Διαχείριση και Ανάλυση Χωροχρονικών Ναυτιλιακών Δεδομένων με MongoDB: Μοντελοποίηση Τροχιών και Πειραματική Αξιολόγηση

Τεχνική Αναφορά

Ελεντίνα Γρίσπου
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Πειραιάς, Ελλάδα
me25006

Γεώργιος Ιατρίδης
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Πειραιάς, Ελλάδα
me25011

Κωνσταντίνος Σπέγκας
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Πειραιάς, Ελλάδα
me25034

Παναγιώτης Χατούπης
Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Πειραιάς, Ελλάδα
me25042

Περίληψη

Η παρούσα τεχνική αναφορά πραγματεύεται τη σχεδίαση, υλοποίηση και αξιολόγηση μιας μη-σχεσιακής βάσης δεδομένων (NoSQL) σε περιβάλλον MongoDB, με στόχο την αποδοτική διαχείριση ναυτιλιακών δεδομένων μεγάλης κλίμακας. Αξιοποιώντας το «Piraeus AIS Dataset», η εργασία εστιάζει στη μετάβαση από το σχεσιακό μοντέλο σε ένα μοντέλο εγγράφων (document-oriented), εφαρμόζοντας το πρότυπο Bucketing για την ομαδοποίηση στιγμάτων σε τροχιές. Η προσέγγιση αυτή επιτυγχάνει βελτιωμένη τοπικότητα δεδομένων και μείωση του κόστους εισόδου/εξόδου (I/O). Στο πλαίσιο της εργασίας υλοποιήθηκαν σύνθετα χωροχρονικά ερωτήματα, όπως η γεωχωρική αναζήτηση εντός λιμένα και η ανίχνευση επικίνδυνων ελιγμών υπό δυσμενείς καιρικές συνθήκες (πλάγιος άνεμος), με χρήση εξειδικευμένων ευρετηρίων (2dsphere, compound). Η πειραματική αξιολόγηση κατέδειξε τη σημαντική βελτίωση της απόδοσης μέσω κατάλληλης ευρετηρίασης, ενώ αναλύθηκε η συμπεριφορά του συστήματος ως προς την κλιμακωσιμότητα, την επιλεξιμότητα και την επίδραση της δομικής πολυπλοκότητας των εμφωλευμένων εγγράφων.

1 Εισαγωγή, κίνητρα, ορισμός προβλήματος

1.1 Εισαγωγή

Η παρούσα τεχνική αναφορά εκπονείται στο πλαίσιο του μαθήματος «Διαχείριση Δεδομένων για Σχεσιακές και Μη-σχεσιακές Βάσεις Δεδομένων» του Π.Μ.Σ. «Πληροφοριακά Συστήματα και Υπηρεσίες». Αντικείμενο της εργασίας είναι η σχεδίαση, υλοποίηση και πειραματική αξιολόγηση μιας μη-σχεσιακής βάσης δεδομένων (NoSQL) στο σύστημα MongoDB, για τη διαχείριση πραγματικών ναυτιλιακών δεδομένων μεγάλης κλίμακας.

Συγκεκριμένα, αξιοποιείται το σύνολο δεδομένων "Piraeus AIS Dataset" [11], το οποίο περιλαμβάνει καταγραφές ετερογενών πηγών πληροφορίας από την ευρύτερη περιοχή του Σαρωνικού Κόλπου. Τα δεδομένα κατηγοριοποιούνται σε τέσσερις βασικούς πυλώνες:

(1) **Κινηματικά δεδομένα (AIS Kinematic):** Δυναμικά μεταβαλλόμενα στοιχεία θέσης, κίνησης και πορείας.
(2) **Στατικά δεδομένα (AIS Static):** Μόνιμα χαρακτηριστικά των πλοίων (vessel_id, τύπος πλοίου).
(3) **Περιβαλλοντικά δεδομένα (NOAA Weather):** Μετεωρολογικές μετρήσεις (άνεμος, ορατότητα, θερμοκρασία κλπ).
(4) **Γεωγραφικά δεδομένα (Geo-related data):** Στατικές χωρικές πληροφορίες που ορίζουν το περιβάλλον κίνησης (λιμάνια, νησιά, χωρικά ύδατα και η λεπτομερής γεωμετρία του λιμένα του Πειραιά).

Ο βασικός στόχος είναι η σχεδίαση και υλοποίηση μιας βάσης δεδομένων MongoDB που να αναδεικνύει την υπεροχή ενός document-oriented μοντέλου στη διαχείριση μεγάλων όγκων χωροχρονικών δεδομένων. Η μετάβαση από μια «παραδοσιακή» δομή αποθήκευσης σε ένα μοντέλο εγγράφων επιτρέπει την ταχύτερη απόκριση σε σύνθετα ερωτήματα και τη διασφάλιση οριζόντιας κλιμάκωσης (sharding).

1.2 Ορισμός προβλήματος

Το κεντρικό πρόβλημα έγκειται στην ανάγκη διαχείρισης «πολύπλοκων αναπαραστάσεων αντικειμένων», όπως ορίζεται στην εκφώνηση της εργασίας. Η διαχείριση τροχιών πλοίων (vessel trajectories) σε μεγάλες κλίμακες παρουσιάζει τρεις κύριες προκλήσεις:

1.2.1 Κατακερματισμός της πληροφορίας (*data fragmentation*). Στο παραδοσιακό σχεσιακό μοντέλο, οι τέσσερις πυλώνες δεδομένων (Kinematic, Static, Weather, Geo) θα έπρεπε να αποθηκευτούν σε ξεχωριστούς πίνακες. Όπως υποδεικνύεται από το διορθωμένο σχήμα του dataset (DB Schema [10]), η ανασύνθεση μιας πλήρους χωροχρονικής διαδρομής απαιτεί συνεχή και δαπανηρά JOINS μεταξύ πινάκων. Το ζητούμενο είναι η ενσωμάτωση (embedding) των σχετικών πληροφοριών σε ενιαία έγγραφα, ώστε να ελαχιστοποιηθεί το I/O overhead.

1.2.2 Διαχείριση «σημειακών» δεδομένων έναντι «τροχιών». Τα ακατέργαστα δεδομένα AIS είναι σημειακά (points). Η αποθήκευση κάθε στίγματος ως αυτόνομο έγγραφο (flat model) οδηγεί σε υπερμεγέθη ευρετήρια (index bloat) και αργή ανάκτηση. Η πρόκληση είναι η εφαρμογή του Bucketing Pattern [7], όπου

πολλαπλά στίγματα ομαδοποιούνται ανά πλοίο και χρονικό παράθυρο, βελτιώνοντας την τοπικότητα των δεδομένων (data locality).

1.2.3 Σύνθετα χωροχρονικά ερωτήματα. Η βάση καλείται να απαντήσει σε ερωτήματα που συνδυάζουν:

- **Χωρικά κριτήρια:** Χρήση GeoJSON για αναζήτηση εντός πολυγώνων (π.χ. λιμένας Πειραιά).
- **Χρονικά κριτήρια:** Χρήση χρονικά διαχωρισμένων buckets που επιταχύνουν την προσπέλαση στην βάση.
- **Περιβαλλοντικές συνθήκες:** Συσχέτιση μετεωρολογικών δεδομένων με κάθε θέση του πλοίου.

1.3 Κίνητρα

Τα κύρια κίνητρα για τη χρήση της MongoDB στη συγκεκριμένη εφαρμογή είναι:

- **Ευελιξία σχήματος (schema flexibility):** Δυνατότητα προσθήκης νέων πεδίων (annotations), επιπλέον μετεωρολογικών μετρήσεων ή αλλαγών στη δομή των δεδομένων χωρίς δαπανηρές μεταβολές σχήματος (schema migrations).
- **Ευκολία ενημέρωσης και επεκτασιμότητα:** Η εισαγωγή νέων αρχείων και η ενσωμάτωση επιπλέον πηγών πληροφορίας πραγματοποιείται άμεσα, καθώς το JSON/BSON-μοντέλο μπορεί να απορροφά νέα δεδομένα χωρίς αυστηρή προ-συμμόρφωση σε στατικούς πίνακες.
- **Αποδοτική ευρετηρίαση:** Αξιοποίηση 2dsphere indexes για γεωχωρικούς υπολογισμούς και αναζητήσεις.
- **Aggregation Framework:** Χρήση pipelines για υπολογισμό στατιστικών (π.χ. μέση ταχύτητα ανά τύπο πλοίου).

2 Βιβλιογραφική Ανασκόπηση

Η διαχείριση κινητικότητας (mobility data management) και η ανάλυση τροχιών πλοίων αποτελούν πεδίο με έντονη ερευνητική δραστηριότητα, λόγω των προκλήσεων που θέτουν ο όγκος και ο ρυθμός εισροής των δεδομένων AIS.

2.1 Το σύνολο δεδομένων Piraeus AIS

Η κύρια πηγή για την παρούσα εργασία είναι το "Piraeus AIS Dataset", το οποίο παρουσιάστηκε από τους Tritsarolis et al. [11]. Οι ερευνητές περιγράφουν τη διαδικασία συλλογής δεδομένων από σταθμό βάσης του Πανεπιστημίου Πειραιώς και τη σημασία του συνδυασμού ετερογενών πηγών (κινηματικών, στατικών και περιβαλλοντικών). Ιδιαίτερη έμφαση δίνεται στο διορθωμένο σχήμα της βάσης [10], το οποίο αναδεικνύει την πολυπλοκότητα των συσχετίσεων που απαιτούνται για την ανάλυση ναυτιλιακών δεδομένων, υπογραμμίζοντας εμμέσως τους περιορισμούς των παραδοσιακών σχεσιακών μοντέλων.

2.2 Μοντελοποίηση τροχιών και bucketing

Η μετάβαση από μεμονωμένα στίγματα (points) σε σημασιολογικές τροχιές (trajectories) αποτελεί κεντρικό θέμα στη βιβλιογραφία. Οι Patroumpas et al. [7] προτείνουν μεθοδολογίες για την online αναγνώριση συμβάντων και τη δημιουργία συνόψεων (summaries) τροχιών. Η προσέγγιση αυτή δικαιολογεί τη χρήση προτύπων σχεδίασης όπως το Bucketing Pattern στη MongoDB, το οποίο επιτρέπει τη συμπύκνωση της πληροφορίας και τη βελτίωση της τοπικότητας των δεδομένων (data locality). Με τον τρόπο αυτό, αποφεύγεται ο κατακερματισμός που θα επέφερε η αποθήκευση κάθε μηνύματος AIS ως ξεχωριστό έγγραφο. Ως αποτέλεσμα, μειώνεται δραματικά το μέγεθος των ευρετηρίων (indexes), βελτιώνοντας σημαντικά την απόδοση και την αξιοποίηση της μνήμης RAM [8].

2.3 Χωροχρονική ανάλυση σε NoSQL συστήματα

Η χρήση NoSQL συστημάτων για Big Data analytics στη ναυτιλία αναδεικνύει την ανάγκη για αποδοτικούς γεωχωρικούς υπολογισμούς (geospatial indexing) και την ανακάλυψη μοτίβων κίνησης. Συστήματα όπως η MongoDB, που υποστηρίζουν εγγενώς 2dsphere ευρετήρια και οριζόντια κλιμάκωση μέσω sharding, ευνοούν τη διαχείριση δυναμικών και ογκωδών συνόλων χωροχρονικών δεδομένων.

2.4 Κλιμακωσιμότητα και Διαμοιρασμός Δεδομένων

Καθώς ο όγκος των δεδομένων AIS αυξάνεται εκθετικά, η βιβλιογραφία NoSQL συστημάτων επικεντρώνεται στη δυνατότητα οριζόντιας κλιμάκωσης (sharding). Όπως επισημαίνεται στη βιβλιογραφία για κατανεμημένα συστήματα NoSQL [3, 4], η επιλογή ενός κατάλληλου `shard key` είναι καθοριστική για την αποφυγή ανισοκατανομής των δεδομένων (*data skewness*) και τη διασφάλιση της ομοιόμορφης κατανομής του φόρτου (*load balancing*). Βάσει των παραπάνω, η μοντελοποίηση που ακολουθείται στοχεύει στη δημιουργία εγγράφων που εν δυνάμει υποστηρίζουν την οριζόντια κλιμάκωση, ακόμη και αν η τρέχουσα υλοποίηση περιορίζεται σε έναν μόνο κόμβο.

2.5 Προηγμένη επεξεργασία χωροχρονικών δεδομένων

Η βιβλιογραφία σχετικά με τη διαχείριση τροχιών πλοίων [7] επισημαίνει ότι η ωμή πληροφορία AIS απαιτεί εμπλουτισμό για να καταστεί επιχειρησιακά χρήσιμη. Στην παρούσα εργασία, ακολουθείται η μεθοδολογία του Feature Engineering για τη συσχέτιση της πορείας του πλοίου (COG) με τις μετεωρολογικές συνθήκες της NOAA. Η προσέγγιση αυτή ευθυγραμμίζεται με τα διεθνή πρότυπα του Παγκόσμιου Μετεωρολογικού Οργανισμού (WMO [13]) για την κατηγοριοποίηση των ανέμων σε 8, 16 ή 32 διευθύνσεις (Wind Rose), εξασφαλίζοντας τη στατιστική εγκυρότητα των δεδομένων. Παράλληλα, η ανάλυση της σχετικής κίνησης πλοίου-ανέμου εδράζεται στις θεμελιώδεις αρχές της ναυτιλιακής επιστήμης, όπως αυτές κωδικοποιούνται στον οδηγό American Practical Navigator (Bowditch [2]), επιτρέποντας στη MongoDB να εκτελεί αποδοτικά ερωτήματα βασισμένα σε σημασιολογικές κατηγορίες αντί για ωμές αριθμητικές τιμές.

3 Σύνολο δεδομένων και προεπεξεργασία

Στην εργασία χρησιμοποιήθηκε υποσύνολο του Piraeus AIS Dataset, περιορισμένο στο πρώτο τρίμηνο του 2019 (Ιανουάριος–Μάρτιος). Η επιλογή αυτή επιτρέπει την διαχείριση ενός ρεαλιστικού όγκου

δεδομένων και επαρκή χρονική κάλυψη για την ανάδειξη χωροχρονικών μοτίβων.

3.1 Καθαρισμός στατικής πληροφορίας (AIS Static)

Η στατική πληροφορία πλοίων αποτελεί το απλούστερο και πιο σταθερό τμήμα του συνόλου δεδομένων και για τον λόγο αυτό η διαδικασία προεπεξεργασίας ξεκίνησε από αυτή. Χρησιμοποιήθηκαν δύο αρχεία:

(1) Το αρχείο στατικών χαρακτηριστικών πλοίων (`unipi_ais_static.csv`)
(2) Το αρχείο αντιστοίχισης κωδικών τύπων πλοίων με περιγραφές (`ais_codes_descriptions.csv`).

3.1.1 Έλεγχος πληρότητας. Τα κενά πεδία `country` συμπληρώθηκαν με την τιμή `Unknown`, ενώ τα κενά `shiptype` αντικαταστάθηκαν με τον κωδικό `0`, ώστε να είναι δυνατή η αντιστοίχισή τους με το λεξικό τύπων πλοίων. Στη συνέχεια, εφαρμόστηκε left join μεταξύ των στατικών δεδομένων και του λεξικού τύπων πλοίων με βάση το πεδίο `shiptype`. Για τους κωδικούς που δεν αντιστοιχούσαν σε γνωστή περιγραφή (π.χ. ειδικοί ή μη τεκμηριωμένοι κωδικοί), η περιγραφή συμπληρώθηκε ως `Unknown Type`.

3.1.2 Έλεγχος διπλοτύπων. Κατά τον έλεγχο μοναδικότητας, εντοπίστηκαν περιπτώσεις *λογικών διπλοτύπων*, όπου το ίδιο `vessel_id` εμφανιζόταν πολλαπλές φορές με διαφορετική ποιότητα πληροφορίας. Για την επίλυση του προβλήματος, υιοθετήθηκε πολιτική προτεραιότητας υπέρ των εγγραφών με έγκυρη και μη-`Unknown` περιγραφή τύπου πλοίου. Οι εγγραφές ταξινομήθηκαν ανά `vessel_id` και ποιότητα πληροφορίας έτσι ώστε να διατηρηθεί μία εγγραφή ανά πλοίο.

3.1.3 Ενοποίηση στατικών δεδομένων. Το τελικό καθαρισμένο σύνολο στατικών δεδομένων αποθηκεύτηκε σε ενιαίο αρχείο (`static.csv`), το οποίο χρησιμοποιήθηκε για τη δημιουργία της συλλογής `vessels` στη MongoDB. Κάθε πλοίο αναπαρίσταται ως ξεχωριστό έγγραφο με βασικά στατικά χαρακτηριστικά (`vessel_id`, χώρα σημαίας και πληροφορία τύπου πλοίου).

3.2 Καθαρισμός και ενοποίηση χρονοσειρών κίνησης (AIS Dynamic & AIS Synopsis)

Για το πρώτο τρίμηνο του 2019 (Ιαν–Μαρ) επεξεργάστηκαν δύο διακριτές χρονοσειρές ανά `vessel_id`:

- Τα πρωτογενή δυναμικά στίγματα AIS (*Dynamic AIS*)
- Οι συνόψεις/επισημειώσεις τροχιάς (*AIS Synopsis / annotations*).

3.2.1 *Dynamic AIS:* Ενοποίηση και κανόνες εγκυρότητας. Τα αρχεία `unipi_ais_dynamic_{jan,feb,mar}2019.csv` συνενώθηκαν σε ενιαίο σύνολο και στη συνέχεια εφαρμόστηκαν:

(1) **Αφαίρεση ακριβών διπλοτύπων** εγγραφών.
(2) **Μετατροπή χρόνου:** το `t` μετατράπηκε από milliseconds σε `datetime`.
(3) **Κανόνες εγκυρότητας:** διαγράφηκαν εγγραφές με `speed` εκτός `[0,60]` knots και `course` εκτός `[0,360]` μοιρών.
(4) **Λογικές συγκρούσεις:** μετά από ταξινόμηση ανά `vessel_id` και χρόνο, διατηρήθηκε μία εγγραφή ανά `(vessel_id, t)` (*keep first*).

Το καθαρισμένο σύνολο αποθηκεύτηκε ως `dynamic.csv`.

3.2.2 *AIS Synopsis:* Ενοποίηση και χρονική ευθυγράμμιση *annotations.* Για τις συνόψεις τροχιών (`unipi_ais_synopses_*.csv`) των μηνών Ιαν–Μαρ 2019 ακολουθήθηκε αντίστοιχη διαδικασία προετοιμασίας με την παταπάνω. Το τελικό σύνολο αποθηκεύτηκε ως `dynamic_synopsis.csv` και χρησιμοποιήθηκε αργότερα για την αντιστοίχιση annotations στα σημεία τροχιάς.

3.2.3 Πολιτική ακρίβειας *(precision policy).* Πριν την αποθήκευση στα CSV εφαρμόστηκε σταθερή πολιτική ακρίβειας για λόγους ομοιομορφίας και ελέγχου μεγέθους BSON:

- **Συντεταγμένες (`lon/lat`):** 5 δεκαδικά ψηφία.
- **Μετρικές κίνησης (`speed/course`):** 2 δεκαδικά ψηφία.

3.3 Χωροχρονική Συσχέτιση και Εμπλουτισμός

Για την ικανοποίηση των αναγκών των ερωτημάτων, πραγματοποιήθηκε συσχέτιση των κινήσεων με το περιβάλλον τους (`weather_with_dynamic2.py`):

(1) **Συσχέτιση Καιρού:** Χρησιμοποιήθηκε ο αλγόριθμος `cKDTree` για την εύρεση του πλησιέστερου μετεωρολογικού σταθμού για κάθε στίγμα πλοίου βάσει γεωγραφικής εγγύτητας.
(2) **Χρονικός Συγχρονισμός:** Οι μετρήσεις της NOAA ευθυγραμμίστηκαν χρονικά με τις θέσεις των πλοίων με στρογγυλοποίηση του χρόνου ανά 3 ώρες. Δηλαδή, βάσει της κοντινότερης μέτρησης σε χρονικό παράθυρο τριών ωρών.
(3) **Υπολογισμός Κατευθύνσεων (Cardinal Directions):** Για την πληρέστερη ερμηνεία των μετεωρολογικών συνθηκών και της κίνησης των πλοίων, υλοποιήθηκε η συνάρτηση `get_cardinal`. Η συνάρτηση αυτή μετατρέπει τις αριθμητικές τιμές των μοιρών ($0°$–$360°$) σε οκτώ κύρια σημεία του ορίζοντα (**N, NE, E, SE, S, SW, W, NW**). Ο υπολογισμός αυτός εφαρμόστηκε τόσο στην κατεύθυνση του ανέμου (`wind_cardinal`) όσο και στην πορεία του πλοίου (`course_cardinal`), επιτρέποντας την εκτέλεση σύνθετων αναλυτικών ερωτημάτων με όρους φυσικής γλώσσας.
(4) **Ενσωμάτωση Annotations:** Οι επισημειώσεις από τις συνόψεις των τροχιών ενσωματώθηκαν στα αντίστοιχα χρονικά σημεία μέσω του script `process_final_trips3.py`.

3.4 Μετασχηματισμός στατικών δεδομένων σε έγγραφα `vessels`

Το καθαρισμένο αρχείο `static.csv` μετατράπηκε σε μορφή JSON (`vessels_ready.json`) ώστε να εισαχθεί απευθείας στη MongoDB ως collection `vessels`. Για κάθε γραμμή δημιουργήθηκε ένα έγγραφο με:

- `vessel_id`
- `country` (με προεπιλογή `Unknown`),
- Εμφωλευμένη πληροφορία τύπου `type_info` με `shiptype_code` και `description`.

3.5 Ανακατασκευή τροχιών και παραγωγή εγγράφων `trips` (Bucketing Pattern)

Από το εμπλουτισμένο σύνολο `dynamic_with_weather.csv` πραγματοποιήθηκε ανακατασκευή των τροχιών και παραγωγή του αρχείου (`trips_ready.json`) για εισαγωγή στη συλλογή `trips`.

Οι τροχιές τμηματοποιούνται (*Trip Splitting*) όταν η χρονική απόσταση μεταξύ δύο διαδοχικών στιγμάτων υπερβαίνει τις 2 ώρες (120 λεπτά) ή όταν αλλάζει το `vessel_id`. Κάθε έγγραφο (document) στη συλλογή `trips` αντιπροσωπεύει ένα πλήρες τμήμα τροχιάς, περιλαμβάνοντας έναν πίνακα από εμπλουτισμένα σημεία (*embedded array of track points*), διασφαλίζοντας την τοπικότητα των δεδομένων (data locality)

Για έλεγχο συνέπειας, επιβλήθηκε σταθερή πολιτική ακρίβειας: `lon/lat` σε 5 δεκαδικά, ενώ μετρικές/καιρός σε 2 δεκαδικά. Επιπλέον, οι τιμές `heading` κρατήθηκαν μόνο όταν ≤ 360 (αλλιώς `null`). Τέλος, τα `trips` εμπλουτίστηκαν με στατική πληροφορία (χώρα/τύπος) μέσω lookup από το `static.csv`.

3.6 Εισαγωγή δεδομένων στη MongoDB και αρχικοποίηση βάσης

Μετά την ολοκλήρωση της προεπεξεργασίας, η βάση δεδομένων `piraeus_ais_db` αρχικοποιήθηκε εκ νέου ώστε να διασφαλιστεί καθαρό περιβάλλον φόρτωσης. Αρχικά διαγράφηκε πλήρως οποιοδήποτε προϋπάρχον περιεχόμενο και στη συνέχεια δημιουργήθηκαν οι συλλογές `vessels` και `trips`. Η εισαγωγή πραγματοποιήθηκε με τμηματική φόρτωση (*chunked loading*) για λόγους ασφάλειας μνήμης και ανθεκτικότητας σε σφάλματα. Τα στατικά δεδομένα φορτώθηκαν στη συλλογή `vessels`, ενώ τα αρχεία JSON των trips εισήχθησαν στη συλλογή `trips`.

3.6.1 Φόρτωση γεωχωρικών επιπέδων *(Geospatial Collections)*.

Για την υποστήριξη χωρικών ερωτημάτων φορτώθηκαν επιπλέον γεωχωρικά επίπεδα στη MongoDB, συγκεκριμένα οι συλλογές: `harbours`, `islands`, `piraeus_port`, `regions` και `territorial_waters`. Τα δεδομένα προήλθαν από shapefiles και μετατράπηκαν σε GeoJSON μορφή. Για κάθε συλλογή δημιουργήθηκε χωρικός δείκτης τύπου `2dsphere` στο πεδίο `geometry`, ώστε να υποστηρίζονται αποδοτικά χωρικά φίλτρα.

3.6.2 Αποθήκευση μετεωρολογικών δεδομένων ως ανεξάρτητη συλλογή.

Εκτός από τον εμπλουτισμό των κινήσεων, τα μετεωρολογικά δεδομένα της NOAA αποθηκεύτηκαν και ως ανεξάρτητη συλλογή `weather`.

4 Μοντελοποίηση, Σχήμα ΒΔ, έγγραφα και μεταφόρτωση δεδομένων

4.1 Στρατηγική Μοντελοποίησης και Υποδομή

Η σχεδίαση της βάσης δεδομένων στη MongoDB έγινε με κύριο γνώμονα την υψηλή διαθεσιμότητα, την οριζόντια κλιμάκωση και την ελαχιστοποίηση του κόστους ανάκτησης δεδομένων.

Καθώς ο όγκος των δεδομένων AIS αυξάνεται εκθετικά, η αρχιτεκτονική της βάσης έχει προβλέψει τη δυνατότητα οριζόντιας κλιμάκωσης μέσω της τεχνικής του sharding, παρόλο που η τρέχουσα υλοποίηση περιορίζεται σε έναν κόμβο. Για την εν δυνάμει εφαρμογή της, το πεδίο `vessel_id` ή ο συνδιασμός του με το πεδίο `start_time` ενδείκνυνται ως το ιδανικό shard key, καθώς η υψηλή πληθικότητά του (high cardinality) διασφαλίζει την ομοιόμορφη κατανομή του φόρτου (load balancing) και την αποφυγή "hotspots".

4.1.1 *Schema-less* Φύση και Ενσωμάτωση (*Embedding*).

Εκμεταλλευόμενοι τη schema-less φύση της MongoDB, ακολουθήθηκε μια στρατηγική Embedding αντί για Linking (referencing). Στόχος είναι η αρχή της "Single Read Principle". Ενσωματώνοντας τα στατικά στοιχεία και τα δεδομένα καιρού μέσα στο έγγραφο της τροχιάς, η βάση δεν χρειάζεται να εκτελέσει δαπανηρές πράξεις σύνδεσης (`Joins/lookup`) κατά την εκτέλεση των ερωτημάτων.

4.2 Κατανομή Συλλογών

Η βάση αποτελείται από τις συλλογές στατικών δεδομένων και τη συλλογή με τις τροχιές.

4.2.1 Δεδομένα ανεξάρτητης πληροφορίας.

- **Islands:** GeoJSON Polygons των νησιών της ευρύτερης περιοχής του Σαρωνικού. Τα δεδομένα αυτά επιτρέπουν τον έλεγχο χωρικών συσχετίσεων, όπως η εγγύτητα ενός πλοίου σε ακτές ή ο προσδιορισμός της διέλευσης από συγκεκριμένα στενά.
- **Harbours:** GeoJSON σημεία (Points) που περιλαμβάνουν την ονομασία και τον μοναδικό κωδικό κάθε λιμένα. Τα δεδομένα αυτά εν δυνάμει χρησιμεύουν ως σημεία αναφοράς για τον υπολογισμό των χρόνων άφιξης και αναχώρησης (ETA/ETD), καθώς και για τον προσδιορισμό των προορισμών των πλοίων.
- **Piraeus Port:** Η εγγραφή της λεπτομερούς γεωμετρικής αναπαράστασης (GeoJSON Polygon) που οριοθετεί την περιοχή δικαιοδοσίας του λιμένα Πειραιά. Η συγκεκριμένη οντότητα είναι καθοριστική για την υλοποίηση ερωτημάτων που αφορούν τη δραστηριότητα εντός του λιμένος (π.χ. υπολογισμός κυκλοφοριακής συμφόρησης ή έλεγχος ταχύτητας εντός των ορίων).
- **Regions:** GeoJSON Polygons που οριοθετούν τις διοικητικές Περιφέρειες της ελληνικής επικράτειας. Η πληροφορία αυτή επιτρέπει την ομαδοποίηση της ναυτιλιακής δραστηριότητας σε ευρύτερο γεωγραφικό επίπεδο, διευκολύνοντας την εξαγωγή στατιστικών στοιχείων ανά διοικητική ενότητα (π.χ. συνολικός αριθμός διελεύσεων πλοίων από την Περιφέρεια Αττικής).
- **Territorial Waters:** Η εγγραφή GeoJSON Polygon που οριοθετεί τα χωρικά ύδατα εντός του Σαρωνικού Κόλπου. Η οντότητα αυτή ορίζει το γεωγραφικό πλαίσιο εντός του οποίου τα πλοία στέλνουν στίγματα και αποτελεί το "φίλτρο" για την οριοθέτηση της περιοχής μελέτης.
- **Vessels:** Στατικά δεδομένα που περιλαμβάνουν την κωδική ονομασία (`vessel_id`), τον τύπο του πλοίου (`type_info`) και τη χώρα που υπάγεται (`Country`). Οι πληροφορίες αυτές είναι θεμελιώδεις για την κατηγοριοποίηση της ναυτιλιακής κίνησης και την ανάλυση της συμπεριφοράς των πλοίων ανάλογα με τον ρόλο τους (π.χ. δεξαμενόπλοια, επιβατηγά).
- **Weather:** Κάθε εγγραφή περιλαμβάνει γεωγραφική θέση σταθμού (GeoJSON Point), χρονική σήμανση, το `cell_id` και βασικές μετεωρολογικές μεταβλητές. Στη συλλογή

δημιουργήθηκαν δείκτες 2dsphere για τη θέση, δείκτης στο metadata.cell_id και χρονικός δείκτης στο πεδίο timestamp, για αποδοτικό φιλτράρισμα και ανάλυση.

4.2.2 Δεδομένα κινητικών χαρακτηριστικών. Η Trips (Bucketed Trajectories) είναι η κεντρική συλλογή της βάσης, η οποία προέκυψε από τον μετασχηματισμό των πρωτογενών δεδομένων AIS Kinematic και NOAA Weather. Κάθε έγγραφο (document) δεν αποτελεί ένα μεμονωμένο στίγμα, αλλά έναν «κάδο» (bucket) που περιέχει την τροχιά ενός βάσει του κανόνα που ορίζεται παραπάνω.

- **Πληροφορίες bucket και πλοίου:**
  - trip_id
  - vessel_id
  - country
  - shiptype
  - vessel_type_description
  - start_time: timestamp πρώτου στίγματος
  - end_time: timestamp τελευταίου στίγματος
  - point_count: Άθροισμα στιγμάτων της τροχιάς
- **trajectory:** Εμφωλευμένος πίνακας (embedded array) εντός κάθε εγγράφου Trip, ο οποίος περιλαμβάνει:
  - t: Timestamp στίγματος
  - loc: Γεωγραφικό στίγμα σε μορφή GeoJSON Point (coordinates: longitude, latitude) και cell_id (κοντινότερο σημείου μετεωρολογικού σταθμού)
  - metrics: Δυναμικά στοιχεία κίνησης (speed - Speed Over Ground, course - Course Over Ground σε μοίρες και στις 8 κατευθήνσεις, heading).
  - weather: Τις μετεωρολογικές συνθήκες: θερμοκρασία (temp_c), ταχύτητα (wind_speed) και διεύθυνση ανέμου σε μοίρες (wind_dir) και στις 8 διευθήνσεις (wind_cardinal), ατμοσφαιρική πίεση (pressure), ρηπές ανέμου (gust), σχετική υγρασία (humidity) και ορατότητα (visibility).
  - annotaions: Ενδείξεις στην μεταβολή της κίνησης του πλοίου

4.2.3 Δομή ενός *Trip* εγγράφου. Στο παρακάτω Listing παρουσιάζεται η τελική ιεραρχική δομή ενός document στη συλλογή Trips. Το παράδειγμα αναδεικνύει την εφαρμογή του **Bucketing Pattern**, όπου ένα ενιαίο έγγραφο περιλαμβάνει 273 στίγματα (point_count), καθώς και την ενσωμάτωση (**embedding**) κινητικών, μετεωρολογικών και σημασιολογικών δεδομένων.

Listing 1: Παράδειγμα αρχείου Bucketed JSON

```
{
  "_id": {
    "$oid": "69833ea96705fce15fef2c30"
  },
  "trip_id": 1,
  "vessel_id": "002351f7584dcb3b6ab87557073727eadd310a...",
  "country": "Greece",
  "shiptype": 89,
  "vessel_type_description": "Tanker, No additional information",
  "start_time": 2019-01-06T11:17:52,
  "end_time": 2019-01-06T14:49:08,
  "point_count": 273,
  "trajectory": [
      {
        "t": 2019-01-06T11:17:52,
        "loc": {
          "type": "Point",
          "coordinates": [
            23.65119,
            37.84399
          ],
          "cell_id": 38
        },
        "metrics": {
          "speed": 9,
          "course": 333,
          "heading": null,
          "course_cardinal": "NW"
        },
        "weather_data": {
          "temp_c": 30.09,
          "wind_speed": 3.26,
          "wind_dir": 259.99,
          "humidity": 70.57,
          "pressure": 1008.33,
          "visibility": 23700,
          "gust": 3.37,
          "wind_cardinal": "W"
        },
        "annotations": []
      },...
    ]
}
```

5 Ερωτήματα, κώδικας και τεχνικές αποδοτικότητας

Στην ενότητα αυτή παρουσιάζεται η υλοποίηση των ερωτημάτων στη MongoDB. Η σχεδίαση αξιοποιεί το document-oriented μοντέλο για την ελαχιστοποίηση των προσπελάσεων δίσκου (disk I/O), ενώ παράλληλα γίνεται χρήση στατικών συλλογών αναφοράς (π.χ. piraeus_port) για τον γεωχωρικό προσδιορισμό.

5.1 Στρατηγική Ευρετηρίασης (Indexing Strategy)

Για την υποστήριξη των ερωτημάτων στη βασική συλλογή trips, δημιουργήθηκαν τα ακόλουθα ευρετήρια, τα οποία επιλέχθηκαν βάσει της ανάλυσης των ερωτημάτων queries:

(1) **Γεωχωρικό Ευρετήριο (2dsphere Index):** Δημιουργήθηκε στο πεδίο trajectory.loc. Είναι απαραίτητο για το Q1 και το Q2, καθώς επιτρέπει την ταχύτατη εύρεση εγγράφων που τέμνουν γεωμετρικά σχήματα (π.χ. πολύγωνα λιμένων ή κύκλους αναζήτησης).

```
db.trips.createIndex({ "trajectory.loc": "2dsphere" })
```

(2) **Ευρετήριο στο Εμφωλευμένο Πεδίο Ανέμου (Nested Field Index):** Για την επιτάχυνση του Q2 (Crosswind), δημιουργήθηκε ευρετήριο στο πεδίο trajectory.weather_data.wind_speed. Αυτό δίνει τη δυνατότητα να εφαρμοστεί φίλτρο στις τιμές της ταχύτητας του ανέμου πριν το ακριβό $unwind, απορρίπτοντας έγγραφα με χαμηλό άνεμο πριν καν φορτωθούν στη μνήμη για γεωχωρικό έλεγχο.

```
db.trips.createIndex({ "trajectory.weather_data.wind_speed":
    1})
```

(3) **Σύνθετο Ευρετήριο (Compound Index):** Για ερωτήματα που αφορούν συγκεκριμένα πλοία και χρονικές περιόδους (π.χ. Q1), χρησιμοποιείται συνδυασμός vessel_id και start_time.

5.2 Υλοποίηση και Ανάλυση Ερωτημάτων

Ακολουθούν τα σενάρια ερωτημάτων που χρησιμοποιήθηκαν για την αξιολόγηση της βάσης.

5.2.1 Ερώτημα *Q1*: Χωροχρονική Αναζήτηση. *«Πόσα πλοία βρέθηκαν εντός της γεωμετρίας του Λιμένα Πειραιά σε συγκεκριμένο χρονικό διάστημα·»*

Το ερώτημα αυτό είναι θεμελιώδες για την απόδοση του συστήματος. Ανακτά τη γεωμετρία του λιμανιού από τη συλλογή `piraeus_port` και εκτελεί τομή (intersection) με τις τροχιές των πλοίων, φιλτράροντας ταυτόχρονα βάσει χρόνου.

Βέλτιστο Ευρετήριο: 2dsphere στο `trajectory.loc` + Ascending στο `start_time`.

Listing 2: Q1: Χωροχρονική Αναζήτηση στον Λιμένα Πειραιά

```
// 1. Fetch port geometry
var port = db.piraeus_port.findOne();

// 2. Aggregate count within time range
db.trips.aggregate([
{
    $match: {
        "start_time": {
            $gte: ISODate("2019-01-01T00:00:00Z"),
            $lt: ISODate("2019-01-05T00:00:00Z")
        },
        "trajectory.loc": {
            $geoIntersects: {$geometry: port.geometry }
        }
    }
},
{$count: "total_trips" }
])
```

5.2.2 Ερώτημα *Q2*: Ανίχνευση Επικίνδυνων Ελιγμών. *«Εντοπισμός μεγάλων πλοίων (Tankers/Passenger) που κινούνται με χαμηλή ταχύτητα ελιγμών (< 5 κόμβους) εντός του λιμανιού, ενώ δέχονται ισχυρό πλευρικό άνεμο (> 8 m/s).»*

Αυτό είναι το πιο σύνθετο ερώτημα της εργασίας. Απαιτεί υπολογισμό της γωνίας πρόσκρουσης του ανέμου στο πλοίο σε πραγματικό χρόνο εκτέλεσης. Χρησιμοποιεί τον τελεστή `$expr` για να συγκρίνει τη διαφορά μεταξύ της πορείας του πλοίου (`course`) και της διεύθυνσης του ανέμου (`wind_dir`). Αν η γωνία πρόσπτωσης βρίσκεται εντός των διαστημάτων $[45°, 135°]$ και $[225°, 315°]$ σε σχέση με την πορεία του πλοίου, ο άνεμος θεωρείται πλευρικός.

Βέλτιστο Ευρετήριο: Συνδυασμός 2dsphere (για το λιμάνι) και ευρετηρίου στο `wind_speed` για γρήγορο pruning.

Listing 3: Q2: Υπολογισμός Πλευρικού Ανέμου (Crosswind)

```
db.trips.aggregate([
{
    $match: {
        // Early Filtering using Index
        "trajectory.weather_data.wind_speed": {$gt: 8},
        "trajectory.loc": {$geoIntersects: {$geometry: port.geometry }
            },
        "vessel_type_description": {$regex: "Passenger|Tanker" }
    }
},
{$unwind: "$trajectory" },
{
    $match: {
        "trajectory.loc": {$geoIntersects: {$geometry: port.geometry }
            },
        "trajectory.metrics.speed": {$lt: 5,$gt: 0.1 },
        "trajectory.weather_data.wind_speed": {$gt: 8},
        // Crosswind Logic: |Course - WindDir| approx 90degrees
        $expr: {
            $let: {
                vars: {
                    delta: {$abs: {$subtract: ["$trajectory.metrics.
                        course", "$trajectory.weather_data.wind_dir"]
                        }}
                },
                in: {
                    $or: [
                    {$and: [{$gte: ["$$delta", 45]},{$lte: ["$$delta",
                        135]}]},
                    {$and: [{$gte: ["$$delta", 225]},{$lte: ["$$delta",
                        315]}]}
                    ]
                }
            }
        }
    }
},
{$count: "critical_events" }
])
```

6 Πειραματική αξιολόγηση: Απόδοση και Κλιμακωσιμότητα

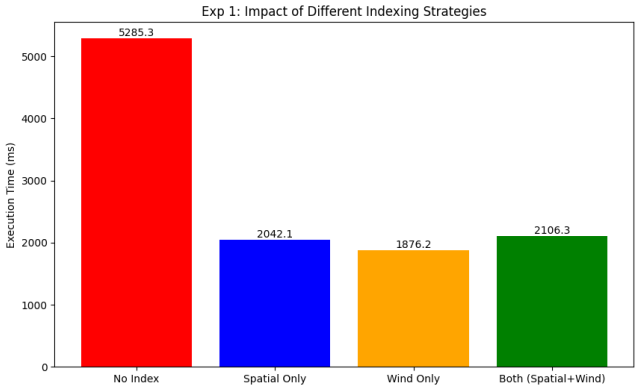
Στην παρούσα ενότητα παρουσιάζεται η πειραματική αξιολόγηση της βάσης δεδομένων. Στόχος είναι η ποσοτικοποίηση της απόδοσης των ευρετηρίων, η ανάλυση της κλιμακωσιμότητας (scalability) καθώς αυξάνεται ο όγκος των δεδομένων και η επίδραση της επιλεκτικότητας (selectivity).

Τα πειράματα εκτελέστηκαν σε τοπικό περιβάλλον με χρήση της γλώσσας προγραμματισμού Python.

6.1 Πείραμα 1: Επίδραση Ευρετηρίων (Indexing)

Εξετάστηκε ο χρόνος εκτέλεσης του σύνθετου ερωτήματος **Q2** κάτω από 4 διαφορετικά σενάρια ευρετηρίασης.

- **Χωρίς Ευρετήριο:** 5285.3 ms
- **Με Χωρικό Ευρετήριο:** 2042.1 ms
- **Με Ευρετήριο στην Ταχύτητα Ανέμου:** 1876.2 ms
- **Συνδυασμό των δύο Ευρετηρίων:** 2106.3 ms



Σχήμα 1: Σύγκριση χρόνου εκτέλεσης με διαφορετικούς συνδυασμούς ευρετηρίων.

Παρατηρήσεις: Η έλλειψη ευρετηρίων καθιστά το ερώτημα εξαιρετικά αργό (≈ 4.9s) λόγω της πλήρους σάρωσης της συλλογής. Η προσθήκη του χωρικού ευρετηρίου μειώνει τον χρόνο στο μισό (≈ 2s). Αξιοσημείωτο είναι ότι το ευρετήριο ανέμου (`wind_speed`) σημείωσε τη βέλτιστη απόδοση, πιθανώς επειδή το φίλτρο `> 8 m/s` ήταν πιο περιοριστικό στο συγκεκριμένο σύνολο δεδομένων. Ο συνδυασμός και των δύο αύξησε ελάχιστα τον χρόνο εκτέλεσης, συγκριτικά με τις εκτελέσεις με τα μεμονωμένα ευρετήρια, μείωσε όμως τον χρόνο εκτέλεσης στην μισή διάρκεια εν συγκρίσει με την περίπτωση χωρίς ευρετήριο.

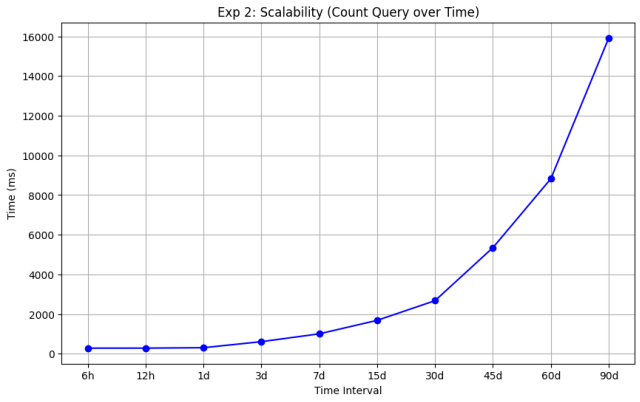
6.2 Πείραμα 2: Κλιμακωσιμότητα Όγκου (Volume Scalability)

Μελετήθηκε η συμπεριφορά του συστήματος στο ερώτημα **Q1** αυξάνοντας σταδιακά το χρονικό παράθυρο αναζήτησης από 6 ώρες έως 90 ημέρες. Τα αποτελέσματα αποτυπώνονται στο Σχήμα 2.

Ανάλυση Αποτελεσμάτων:

(1) **Γραμμική Φάση (6h - 30d):** Παρατηρείται μια ομαλή, γραμμική αύξηση του χρόνου (από 278ms σε 2.8s), γεγονός που δείχνει ότι το σύστημα κλιμακώνεται σωστά όσο τα δεδομένα χωρούν στη μνήμη RAM.

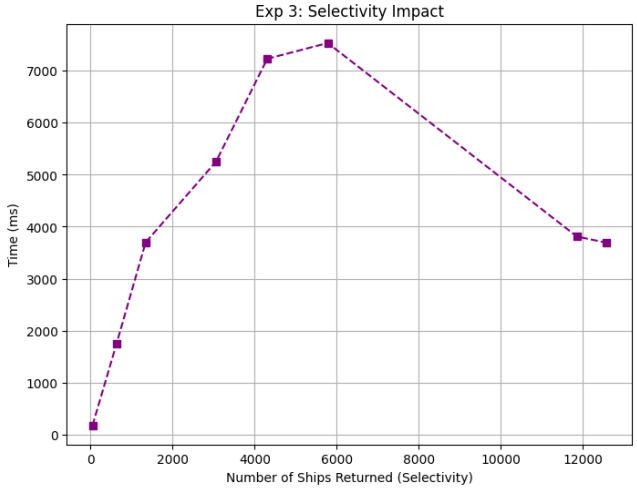
(2) **Φάση Κορεσμού (45d - 90d):** Παρατηρείται απότομη αύξηση ("knee of the curve"). Από τις 45 ημέρες (5.3s) στις 90 ημέρες (15.8s), ο χρόνος τριπλασιάζεται ενώ ο όγκος απλά διπλασιάζεται. Αυτό υποδεικνύει ότι ο όγκος των δεδομένων που υποβάλλονται σε επεξεργασία ξεπέρασε τη διαθέσιμη μνήμη του συστήματος, προκαλώντας σημαντική καθυστέρηση στην απόκριση της βάσης.



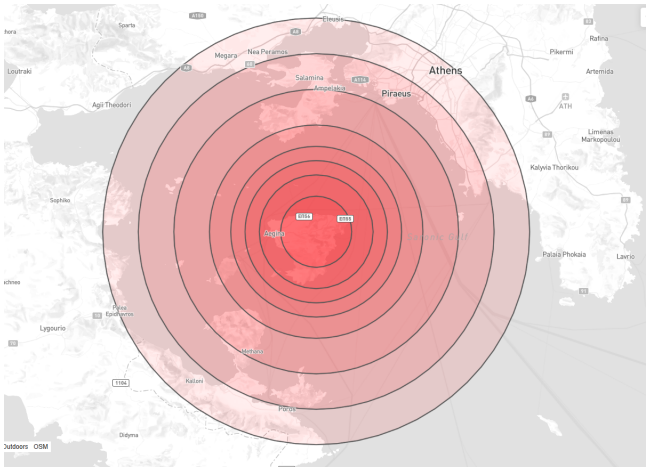
Σχήμα 2: Χρόνος εκτέλεσης σε συνάρτηση με το χρονικό εύρος (όγκος δεδομένων).

6.3 Πείραμα 3: Επιλεκτικότητα (Selectivity)

Το πείραμα αυτό μετρά τον χρόνο απόκρισης καθώς αυξάνεται η ακτίνα της περιοχής αναζήτησης (προσεγγιζόμενη με κυκλικά πολύγωνα, Σχήμα 4), επιστρέφοντας από ≈ 200 έως 12800 **εγγραφές τμημάτων τροχιών (trajectory segments)**.



Σχήμα 3: Επίδραση της επιλεκτικότητας (πλήθος ανακτώμενων εγγραφών) στον χρόνο απόκρισης.



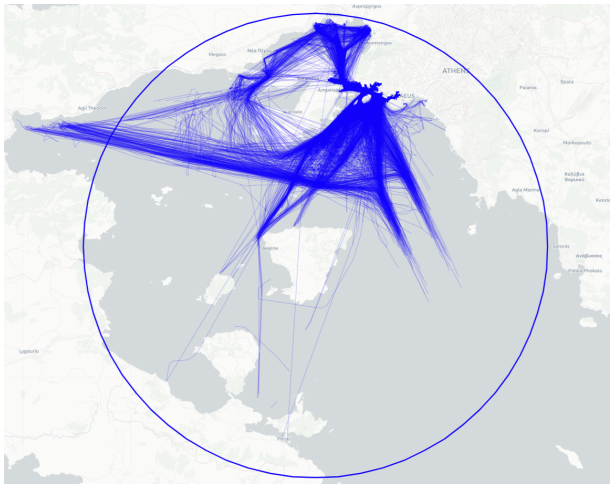
Σχήμα 4: Τα διαφορετικά χωρικά εύρη (ακτίνες) του πειράματος.

Παρατηρήσεις: Η ανάλυση των αποτελεσμάτων αποκαλύπτει τρία διακριτά φαινόμενα κατά τη διαδικασία ανάκτησης, τα οποία εξελίσσονται καθώς αυξάνεται ο όγκος των δεδομένων:

(1) **Αρχική Γραμμική Αύξηση:** Στα πρώτα στάδια του πειράματος (έως 4.100 εγγραφές), παρατηρείται μια φυσιολογική, σχεδόν γραμμική αύξηση του χρόνου απόκρισης. Καθώς το πλήθος των ανακτώμενων εγγραφών μεγαλώνει, το σύστημα απαιτεί αναλογικά περισσότερο χρόνο για την ανάγνωση και μεταφορά των δεδομένων.

(2) **Φαινόμενο Χωρικής Διασποράς (Spatial Dispersion):** Κατά τη μετάβαση από τις 4.100 στις περίπου 5.800 εγγραφές, παρατηρείται μια δυσανάλογη αύξηση του χρόνου (από 7.2s σε 7.4s), παρόλο που προστέθηκαν μόλις 1.700 νέες

Σχήμα 5: Οπτικοποίηση της χωρικής διασποράς.



Σχήμα 6: Το σύνολο των τροχιών (Cache Hit).

εγγραφές. Το γεγονός αυτό οφείλεται στη γεωγραφική κατανομή των νέων δεδομένων (Σχήμα 5). Οι επιπλέον εγγραφές βρίσκονται στην περιφέρεια της περιοχής αναζήτησης (δακτύλιος) και είναι έντονα διασκορπισμένες. Αυτή η διασπορά αναγκάζει τον δίσκο να εκτελέσει πολλαπλές **τυχαίες προσπελάσεις (Random I/O)** [5] για να τις ανακτήσει, καθώς δεν βρίσκονται σε συνεχόμενα μπλοκ μνήμης, σε αντίθεση με τις αρχικές 4.100 που ήταν ήδη αποθηκευμένες στην προσωρινή μνήμη (cache).

(3) **Επίδραση της Προσωρινής Μνήμης (Cache Warming):** Στο τελικό στάδιο, παρόλο που το πλήθος των τροχιών διπλασιάζεται (από 5.800 σε 12.000+), ο χρόνος εκτέλεσης μειώνεται δραματικά από τα 7.4s στα 3.8s. Αυτό συμβαίνει διότι τα δεδομένα του μεγαλύτερου κύκλου (Σχήμα 6) είχαν ήδη φορτωθεί στην προσωρινή μνήμη κατά την εκτέλεση του προηγούμενου, χρονοβόρου ερωτήματος. Έτσι, η βάση εξυπηρέτησε το μεγαλύτερο ερώτημα ανακτώντας τα δεδομένα απευθείας από τη μνήμη RAM, αποφεύγοντας την αργή προσπέλαση στον δίσκο [6].

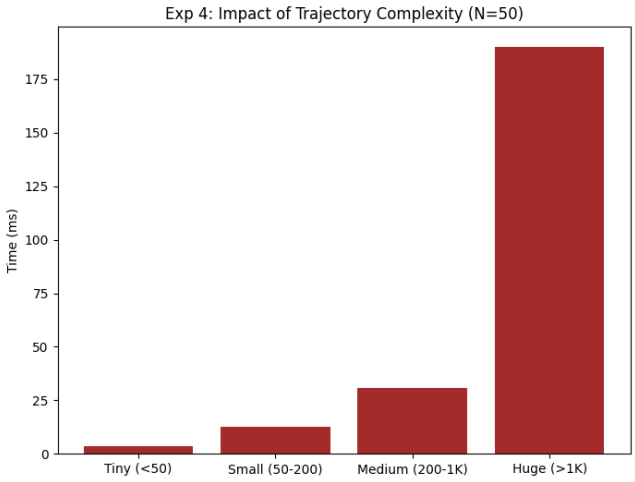
6.4 Πείραμα 4: Πολυπλοκότητα Δομής (Trajectory Complexity)

Συγκρίθηκε ο χρόνος επεξεργασίας για σταθερό αριθμό εγγράφων ($N = 50$ trips), τα οποία όμως διαφέρουν στο πλήθος των στιγμάτων (`point_count`).

Παρατηρήσεις: Τα αποτελέσματα είναι αποκαλυπτικά για το κόστος των εμφωλευμένων πινάκων.

- **Tiny (<50 points):** 2.91 ms
- **Huge (>1000 points):** 204.11 ms

Η επεξεργασία τροχιών με πολλά στίγματα είναι σημαντικά πιο αργή, κυρίως λόγω της εντολής `$unwind` στο Aggregation Pipeline [1], η οποία αναγκάζει τη βάση να δημιουργήσει χιλιάδες προσωρινά έγγραφα στη μνήμη. Αυτό αποδεικνύει ότι στην απόδοση των NoSQL βάσεων ρόλο δεν παίζει μόνο το πλήθος των εγγράφων, αλλά και η δομική τους πολυπλοκότητα.



Σχήμα 7: Χρόνος επεξεργασίας ανάλογα με το μέγεθος της τροχιάς.

7 Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Στην εργασία αυτή σχεδιάστηκε και υλοποιήθηκε μια μη-σχεσιακή βάση δεδομένων σε MongoDB για διαχείριση *χωροχρονικών* ναυτιλιακών δεδομένων μεγάλης κλίμακας, αξιοποιώντας υποσύνολο του Piraeus AIS Dataset [10, 11]. Ο πυρήνας της υλοποίησης είναι η συλλογή `trips`, όπου οι τροχιές δεν αποθηκεύονται ως μεμονωμένα στίγματα αλλά ως **bucketed trajectories** (έγγραφα με embedded πίνακα `trajectory`). Η επιλογή αυτή ευθυγραμμίζεται με το **Bucket Pattern** και εξυπηρετεί την αρχή **single-read** για τα πιο συχνά ερωτήματα, μειώνοντας την ανάγκη για "joins" (`$lookup`) κατά την ανάκτηση [3, 8].

Σε επίπεδο ερωτημάτων, υλοποιήθηκαν χωροχρονικά φίλτρα (π.χ. εντός γεωμετρίας λιμένα σε χρονικό διάστημα) και ένα πιο

σύνθετο αναλυτικό ερώτημα ανίχνευσης επικίνδυνων ελιγμών υπό ισχυρό πλευρικό άνεμο (*crosswind*). Τα πειράματα έδειξαν ότι η κατάλληλη ευρετηρίαση (γεωχωρικό 2dsphere και ευρετήρια σε επιλεκτικά scalar πεδία όπως `wind_speed`) μειώνει σημαντικά τον χρόνο εκτέλεσης.

7.2 Τι μάθαμε

Από τη διαδικασία προεπεξεργασίας, μοντελοποίησης και αξιολόγησης προέκυψαν τα εξής:

- **Το σχήμα πρέπει να «χτίζεται πάνω» στα ερωτήματα.** Η επιλογή embedding (στατικά στοιχεία + καιρός + σημεία τροχιάς στο ίδιο document) επιταχύνει αναλυτικά ερωτήματα που απαιτούν πολλά πεδία μαζί, αλλά αυξάνει το μέγεθος του εγγράφου και κάνει την επεξεργασία μεγάλων πινάκων πιο απαιτητική [1, 3].
- **Η επιλεκτικότητα φίλτρων μπορεί να είναι πιο κρίσιμη από το «είδος» του ευρετηρίου.** Σε περιπτώσεις όπου ένα scalar φίλτρο (π.χ. `wind_speed > 8`) είναι πολύ επιλεκτικό, μπορεί να προσφέρει ισχυρό *pruning* πριν από ακριβά στάδια όπως `$unwind` [1].
- **Η δομική πολυπλοκότητα επηρεάζει την απόδοση όσο και το πλήθος εγγράφων.** Τα πολύ μεγάλα `trajectory` arrays αυξάνουν το κόστος επεξεργασίας pipelines (ιδίως με `$unwind`), άρα χρειάζεται ισορροπία μεταξύ «μεγάλων buckets» και πρακτικής επεξεργασίας [1, 8].
- **Η μετατροπή ωμών μετρήσεων σε κατηγορίες βελτιώνει την αναλυσιμότητα.** Η χαρτογράφηση πορείας/ανέμου σε κατευθύνσεις (*cardinal directions*) υποστηρίζει πιο ερμηνεύσιμα ερωτήματα και στατιστικά, και είναι σύμφωνη με πρακτικές τυποποίησης μετεωρολογικών παρατηρήσεων [2, 12, 13].

7.3 Τι δεν υλοποιήθηκε και πώς θα είχε υλοποιηθεί

Παρότι η σχεδίαση «προβλέπει» κλιμάκωση και πιο εκτεταμένα analytics, ορισμένα στοιχεία δεν υλοποιήθηκαν πλήρως:

- **Πραγματικό sharding/multi-node πείραμα.** Η αξιολόγηση πραγματοποιήθηκε σε μονό κόμβο. Για πλήρη μελέτη κλιμακωσιμότητας θα απαιτούνταν sharded cluster (config servers, mongos, πολλαπλά shards) και επανάληψη των πειραμάτων με αυξανόμενους κόμβους/όγκο, εξετάζοντας ισορροπία φόρτου και hotspots [3, 4].
- **Πλουσιότερα ερωτήματα πάνω στις synopses/annotations.** Παρότι ενσωματώθηκαν annotations, δεν παρουσιάστηκε ξεχωριστό ερώτημα που να αξιοποιεί συστηματικά «γεγονότα» τροχιάς (π.χ. στάση, απότομη αλλαγή πορείας) για εξαγωγή συμπερασμάτων. Αυτό θα υλοποιούνταν ως pipelines με `$filter` στο `trajectory` και ομαδοποίηση/στατιστικά σε επίπεδο trip ή vessel, σε λογική *event recognition* [7, 14].

7.4 Μελλοντικές επεκτάσεις

Με βάση τα ευρήματα, προτείνονται οι ακόλουθες επεκτάσεις:

- **Κατανεμημένη υλοποίηση και αξιολόγηση.** Εφαρμογή sharding με κατάλληλο shard key (π.χ. `vessel_id` ή σύνθετο (`vessel_id, start_time`)) και επανάληψη των πειραμάτων ώστε να αξιολογηθεί οριζόντια κλιμάκωση και συμπεριφορά υπό αυξανόμενο φόρτο [3, 4].
- **Βελτιστοποίηση bucketing για πολύ μεγάλες τροχιές.** Χρήση δυναμικού κανόνα δημιουργίας buckets (π.χ. όριο μέγιστου πλήθους σημείων ή μέγιστης διάρκειας), ώστε να αποφεύγονται «huge» έγγραφα που επιβαρύνουν pipelines και μνήμη [8].
- **Προχωρημένη εξόρυξη γνώσης σε τροχιές.** Εξαγωγή αγκυροβολιών, co-movement και μοτίβων κίνησης (π.χ. ροές σε διαύλους, συχνές διαδρομές), αξιοποιώντας τεχνικές *trajectory mining* και σχετικές προσεγγίσεις σε streaming/συμβάντα [7, 9, 14].
- **Ακριβέστερη συσχέτιση καιρού (spatio-temporal enrichment).** Μετάβαση από «πλησιέστερο σταθμό + χρονική στρογγυλοποίηση» σε παρεμβολή (χωρική/χρονική), όπου αυτό είναι εφικτό, ώστε οι αναλύσεις να είναι πιο αξιόπιστες σε τοπικές μεταβολές.
- **Αναπαραγωγιμότητα και έλεγχοι ποιότητας.** Αυτοματοποίηση ETL/πειραμάτων (pipelines εκτέλεσης, καταγραφή παραμέτρων, επαναλήψεις), και αξιοποίηση μηχανισμών validation/monitoring που προσφέρει το οικοσύστημα της MongoDB [1].

Συνολικά, η εργασία ανέδειξε ότι μια document-oriented αναπαράσταση με προσεκτικό bucketing και κατάλληλη ευρετηρίαση μπορεί να υποστηρίξει αποδοτικά χωροχρονικά ερωτήματα σε ναυτιλιακά δεδομένα μεγάλης κλίμακας, ενώ οι πιο κρίσιμες μελλοντικές βελτιώσεις αφορούν την κατανεμημένη κλιμάκωση και τη μείωση του κόστους επεξεργασίας σε σύνθετα aggregation pipelines.

Βιβλιογραφία

[1] [n. d.]. MongoDB Manual. https://www.mongodb.com/docs/.

[2] Nathaniel Bowditch. 2019. *The American Practical Navigator: An Epitome of Navigation (Pub. No. 9).* National Geospatial-Intelligence Agency (NGA), Springfield, VA, USA.

[3] Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow. 2019. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage.* O'Reilly Media.

[4] Ali Davoudian, Leyli Chen, and Mengchi Liu. 2018. A survey on NoSQL stores. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 1–43.

[5] MongoDB Inc. 2024. *Analyze Query Performance - MongoDB Manual.* https://www.mongodb.com/docs/manual/tutorial/analyze-query-performance/ Accessed: 2024-05-20.

[6] MongoDB Inc. 2024. *WiredTiger Storage Engine - MongoDB Manual.* https://www.mongodb.com/docs/manual/core/wiredtiger/ Accessed: 2024-05-20.

[7] Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Marios Vodas, Nikos Pelekis, and Yannis Theodoridis. 2017. Online event recognition from moving vessel trajectories. *Geoinformatica* 21, 2 (2017), 389–427. doi:10.1007/s10707-016-0266-x

[8] MongoDB Editorial Team. 2020. *Time Series Data Modeling: The Bucket Pattern.* https://www.mongodb.com/blog/post/time-series-data-modeling-the-bucket-pattern

[9] Andreas Tritsarolis et al. 2021. MaSEC: Discovering anchorages and co-movement patterns on streaming vessel trajectories. In *Proceedings of the 17th International Symposium on Spatial and Temporal Data (SSTD '21).* ACM. doi:10.1145/3469830.3470909

[10] Andreas Tritsarolis, Yannis Kontoulis, and Yannis Theodoridis. 2022. Corrigendum to "The Piraeus AIS dataset for large-scale maritime data analytics" [Data in Brief, 40 (2022), 107782]. *Data in Brief* 41 (2022), 107940. doi:10.1016/j.dib.2022.107940

[11] Andreas Tritsarolis, Yannis Kontoulis, and Yannis Theodoridis. 2022. The Piraeus AIS dataset for large-scale maritime data analytics. *Data in Brief* 40 (2022), 107782. doi:10.1016/j.dib.2021.107782

[12] World Meteorological Organization. 2018. *Guide to Marine Meteorological Services (WMO-No. 471).* WMO, Geneva, Switzerland.

[13] World Meteorological Organization. 2021. *Guide to Instruments and Methods of Observation (WMO-No. 8)* (2021 update ed.). WMO, Geneva, Switzerland.

[14] Zheng Yu. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 1–41.