Fast and Efficient Cluster Based Map for Ship Tracking

Andi M Ali Mahdi Akbar
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mahdi11@mhs.ee.its.ac.id

I Ketut Eddy Purnama
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ketut@ee.its.ac.id

Supeno Mardi Susiki Nugroho
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mardi@te.its.ac.id

Mochamad Hariadi
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mochar@te.its.ac.id

***Abstract*—Automatic Identification System (AIS) is a system designed to improve maritime security by enabling the ship navigator to view the identity, position, and direction of other ships nearby. AIS data can be used to monitor ship activities. AIS data send simultaneously from multiple ship at very fast intervals. With the increased amount of data received, the performance of data retrieval using traditional RDBMS often decreased significantly. Also, storing geospatial data in traditional RDBMS or NoSQL lead to persistence of data, which is not required when tracking ship position, as majority of ship always moving. The increased amount of ship shown in monitoring map make it less informative, hence affecting users who use monitoring system. This paper proposes a fast cluster-based method to store and query AIS geospatial information using Redis and also maps it as a cluster into a web-based map to provide a faster and more efficient display.**

***Index Terms*—AIS, Automatic Identification System, Ship Tracking, Server side cluster, Geohash, Redis.**

I. Introduction

Automatic Identification System (AIS) is used to enhance safety and efficiency of navigation, safety of life at sea, and maritime environmetal protection [1]. AIS already implemented on ships that are at least 300 GT (Gross Tonnage) for all ships that travel internationally and 500 GT for doing national shipping [2] [3].

AIS information can be categorize into two types of data, static information and dynamic information. Static information are information about ship that rarely changes such as ship type, ship size, ship name, call sign, and maritime mobile service identity (MMSI). Meanwhile the dynamic information are data that dynamically change everytime, such as ship position, vessel direction, speed, and time. AIS geospatial information can be stored and processed to monitor position of the ship so it can be used to avoid collision, track the ship;s location, predict their track, to observe their movement, etc. providing very usefull information. [4] [5].

AIS geospatial data sent simultaneously from multiple ship at very fast intervals. With the increase of the amount of data received, the performance of retrieval data using traditional RDBMS often decreased significantly. PostGIS [6], Oracle Spatial [7], etc. are a few of geospatial databases that are capable of storing, reading, and indexing geospatial data. There are also NoSQL [8], an alternative storage which is not based on relational models become a solution to the problem the the relational databases have when storing massive data.

Storing geospatial data in traditional RDBMS or NoSQL lead to persistence of data, which is not required when tracking ship position, as majority of ship always moving. Redis [9] [10] as memory based cache provides a better and efficient system to store a dynamic data like geospatial since the data is stored in memory so the system do not have the overhead of disk write.

The approach described in this paper shows a highly efficient method for ship tracking. This paper mainly discusses about how to store and retrieve ship position, and also efficiently plotting it on a digital map.

II. Literature Review

Real-time ship monitoring is used to see the condition of the ship such as position, speed, direction, etc. Because of the many benefits of ship monitoring for marine security, it is necessary to have a reliable ship activity monitoring system [11]. However, there are still few of capable system that can be used to monitor ship's activities.

The latest position of ship is plotted on a web map using a marker. If the number of ship in the current viewing area is very large, then more and more markers will be displayed on the map (Figure 6). On the other hand, the burden on the server due to the large amount of data processed for every request will increase in number of users. Moreover, when the marker displayed exceeds the capability of user device, it will make the user's side crash or fail [12].

Usually, the user who monitor the ship's activities, works on a smaller areas rather than a large areas. Limiting the scope of processed data into user current view shall reduce the amount of data processed by the system, thus increasing the overall performance [13].

III. Research Method

We use Redis as databases to store and query AIS geospatial information. Redis is an open source in-memory

978-1-5386-7509-0/18/$31.00 ©2018 IEEE

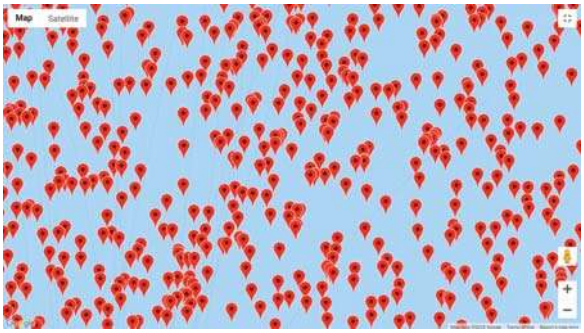

Fig 1. Map with crowded marker

data structure store used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets, and also support geospatial indexes with radius queries. The idea behind using Redis as our storage is not only because Redis is a memory based cache, but also the capability of Redis to store and query spatial information in form of geospatial indexes.

The ship tracking is split into three major processes, storing process, query process, and plotting the ship position into the map. There are two kind of storing process, the first one is to store ship's position and the other is to store the cluster points into databases. To plot the ship into the web map, The cluster point query is performed to retrieve all cluster points within current view. Next, the ship position query is performed to retrieve all ship within radius of cluster point. The query process shown in figure 2.

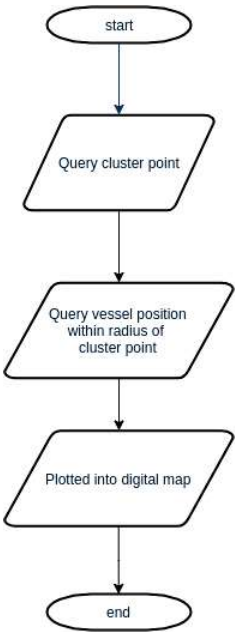

Fig 2. Query and plotting flow diagram

Next subsection will describe how to store and query AIS geospatial information in detail way.

A. Storing ship position

Ship geospatial information or ship's position in form of longitude and latitude were stored using Redis GEOADD. Redis GEOADD is used to store spatial information in form of geohash. Latitude and longitude are encoded into geohash so it can be indexes as geo-index. Geohash [14] is a geocoding algorithm that converts spatial coordinates into a single string. Instead of using string, geohash is converted into a unique 52 bit integer and GEOADD use it as score in sorted set. Redis sorted set score can represent a 52 bit integer without losing precision.

We use continent as a key to group ship's information based on its continent. In example, when ship in Indonesia sent its position, any AIS receiver who receives it will store the data in ASIA region. The grouping can be made smaller (ie. country or nation), but for the sake of simplicity, the continent will be used as the largest grouping.

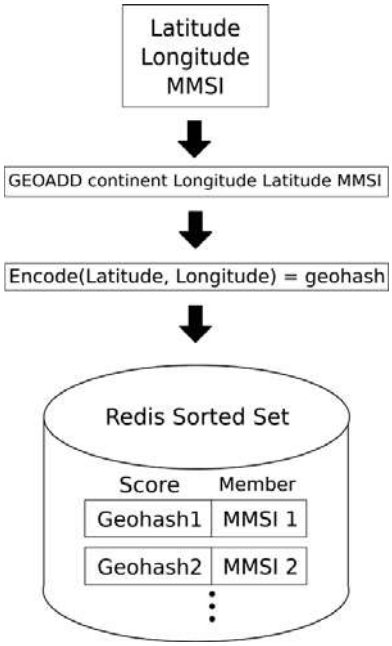

Fig 3. Process of storing ship position into storage

Redis command:

GEOADD *continent longitude latitude mmsi*

B. Storing cluster point

Cluster point is a center coordinate of clustering point. This cluster point is stored into sorted set, similar to ship position. Instead of using GEOADD, we use ZADD to insert cluster point into sorted set. ZADD is a Redis command to store value into sorted set with score. This score is used as indexing, in this case, we use Geohash as indexing. First, we need to encode the coordinate into 52 bit-geohash.

$$geohash = geohash\_encode(longitude, latitude) \quad (1)$$

The latitude and longitude are converted into geohash value using (1). Geohash is set as score, and cluster point id is set as a member to sorted set. If a specified member is already a member of the sorted set, the score is updated and the element reinserted at the right position to ensure the correct ordering. This point will later be use to determine how much cluster point in current viewing map.

ZADD is used to store position of cluster point. "CLUSTERPOINT" is use as key to sorted set, geohash as score to represent cluster point coordinate, and cluster id as name of the member.

Redis command:

ZADD *CLUSTERPOINT geohash cluster_id*

C. Cluster point query

To perform ship position query, first we need to retrieve all cluster point within current view.

Fig 4. Boundary of map view

Figure 4 shows boundary of current viewing map. To return any cluster point within the current view, ZRANGEBYSCORE is used to return any cluster point in sorted set. ZRANGEBYSCORE is Redis command that retrieves all member in the sorted set having score within range. WITHSCORE option is used to return the score of each member in queried range. The geohash value of north-east corner and south-west corner calculated using (1). We can return all member within current view by using ZRANGEBYSCORE with geohash value of south-west corner and geohash value of north-east corner as input.

$$\forall e \in SortedSet = \begin{cases} e_{score} \geq GeohashSW \\ e_{score} \leq GeohashNE \end{cases} \quad (2)$$

ZRANGEBYSCORE will return all member with score greater than geohash value of south-west and less than geohash value of north-east as shown in (2).

Redis command:

ZRANGEBYSCORE *CLUSTERPOINT* $Geohash_{SW}$ $Geohash_{NE}$ *WITHSCORE*

D. Ship Position query

GEORADIUS is used to query ship position. GEORADIUS is Redis command to return the members of sorted set which is populated with geospatial information using GEOADD. We use GEORADIUS to return any members within radius of input coordinate in sorted set.

$$\{lon, lat\} = geohash\_decode(geohash) \quad (3)$$

Cluster point score, which is geohash, is decoded using (3) into longitude and latitude. Cluster point serves as center of radius. Cluster point coordinate (Longitude and latitude) is then used as input to retrieve any ship within radius of cluster point.

Continent is used as key, cluster point longitude and latitude as center of radius, radius in kilometer, and WITHCOORD parameter to return coordinate from any returned member.

Redis command:

GEORADIUS *Continent longitude latitude radius km WITHCOORD*

E. Plotting clustered ship into map

The ship queried from storage are plotted in a web map using marker. Cluster point query is performed to return any cluster point within current viewing map. Using cluster point we got from previous query, ship position query is performed to retrieve all ship within radius of cluster point. The response from query is then plotted into web map with marker for each data. Figure 5 shows web map without clustering system, and figure 6 shows web map using clustering system.

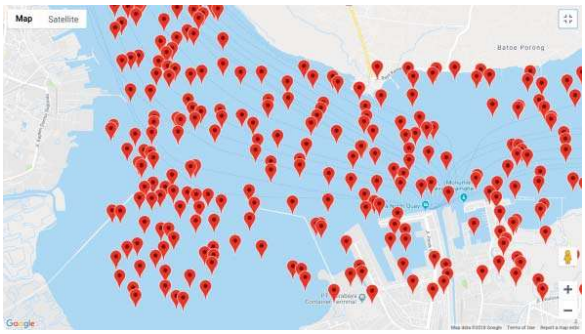

Fig 5. Unclustered ship plotted into web map

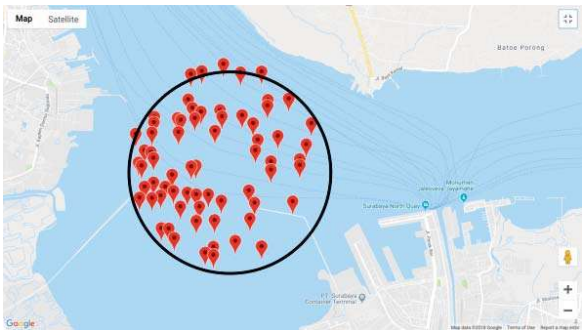

Fig 6. Clustered ship plotted into web map

Next, marker within cluster point is clustered into one big cluster marker. We are using Google maps cluster marker [15] to cluster all ship marker. The number on a cluster marker shows how many markers it contains. We can drill down the cluster marker to zoom in into next level, revealing the next smaller cluster marker with number on the cluster decrease. This can be repeated until the individual ship is shown in the map. Zooming out of the map consolidates the markers into clusters again.

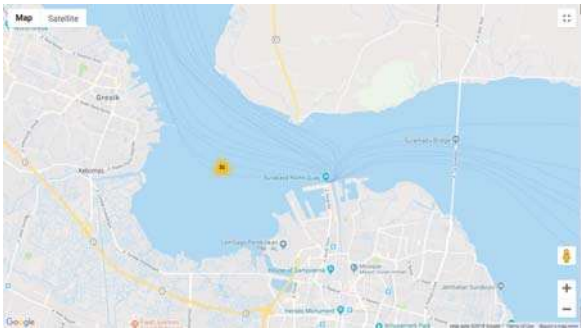

Fig 7. Web map with clustered marker

IV. Experiment And Results

We perform the experiment on Redis version 4.0.7 which run on 64-bit Ubuntu server with Intel Xeon, 8 GB memory, and 100 GB disk. The experimental data set was collected from AIS receiver of Intelligent Maritime Transportation System since 2016. The experiment was conducted to prove the concept and show the performance of the proposed method.

We use region of indonesia as our test location. When we conduct our experiment, there are 3372 ship around indonesia recorded in our storage. The map with unclustered query shown in figure 8.

Fig 8. Map with unclustered ship

To prove the concept, we use port as cluster point. We use two large ports in indonesia, Tanjung Priok Port in Jakarta (latitude -6.092115, longitude 106.8816) and Tanjung Perak Port in Surabaya (latitude -7.196120, longitude 112.733468). Both of these ports coordinate are stored into database as cluster point using storing cluster point method. Using (1), we get both port geohashes. Tanjung Priok Port has geohash value 3196628483040700 and Tanjung Perak Port has geohash value 3220562048275474.

Our current view is on indonesia with assumption for North-East corner coordinate is at latitude 6.172201 and longitude 139.594200, and South-West corner coordinate is at latitude -8.517442 and longitude 94.724562. By using (1), we got geohashNE with value 4086502524091493 and geohashSW with value 3185248313893645.

Performing cluster point query by using geohashNE and geohashSW as input, both Tanjung Priok Port and Tanjung Perak Port will be returned as response because both port has geohash value between geohashNE and geohashSW as explained in equation (2).

Using (3), we decoded both ports geohashes back into latitude and longitude. The ship position query is performed for each cluster point. The response from the ship position query is then plotted onto the web map with marker as shown in the figure 9.

Fig 9. Map with clustered ship

After the marker is plotted into the map, all the marker is clustered to make it more efficient and informative, shown in figure 10. The marker is clustered based on how close they are from each other. The clustered marker also shows how many ships are located in those area, providing a quick information about the number of ship in those cluster point.

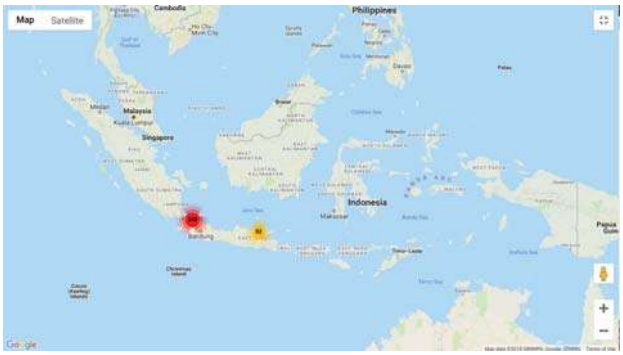

Fig 10. Map with clustered marker

V. Conclusion

This paper shows how to store, process, and retrieve ship position sent by AIS using Redis. We also showed how to plot it into web based map. The server side cluster is done at the time the ship's position query is performed. The cluster point is required before performing the ship position query. The cluster point is used to determine the center point of clustered ship. Then the result of ship position query is plotted into web based map using marker. The marker is also clustered with its nearest marker into one big marker. This provides an informative view to convey the number of ship for each cluster point. The user can drill down the clustered marker to zoom into next level, revealing the next smaller cluster marker. This can be repeated until the individual ship is shown in the map. Because Redis is a memory based cache, the data requested by the first user is kept in memory, so writing and reading activity shall be faster. The increasing number of concurrent user should not affect the performance of

system. And also, because of clustered marker, the user who viewed the map should not experience slow down because of crowded marker.

ACKNOWLEDGMENT

The authors would like to thank IPTEK DIKTI, B201 Telematic Laboratory for research funding and support.

REFERENCES

[1] Tetreault, B.J., 2005, September.*Use of the Automatic Identification System (AIS) for maritime domain awareness (MDA)*. In OCEANS, 2005. Proceedings of MTS/IEEE (pp. 1590-1594). IEEE.
[2] http://www.yachtinsure.com/news-auto-identification-systems.htm, Automatic Identification System (AIS):Complete Guide to AIS
[3] International Maritime Organization (IMO), Annex 3, Recommendation On Performance Standards For An Universal Shipborne Automatic Identification Systems (AIS),IMO Resolution MSC.74(69).
[4] Deng, F., Guo, S., Deng, Y., Chu, H., Zhu, Q. and Sun, F., 2014, September.*Vessel track information mining using AIS data. In Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014 International Conference on (pp. 1-6). IEEE.
[5] Tichavska, M., Cabrera, F., Tovar, B. and Araa, V., 2015, February.*Use of the Automatic Identification System in academic research*. In International Conference on Computer Aided Systems Theory (pp. 33-40). Springer, Cham.
[6] Zhang, L. and Yi, J., 2010, August. Management methods of spatial data based on PostGIS. In Circuits, Communications and System (PACCS), 2010 Second Pacific-Asia Conference on (Vol. 1, pp. 410-413). IEEE.
[7] Barman, S., Barman, A.K., Panda, B. and Sharma, P., 2015, February. Implementation of a smart map using spatial oracle. In Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on (pp. 1-5). IEEE.
[8] Zafar, R., Yafi, E., Zuhairi, M.F. and Dao, H., 2016, May. Big Data: The NoSQL and RDBMS review. In Information and Communication Technology (ICICTM), International Conference on (pp. 120-126). IEEE.
[9] Hao Yu, Yuehu Liu, Chuan Tian, Liang Liu, Mingchao Liu and Yong Gao, "A cache framework for geographical feature store," 2012 20th International Conference on Geoinformatics, Hong Kong, 2012, pp. 1-4.
[10] Carlson, J.L., 2013. Redis in action. Manning Publications Co..
[11] De Vreede, I., 2016. *Managing historic Automatic Identification System data by using a proper Database Management System structure*.
[12] S. Meier and F. Heidmann, "Too Many Markers, Revisited: An Empirical Analysis of Web-Based Methods for Overcoming the Problem of Too Many Markers in Zoomable Mapping Applications," 2014 14th International Conference on Computational Science and Its Applications, Guimaraes, 2014, pp. 121-125.
[13] Thomas, G., Alexander, G. and Sasi, P.M., 2017, July. *Design of high performance cluster based map for vehicle tracking of public transport vehicles in smart city*. In IEEE Region 10 Symposium (TENSYMP), 2017 (pp. 1-5). IEEE.
[14] Niemeyer, G., 2008. Geohash.
[15] Mahe, L., & Broadfoot, C. (2010, December). Too Many Markers! - Google Maps API Google Developers. Retrieved March 2, 2014, from https://developers.google.com/maps/articles/toomanymarkers