Scientific Journals
of the Maritime University of Szczecin

Zeszyty Naukowe
Akademii Morskiej w Szczecinie

2022, 71 (143), 93–101
ISSN 1733-8670 (Printed)
ISSN 2392-0378 (Online)
DOI: 10.17402/522

Received: *25.10.2021*
Accepted: *19.05.2022*
Published: *30.09.2022*

Performance of AIS geoinformation extraction using SQL and NoSQL TranStat databases

Wojciech Czapliński[1], Wojciech Gąsowski[2]

[1] https://orcid.org/0000-0002-9503-4742
[2] https://orcid.org/0000-0001-6804-8324

Maritime University of Szczecin
1-2 Wały Chrobrego St., 70-500 Szczecin, Poland
e-mails: [1]w.czaplinski@am.szczecin.pl, [2]w.gasowski@am.szczecin.pl
corresponding author

Keywords: database, AIS, spatial, geospatial, PostgreSQL, postgis, MongoDB

JEL Classification: R41, R42, Y10, R10

Abstract

The Automatic Identification System (AIS) device is mandatory for ships that comply with the International Convention for the Safety of Life at Sea (SOLAS). AIS is intended for vessel traffic monitoring to improve shipping safety. In the examined area, the base station received 22 128 345 messages in April 2019. Approximately 80% of these messages included position reports, which were subjected to geospatial analysis. One possible utilization of AIS messages is used in an intelligent maritime transport statistics production system called TranStat in the Gospostrateg project. This specific study compares the speed of executing geospatial queries in a relational PostgreSQL database engine and a non-relational MongoDB database engine. For the purpose of this research, we have defined four AIS datasets, four test polygons of varied number of vertices, and a reference point on a fairway. The tests were used to assess the execution of the queries in a database that returns the number of ships located in a predefined area and the number of ships located at a preset distance from the defined point. It has been determined from the test results that test queries are performed faster and data stored in the database occupy less disk space in MongoDB than in PostgreSQL. Faster geospatial analysis of AIS messages may improve the navigation safety by earlier detection of dangerous situations.

Introduction

The amount of geo-spatial information, such as AIS messages, has increased over the last few years. Marine monitoring services store AIS messages from about 100 000 vessels in real time. This creates new challenges for the efficient processing of such data (Varlamis, Tserpes, and Sardianos, 2018). BigData solutions or databases are required to manage this amount of data. In this paper, the authors compare relational and NoSQL databases.

A TranStat system is based on an AIS system as a source of information, while an Automatic Identification System (AIS) identifies the ships. It is composed of shore base stations and shipboard devices. The system provides automatic two-way ship-to-ship and ship-to-shore data exchange. AIS devices are also installed as aids for navigation. AIS was developed to enhance the safety of navigation and it is used for monitoring ship traffic and as an anticollision tool. There are three types of data exchanged in AIS (Xu et al., 2016):
- static,
- dynamic,
- voyage related.

Twenty seven basic types of AIS messages have been defined. Each type has a specific purpose, e.g. messages 1, 2, and 3 are position reports – i.e. the position in the reference system WGS-84 (World Geodetic System of 1984) – containing ship dynamic data. Messages 6, 8, 25, and 26 are binary data transmissions.

Depending on the type of message, various transmission frequencies are used. The following transmission intervals (ITU, 2014) are adopted:

- static data – every 6 minutes, if changes take place – on request;
- dynamic data – depending on ship speed and course alterations (every 2 seconds to 3 minutes), with satellite AIS (long-range broadcast message) – every 3 minutes;
- voyage related information – every 6 minutes, if changes take place – on request;
- safety information – if necessary.

The description of specific messages, and the number recorded in April 2019, are given in Table 1. The data derives from the AIS base station located at the main building of the Maritime University of Szczecin (MUS). Messages were decoded from the NMEA RAW files stored on the base station's hard drive.

Each of the 27 types of messages was defined for a purpose, and almost all of them have a different structure. The same type of message may have various attributes, depending on the previously set parameters. Approximately 80% of the AIS messages are position reports (Table 1), which enables analysis in the spatial context.

The nature of the data sent via AIS requires a proper database management system (DBMS) for their optimal storage and processing. Because of this, the following factors should be considered:

- structure of stored data;
- quantity of stored data;
- rate at which data flows;
- method of data processing, in particular the processing of spatial data in the WGS-84 reference system.

Table 1. Description and number of AIS messages by type in April 2019, from the base station at Maritime University of Szczecin (ITU, 2014)

No.	Description	Number of messages	%
1	Routine position report	5 422 185	24.50e+0
2	Assigned position report	11 862 111	53.61e+0
3	Special position report broadcast on request	1 438 710	06.50e+0
4	Position, date and UTC, current slot number of the base station	1 417 394	06.41e+0
5	Static and voyage related data	436 558	01.97e+0
6	Addressed message, binary data	60 605	27.39e–2
7	Acknowledgement of addressed binary data receipt	158 568	71.66e–2
8	Dissemination message to all stations, binary data	125 634	56.78e–2
9	Position report sent by aircraft stations in SAR operations	285	12.88e–4
10	Requests to state date and UTC	1	04.52e–6
11	Specifying current date and UTC, if available	180	08.13e–4
12	Addressed message on safety	9	40.67e–6
13	Acknowledgement of addressed safety data receipt	5	22.60e–6
14	Dissemination message on safety	9	40.67e–6
15	Request for sending a specific message	271	12.25e–4
16	Specific method of transmission assigned by base station	0	00.00e+0
17	DGNSS corrections disseminated by base station	66 716	30.15e–2
18	Routine position report	189 745	85.75e–2
19	Extended position report including static data	0	00.00e+0
20	Reserved for base station	475 292	02.15e+0
21	Dissemination of slot reservation for base station	51 224	23.15e–2
22	Report on the position and status sent by aid to navigation	0	00.00e+0
23	Management of transmission parameters of mobile AIS stations in a specific area by base station	222 788	01.01e+0
24	AIS station data	125 292	56.62e–2
25	Short unplanned transmission of binary data	0	00.00e+0
26	Short unplanned transmission of binary data	0	00.00e+0
27	Long-range (satellite) AIS message	74 763	33.79e–2
	Total number of messages	22 128 345	

A large number of AIS messages received by the shipboard devices, as well as by the shore-based stations, requires efficient databases. Basically, the databases can be divided into:
- relational databases;
- non-relational databases, so called Not only SQL (NoSQL).

In terms of performance, the optimal choice for systematized data is relational databases. For data with an unknown structure at the design stage, or where the structure changes with time, NoSQL databases are more appropriate (Baralis et al., 2017).

The relational database – PostgreSQL

Relational databases are commonly used systems for storing and processing large volumes of data. In the relational model, the internal structure of data always consists of a set of columns and rows, and the data in a table (records) must have a strictly defined structure. The Structured Query Language (SQL) is the most common mechanism for defining queries and modifying a relational database. Indexes are used to search for records with specified values of an attribute. The index is a data structure that – after extracting a certain record property, usually the value of one or more fields – enables faster extraction of records with that property. Correctly defined indexes exclude the need to search all the records in the table (Garcia-Molina, et al., 2006). Increasing volumes of collected data led to a seeking of effective solutions for data processing, including spatial data.

PostGIS is a module extension for the relational database PostgreSQL, which makes it possible to define geographical data and execute queries for locations via SQL. This extension was designed to consider rules for SQL defined by the Open Geospatial Consortium (OGC, 2019) (OGC). It supports all the standard types of geometric objects (e.g. point, line, and polygon) and standard spatial operations (e.g. distance and intersection). For optimized spatial queries, PostGIS enables (PostGIS, 2019) the use of the indexes:
- Generalized Search Trees (GiST),
- binary trees (B-trees),
- sub-rectangle trees (R-trees).

NoSQL Database – MongoDB

NoSQL databases are increasingly used for storing non-structured data, such as documents. In a NoSQL database, documents belonging to the same set may have different names of attributes – the data schema. No attributes indicate that it has not been created or it is irrelevant for the document. Modification of the document attributes does not require them to be defined beforehand. One of the most popular NoSQL databases, with documents stored in collections, is MongoDB. It provides operations on spatial data.

Spatial data in MongoDB comply with the notation of Geospatial JSON (GeoJSON) defined by OGC (OGC, 2019). Basic spatial operations are supported (containment, intersection, and neighborhood) and there are seven types of geometric objects (MongoDB, 2019):
- Point,
- MultiPoint,
- LineString,
- MultiLineString,
- Polygon,
- MultiPolygon,
- GeometryCollection.

MongoDB offers embedded mechanisms for analyzing a spherical surface (in the WGS-84 reference system) and a planar (Euclidean) surface. For this purpose, the database uses indexes that are named 2dsphere and 2d, respectively.

Methodology

Test environment

Tests were performed in Windows 10×64 with a Docker 2.0.0.3 environment. Docker is a platform for the development, implementation, and start-up of applications in an isolated environment called a container. Three containers have been prepared for test purposes:
- database server Postgres 11.2 with PostGIS 2.5.2 extension,
- database server MongoDB 4.0.10,
- client application Python 3.5.7 with libraries (i.e. pymongo, psycopg2, and libais).

Software was run on a computer with the following hardware configuration:
- processor: Intel Core i7-8650U,
- RAM: 16 GB,
- hard disk: Solid State hard Drive (SSD) M.2, capacity 512 GB.

The resources allocated to the Docker environment were as follows:
- 4 processor cores,
- 8 GB of RAM,
- 2 GB size of swap file.

Each container had access to the full resources allocated to the Docker application.

Test data

The scope of the test data includes a period of one full month from 00:00:00 UTC, 1 April 2019 to 23:59:59 UTC, 30 April 2019. Over this period, 22 183 325 AIS messages were received. To standardize the data, only messages 1, 2, and 3 were recorded in the database. These messages include spatial data, have a similar structure, and are transmitted most frequently. In April 2019, 18 723 006 of these messages were received, which is 84.611% of all incoming traffic. The AIS station installed at the Maritime University of Szczecin covers the area up to Tower Gate 1 of the Szczecin–Świnoujście fairway to the north, and the town of Schwedt by the Odra River to the south (Figure 1).

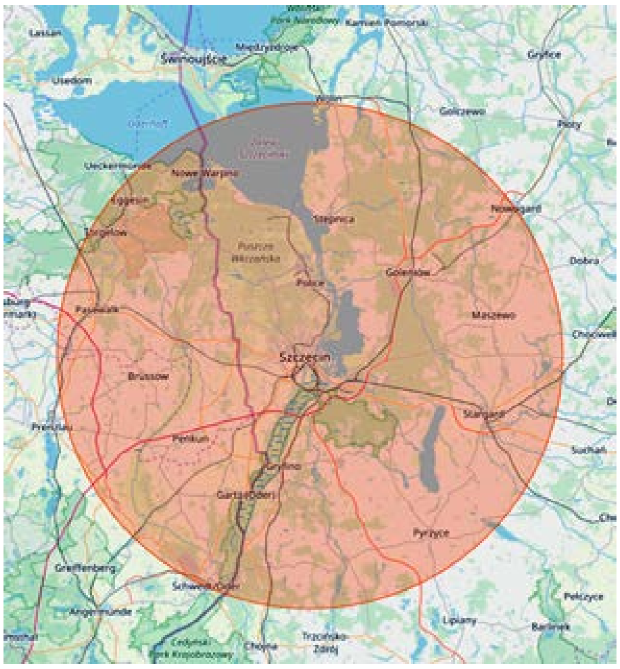

Figure 1. Coverage of the AIS station installed at the Maritime University of Szczecin

To simplify the nomenclature used herein, it is assumed that:

- a record means a row in the table (PostgreSQL) or a document in a collection (MongoDB),
- a set means the table (PostgreSQL) or a collection (MongoDB).

Four datasets containing AIS messages No. 1, 2, and 3 were prepared from an analysis of the received messages. The sets were divided to cover four different time intervals, all commencing from the same moment. The names of the sets correspond to the time interval of the message reception. Details concerning each set, time intervals, and number of messages in the sets are given in the Table 2.

Table 2. Details of the test datasets

Name of set	Time interval	Number of messages
1 hour (1H)	⟨01.04.2019 00:00; 01.04.2019 01:00⟩	24 844
1 day (1D)	⟨01.04.2019 00:00; 02.04.2019 00:00⟩	653 283
1 week (1W)	⟨01.04.2019 00:00; 08.04.2019 00:00⟩	4 974 151
1 month (1M)	⟨01.04.2019 00:00; 01.05.2019 00:00⟩	18 723 006
Total		24 375 284

Indexes were defined on the datasets to optimize queries. Spatial objects in the PostgreSQL used a GIST, while in MongoDB a 2dsphere index was employed. Tests were performed to determine the times for executing a single query on datasets with and without indexes. These instances are presented at Figure 2.

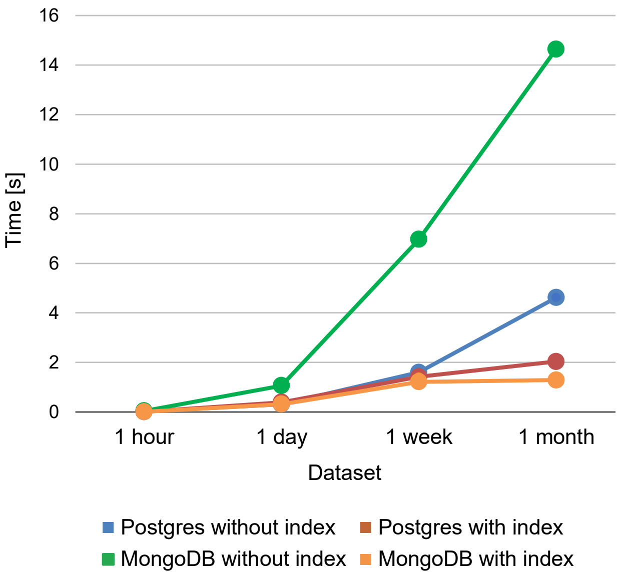

Figure 2. Comparison of the query execution time (in the databases PostgreSQL and MongoDB) for the datasets with and without an index

The results show that the queries that use indexes are executed faster than those without indexing. Therefore, later in this study, query performance tests were conducted for indexed objects only.

Performance tests

The following operations were performed on an identical dataset to compare the performance of the PostgreSQL (Matthew & Stones, 2005) and MongoDB (Plugge, Membrey & Hawkins, 2010) databases:

1. adding records to the database (time length of addition, use of the processor and memory),
2. counting the number of ships in the defined geographical area,
3. counting the number of ships in the defined geographical area and time interval,
4. counting the number of ships located at a defined distance from an indicated point,
5. counting the number of ships located at a defined distance from an indicated point in a specified time interval (Pandley et al., 2018).

Results

The blue color in the diagram indicates values for the PostgreSQL database, while green represents MongoDB. The operation execution time is an arithmetic mean of five consecutive attempts of the same query.

Ad. 1. Records to the databases were added by reading files containing AIS data for the whole month. A single file contained data from one hour. All messages No. 1, 2, and 3 from each file were added to the database in one bulk insert. On average, 26 000 AIS messages were added in one query. The test of record adding is shown in Table 3.

Table 3. The test results of adding records to a database

		PostgreSQL	MongoDB
Time of addition [s]		22 353.26	5094.44
Average use of the CPU [%]		50.47	9.54
Volume of data on the disk [MB]	without index	4647	2655
	with index	12 131	4271

Results in Table 3 show that MongoDB executes the operations of adding records significantly faster than PostgreSQL, using less resources of the processor and the disk.

Ad. 2. The geographical area chosen for the test covered *Basen Górniczy* (Miners' Basin) in the port of Szczecin. Four test polygons were defined for the selected test area:

a) polygon P1 composed of 4 pairs of coordinates,
b) polygon P2 composed of 44 pairs of coordinates,
c) polygon P3 composed of 131 pairs of coordinates,
d) polygon P4 composed of 314 pairs of coordinates.

All areas are presented in Figure 3.

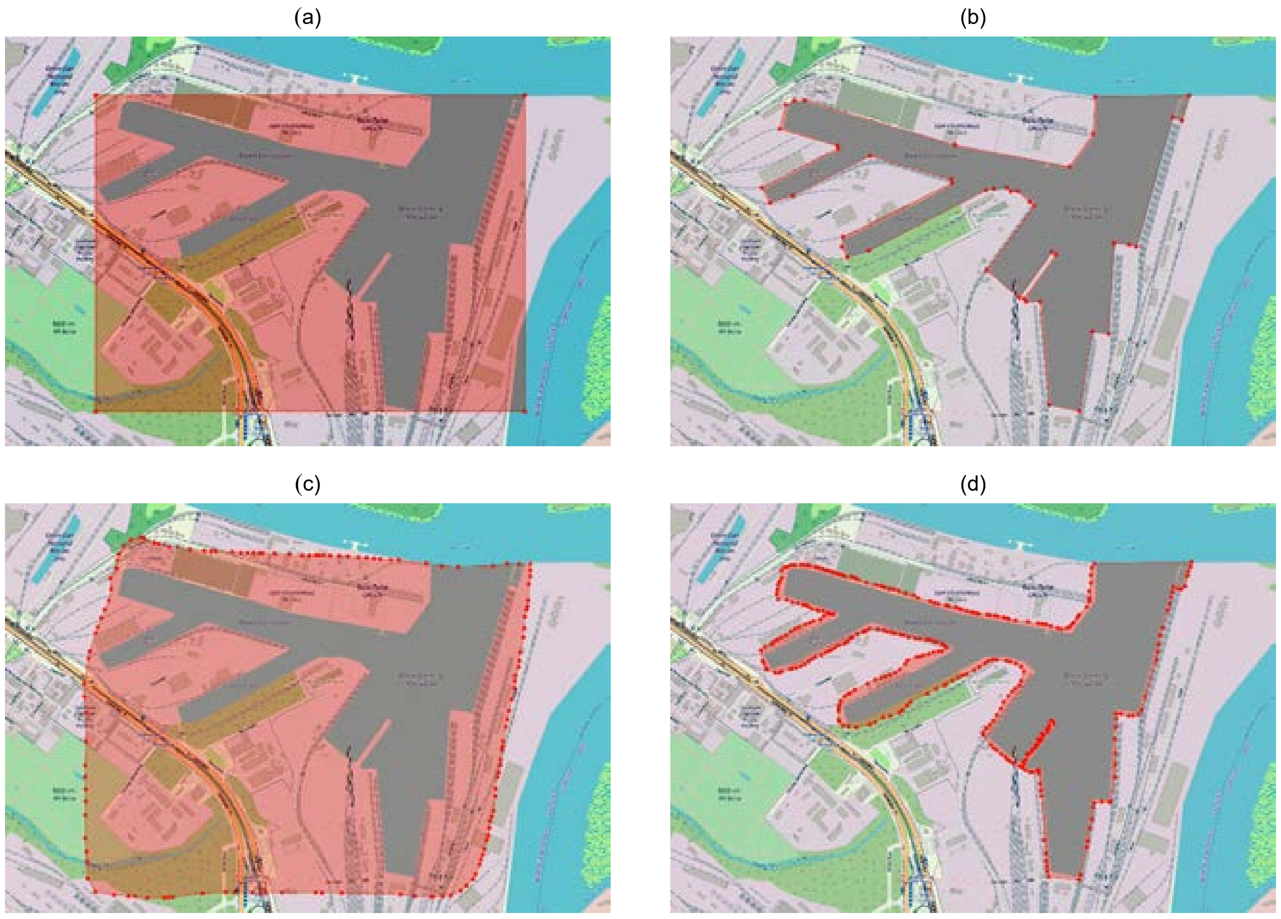

Figure 3. Test polygons (a) P1, (b) P2, (c) P3, and (d) P4 superimposed onto an OpenStreetMap

Each defined polygon covered a similar port area. Various numbers of vertices were used to determine differences in the execution of the queries to databases. A test query returns the amount of Maritime Mobile Service Identity (MMSI) numbers (number of ships) that were found in the defined area within the whole dataset. Test query execution times for each set are collected at Figure 4.

For simple shapes, roughly rectangular areas (a) and (c), the PostgreSQL database executes the queries faster than MongoDB. While queries addressing complex areas (b) and (d), which may reflect the profile of port basin shapes, are executed twice as fast with (b) and nearly four times as quick with (d), by MongoDB.

Ad. 3. This test query referred to the polygons defined in the previous point. The query results were additionally limited ⟨01.04.2019 00:00; 06.04.2019 00:00⟩ by the time interval. Test query execution times for each dataset are collected for Figure 5.

Based on the results of the tests, it can be stated that MongoDB is faster than PostgreSQL in executing queries for the unique amount of MMSI numbers found in a given area in a specified time interval. The difference in execution time is less noticeable for queries that refer to the rectangle shaped areas (P1 and P3). For queries made in complex areas (P2 and P4), MongoDB is almost four times faster.

Ad. 4. For this test, the selected reference point for the queries was at a distance of 500 m away, NNE of the EWA grain silo. The geographic coordinates of that point are $\varphi = 53°26.455'$ N and $\lambda = 014°35.346'$ E. The test query was to find all the unique MMSI numbers within a radius of 300 m from the set point. The practical value of such a query is that the surrounding area can be searched for ships that may pose a risk of collision. The area is shown at Figure 6.

The execution times of the test query by MongoDB and PostgreSQL are presented in Figure 7.

The query returning the number of ships, located within a 300 m radius, is executed in a similar time by both PostgreSQL and MongoDB. The query on a dataset, which includes AIS messages from 1 week (1W), is performed by PostgreSQL 0.099 s faster than MongoDB. The dataset of 1 month data (1M) is handled by PostgreSQL 0.169 s slower than MongoDB.

Ad. 5. Similar to test 4, the previously defined point and radius were used. The query results were

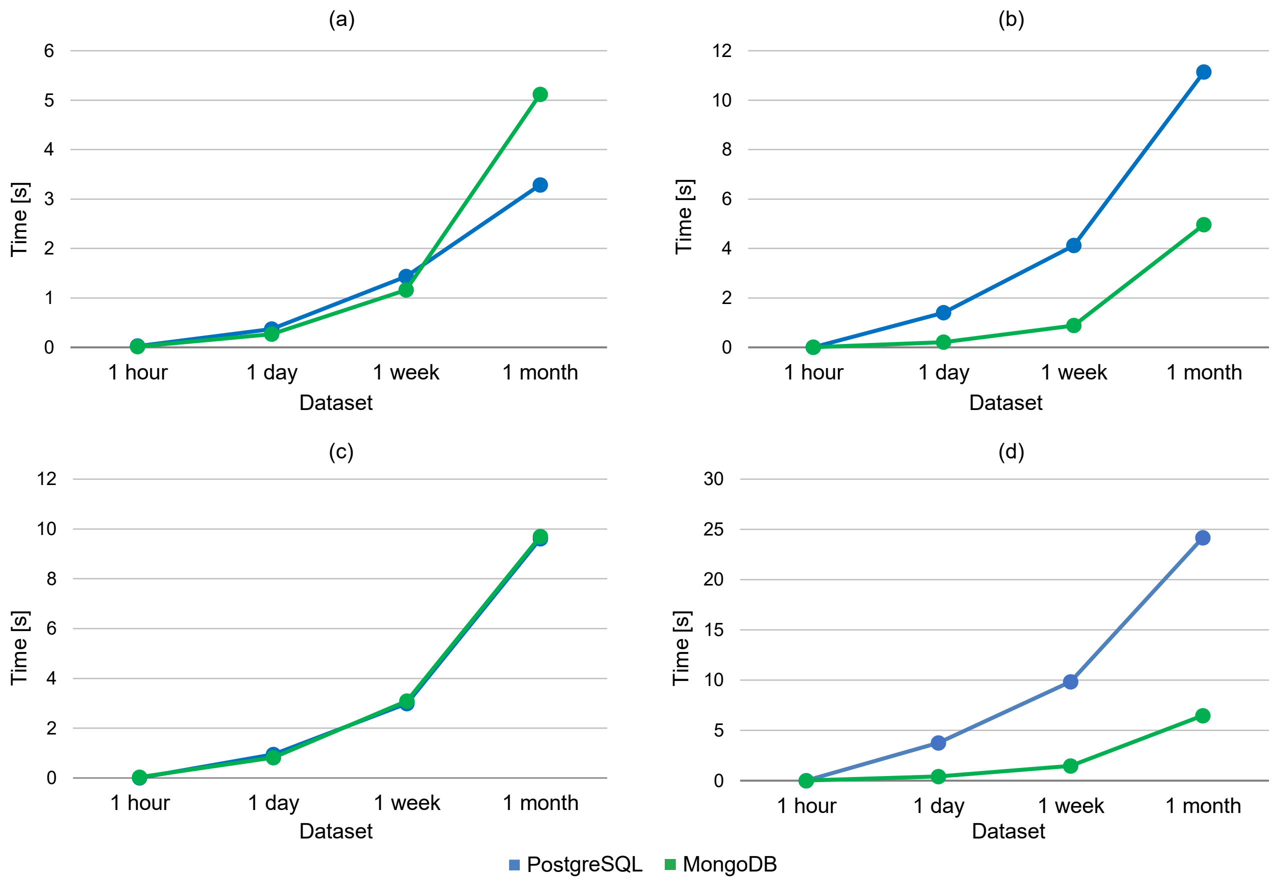

Figure 4. Execution times of the query, which counts the number of ships in the defined test polygons: (a) P1, (b) P2, (c) P3, and (d) P4

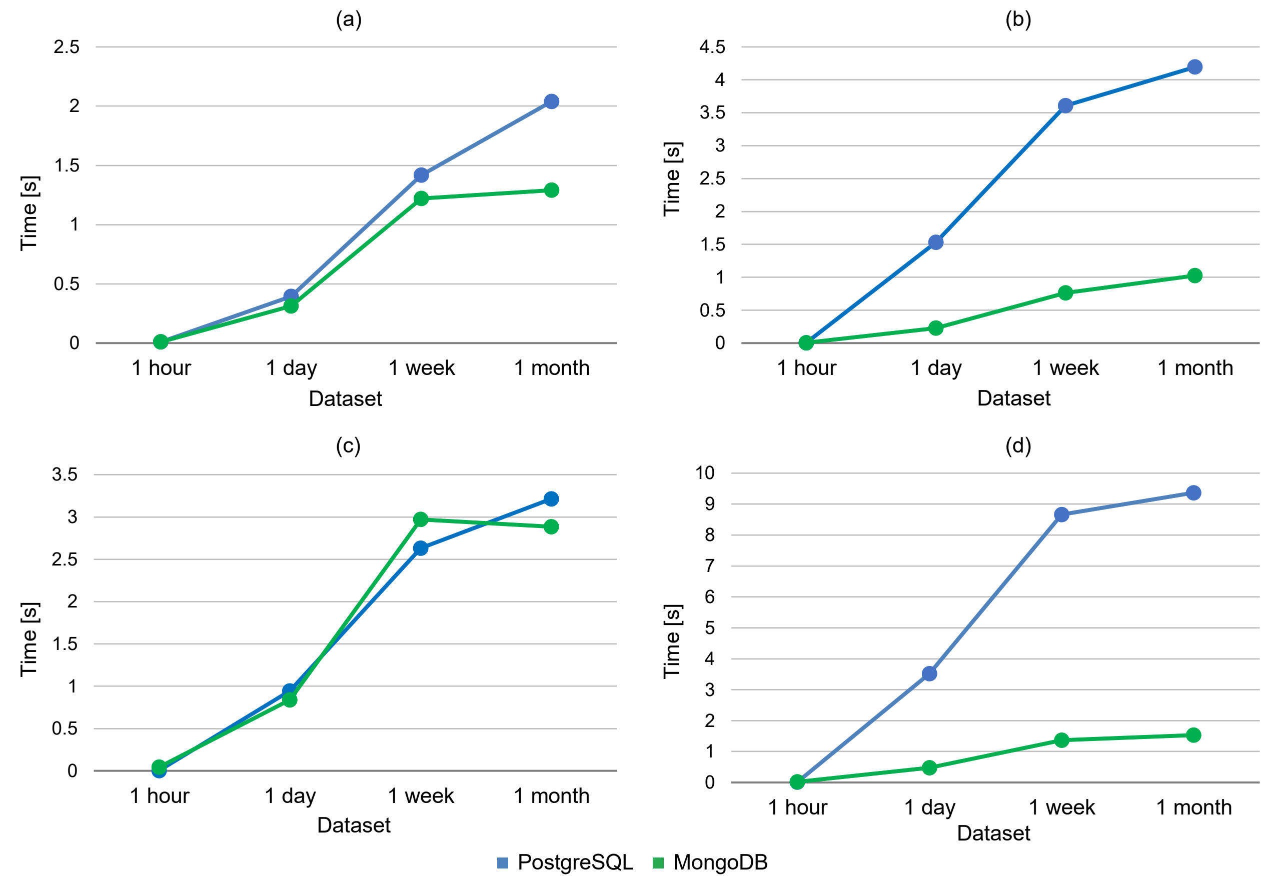

Figure 5. Execution times of the query, which counts the number of ships in the *Basen Górniczy* area from 1. to 6. April 2019 in the defined test polygons: (a) P1, (b) P2, (c) P3, and (d) P4

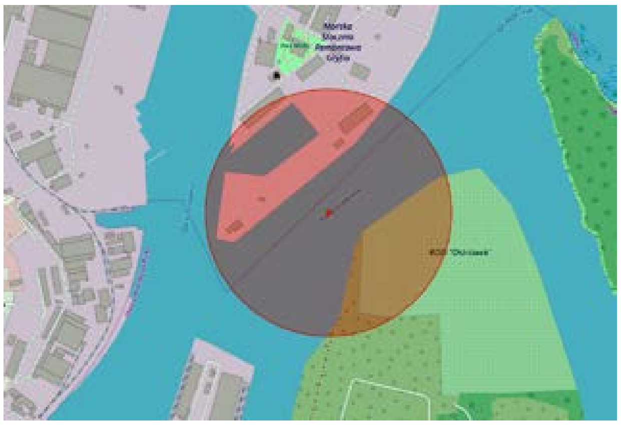

Figure 6. The selected reference point, centered in a circle that represents a 300 m radius, charted on an OpenStreetMap

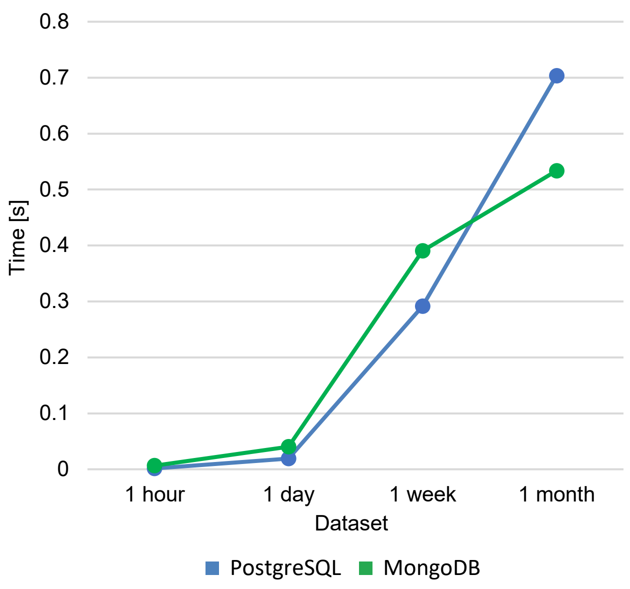

Figure 7. Execution time for counting the number of ships within a radius of 300 m from the set point

further limited by indicating the following time interval ⟨01.04.2019 00:00; 06.04.2019 00:00⟩. Test query execution times for each dataset are collected for Figure 8.

Compared to test 4, the additional criterion reduces the time differences between the performance of the test query by both databases. On a 1 week dataset (1W), PostgreSQL executes the query in 0.275 s, while MongoDB completes it in 0.387 s. In the case of 1 month dataset (1M), the difference is 0.079 s – the execution time by MongoDB was 0.421 s, while PostgreSQL was 0.500 s. Average times for the examined queries are presented in Table 4.

Table 4. Average execution times for the tested queries

	PostgreSQL	MongoDB
Average execution time: counting the number of ships in the defined test polygons [s]	4.57	2.17
Average execution time: counting the number of ships in the Basen Górniczy area from 1. to 6. April 2019 in the defined test polygons [s]	2.60	0.94
Average time of counting the number of ships within 300 m from the set point [s]	0.25	0.24
Time of counting the number of ships in a 300 m radius from the set point in the period 1–6 April 2019 [s]	0.20	0.21

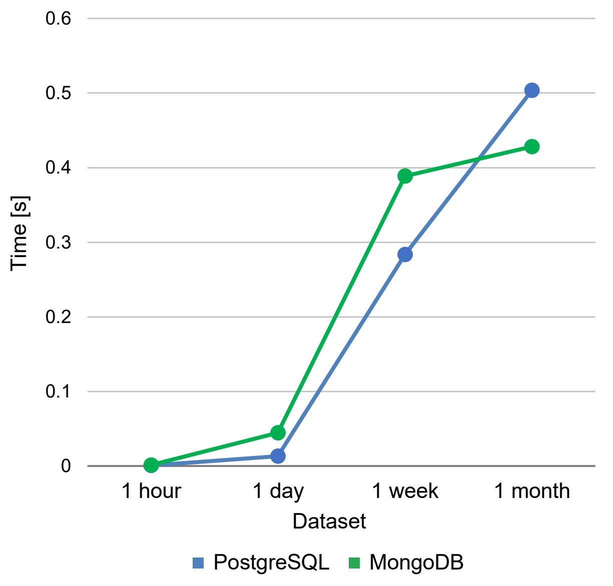

Figure 8. Execution time for counting the number of ships in a 300 m radius from the set point, over the period 1–6 April 2019

Conclusions

This article discusses the results of comparative tests for two databases, i.e. MongoDB and PostgreSQL, in which the storage and processing of position data from an AIS were examined. The tests and analyses of the results lead to the following conclusions:

- As AIS messages differ in their structure, storing the data in a MongoDB enables the use of one collection for all types of messages. In the case of PostgreSQL, it is necessary to predefine tables conforming to the structures of each message type;
- In April 2019, 22 128 345 AIS messages were received in the examined area (on average, 737 612 per day). It should be noted that, in an area with heavy vessel traffic, the number of received messages will be significantly larger; the data recorded in MongoDB occupy nearly three time less disk space than in PostgreSQL;
- MongoDB executes queries adding records 4 times faster than PostgreSQL, particularly when a large number of records are added in a short time;
- The defined indexes should match the queries; MongoDB responds to queries with indexes 11 times faster, PostgreSQL does it twice faster;
- Areas such as port basins, fairways, rivers, bays, and gulfs etc. have irregular shapes. Queries for objects searched for inside irregular shapes are executed much faster in MongoDB than PostgreSQL;
- Queries returning points located within a specified radius from the set point are performed by both databases in a similar time.

The use of MongoDB should be considered for storage and processing of large volumes of AIS data. Data in this database requires less space on the disk, and the queries take less time than in the case of a PostgreSQL database. This may lead to much faster analysis of a navigational situation, enabling earlier detection of dangerous situations, particularly in the areas of heavy traffic.

References

1. Baralis, E., Dalla Valle A., Garza, P., Rossi, C. & Scullina, F. (2017) *SQL versus NoSQL databases for geospatial applications*. Boston, MA, USA, 2017 IEEE International Conference on Big Data.
2. Garcia-Molina, H., Ullman, J.D. & Widom, J. (2006) *Systemy baz danych. Pełny Wykład.* Warszawa: Wydawnictwo Naukowo-Techniczne.
3. ITU (2014) *Recommendation ITU-R M.1371-5: Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band.* [Online] Available at: https://www.itu.int/rec/R-REC-M.1371-5-201402-I/en [Accessed: 15th June 2019].
4. Matthew, N. & Stones, R. (2005) Beginning Databases with PostgreSQL: From Novice to Professional, Berkley (US): Apress.
5. MongoDB Inc. (2019) *MongoDB Documentation*. [Online] Available at: https://www.mongodb.com [Accessed: June 13, 2019].

6. OGC (2019) Open Geospatial Consortium. [Online] Available at: http://www.opengeospatial.org [Accessed: 15th June 2019].
7. Pandley, V., Kipf, A., Neuman, T. & Kemper, A. (2018) How good are modern spatial analytics systems? *Proceedings of the VLDB Endowment* 11, pp. 1661–1673.
8. Plugge, E., Membrey, P. & Hawkins, T. (2010) *The Definitive Guide to MongoDB: The noSQL Database for Cloud and Desktop Computing*. Berlin: Springer-Verlag.
9. PostGIS (2019) *PostGIS Documentation*. [Online] Available at: https://postgis.net [Accessed June 2019].
10. Varlamis, I., Tserpes, K. & Sardianos, C. (2018) *Detecting search and rescue missions from AIS Data*. 2018 IEEE 34th International Conference on Data Engineering Workshops (ICDEW), pp. 60–65.
11. Xu, T., Hu, Q., Xiang, Z., Yang, C. & Wang, D. (2016) The comparison study on AIS signal reception rate with directional antenna and omni antenna. *TransNav The International Journal on Marine Navigation and Safety of Sea Transportation* 10, pp. 205–211.

Cite as: Czapliński, W., Gąsowski, W. (2022) Performance of AIS geoinformation extraction using SQL and NoSQL TranStat databases. *Scientific Journals of the Maritime University of Szczecin, Zeszyty Naukowe Akademii Morskiej w Szczecinie* 71 (143), 93–101.