traditional relational databases. The number of applications of traffic data and the number of their users grow rapidly every day. Therefore, there is a growing demand to develop new and enhance existing applications. Flexible model of MongoDB shortens the time required for the application development. Additionally, MongoDB provide a rich set of indexing options to optimize a wide variety of queries, including text indexes, geospatial indexes, etc. Traffic and transport generate huge amount of data every minute and the good performances of MongoDB databases to process those huge amount of data is another great advantage. Flexibility and good performances make this type of database suitable for developing traffic and transportation databases.

ACKNOWLEDGMENT

This paper has been partially supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia project under No. 36012, and the European Commission "Horizon 2020 – the Framework Programme for Research and Innovation (2014-2020)" project under the name "Novel Decision Support tool for Evaluating Strategic Big Data investments in Transport and Intelligent Mobility Services – NOESIS". The data generated by automatic traffic counters provided by the company MHM - Project from Novi Sad.

REFERENCES

[1] SMARTM2MSOLUTIONS. (2018, January 4). Logistics & Transport. Available: http://www.smartm2msolutions.com/applications/logistics-transport/.

[2] E. Vlahogianni, B. Park and J.W.C. Lint, "Big data in transportation and traffic engineering," Transportation Research Part C: Emerging Technologies, Vol. 58, Part B, 2015, pp. 161.

[3] D. Simić, V. Ilin, M. Kopić, V. Svirčević, R. Jovičić, "Big data on large urban logistics for sustainability development cities," The 5th International Conference "Towards A Humane City", Novi Sad, Serbia, 2015, pp. 363-368.

[4] MongoDB. (2018, January 5). Big Data Explained. Available: https://www.mongodb.com/big-data-explained.

[5] K. Banker, P. Bakkum, S. Verch, D. Garrett and T. Hawkins, MongoDB in Action: Covers MongoDB version 3.0, Manning Publications, USA, 2016.

[6] D. Sullivan, NoSQL for Mere Mortals, Addison-Wesley, USA, 2015.

[7] MongoDB. (2018, January 15). MongoDB for Time Series Data. Available: https://www.mongodb.com/presentations/mongodb-time-series-data-part-1-setting-stage-sensor-management.

[8] Bosch Group. (2018, January 23). Community-based parking. Available: http://www.bosch-mobility-solutions.com/en/products-and-services/mobility-services/connected-parking/community-based-parking/.

[9] Bosch Group. (2018, January 23). Asset Monitoring. Available: http://www.bosch-engineering.de/en/de/einsatzgebiete/schienenfahrzeuge/amra/amra_2.html.

[10] Bosch Group. (2018, January 23). Bosch Connected Services. Available: http://www.boschconnectedvehicle.com/us/en/index.html.