Archiving AIS messages in a Geo-DBMS

Martijn Meijers, Wilko Quak & Peter van Oosterom
Delft University of Technology
b.m.meijers|c.w.quak|p.j.m.vanoosterom@tudelft.nl

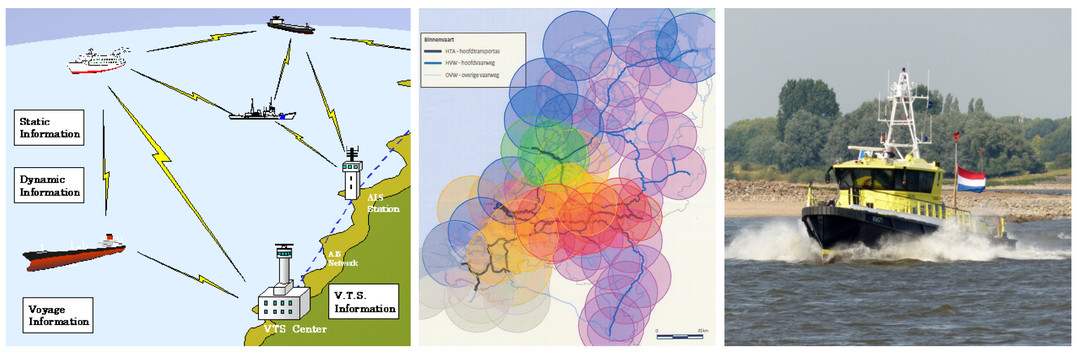
Abstract

Rijkswaterstaat has developed DIAMONIS, a network for AIS message reception. Vessels broadcast – amongst other information – their position and identity in intervals ranging from 2 seconds to 3 minutes. This leads to a large volume of spatio-temporal data. The current system architecture is not suited for archiving and analysing this large volume of historic AIS messages. This poster shows the result of 2 studies [1, 2] into efficient storage of these historic AIS messages using a Geo-DBMS (MongoDB and PostgreSQL).

Introduction

Within Rijkswaterstaat (RWS) AIS (Automated Identification System) messages are received in real time with the DIAMONIS (Dutch Inland AIS monitoring Infrastructure) network. The current architecture of this system is not suited for archiving large amounts of historic AIS messages. The aim of this research is to investigate how the messages can be stored, so that they can be queried efficiently [1, 2].

Figure 1: The DIAMONIS network receives AIS messages transmitted by vessels

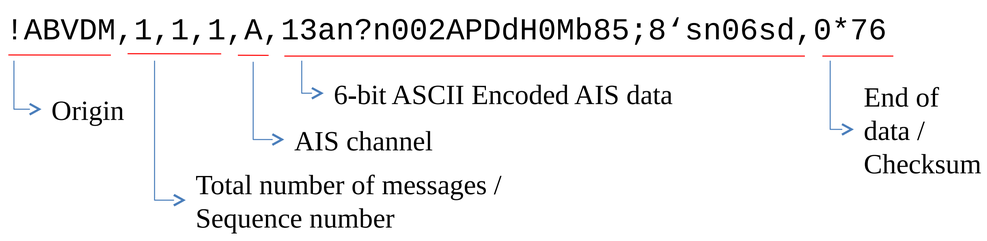


Sources: Kanmon Kaikyo Traffic Advisory Servic Center · Wikipedia · Vessel Finder (Roeland.J)

Main Objectives

1. Data volume on par with file based solution
2. Queries execute reasonably efficient for variety of use cases
3. All data stored for future use, but 4 message parameters directly available: timestamp, message type (more than 20 message types, hence efficient subselection needed), vessel identity and position

Figure 2: AIS message as received by DIAMONIS (see [3] for decoding)



Data and Methods

As AIS messages are received frequently for many vessels, the total data volume is significant. Per week more than 80 million messages are received by DIAMONIS (leading to over 1.5GB of raw message data per week). A limited dataset with 523,477 AIS messages received for 30 different RWS vessels for 2 days in November 2015 was loaded into both MongoDB and PostgreSQL. After an initial test for analysing storage requirements, spatial indexing and query capabilities were investigated.

Results

Storage requirements

Test data was loaded into both MongoDB, as well as in PostgreSQL to find how both systems would perform with respect to storage size.

Table 1: Data types with their resulting table size

Solution	Data type	Size (MB)	Factor to File
File	Raw ASCII message + timestamp	27	—
MongoDB	JSON	58	2.1×
PostgreSQL	JSON	213	7.9×
PostgreSQL	JSONB	273	10.1×
PostgreSQL	varchar	35	1.3×
PostgreSQL	bit vector	35	1.3×

Indexing, which is necessary to guarantee efficient access to individual records, also requires additional storage space. In PostgreSQL, we defined database functions to access the parameters of the AIS messages, stored as bit vector in the DBMS. Using these functions, we created 4 functional indexes on the table.

Table 2: Functional indexes and their size

Index	Type	Column/Function	Size (MB)	Share (%)
Vessel ID	B-Tree	`ais_mmsi(payload)`	11	19
Message type	B-Tree	`ais_type(payload)`	11	19
Timestamp	B-Tree	`timestamp`	11	19
Geometry	R-Tree	`ais_point(payload)`	25	43

The total amount needed for storing the AIS messages in PostgreSQL based on the bit vector data type, including functional indexes, is a factor 4 larger compared to file based storage.

Table 3: Total storage size for PostgreSQL

Solution	What	Size (MB)	Factor to File
PostgreSQL	Table with bit vector	35	
	Indexes	58	
		93	3.5×

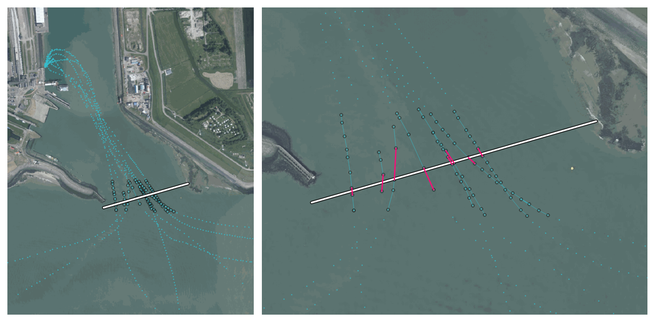
Queries

For a variety of use cases 2 queries are a starting point:

1. Location query: Find the last known position of a vessel (at a specific time).
2. Trajectory query: Give the historic positions of a vessel, i. e. points ordered by timestamp and subsequently connected to straight line segments.

For example, using these queries an analysis can be carried out how many vessels cross a line.

Figure 3: Analyse how many vessels cross a line



Discussion

- Both MongoDB and PostgreSQL can offer compact storage for AIS messages
- Structure developed in PostgreSQL based on bit vector type provides history of all message content, while providing *efficient* access to often used parameters
- This takes a factor 4 more storage space than raw text file storage
- Data structure allows advanced spatial query capabilities (due to PostGIS)

Forthcoming Research

We bulkloaded AIS data that was already collected. However, archiving data in real-time will be different: Re-indexing / re-clustering operations will be needed. A possibility is to structure the data in 2 tables: An (unstructured) heap and an indexed and clustered historic archive table.
How often to perform re-organization of the heap table into the historic archive?

References

[1] Irene de Vreede. Managing historic Automatic Identification System data by using a proper Database Management System structure. Master’s thesis, TU Delft, november 2016.

[2] Martijn Meijers, Peter van Oosterom, and Wilko Quak. Management of AIS messages in a Geo-DBMS. Technical report, Delft University of Technology, 2016.

[3] Eric S. Raymond. AIVDM/AIVDO protocol decoding, 2016.

Acknowledgements

This work has been carried out in the framework of the ‘Rijkswaterstaat-TU Delft Raamovereenkomst betreffende Samenwerking en Kennisuitwisseling op gebied van Ruimtelijke Informatievoorziening’ (Reference 31103836).