Vessel Trajectories Application using MongoDB as Storage Layer

Εμμανουήλ Διλμπέρης
Τμήμα Ψηφιακών Συστημάτων Πανεπιστήμιο Πειραιώς
Πειραιάς, Ελλάδα
me2104@unipi.gr

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας εργασίας είναι η δημιουργία εφαρμογής, η οποία θα είναι ικανή να απαντά με ακρίβεια χωρικά και χωροχρονικά ερωτήματα που αφορούν τροχιές σκαφών. Η λήψη των δεδομένων που χρησιμοποιήθηκαν έγινε από το αποθετήριο Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance and Reconnaissance και περιέχουν πληροφορίες πλοίων, οι οποίες συλλέγονται μέσω του Συστήματος Αυτόματης Αναγνώρισης (AIS). Η αποθήκευση τους έγινε στη μη σχεσιακή βάση δεδομένων MongoDB. Ύστερα υλοποιήθηκε η τελική μοντελοποίηση και ευρετηρίαση τοπικά. Στη συνέχεια δημιουργήθηκε ένα cluster, όπου έγινε εισαγωγή των δεδομένων και ο διαμοιρασμός τους σε 2 shards. Τέλος αναπτύχθηκε εφαρμογή στη γλώσσα προγραμματισμού Python η οποία επεξεργάζεται τα δεδομένα που ανακτά από την βάση δεδομένων με σκοπό την οπτικοποίηση τους σε ένα γεωγραφικό χάρτη.

KEYWORDS

Python, MongoDB, NoSQL, spatio-temporal data, trajectories

1 ΕΙΣΑΓΩΓΗ

Η γρήγορη ανάπτυξη τεχνολογιών ασύρματης επικοινωνίας σε συνδυασμό με την εξέλιξη των τεχνολογιών που επιτρέπουν την αποθήκευση και την επεξεργασία μεγάλων όγκων δεδομένων, έχουν συμβάλει στην σημαντική ανάπτυξη εφαρμογών που ασχολούνται με δεδομένα τροχιάς. Τα δεδομένα τροχιάς καταγράφουν τη θέση ενός αντικειμένου στο χώρο σε μια συγκεκριμένη χρονική στιγμή.

Τα αντικείμενα που περιγράφονται από τροχιές είναι συνήθως κινούμενα, καθώς η χωρική τους θέση μεταβάλλεται με τον χρόνο και αυτές οι αλλαγές συχνά είναι συνεχείς στο χρόνο. Όμως, για να αποθηκευτούν σε ένα σύστημα βάσης δεδομένων, αναπαρίστανται ως διακριτές τοποθεσίες.

Στον τομέα της ναυτιλίας συγκεκριμένα παράγεται τεράστιος όγκος ετερογενών δεδομένων και η ανάλυση αυτών είναι εξαιρετικά σημαντική καθώς επιτρέπει στις επιχειρήσεις να εξάγει χρήσιμες πληροφορίες. Τα μεγάλα δεδομένα (Big Data) χρησιμοποιούνται για τη διαχείριση των αισθητήρων των πλοίων και για προγνωστικές αναλύσεις, οι οποίες είναι απαραίτητες για την αποφυγή καθυστερήσεων και τη βελτίωση της συνολικής λειτουργικής αποτελεσματικότητας του κλάδου. Επίσης αντλούνται πληροφορίες για την ενεργειακή απόδοση, την κατανάλωση καυσίμων και τις επιδόσεις των πλοίων, βοηθώντας έτσι και θέματα που έχουν να κάνουν με την οικολογία και το περιβάλλον. Στη ναυτιλιακή βιομηχανία, η σωστή παρακολούθηση του φορτίου είναι ουσιαστική, για τη διασφάλιση της απαραίτητης ασφάλειας και εμπιστευτικότητας.

Για τους παραπάνω λόγους, μια μη σχεσιακή βάση(NOSQL) είναι ιδανική για την αποθήκευση τέτοιων δεδομένων, αφού μπορεί να διαχειριστεί καλά μεγάλες ποσότητες, πολύπλοκων δεδομένων διαφορετικού τύπου.

2 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Σχετική με την παρούσα εργασία είναι η διπλωματική διατριβή [2] η οποία επικεντρώνεται στην επεξεργασία χωροχρονικών ερωτημάτων στη μη σχεσιακή βαση δεδομένων MongoDB και χρήση NoSQL τεχνικών. Κάποια από αυτά τα ερωτήματα είναι τα Circle Range και Box Range που αφορούν την εύρεση δεδομένων μέσα σε ένα κύκλο ή ορθογώνιο αντίστοιχα για ένα συγκεκριμένο χρονικό διάστημα.

Επιπλέον μελετήθηκε το επιστημονικό άρθρο του Ernest Batty, με τίτλο “Data Analytics Enables Advanced AIS Applications” [10], όπου πραγματοποιείται αναλυτική περιγραφή της βοήθειας της ανάλυσης δεδομένων για τη δημιουργία προηγμένων εφαρμογών. Συγκεκριμένα τονίζεται η μεγάλη ανάγκη, τα δεδομένα που συλλέγονται, επεξεργάζονται και αποθηκεύονται από το AIS να είναι καθαρά και πλήρως ενημερωμένα ως προς την αλλαγή στατικών δεδομένων, όπως είναι το όνομα ή άλλα χαρακτηριστικά του πλοίου.

3 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Τα σύνολο δεδομένων που ήταν διαθέσιμο για την παρούσα εργασία περιέχει πληροφορίες για 5055 πλοία, που συλλέγονται μέσω του AIS (Automatic Identification System). Τα δεδομένα χωρίζονται σε τέσσερις κατηγορίες, την πλοήγηση των πλοίων (navigation data), τις πληροφορίες των πλοίων (vessel-oriented data), για παράδειγμα όνομα και τύπος πλοίου, τα γεωγραφικά δεδομένα (geographic data) και τα περιβαλλοντικά δεδομένα (environmental data). Αφορούν το χρονικό διάστημα μεταξύ 01/10/2015 και 31/3/2016 στην περιοχή της Κέλτικης θάλασσας και του Βισκαϊκού κόλπου.

Από αυτό το σύνολο δεδομένων χρησιμοποιήθηκαν από την κατηγορία navigational data το αρχείο nari_dynamic, το οποίο περιέχει το sourcemmsi που είναι μια σειρά από εννιά ψηφία που αποστέλλονται σε ψηφιακή μορφή μέσω ενός καναλιού ραδιοσυχνοτήτων προκειμένου να προσδιορίσουν μοναδικά σταθμούς πλοίων, μια συγκεκριμένη χρονική στιγμή, το γεωγραφικό πλάτος και μήκος που βρίσκεται εκείνη τη στιγμή και την κατάσταση πλοήγησης. Επίσης τα αρχεία nari_static και anfr που περιέχουν πληροφορίες όπως το όνομα του πλοίου, τον αριθμό του τύπου κ.α. και τα αρχεία Navigational Status, MMSI Country Code, Ship Type List για την αντιστοίχιση του κωδικού της χώρας, του τύπου του πλοίου και της κατάστασης πλοήγησης που περιέχεται στα τρία πρώτα αρχεία με την πλήρη ονομασία.

4 ΟΡΙΣΜΟΣ ΕΡΩΤΗΜΑΤΩΝ

Τα ερωτήματα που θα μπορούν να απαντηθούν με την υλοποίηση της παρούσας εργασίας είναι σχεσιακά, χωρικά, χωροχρονικά και ερωτήματα βασιζόμενα σε τροχιές.

Τα σχεσιακά ερωτήματα αφορούν εύρεση τροχιών πλοίων με βάση συγκεκριμένα χαρακτηριστικά, για παράδειγμα τα ονόματα των πλοίων, τον τύπο τους, τη σημαία τους και την κωδικό sourcemmsi.

Χωρικά ερωτήματα τα οποία μπορούν να χωριστούν σε ερωτήματα εύρους και k κοντινότερων γειτόνων. Τα ερωτήματα χωρικού εύρους αναζητούν συγκεκριμένα χωρικά αντικείμενα που σχετίζονται με άλλα χωρικά αντικείμενα σε μια συγκεκριμένη απόσταση [11]. Ο παραπάνω υπολογισμός γίνεται εφικτός, ορίζοντας στη σφαίρα ένα κύκλο, δοθέντος το γεωγραφικό πλάτος και μήκος ενός σημείου και μιας ακτίνας ορισμένη σε μέτρα, με αποτέλεσμα την επιστροφή όλων των σημείων εντός αυτού του κύκλου. Για την εύρεση των k κοντινότερων γειτόνων[13], ομοίως ορίζοντας έναν κύκλο υπολογίζονται τα k κοντινότερα σημεία από το κέντρο του.

Στη συνέχεια θα πρέπει στα χωρικά ερωτήματα που αναφέρθηκαν προηγουμένως να εισαχθεί και η έννοια του χρόνου. Δηλαδή να καταστεί εφικτός για ένα συγκεκριμένο χρονικό διάστημα ο εντοπισμός ενός συνόλου από χωρικά δεδομένα και να εφαρμοστούν αντίστοιχα, ερωτήματα εύρους και κοντινότερου γείτονα.

Τα ερωτήματα που βασίζονται σε δοθείσες τροχιές αφορούν την εύρεση k παρόμοιων τροχιών για ένα συγκεκριμένο χρονικό διάστημα.

5 MONGODB

Τα δεδομένα αποθηκεύτηκαν στη MongoDB, όπου έγινε και η μοντελοποίηση τους. Η MongoDB είναι μια μη σχεσιακή βάση δεδομένων προσανατολισμένη στα έγγραφα (document-oriented). Αυτό σημαίνει ότι δεν χρησιμοποιεί πίνακες, αλλά συλλογές εγγράφων τύπου JSON (BSON). Κάθε έγγραφο δεν είναι απαραίτητο να περιέχει τα ίδια πεδία (fields). Επίσης δίνει τη δυνατότητα εμφωλευμένων εγγράφων σε ένα πεδίο.

5.1 Περιγραφή Μοντελοποίησης

Το στάδιο της μοντελοποίησης παίζει καθοριστικό ρόλο, διότι σε αυτή βασίζεται η σωστή λειτουργία και η βέλτιστη αποδοτικότητα της εφαρμογής[3]. Το πρώτο βήμα στη βελτιστοποίηση απόδοσης είναι να γίνουν πλήρως κατανοητά τα μοτίβα ερωτημάτων της εφαρμογής, ώστε να σχεδιαστεί το κατάλληλο μοντέλο με τα κατάλληλα ευρετήρια. Η προσαρμογή της μοντελοποίησης στα μοτίβα των ερωτημάτων παράγει πιο αποτελεσματικά ερωτήματα και κατανέμει πιο αποτελεσματικά τον φόρτο εργασίας σε ένα cluster.

Πριν ξεκινήσει η μοντελοποίηση, έγινε η εισαγωγή των αρχείων που επιλέχθηκαν στη βάση. Σκοπός είναι η δημιουργία μόνο μιας συλλογής, η οποία θα είναι δομημένη με τέτοιο τρόπο, έτσι ώστε να μπορεί να απαντήσει τα ερωτήματα που παρουσιάστηκαν στην ενότητα 3 σε όσο το δυνατόν χαμηλότερο χρόνο απόκρισης. Δεν επιλέχθηκε η χρήση περισσότερων από μία συλλογή για δύο λόγους. Πρώτον αν χρειαζόταν ένα ερώτημα τη χρήση δύο ή παραπάνω συλλογών, θα έπρεπε να χρησιμοποιήσουμε την εντολή **$lookup** η οποία θα κόστιζε πολύ σε χρόνο μειώνοντας έτσι την απόδοση της εφαρμογής. Δεύτερον αν υπήρχαν συλλογές οι οποίες περιείχαν ίδια δεδομένα, αλλά διαφορετική μοντελοποίηση έτσι ώστε να χρησιμοποιείται η κατάλληλη για το αντίστοιχο ερώτημα, για το οποίο φτιάχτηκε, τότε θα υπήρχε πλεονασμός δεδομένων.

Η συλλογή στήθηκε πάνω στο αρχείο nari_dynamic, το οποίο περιέχει τα δεδομένα πλοήγησης για κάθε πλοίο και από τα υπόλοιπα αρχεία πήραμε τις πληροφορίες που μας ενδιέφεραν ενσωματώνοντας τες σε αυτό. Το μοτίβο σχεδίασης σχήματος που επιλέχθηκε είναι το **bucket**, μιας και είναι ιδανικό για δεδομένα ροής (streaming data) όπως οι τροχιές. Τα έγγραφα θα ομαδοποιηθούν ανά sourcemmsi και ανά μήνα, δηλαδή για ένα πλοίο θα υπάρχουν το πολύ έξι έγγραφα που θα απευθύνονται σε αυτό, και πέρα από τα πεδία για τα χαρακτηριστικά του πλοίου θα υπάρχει και ένα ακόμα, στο οποίο θα περιέχεται ένα εμφωλευμένο έγγραφο με τα χωροχρονικά δεδομένα του πλοίου τον αντίστοιχο μήνα.

Το nari_dynamic αποτελείται από 19 εκατομμύρια έγγραφα. Από αυτά βρεθήκαν περίπου 560 χιλιάδες διπλότυπα, τα οποία αφαιρέθηκαν. Η MongoDB έχει ως περιορισμό πως ένα έγγραφο μπορεί να έχει μέγεθος το πολύ 16mb, κάτι το οποίο εμποδίζει το αρχικό πλάνο για τη δημιουργία μοτίβου bucket, διότι υπάρχουν πλοία που έχουν πολλή πληροφορία στο διάστημα ενός μήνα που ξεπερνάει το όριο χωρητικότητας. Για παράδειγμα το πλοίο με sourcemmsi 228186700 έχει στο χρονικό διάστημα από 20/01/2016 μέχρι 24/01/16 214.464 σημεία. Κάνοντας έναν έλεγχο παρατηρήθηκε ότι για το συγκεκριμένο πλοίο υπάρχουν δεδομένα με χρονική διαφορά δυο δευτερολέπτων μεταξύ τους. Βρέθηκαν τα πλοία που είχαν αντίστοιχα πυκνή πληροφορία σε μικρό χρονικό διάστημα και αφαιρέθηκαν τα έγγραφα ανά sourcemmsi που είχαν χρονική διαφορά μικρότερη από δέκα δευτερόλεπτα. Αυτή η ενέργεια υλοποιήθηκε μέσω της MongoDB. Πρώτα ταξινομήθηκε η συλλογή κατά αύξουσα σειρά για τα πεδία του sourcemmsi και της χρονικής στιγμής, ύστερα αρχικοποιώντας μια μεταβλητή time με τιμή ίση με 1 και με τη βοήθεια μιας συνάρτησης στη γλώσσα

προγραμματισμού JavaScript, δημιουργήθηκε μια λούπα η οποία σύγκρινε την τιμή του εκάστοτε εγγράφου με του προηγούμενου. Αν δεν ικανοποιούσε η διαφορά των χρονικών στιγμών τη συνθήκη, δηλαδή να ήταν μικρότερη των δέκα δευτερολέπτων, τότε το έγγραφο διαγραφόταν από τη συλλογή. Με αυτόν τον τρόπο έγινε εφικτή η επίτευξη του bucket μοτίβου, δίχως να χάσουμε σημαντικά δεδομένα για τις τροχιές των πλοίων.

Στη συνέχεια δημιουργήθηκαν δύο νέα πεδία. Το πρώτο (**Month**) σαν τιμές έπαιρνε το όνομα του μήνα που αντιστοιχεί στην τιμή του πεδίου timestamp και το δεύτερο (**location**) εμφωλεύθηκε ένα έγγραφο το οποίο σαν πεδία είχε το **type** όπου έπαιρνε την τιμή Point και το **coordinates** που είχε μία λίστα με δύο στοιχεία, το γεωγραφικό πλάτος και το γεωγραφικό μήκος (lon, lat). Το πεδίο **location** είναι απαραίτητο για τις εντολές που θα χρησιμοποιήσουμε για τα χωρικά ερωτήματα και δίχως αυτό δεν είναι εφικτή η χρήση τους. Έπειτα έγινε ομαδοποίηση με βάση τα πεδία του sourcemmsi και του μήνα και σε ένα πεδίο με την ονομασία **spatiotemp** εισήχθησαν μέσω της εντολής **$push** έγγραφα που το κάθε ένα περιείχε τη χωροχρονική πληροφορία του πλοίου. Αυτό είχες ως αποτέλεσμα η συλλογή από 19 εκατομμύρια έγγραφα να μειωθεί σε 9.5 χιλιάδες.

Επειδή το πεδίο Month δημιουργήθηκε μόνο για να επιτευχθεί το μοτίβο που θέλαμε, δεν χρησίμευε κάπου πια. Για αυτό το λόγο αντικαταστάθηκε από δυο νέα, το **min_timestamp** και το **max_timestamp** με τιμές τη μικρότερη και αντίστοιχα τη μεγαλύτερη από το πεδίο **spatiotemp.timestamp.** Αυτά τα δυο νέα πεδία θα μπορούσαν να βοηθήσουν και να βελτιστοποιήσουν την απόδοση σε ερωτήματα που περιέχουν την παράμετρο του χρόνου.

Για τον κωδικό sourcemmsi ισχύει πως τα τρία πρώτα ψηφία του είναι ένας κωδικός που αντιστοιχεί στην ονομασία μίας χώρας. Με το συνδυασμό αυτού του στοιχείου και του αρχείου MMSI Country Code, το οποίο περιέχει την κανονική ονομασία της χώρας και τον κωδικό της, δημιουργήθηκε το πεδίο **shipflag** με τιμές την ονομασία την χώρας που ανήκει το κάθε πλοίο.

Από το αρχείο anfr επιλέχθηκαν τα πεδία με το όνομα του πλοίου (**ship_name**), τον τύπο του πλοίου (**shiptype**) και το sourcemmsi έτσι ώστε να γίνει η σύνδεση με την συλλογή που περιέχει τη χωρική και χρονική πληροφορία και να ενσωματωθούν σε αυτή. Το αρχείο nari_static περιέχει όμοια πληροφορία με το anfr, δηλαδή περιέχει δεδομένα όπως το όνομα του πλοίου και τον τύπο του. Μόνο που ο τύπος του πλοίου έχει σαν τιμή μια κωδική ονομασία, η οποία αντικαταστάθηκε με την κανονική ονομασία μέσω του Ship Type List αρχείου . Στη συνέχεια επαναλήφθηκε η ίδια διαδικασία με πριν για όσα έγγραφα μπόρεσαν να αντιστοιχηθούν με τα sourcemmsi του nari_static και δεν είχαν πάρει από το anfr τα χαρακτηριστικά του ονόματος και του τύπου του πλοίου. Τέλος η συλλογή που έχει δημιουργηθεί μέχρι στιγμής περιέχει και το navigational status με κωδική ονομασία η οποία αντικαταστάθηκε με την πραγματική από το αρχείο Navigational Status.

Τα έγγραφα της τελικής συλλογής έχουν την παρακάτω μορφή:


```
{ _id:
  { sourcemmsi: '205655000',
    min_timestamp: 1458905631,
    max_timestamp: 1458927231 },
 shipname: 'WAASMUNSTER',
 shiptype: 'Tanker',
 nav_status: 'under way using engine',
shipflag: 'Belgium',
 spatiotemp:
[{ location: { type: 'Point', coordinates: [ -6.227605, 48.052174 ] },
     timestamp: 1458905631 },
   { location: { type: 'Point', coordinates: [ -4.52478, 49.1831 ] },
     timestamp: 1458927231 } ]}
```


5.2 Ευρετηρίαση

Σε οποιαδήποτε βάση δεδομένων, τα ευρετήρια υποστηρίζουν την αποτελεσματική εκτέλεση των ερωτημάτων. Χωρίς αυτά, η βάση δεδομένων πρέπει να σαρώσει κάθε έγγραφο σε μια συλλογή για να επιλέξει αυτά που ταιριάζουν με την δήλωση του ερωτήματος. Εάν υπάρχει το κατάλληλο ευρετήριο για ένα ερώτημα, η βάση δεδομένων μπορεί να το χρησιμοποιήσει για να περιορίσει τον αριθμό των εγγράφων που πρέπει να επιθεωρήσει. Η MongoDB προσφέρει μια μεγάλη γκάμα από ευρετήρια. Μπορούν να δημιουργηθούν και να διαγραφούν κατά απαίτηση, για να ικανοποιήσουν τις εξελισσόμενες απαιτήσεις εφαρμογών και τα μοτίβα των ερωτημάτων. Υπάρχει η δυνατότητα δήλωσής τους σε οποιοδήποτε πεδίο στα έγγραφα, συμπεριλαμβανομένων των πεδίων που είναι εμφωλευμένα.

Η στρατηγική που επιλέχθηκε για τη διαδικασία της ευρετηρίασης βασίστηκε εξ’ ολοκλήρου στα ερωτήματα της εφαρμογής[1]. Τα πεδία στα οποία δημιουργήθηκαν ευρετήρια είναι:

1. _id: Η MongoDB δημιουργεί αυτό ένα μοναδικό ευρετήριο στο πεδίο _id κατά τη διάρκεια της δημιουργίας της συλλογής ώστε να αποφευχθεί η εισαγωγή δύο εγγράφων με την ίδια τιμή σε αυτό το πεδίο. Στην παρούσα συλλογή το πεδίο αυτό αποτελείται από το sourcemmsi, το min_timestamp και max_timestamp και η ευρετηρίαση συμβάλλει στην γρηγορότερη εκτέλεση ερωτημάτων που αφορούν χρόνο.
2. spatiotemp.location: Έγινε η ανάθεση ενός ευρετηρίου Geospatial 2dsphere, ιδανικό για ερωτήματα γεωχωρικών συντεταγμένων, διότι χρησιμοποιεί σφαιρική γεωμετρία για να επιστρέψει αποτελέσματα. Επίσης οι εντολές που χρησιμοποιούνται στα χωρικά ερωτήματα έχουν ως απαραίτητη προϋπόθεση την ύπαρξή του.
3. spatiotemp.timestamp: Η δημιουργία ενός ευρετηρίου σε αυτό το πεδίο βελτιώνει κατά πολύ την απόκριση των ερωτημάτων που εμπεριέχουν το χαρακτηριστικό του χρόνου.
4. shipname: Η χρήση σε αυτό το πεδίο βοηθά τη γρηγορότερη αναζήτηση με βάση την ονομασία ενός πλοίου.

Στα υπόλοιπα πεδία (shipflag, shiptype, nav_status) δεν αποδόθηκε κάποιο ευρετήριο, διότι το πλήθος των μοναδικών τιμών τους είναι μικρό. Η χρήση των ευρετηρίων δεν πρέπει να είναι αλόγιστη. Απαιτούν πόρους και καταναλώνουν μνήμη RAM και δίσκο.

5.3 Κατάτμηση

Είναι μια μέθοδος για το διαμοιρασμό των δεδομένων σε πολλαπλά μηχανήματα (shards). Η MongoDB χρησιμοποιεί την κατάτμηση για να υποστηρίζει αποθήκευση πολύ μεγάλων συνόλων δεδομένων και διαχείριση μεγάλου φόρτου εργασίας. Υπάρχουν δύο είδη κλιμάκωσης, η οριζόντια και η κατακόρυφη. Η MongoDB υιοθετεί την οριζόντια. Περιλαμβάνει τη διαίρεση του συνόλου δεδομένων του συστήματος και τη φόρτωση τους σε πολλούς διακομιστές (shard servers). Αν και η ταχύτητα και η χωρητικότητα ενός μεμονωμένου μηχανήματος μπορεί να μην είναι υψηλή, κάθε μηχάνημα διαχειρίζεται ένα υποσύνολο του φόρτου εργασίας, παρέχοντας δυνητικά καλύτερη απόδοση από ένα μεμονωμένο διακομιστή υψηλής ταχύτητας και χωρητικότητας.

Στην παρούσα εργασία δημιουργήθηκε ένα cluster το οποίο αποτελείται από τρεις συνιστώσες:

1) **Config Server:** Αποθηκεύονται τα μεταδεδομένα του cluster. Τα μεταδεδομένα αντικατοπτρίζουν την κατάσταση και την οργάνωση για όλα τα δεδομένα και τα components μέσα στο cluster. Τα μεταδεδομένα περιλαμβάνουν τη λίστα των κομματιών (chunks) σε κάθε shard και τις περιοχές που αυτά ορίζουν.

2) **QueryRouter (mongos):** Λειτουργεί ως διεπαφή και σημείο εισόδου στο cluster. Η εφαρμογή συνδέεται σε αυτό αντί να συνδέεται με τα υποκείμενα shards. Το mongos εκτελεί ερωτήματα, συλλέγει αποτελέσματα και τα μεταβιβάζει πίσω στην εφαρμογή.

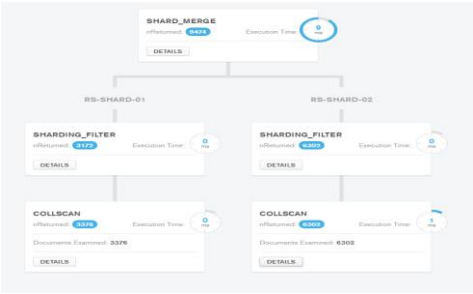
3) **Shards**: Στο κάθε ένα περιέχεται ένα υποσύνολο των διαχωρισμένων δεδομένων. Κάθε shard μπορεί να δημιουργηθεί ως replica set.

Το cluster κατασκευάστηκε με τη βοήθεια ενός docker-compose. Σηκώθηκαν συνολικά τέσσερα containers τα οποία επικοινωνούσαν μεταξύ τους, ένα QueryRouter, ένας Config Server και δύο shards.

Στη MongoDB πρέπει να επιλεχθεί το πεδίο πάνω στο οποίο θα γίνει η κατάτμηση. Η επιλογή του είναι πολύ σημαντική καθώς καθορίζεται ο τρόπος με τον οποίο θα διαχωριστούν τα δεδομένα. Η συλλογή διαχωρίστηκε με βάση το πεδίο _id.sourcemmsi, το οποίο χρησιμοποιείται στα σχεσιακά ερωτήματα και στα ερωτήματα για δοθείσες τροχιές. Η αρχική σκέψη ήταν να διαχωριστούν ως προς το πεδίο του χρόνου (spatiotemp.timestamp), όμως δεν είναι δυνατός ο διαχωρισμός με βάση ένα εμφωλευμένο πεδίο. Επίσης η κατάτμηση με κάποιο από τα πεδία _id.min_timestamp, _id.max_timestamp δεν θα ωφελούσε κάπως την απόδοση αφού δεν χρησιμοποιήθηκαν σε κανένα ερώτημα. Απαραίτητο βήμα για το διαμοιρασμό είναι και η δημιουργία ενός ευρετηρίου σε αυτό το πεδίο. Το ευρετήριο που επιλέχθηκε είναι το Hashed. Αυτή η προσέγγιση εγγυάται μια ομοιόμορφη κατανομή των δεδομένων στα shards.

Πίνακας 1: Διαχωρισμός των δεδομένων στα Shards

	Data	Docs	Chunks
Shard 1	576.28 MB	4597	6
Shard 2	675.15 MB	4877	6



Εικόνα 1: Αναπαράσταση του Cluster στο MongoDBCompass

6 Εφαρμογή

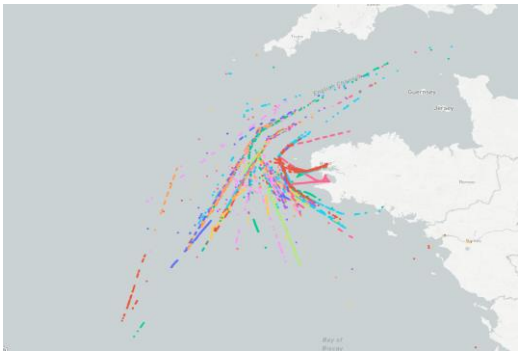
Αναπτύχθηκε με την χρήση της γλώσσας προγραμματισμού Python και κατάλληλων βιβλιοθηκών. Σκοπός της εφαρμογής είναι η λήψη των αποτελεσμάτων των ερωτημάτων που ορίσαμε στην αρχή από το cluster, η επεξεργασία τους και η οπτικοποίηση τους.

6.1 Σχεσιακά ερωτήματα

Σε αυτό τον τύπο ερωτημάτων, ζητείται από το χρήστη να εισάγει τα χαρακτηριστικά που τον ενδιαφέρουν. Τα διαθέσιμα χαρακτηριστικά είναι η χώρα προέλευσης, ο τύπος , το όνομα και η κατάσταση πλοήγησης του πλοίου. Έχει τη δυνατότητα να επιλέξει από ένα έως και τα τέσσερα. Στη συνέχει καλείται να δώσει τιμές σε κάθε ένα από αυτά. Η τιμή για κάθε χαρακτηριστικό μπορεί να είναι μία ή πολλές. Στη συνέχεια η εφαρμογή καλεί το κατάλληλο ερώτημα για να πάρει τα αποτελέσματα με βάση τις τιμές που εισήγαγε ο χρήστης. Για παράδειγμα αν έχουν επιλεχθεί δυο χαρακτηριστικά το ερώτημα που θα σχηματιστεί είναι το εξής:

```
query = collection.find({
        relations[0]:{"$in": relations_values [0]}
        ,relations[1]:{"$in": relations_values[1]}})
```

Το relations είναι μια λίστα όπου περιέχονται τα ονόματα των πεδίων που θα χρησιμοποιηθούν. Από το μήκος της, η εφαρμογή βρίσκει το πλήθος των πεδίων που δηλώθηκαν. Ύστερα ζητώνται τιμές για κάθε στοιχείο της relations και αυτές αποθηκεύονται σε μια λίστα, η οποία θα αποθηκευτεί στη relations_values (λίστα).



Εικόνα 2: Απεικόνιση όλων των πλοίων με σημαία Μάλτας και τύπο πλοίου Cargo.

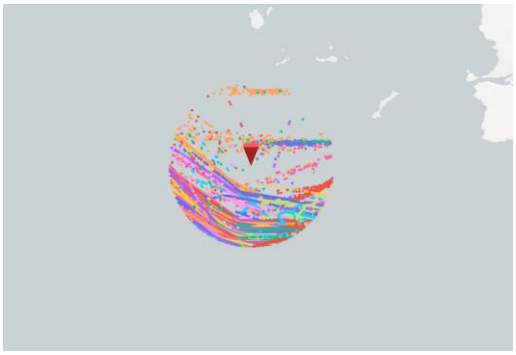
Στα σχεσιακά ερωτήματα δίνεται και η δυνατότητα της αναπαράστασης της τροχιάς ενός μόνο πλοίου μέσα στο συγκεκριμένο εξάμηνο, δίνοντας τον κωδικό sourcemmsi που του αναλογεί.

6.2 Χωρικά ερωτήματα

Ερωτήματα εύρους και k κοντινότερων γειτόνων είναι οι επιλογές που έχει ο χρήστης στην κατηγορία των χωρικών ερωτημάτων. Στα ερωτήματα εύρους ζητείται από το χρήστη να δώσει το γεωγραφικό πλάτος και μήκος ενός σημείου στο χάρτη και μια ακτίνα σε μέτρα. Η δομή του ερωτήματος είναι η εξής:

```
query = collection.aggregate([{
'$geoNear': {'near': {'type': 'Point',
                      'coordinates': [lon,lat]},
    'distanceField': 'dist.calculated', 'maxDistance': max_distance,
                    'includeLocs': 'dist.location', 'spherical': True,
                        'key': 'spatiotemp.location'}}, { '$unwind': {'path':
'$spatiotemp',
            'preserveNullAndEmptyArrays': True}}, {
'$match': {'spatiotemp.location': { '$geoWithin': {
            '$centerSphere':[[lon, lat],
            max_distance/(6371 * 1000)]}}}}])
```

Με τη βοήθεια της συνάρτησης **$geonear** της MongoDB, η οποία τροφοδοτείται με τις συντεταγμένες (**coordinates**) που δόθηκαν και το μήκος της ακτίνας (max_distance), επιστρέφονται όλα τα έγγραφα που έχουν τουλάχιστον ένα ζεύγος συντεταγμένων που περιέχεται στο εύρος που ορίστηκε. Στα έγγραφα αυτά όμως, περιέχονται και συντεταγμένες που ξεπερνούν το επιτρεπτό εύρος. Ο τελεστής **$unwind** αποδομεί ένα πεδίο πίνακα για να εξάγει ένα έγγραφο για κάθε στοιχείο του. Κάθε εξαγόμενο έγγραφο είναι το έγγραφο εισόδου με την τιμή του πεδίου πίνακα να αντικαθίσταται από το στοιχείο. Με τη συνάρτηση **$geoWithin**, σε συνδυασμό με τη συνάρτηση **$centerSphere** η οποία δέχεται τις συντεταγμένες του χρήστη και την ακτίνα, επιλέγονται τα κατάλληλα έγγραφα εντός του εύρους.



Εικόνα 3: Απεικόνιση όλων των σημείων εντός ενός κύκλου με κέντρο το σημείο [48.315828, -4.96444] και ακτίνα 5 χιλιομέτρων.

Για το ερώτημα με τους k κοντινότερους γείτονες, πέρα από τις συντεταγμένες και την ακτίνα, ο χρήστης πρέπει να εισάγει και το πλήθος των κοντινότερων γειτόνων που επιζητά. Η δομή του ερωτήματος παρόμοια με του προηγούμενου.

```
query = collection.aggregate([{
            '$geoNear': {'near': {'type': 'Point',
             'coordinates': [lon,lat]},
             'distanceField': 'dist.calculated',
            'maxDistance': max_distance,
            'includeLocs': 'dist.location',
            'spherical': True, 'key': 'spatiotemp.location'}}
           , {'$limit':k}])
```

Η συνάρτηση **$geonear** έχει την ιδιότητα να επιστρέφει τα έγγραφα σε σειρά από το πλησιέστερο προς το πιο απομακρυσμένο από το προκαθορισμένο σημείο. Επίσης μέσω της παραμέτρου **“includeLocs”,** δημιουργεί νέο πεδίο στα έγγραφα το οποίο περιέχει τις συντεταγμένες, για τις οποίες το έγγραφο επιλέχθηκε**.** Με αυτόν τον τρόπο αποφεύγεται η χρήση του τελεστή **$unwind**. Θέτοντας ένα όριο με τον τελεστή **$limit** επιστρέφονται μόνο τα k πιο κοντινά σημεία. Στην εφαρμογή έχει οριστεί ένας έλεγχος ο οποίος αποτρέπει την ύπαρξη ενός sourcemmsi δεύτερη φορά.

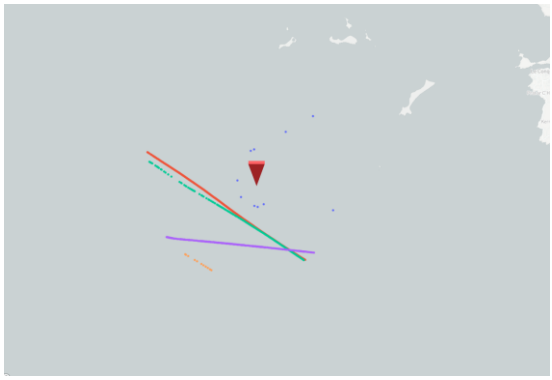
Εικόνα 4: Απεικόνιση των οχτώ κοντινότερων γειτόνων του σημείου με συντεταγμένος [48.307, -4.641] σε ακτίνα τριών χιλιομέτρων.

6.3 Χωροχρονικά ερωτήματα

Τα χωροχρονικά ερωτήματα είναι τα ίδια με τα χωρικά, δηλαδή εύρους και k κοντινότερων γειτόνων, με τη διαφορά ότι χρησιμοποιείται και η παράμετρος του χρόνου για φιλτράρισμα. Το αντίστοιχο ερώτημα εύρους είναι το εξής:

```
query = collection.aggregate([{'$geoNear': {
'near':{'type':'Point','coordinates':[lon, lat]]},
'distanceField': 'dist.calculated', 'maxDistance': max_distance,
'includeLocs': 'dist.location', 'spherical': True,
'key': 'spatiotemp.location'}},
{'$match': {'spatiotemp.timestamp': {'$gte': date0.timestamp(), '$lte': date1.timestamp()}}}
,{'$unwind':{'path': '$spatiotemp', 'preserveNullAndEmptyArrays': True}}
,{'$match': {'spatiotemp.location': {'$geoWithin':
{'$centerSphere': [[lon,lat]], max_distance/(6378.1*1000)]}}
,'spatiotemp.timestamp':{'$gte':1454278929,'$lte':1457994129}}])
```

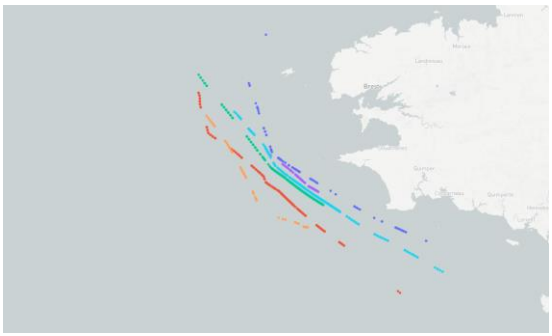
Τα ερωτήματα είναι με τέτοιο τρόπο δομημένα έτσι ώστε να επιτυγχάνεται η χρήση του index στο spatiotemp.timestamp. Χρησιμοποιείται ένα φιλτράρισμα με το πεδίο του χρόνου πριν τη χρήση του $unwind, για να περιοριστούν περισσότερο τα έγγραφα με βάση το χρόνο πριν την αποδόμηση, καθώς μετά το $unwind δεν γίνεται η χρήση των ευρετηρίων.



Εικόνα 5: Απεικόνιση ίδιου σημείου με την Εικόνα 3 για τη χρονική περίοδο 10/02/16 – 15/02/16.

6.4 Ερωτήματα Τροχιών

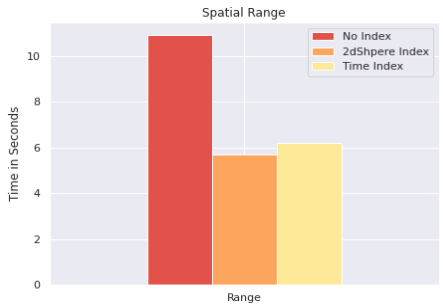
Σε αυτή την κατηγορία ανήκει η εύρεση των k όμοιων τροχιών για μια δοθείσα σε ένα συγκεκριμένο χρονικό διάστημα. Στη διαδικασία αυτή χρησιμοποιούνται δυο ερωτήματα. Το πρώτο αφορά την κύρια τροχιά. Αυτή ανακτάτε με την κατάλληλη τιμή στο πεδίο sourcemmsi και ένα φιλτράρισμα στο χρονικό διάστημα που ορίστηκε. Το δεύτερο ερώτημα είναι το ίδιο σαν δομή, με τη διαφορά σαν φίλτρο για το πεδίο sourcemmsi επιλέγονται όλα τα υπόλοιπα εκτός αυτού που χρησιμοποιήθηκε στο πρώτο ερώτημα. Έτσι συλλέγονται όλες οι τροχιές για το συγκεκριμένο χρονικό διάστημα, οι οποίες θα συγκριθούν με την αρχική. Η σύγκριση και η επιλογή των k πιο όμοιων τροχιών γίνεται μέσω της βιβλιοθήκης tslearn και της συνάρτησής της dtw [9] στην Python, η οποία επιστρέφει τη μέση χιλιομετρική διαφορά μεταξύ των τροχιών. Τα αποτελέσματα αποθηκεύονται σε μια λίστα η οποία ταξινομείται κατά αύξουσα σειρά. Από τη λίστα αυτή αποκλείονται τροχιές πλοίων με πολύ μικρό αριθμό σημείων. Το ερώτημα απαντάται παίρνοντας τα k πρώτα στοιχεία τις λίστας και τα sourcemmsi που αντιστοιχούν σε αυτά, οπτικοποιώντας τις τροχιές τους μαζί με την αρχική.



Εικόνα 6: Δοθείσα τροχιά (πράσινη) και οι πέντε πιο όμοιες τροχιές

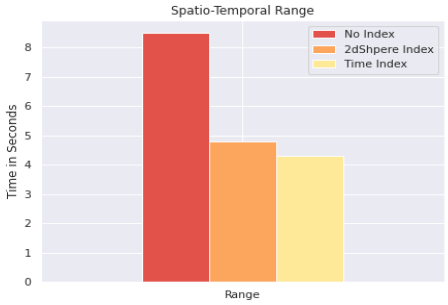
7 ΑΠΟΤΕΛΕΣΜΑΤΑ

Παρακάτω παρουσιάζονται αναλυτικά οι χρόνοι απόκρισης της εφαρμογής για τα ερωτήματα εύρους σε χωρικό αλλά και σε χωροχρονικό πλαίσιο. Η σύγκριση γίνεται με βάση τη χρήση ευρετηρίων που χρησιμοποιήθηκαν. Επειδή δεν γίνεται να χρησιμοποιηθεί η συνάρτηση **$geonear** χωρίς 2dSphere ευρετήριο, για την no index εκδοχή, χρησιμοποιήθηκε η συνάρτηση **$geoWithin** για την κατασκευή του αντίστοιχου ερωτήματος.



Εικόνα 7: Αποκρίσεις της εφαρμογής για το ερώτημα εύρους χωρίς την παράμετρο του χρόνου.

Είναι φανερό πως προσθέτοντας και το ευρετήριο στο πεδίο του χρόνου, αυξάνεται ο χρόνος απόκρισης της εφαρμογής για το παρόν ερώτημα. Όμως το ευρετήριο διατηρήθηκε γιατί το συγκεκριμένο ευρετήριο λειτούργησε θετικά στο ίδιο ερώτημα με την παράμετρο του χρόνου, αλλά και στο ερώτημα για την εύρεση όμοιων τροχιών για κάποιο χρονικό διάστημα



Εικόνα 8: Αποκρίσεις της εφαρμογή για το ερώτημα εύρους με την παράμετρο του χρόνου

Οι αποκρίσεις της εφαρμογής άλλαζαν με βάση τον όγκο δεδομένων που επέστρεφε το ερώτημα σε αυτή. Αυτό εξαρτιόταν από τις τιμές που θα δέχονταν οι παράμετροι του ερωτήματος. Για παράδειγμα δίνοντας σαν σημείο συντεταγμένες που βρίσκονται κοντά σε λιμάνι, είναι λογικό να βρεθούν περισσότερα έγγραφα σε σχέση με ένα σημείο κάπου ανοιχτά του ωκεανού. Επίσης οι παράμετροι του χρόνου και της ακτίνας επηρέαζαν τον όγκο των δεδομένων. Όσο μεγαλύτερο το χρονικό διάστημα και η ακτίνα τόσο περισσότερα δεδομένα θα ανακτούσε η εφαρμογή.
Λόγω της χρήσης της συνάρτησης **$geonear** δεν δινόταν η δυνατότητα μέσω της εντολής **'executionStats'** να πάρουμε μετρήσεις για κάθε shard ξεχωριστά.

8 ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Στα πλαίσια της παρούσας εργασίας έγινε αισθητή η αποτελεσματικότητα των μη σχεσιακών βάσεων δεδομένων και συγκεκριμένα της MongoDB για την ανάπτυξη εφαρμογής στην οποία γίνεται απεικόνιση τροχιών πλοίων σε γεωγραφικό χάρτη. Η δυνατότητα που παρέχει για ευέλικτα σχήματα εγγράφων καθιστά δυνατή την κατάλληλη μοντελοποίηση για τις ανάγκες των ερωτημάτων. Ιδιαίτερα η εμφώλευση εγγράφων κατέστησε ικανή την ομαδοποίηση των δεδομένων διατηρώντας έτσι τα δεδομένα που είχαν κάποια σχέση μεταξύ τους σε κοινό έγγραφο και μειώνοντας δραματικά τον όγκο των συνολικών εγγράφων. Η ευρετηρίαση και η οριζόντια κλιμάκωση βελτίωσαν ακόμα περισσότερο την συνολική απόδοση.
Σε μελλοντικές επεκτάσεις πάνω στην παρούσα εργασία, θα ήταν ενδιαφέρουσα η χρήση δεδομένων ροής σε συνδυασμό με τη χρήση τεχνικών μηχανικής μάθησης και τεχνητής νοημοσύνης. Με την εφαρμογή τέτοιων τεχνικών και δεδομένων θα μπορούσαν να αναπτυχθούν εφαρμογές, οι οποίες θα πρότειναν βέλτιστες διαδρομές, θα προέβλεπαν την κίνηση (traffic) για κάποιο χρονικό διάστημα σε συγκεκριμένα χωρικά ύδατα αλλά και σε λιμάνια, θα προέτρεπαν πιθανά ατυχήματα ή συγκρούσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Berkay Aydin, Vijay Akkineni, Rafal A.Angryk, 2016. Modeling and Indexing Spatiotemporal Trajectory Data in Non-Relational Databases
[2] Χρήστος Γιαννόγλου. 2019. Επεξεργασία χωροχρονικών ερωτημάτων στη MongoDB
[3] Rober T Mason. 2015. NoSQL Databases and Data Modeling Techniques for a Document-oriented NoSQL Databse
[4] Pavel Senin, 2009, Dynamic Time Warping Algorithm Revie
[5] MongoDB documentation
[6] PyMongo documentation
[7] Koutroumanis, Nikolaos, NoDA: Unified NoSQL Data Access Operators for Mobility Data
[8] Nikolaos Koutroumanis, Christos Doulkeridis. 2021. Scalable Spatio-temporal Indexing and Query
[9] Yasushi Sakurai, Masatoshi Yoshikawa, Christos Faloutsos.. FTW: Fast Similarity Search under the Time Warping Distance
[10] Ernest Batty. 2017. Data Analytics Enables Advanced AIS Applications
[11] Rouma Rathore, Spatial Range Query
[12] Plotly documentation
[13] Coskunn B., Sertok S., Anbaroglu B. 2019. K-NEAREST NEIGHBOUR QUERY PERFORMANCE ANALYSES ON A LARGE SCALE TAXI DATASET: POSTGRESQL vs. MONGODB