Πανεπιστήμιο Πειραιώς
Σχολή Τεχνολογιών Πληροφορικής και Επικοινωνιών
Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Συγκριτική Ανάλυση Εργαλείων Για Μεγάλα Σύνολα Δεδομένων Τροχιάς** **Comparative analysis of tools for big trajectory datasets**
Ονοματεπώνυμο Φοιτητή	**Δέδες Χρήστος**
Πατρώνυμο	**Απόστολος**
Αριθμός Μητρώου	**ΜΠΠΛ18016**
Επιβλέπων	**Πελέκης Νικόλαος, Αναπλ. Καθηγητής**

Ημερομηνία Παράδοσης **Μάρτιος 2023**

Τριμελής Εξεταστική Επιτροπή

Νικόλαος Πελέκης Αναπληρωτής Καθηγητής Τμήμα Στατιστικής και Ασφαλιστικής Επιστήμης	Άγγελος Πικράκης Επίκουρος Καθηγητής Τμήμα Πληροφορικής	Ιωάννης Θεοδωρίδης Καθηγητής Τμήμα Πληροφορικής

Περίληψη

Με την ολοένα αυξανόμενη χρήση συσκευών GPS που ενσωματώνονται σε μέσα μεταφοράς και συσκευές, όπως αυτοκίνητα, πλοία, αεροπλάνα και κινητά τηλέφωνα, διατίθενται όλο και μεγαλύτεροι όγκοι δεδομένων τροχιάς που καταγράφουν τις κινήσεις οχημάτων και ανθρώπων, κάτι που είναι χρήσιμο σε πολλούς τομείς εφαρμογών, όπως οι μεταφορές, η διαχείριση της κυκλοφορίας (εναέριας και επίγειας) και οι υπηρεσίες που βασίζονται στην τοποθεσία. Ως αποτέλεσμα, έχουν προταθεί και έχουν προκύψει πολλά συστήματα διαχείρισης και ανάλυσης δεδομένων τροχιάς.

Πολλοί όμως από τους υπάρχοντες αλγορίθμους επικεντρώνονται στη βελτιστοποίηση των τεχνικών αποθήκευσης και επεξεργασίας των μεγάλων δεδομένων σε ένα μόνο μηχάνημα. Ωστόσο, λόγω της τεράστιας αύξησης του όγκου των δεδομένων, ο αριθμός των τροχιών υπερβαίνει την ικανότητα αποθήκευσης και επεξεργασίας ενός μόνο μηχανήματος και απαιτεί ανάλυση τροχιών μεγάλης κλίμακας σε κατανεμημένα περιβάλλοντα.

Με κίνητρο την ανάγκη εύρεσης κατάλληλων συστημάτων, τόσο απλών όσο και κατανεμημένων, προχωράμε στην παράθεση και τη σύγκριση, μέσω πινάκων, ορισμένων πρόσφατων συστημάτων επεξεργασίας και ανάλυσης μεγάλων δεδομένων τροχιάς. Ιδιαίτερη έμφαση δίνεται σε τεχνικές όπως η ευρετηρίαση, η διαμέριση και η επεξεργασία ερωτημάτων. Με αυτόν τον τρόπο, αυτή η εργασία αποτελεί έναν χρήσιμο συγκεντρωτικό οδηγό και μια συνεπτυγμένη ανάλυση των ήδη υπαρχόντων συστημάτων που υπάρχουν στη βιβλιογραφία.

Abstract

With the accelerating use of GPS devices embedded in vehicles and devices such as cars, ships, airplanes, and mobile phones, ever-increasing amounts of trajectory data are available that record the movements of vehicles and people, which is useful in many application areas, such as transportation, traffic management (air and ground) and location-based services. As a result, many trajectory data management and analysis systems have been proposed and emerged.

Nonetheless, many of the existing algorithms focus on optimizing the storage and processing techniques of big data on a single machine. However, due to the huge increase in data volume, the number of trajectories exceeds the storage and processing capacity of a single machine and requires large-scale trajectory analysis in distributed environments.

Motivated by the need to find suitable systems, both single and distributed, we proceed to list and compare, through tables, some modern and recent systems for processing and analyzing large trajectory data. Special emphasis is placed on techniques such as indexing, partitioning, and query processing. In this way, this paper provides a useful and concentrated guide and a comprehensive analysis of already existing systems present in the literature.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Εισαγωγή

Δεν υπάρχει καμία αμφιβολία πλέον, ότι ζούμε στην εποχή της πληροφορίας, όπου ολοένα και περισσότερα κομμάτια της ανθρώπινης δραστηριότητας εκτίθενται στο διαδίκτυο. Αυτό έχει ως συνέπεια να παράγεται καθημερινά ένας τεράστιος όγκος δεδομένων, η εκμετάλλευση του οποίου αποτελεί μια πρόκληση για την τεχνολογία της πληροφορικής, τις επιχειρήσεις, το κράτος, την επιστήμη και τον σύγχρονο άνθρωπο εν γένει.

Με την ευρεία διαθεσιμότητα συσκευών με δυνατότητα GPS και την αυξανόμενη χρήση των φορητών υπολογιστικών υπηρεσιών, τα δεδομένα τροχιάς από διάφορα κινούμενα αντικείμενα (π.χ. άνθρωποι, μέσα μεταφοράς και ζώα) αυξάνονται συνεχώς με υψηλή ταχύτητα [1]. Τα δεδομένα τροχιάς με τα αναλυτικά στοιχεία τους μπορεί να ωφελήσουν πολλές εφαρμογές της καθημερινής ζωής, συμπεριλαμβανομένων των υπολογισμών για τον αστικό τρόπο ζωής [2], των μεταφορών [3], των μελετών συμπεριφοράς ζώων [4] και της ασφάλειας [5], για να αναφέρουμε μόνο μερικές.

Το σύνολο των δεδομένων που παράγεται από τον σύγχρονο άνθρωπο μας βάζει αναμφίβολα στην εποχή των δεδομένων μεγάλης κλίμακας (Big data). Σε αυτή την εποχή, ο αυξανόμενος όγκος των δεδομένων, η απαίτηση για γρήγορη επεξεργασία τους και η μεγάλη ποικιλία στον τύπο, την δομή και την προέλευση των δεδομένων δημιουργούν πολλές και νέες τεχνολογικές προκλήσεις στις οποίες και δεν μπορούν να ανταπεξέλθουν τα παραδοσιακά συστήματα διαχείρισης δεδομένων. Η ανάγκη για αποτελεσματική αποθήκευση και επεξεργασία των δεδομένων οδηγεί στην εμφάνιση και ανάπτυξη νέων τεχνολογιών για μεγάλα δεδομένα οι οποίες χωρίζονται σε τρεις γενικές κατηγορίες:

1. Hadoop [6] και συναφείς τεχνολογίες (δηλαδή το οικοσύστημα που έχει αναπτυχθεί γύρω από το Hadoop και περιλαμβάνει γλώσσες υψηλού επιπέδου πάνω από το Hadoop αλλά και SQL στο Hadoop).

2. Συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων και μη σχεσιακές βάσεις δεδομένων (NoSQL) που χρησιμοποιούνται ευρέως από διαδικτυακές εφαρμογές αλλά και για ανάλυση δεδομένων.

3. Εξειδικευμένα συστήματα για την επεξεργασία δεδομένων διασυνδεμένων γράφων, συνεχών ροών δεδομένων (streams) και σύνθετων επιστημονικών δεδομένων.

Σε αυτό το πλαίσιο, η σύγκριση και αξιολόγηση των χαρακτηριστικών των συστημάτων επεξεργασίας δεδομένων τροχιάς μεγάλης κλίμακας αποκτά καθοριστική σημασία. Και αυτό γιατί έτσι αξιολογούνται οι υπάρχουσες τεχνολογίες μεγάλων δεδομένων και δίνεται το έναυσμα για περαιτέρω βελτίωση της απόδοσης τους. Ενώ, επιπλέον, εφαρμογές που επεξεργάζονται μεγάλα δεδομένα διευκολύνονται στο να εκτιμήσουν τις ανάγκες τους.

Το Hadoop κυριαρχεί στην περιοχή της επεξεργασίας δεδομένων σε κατανεμημένα συστήματα (distributed systems). Επικεντρώνεται κυρίως στο να επιτύχει κλιμακωτή απόδοση σε μεγάλα διασυνδεδεμένα υπολογιστικά συστήματα (cluster) με πολλούς κόμβους (nodes). Η υπολογιστική μηχανή του Hadoop εκτελεί υπολογισμούς σε δύο φάσεις, την συλλογή (Map) και την ελαχιστοποίηση (Reduce), ενώ σε κάθε φάση από ένα ζεύγος κλειδιού-τιμής παράγεται ένα ζεύγος με το αποτέλεσμα. Επειδή σε κάθε υπολογιστικό στάδιο κρατούνται στην μνήμη οι ενδιάμεσοι υπολογισμοί, δημιουργείται πρόβλημα στην απόδοση του Hadoop. Για αυτό τον λόγο έχουν εμφανιστεί και άλλα συστήματα που στοχεύουν στο να ξεπεράσουν αυτούς τους περιορισμούς στην απόδοση.

Ένα παράδειγμα είναι το Spark [7], το οποίο έχει δανειστεί από το Hadoop την επεξεργασία δεδομένων στην κύρια μνήμη αλλά χρησιμοποιεί υπολογιστικά φθηνότερες διαδικασίες ανακατάταξης των δεδομένων που επεξεργάζεται. Η βασική ιδέα γύρω από το Spark είναι το Resilient Distributed Dataset (RDD) που αναπαριστά μια συλλογή δεδομένων που είναι κατανεμημένη στους κόμβους ενός υπολογιστικού συμπλέγματος (cluster).

Επιπλέον, ένα παράδειγμα επεξεργασίας δεδομένων μεγάλης κλίμακας που κερδίζει έντονο ενδιαφέρον τελευταία λόγω της ανάπτυξης αντίστοιχων εφαρμογών είναι η επεξεργασία ροών δεδομένων. Το Apache Storm [8] είναι ένα τέτοιο παράδειγμα και έχει δώσει το έναυσμα για δημιουργία μηχανών επεξεργασίας ροών δεδομένων ανθεκτικών σε λάθη. Άλλο ένα παράδειγμα είναι το Apache Flink [9], που έχει την δυνατότητα να επεξεργάζεται ροές δεδομένων που προέρχονται από διαφορετικές πηγές χρησιμοποιώντας έναν σχεσιακό πίνακα ο οποίος και γεμίζει συνεχόμενα με δεδομένα σε

πραγματικό χρόνο. Τέλος, άλλο ένα σύστημα που ανήκει σε αυτήν την κατηγορία είναι το Apache Kafka [10], το οποίο βασίζεται σε ένα αρχείο καταγραφής, που επιτρέπει στους χρήστες να διαβάζουν και να γράφουν σε οποιοδήποτε αριθμό συστημάτων ή εφαρμογών, σε πραγματικό χρόνο. Παρέχει, δηλαδή, μια ενιαία πλατφόρμα υψηλής απόδοσης και χαμηλής καθυστέρησης για το χειρισμό ροών δεδομένων σε πραγματικό χρόνο.

Αυτή η εργασία παρέχει μια επισκόπηση συστημάτων τελευταίας τεχνολογίας στην αποθήκευση και επεξεργασία μεγάλων δεδομένων, εστιάζοντας κυρίως σε επεκτάσιμες λύσεις για δεδομένα τροχιάς. Παρά την πλούσια βιβλιογραφία για τη διαχείριση χωροχρονικών δεδομένων και δεδομένων κινητικότητας, μόνο ένας περιορισμένος αριθμός ερευνητικών πρωτοτύπων επιχειρεί να αντιμετωπίσει τις ιδιαιτερότητες που εμφανίζουν τα μεγάλα δεδομένα τροχιάς. Αν και η πλειονότητα των ανεπτυγμένων πρωτοτύπων αποτελούν επεκτάσεις του Hadoop ή του Spark προκειμένου να είναι εφαρμόσιμα για χωρικά δεδομένα, σε αυτήν την εργασία καλύπτουμε προσεγγίσεις μεγάλων δεδομένων που χειρίζονται επίσης τη χρονική διάσταση.

Στη συνέχεια η δομή της εργασίας είναι η εξής: το κεφάλαιο 2 περιλαμβάνει ένα θεωρητικό υπόβαθρο για τα μεγάλα δεδομένα. Στο κεφάλαιο 3 εξετάζουμε, ορισμένα υπάρχοντα συστήματα που προσεγγίζουν τη διαχείριση μεγάλων δεδομένων τροχιάς, αναλύουμε τη δομή και τον τρόπο λειτουργίας τους. Στο κεφάλαιο 4, παρουσιάζουμε τις υποκείμενες τεχνικές που χρησιμοποιούνται για τη διαμέριση, την ευρετηρίαση και την επεξεργασία των ερωτημάτων, καθώς και τον τρόπο με τον οποίο εφαρμόζονται στα συστήματα που περιγράφουμε στο προηγούμενο κεφάλαιο. Στο κεφάλαιο 5 γίνεται μια ταξινόμηση των συστημάτων σε πίνακες, βάσει των χαρακτηριστικών τους και των τεχνικών επεξεργασίας που χρησιμοποιούν και πραγματοποιείται μια σύγκριση βασισμένη σε ορισμένες παραμέτρους. Τέλος, το κεφάλαιο 6, περιλαμβάνει ορισμένα συμπεράσματα που προκύπτουν από την παράθεση των συστημάτων της βιβλιογραφίας, καθώς και ορισμένες προτάσεις για μελλοντική έρευνα.

2. Θεωρητικό και Τεχνικό Υπόβαθρο

Σε αυτή την ενότητα παρέχουμε ορισμένες έννοιες τόσο θεωρητικές όσο και τεχνικές, που σχετίζονται με τα είδη των μεγάλων δεδομένων και με τους τύπους των ερωτημάτων τροχιάς.

Δεδομένα κινητικότητας (mobility data)

Ο όρος "δεδομένα κινητικότητας" περιγράφει πληροφορίες που δημιουργούνται από δραστηριότητα, συμβάντα ή συναλλαγές χρησιμοποιώντας συσκευές ή υπηρεσίες κινητικότητας με ψηφιακή δυνατότητα. Αυτά τα δεδομένα καταγράφονται συχνά ως μια σειρά σημείων με γεωγραφικό πλάτος και μήκος που συλλέγονται σε τακτά χρονικά διαστήματα από συσκευές όπως smartphone, κοινόχρηστα οχήματα μικροκινητικότητας (κοινόχρηστα ποδήλατα, ηλεκτρονικά ποδήλατα, σκούτερ κ.λπ.), υπολογιστές ενσωματωμένοι επί οχημάτων, αισθητήρες, δίκτυα ασφαλείας, ραντάρ ή συστήματα πλοήγησης που βασίζονται σε εφαρμογές (π.χ. GoogleMaps κ.λπ.). Τα δεδομένα κινητικότητας έχουν συχνά ένα χρονικό στοιχείο, που προσδίδει χρόνο καθώς και τοποθεσία σε κάθε σημείο. [11]

Χωρικά Δεδομένα

Τα χωρικά δεδομένα (Spatial Data) αναφέρονται σε όλους εκείνους τους τύπους δεδομένων που μπορούν να περιγράψουν μία γεωγραφική τοποθεσία στο επίπεδο ή στο χώρο. Συνήθως αποτυπώνονται ως συντεταγμένες σε ένα γεωγραφικό σύστημα συντεταγμένων και με την βοήθεια ενός GIS (Geographic Information System) εργαλείου μπορούν να επεξεργαστούν, να αναλυθούν και να αναπαρασταθούν σε μορφές κατανοητές για τους ανθρώπους. Χωρικά δεδομένα παράγονται καθημερινά από τον κάθε άνθρωπο που μεταφέρει ένα κινητό τηλέφωνο, είτε μέσω του GPS, είτε μέσω των σημάτων που στέλνονται στις κυψέλες των εταιριών κινητής τηλεφωνίας. Ανά πάσα στιγμή κάποιος μπορεί να βρει το στίγμα του σε έναν χάρτη και να κατευθυνθεί οπουδήποτε θελήσει. Όλα αυτά με την βοήθεια της ανάλυσης των χωρικών δεδομένων που έχουν συλλεχθεί. Τέτοιες μορφές δεδομένων μπορούν να χρησιμοποιηθούν και από οργανισμούς για κρίσιμους σκοπούς.

Χρονικά Δεδομένα

Τα χρονικά δεδομένα είναι δεδομένα που αντιπροσωπεύουν μια κατάσταση στο χρόνο, όπως για παράδειγμα οι συνολικές βροχοπτώσεις σε μια περιοχή, σε μια συγκεκριμένη ημερομηνία. Το κύριο χαρακτηριστικό των χρονικών δεδομένων είναι ο χρόνος, ο οποίος αποτυπώνεται με την μορφή ώρας ή ημερομηνίας. Μερικές φορές είναι πιο εύστοχο να παρατηρηθεί ένα φαινόμενο στο πως εξελίσσεται κατά την πάροδο του χρόνου. Τα χρονικά δεδομένα συλλέγονται για την ανάλυση των καιρικών προτύπων και άλλων περιβαλλοντικών μεταβλητών, την παρακολούθηση κυκλοφοριακών συνθήκων, τη μελέτη δημογραφικών τάσεων κ.λπ. Αυτά τα δεδομένα προέρχονται από πολλές πηγές που κυμαίνονται από τη χειροκίνητη εισαγωγή δεδομένων έως τα δεδομένα που συλλέγονται με χρήση αισθητήρων παρατήρησης ή δημιουργούνται από μοντέλα προσομοίωσης.

Η ανάλυση των χρονικών δεδομένων είναι μία μεγάλη πρόκληση καθώς στις περισσότερες εφαρμογές πρέπει να αναλύονται σε πραγματικό χρόνο. Όπως προαναφέρθηκε μερικές φορές η ροή της πληροφορία που συλλέγεται γίνεται αρκετά μεγαλύτερη από την ταχύτητα επεξεργασίας των συστημάτων που την αναλύουν.

Τέλος, είναι άξιο αναφοράς ότι υπάρχουν αφοσιωμένες βάσεις δεδομένων σε χρονικά δεδομένα.

Χώρο-Χρονικά Δεδομένα

Τα χώρο-χρονικά δεδομένα είναι ο συνδυασμός των χρονικών και χωρικών δεδομένων, δηλαδή μπορούν να αναπαρασταθούν σε 4 διαστάσεις (μήκος, πλάτος, ύψος, χρόνος). Συνήθως αποτυπώνουν παρατηρήσεις αντικειμένων με αυτά τα χαρακτηριστικά, μία γεωγραφική απεικόνιση του αντικειμένου σε ένα GIS και ο χρόνος παρατήρησης αυτού. Ένα παράδειγμα μπορεί να είναι οι αποβιβάσεις πελατών ταξί σε μία πόλη όπως μελετήθηκαν στα πλαίσια του διαγωνισμού GISCup 2016. [12]

Δεδομένα τροχιάς

Άλλος ένας παρεμφερής αλλά πιο σύνθετος όρος που χρησιμοποιείται, ο οποίος θα απασχολήσει και τη συγκεκριμένη εργασία, είναι οι τροχιές (Trajectories). Αν και έχει γίνει μελέτη από πολλούς ερευνητές για δεδομένα τροχιών, υπάρχουν ακόμα αρκετές δυσκολίες στην αποτύπωση, επεξεργασία και ανάλυση αυτών.

Υπάρχουν διαφορετικές προσεγγίσεις ως προς την αποτύπωση των δεδομένων αναλόγως την τεχνολογία που χρησιμοποιείται και τα αντικείμενα τα οποία παρατηρούνται. Και αυτό ισχύει αν αναλογιστεί κανείς την ευρύτητα που μπορούν να έχουν τα χώρο-χρονικά δεδομένα. Σίγουρα χρειάζονται οι συντεταγμένες στις οποίες παρατηρήθηκε το αντικείμενο και ο χρόνος παρατήρησης για να θεωρηθεί κάποιο σύνολο από δεδομένα, χώρο-χρονικό. Από κει και έπειτα μπορεί κάποιος να θεωρήσει οποιαδήποτε άλλη μεταβλητή η στοιχείο που μπορεί να θέλει να μελετήσει όπως ταχύτητα, κατεύθυνση και άλλα, τα οποία περιγράφουν την κατάσταση ενός αντικειμένου εκείνη την χρονική στιγμή.

Άρα αυτά τα δεδομένα μπορούν να εκφραστούν ως μία ακολουθία σημείων/παρατηρήσεων ενός κινούμενου αντικειμένου τα οποία συνδέονται μεταξύ τους με κάποιο μοναδικό αναγνωριστικό και διαθέτουν πληροφορίες για τουλάχιστον την τοποθεσία στην οποία παρατηρήθηκαν την χρονική στιγμή που παρατηρήθηκαν.

Η χωρική τροχιά είναι ένα ίχνος που δημιουργείται από ένα κινούμενο αντικείμενο σε γεωγραφικούς χώρους και συνήθως αντιπροσωπεύεται από μια σειρά χρονολογικά διατεταγμένων σημείων. [13]

Οι Mazimpaka και Timpf παρέχουν επίσης έναν ορισμό των δεδομένων τροχιάς, αναφέροντας ότι η τροχιά ενός κινούμενου αντικειμένου είναι ένα διακριτό ίχνος που ταξιδεύει το κινούμενο αντικείμενο σε γεωγραφικούς χώρους. [14]

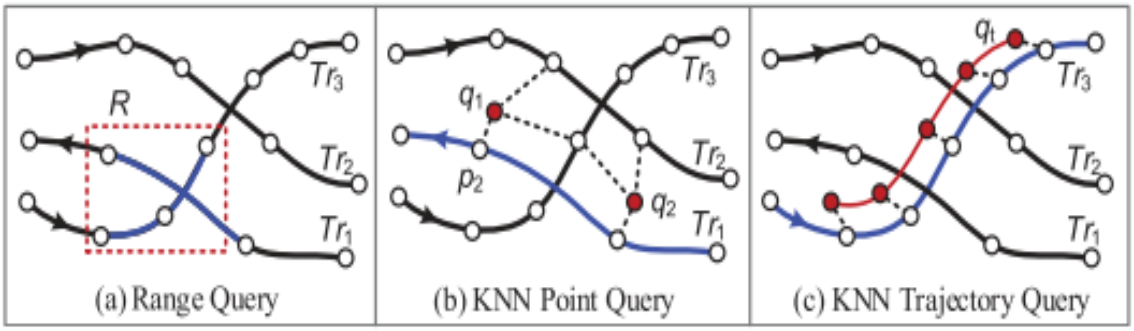
Στην εργασία των Wang et al. [15], δίνεται ο ορισμός των δεδομένων τροχιάς ως εξής: Οι συσκευές που παρακολουθούν κινούμενα αντικείμενα δημιουργούν συχνά μια σειρά από διατεταγμένα σημεία που αντιπροσωπεύουν μια τροχιά. Πιο επιστημονικά, μια τροχιά T αποτελείται από δύο ή περισσότερα σημεία μόνο στο χώρο και ορίζεται ως:

Ορισμός. (Σημείο) Ένα σημείο p = {*x, y, t*} καταγράφει το γεωγραφικό πλάτος x, το γεωγραφικό μήκος y στη χρονική στιγμή t.

Τύποι ερωτημάτων

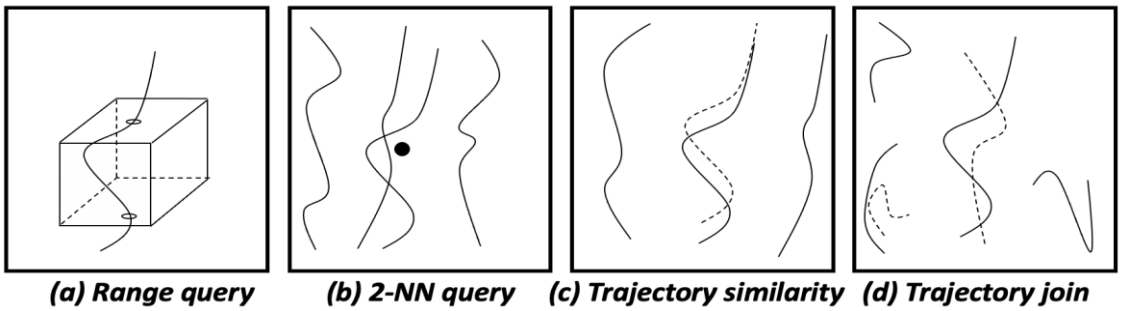
Στο τεχνικό κομμάτι, υπάρχουν ορισμένοι βασικοί τύποι ερωτημάτων για την επεξεργασία των δεδομένων τροχιάς. Στην εργασία [13] παρουσιάζονται δύο από τους βασικότερους, οι οποίοι είναι τα

ερωτήματα διαστήματος και τα ερωτήματα πλησιέστερων γειτόνων. Από το σχήμα 1, στην περίπτωση a, παρατηρούμε ότι τα ερωτήματα διαστήματος υπολογίζουν τις τροχιές που βρίσκονται (ή τέμνονται) σε ένα χωρικό (ή χωροχρονικό) εύρος. Στην περίπτωση b, βλέπουμε ένα ερώτημα k-NN, όπου υπολογίζεται η τροχιά που έχει την ελάχιστη απόσταση σε σχέση με τα δύο δοσμένα σημεία. Στην περίπτωση c, εμφανίζεται πάλι ένα ερώτημα k-NN, όπου υπολογίζεται η τροχιά που έχει την ελάχιστη απόσταση σε σχέση με μία δοσμένη τροχιά.



Σχήμα 1. Δύο κατηγορίες ερωτημάτων για δεδομένα τροχιάς *[13]*

Όμως και στην εργασία [16] αναφέρονται, πέρα από τα βασικά ερωτήματα που είδαμε προηγουμένως, κάποια επιπλέον ερωτήματα για την επεξεργασία των δεδομένων τροχιάς. Στο σχήμα 2, παρουσιάζεται στην περίπτωση a ένα χωροχρονικό ερώτημα διαστήματος για δεδομένα τροχιάς, όπου υπολογίζονται όλα τα τμήματα τροχιών μέσα σε μια χωρική περιοχή κατά τη διάρκεια ενός χρονικού διαστήματος. Στην περίπτωση b, εμφανίζεται ένα ερώτημα k-NN, το οποίο υπολογίζει τις δύο τροχιές που βρίσκονται πιο κοντά σε ένα δεδομένο σημείο. Στην περίπτωση c, απεικονίζεται ένα ερώτημα ομοιότητας τροχιάς το οποίο, δεδομένης μιας συνάρτησης ομοιότητας τροχιάς, υπολογίζει την πιο όμοια τροχιά με μια δεδομένη τροχιά ερωτήματος. Οι παραλλαγές αυτού του ερωτήματος μπορούν να χρησιμοποιήσουν ένα όριο απόστασης στην ομοιότητα ή να ανακτήσουν τις k πιο όμοιες τροχιές. Τέλος, στην περίπτωση d, παρουσιάζεται η ένωση τροχιάς, όπου δίνονται δύο σύνολα δεδομένων τροχιών, και το ζητούμενο είναι να εντοπιστούν τα ζεύγη τροχιών που ικανοποιούν μια συνθήκη (συνήθως εκφράζεται ως περιορισμός ομοιότητας).



Σχήμα 2. Βασικά ερωτήματα δεδομένων τροχιάς *[16]*

3. Συστήματα Αποθήκευσης και Επεξεργασίας Μεγάλων Δεδομένων Τροχιάς

Όπως αναφέρεται στην εργασία [15],υπάρχουν αρκετά ερευνητικά συστήματα ειδικά σχεδιασμένα για τη διαχείριση των δεδομένων τροχιάς ([17],[1],[18],[19],[20],[21]), την επεξεργασία τους και την αποθήκευση τροχιών ως σύνολο σημείων, καθώς και αρκετά εμπορικά συστήματα που δεν σχεδιάστηκαν απαραίτητα μόνο για δεδομένα τροχιάς, αλλά μπορούν να αποθηκεύσουν και να χειριστούν τέτοιου είδους δεδομένα μέσω πρόσθετων επεκτάσεων. Επιπλέον, ορισμένα από αυτά τα συστήματα αφορούν μόνο την αποθήκευση των δεδομένων, ενώ υπάρχουν και συστήματα που πέρα από την αποθήκευση, διαθέτουν και κάποια διαδικασία επεξεργασίας των αποθηκευμένων δεδομένων.

Αυτά τα συστήματα διαχείρισης τροχιάς ([1],[18],[20]) υποστηρίζουν μόνο μια μορφή αποθήκευσης και συχνά μόνο έναν τύπο ερωτήματος, όπως ένα ερώτημα περιοχής (εύρεση τροχιών σε ένα ορθογώνιο). Πιο πρόσφατα, κατανεμημένα συστήματα ([17],[19],[22]) αναπτύχθηκαν για να αποθηκεύουν μεγάλα σύνολα δεδομένων τροχιάς που βασίζονται σε σημεία και μπορούν να εκτελούν πιο προηγμένα ερωτήματα, όπως αναζήτηση πλησιέστερων γειτόνων kNN σε τροχιές.

Ορισμένα υπάρχοντα εμπορικά συστήματα ή συστήματα ανοιχτού κώδικα (π.χ. μια γενική βάση δεδομένων όπως Oracle Spatial, SQL Server και MySQL, ή γεωχωρικές βάσεις δεδομένων όπως ArcGIS, PostGIS, GeoMesa) μπορούν επίσης να επεκταθούν για τη διαχείριση των δεδομένων τροχιάς, κάτι το οποίο δε θα μας απασχολήσει στη συγκεκριμένη εργασία.

CloST [23]

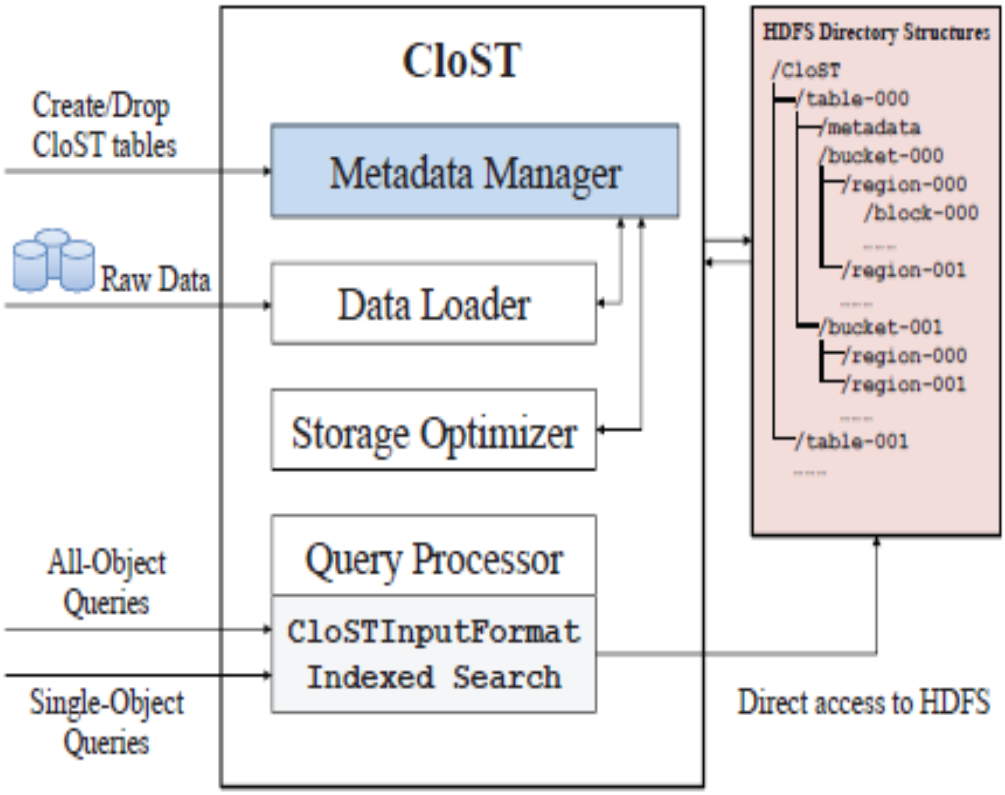
Το CloST είναι ένα επεκτάσιμο σύστημα αποθήκευσης μεγάλων χωροχρονικών δεδομένων, για την υποστήριξη της ανάλυσης των δεδομένων και βασίζεται στο Hadoop. Ο κύριος στόχος του CloST είναι να αποφύγει τη σάρωση ολόκληρου του συνόλου των δεδομένων όταν δίνεται ένα χωροχρονικό διάστημα. Για το σκοπό αυτό, προτείνεται ένα μοντέλο δεδομένων που επεξεργάζεται τρία βασικά χαρακτηριστικά, όπως το αναγνωριστικό του αντικειμένου, μια τοποθεσία και έναν χρόνο.

Με βάση αυτό το μοντέλο δεδομένων, το CloST διαχωρίζει ιεραρχικά τα δεδομένα, σε τρία επίπεδα, χρησιμοποιώντας όλα τα βασικά χαρακτηριστικά, πράγμα που επιτρέπει την αποτελεσματική παράλληλη επεξεργασία των σαρώσεων της χωροχρονικής περιοχής. Επομένως, η αποτελεσματικότητα της διαμέρισης των δεδομένων δεν εξαρτάται από το εάν το χωρικό κατηγόρημα είναι πιο επιλεκτικό σε σχέση με το χρονικό κατηγόρημα ή το αντίθετο. Σύμφωνα με τα χαρακτηριστικά των δεδομένων, προτείνεται μια συμπαγής δομή αποθήκευσης δεδομένων σε HDFS που μειώνει το μέγεθος της αποθήκευσης κατά μια τάξη μεγέθους σε σύγκριση με αρχεία ακατέργαστων δεδομένων. Σε τέτοια αρχεία, οι εγγραφές από το ίδιο αντικείμενο ομαδοποιούνται και στη συνέχεια αποθηκεύονται ανά στήλη. Τα δεδομένα συμπιέζονται δύο φορές σε επίπεδο στήλης και σε επίπεδο ομάδας αντίστοιχα. Επιπλέον, προτείνονται επεκτάσιμοι και αποτελεσματικοί αλγόριθμοι, χρησιμοποιώντας το MapReduce, για την εκτέλεση όλων των ερωτημάτων αντικειμένων, της αρχικής φόρτωσης δεδομένων και της σταδιακής φόρτωσης του συστήματος με νέα δεδομένα.

Στο CloST, όλα τα δεδομένα και τα μεταδεδομένα αποθηκεύονται στο HDFS ως κανονικά αρχεία με προκαθορισμένη δομή καταλόγου και το σύστημα παρέχει έναν αριθμό στοιχείων για τη διαχείρισή τους. Όπως φαίνεται στο σχήμα 3, τα δομικά στοιχεία του CloST περιλαμβάνουν το διαχειριστή μεταδεδομένων, το πρόγραμμα φόρτωσης δεδομένων, το βελτιστοποιητή αποθήκευσης και τον επεξεργαστή ερωτημάτων.

Το CloST, για να υποστηρίξει την αποτελεσματική σάρωση του χωροχρονικού διαστήματος, χρησιμοποιεί ένα νέο μοντέλο δεδομένων. Αυτό το μοντέλο δεδομένων στοχεύει σε πολύ μεγάλα χωροχρονικά σύνολα δεδομένων που μοιράζονται παρόμοια χαρακτηριστικά με τα δεδομένα καταγραφής μιας συσκευής GPS. Αποθηκεύει χωροχρονικά δεδομένα σε πίνακες. Ένας πίνακας έχει το σχήμα με τη μορφή (*Oid, Loc, Time, A1, . . . ,An*). Τα πρώτα τρία χαρακτηριστικά αναφέρονται ως χαρακτηριστικά πυρήνα, όπου το Oid είναι ένα αναγνωριστικό αντικειμένου, το Loc είναι ένα χωρικό

σημείο και το Time είναι μια χρονική σήμανση. Το πιο σημαντικό χαρακτηριστικό του μοντέλου δεδομένων CloST είναι ότι τα βασικά χαρακτηριστικά είναι υποχρεωτικά και προκαθορισμένα για οποιονδήποτε πίνακα. Ως εκ τούτου, τα συστήματα αποθήκευσης που βασίζονται σε αυτό το μοντέλο δεδομένων μπορούν να βελτιστοποιηθούν ιδιαίτερα για τα βασικά χαρακτηριστικά. Σε αντίθεση με τα βασικά χαρακτηριστικά, τα A1 έως An ορίζονται από τους χρήστες και αναφέρονται ως κοινά χαρακτηριστικά.

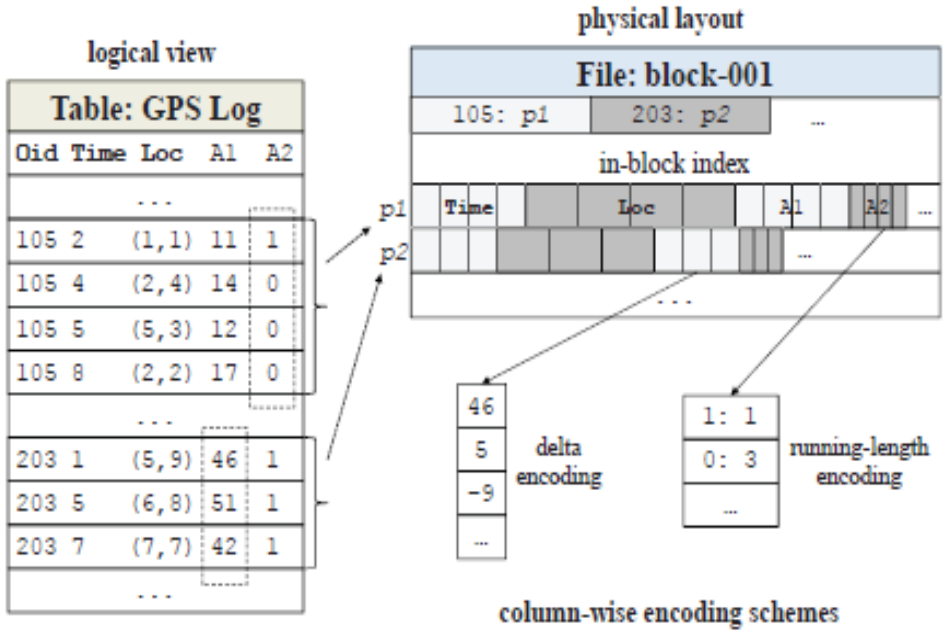


Σχήμα 3. Επισκόπηση συστήματος CloST [23]

Για την αποθήκευση των δεδομένων, μετά τη διαμέρισή τους, χρησιμοποιείται το αρχείο μπλοκ. Το αρχείο μπλοκ αποτελεί μια αποτελεσματική και συμπαγή διάταξη αποθήκευσης για την πραγματική αποθήκευση των εγγραφών. Το σχήμα 4 δείχνει ένα παράδειγμα των εσωτερικών δομών ενός αρχείου μπλοκ. Οι εγγραφές ομαδοποιούνται ανά Oid και κάθε ομάδα αποθηκεύεται σε ένα τμήμα αρχείου. Ένα ευρετήριο εντός του μπλοκ τοποθετείται στην αρχή του αρχείου για να αντιστοιχίσει ένα Oid στην αντίστοιχη θέση του τμήματος. Για κάθε τμήμα, οι εγγραφές ταξινομούνται πρώτα κατά χρόνο και στη συνέχεια οργανώνονται με τρόπο αποθήκευσης σε στήλη, δηλαδή, οι τιμές από το ίδιο χαρακτηριστικό αποθηκεύονται συνεχώς. Για εξοικονόμηση χώρου, δεν αποθηκεύεται το Oid σε τμήματα επειδή μπορεί να το προκύψει από το ευρετήριο εντός του μπλοκ.

Για να βελτιώσει την απόδοση αποθήκευσης, το CloST συμπιέζει τα δεδομένα πρώτα σε επίπεδο στήλης και μετά σε επίπεδο τμήματος. Για τη συμπίεση σε επίπεδο στήλης, οι αριθμητικές τιμές κωδικοποιούνται είτε με κωδικοποίηση δέλτα είτε με κωδικοποίηση μήκους λειτουργίας [18]. Οι κωδικοποιήσεις αυτές μπορούν να μειώσουν σημαντικά το μέγεθος αποθήκευσης όταν οι τιμές

αλλάζουν αργά ή παραμένουν σταθερές με την πάροδο του χρόνου, κάτι που είναι κοινό για πολλά χωροχρονικά σύνολα δεδομένων.



Σχήμα 4. Αρχείο μπλοκ [23]

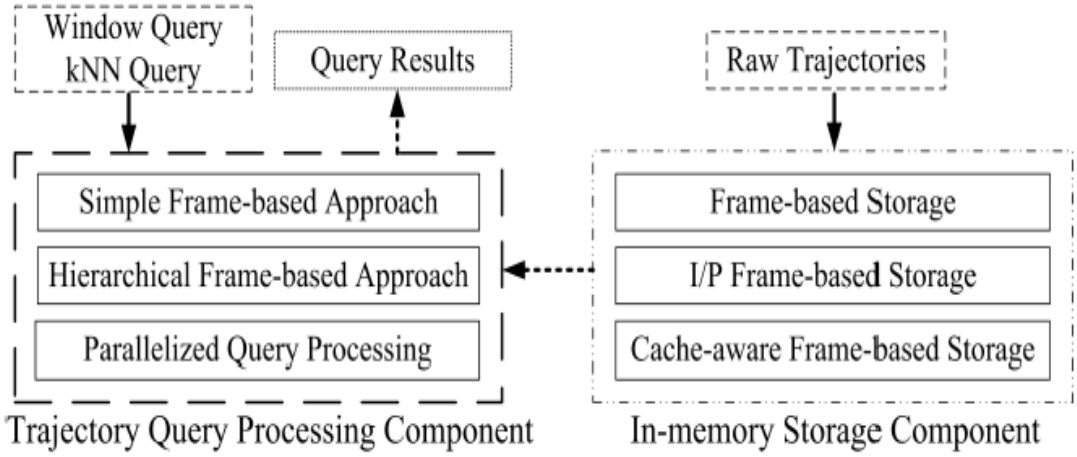
SharkDB [20]

Οι Wang et al. προτείνουν την προσανατολισμένη σε στήλες αποθήκευση τροχιών στην εσωτερική μνήμη, η οποία είναι μια οπτικά βελτιωμένη δομή για την αποθήκευση και την επεξεργασία μεγάλων δεδομένων τροχιάς. Αυτός ο τρόπος αποθήκευσης μπορεί να συνδυάσει τα πλεονεκτήματα της υψηλής απόδοσης της κύριας μνήμης και τα οφέλη της αποθήκευσης σε στήλη για τις εργασίες ανάλυσης. Εν τω μεταξύ, ο σχεδιασμός του συστήματος υποστηρίζει επίσης την αποτελεσματική συμπίεση των δεδομένων τροχιάς και την επεξεργασία των ερωτημάτων στα συμπιεσμένα δεδομένα τροχιάς.

Προκειμένου να εκμεταλλευτεί τις δομές δεδομένων σε στήλη και τις τεχνικές συμπίεσης για την αποθήκευση και την ανάλυση των δεδομένων τροχιάς, το SharkDB αντιμετωπίζει δύο κύριες προκλήσεις. Η πρώτη είναι ο τρόπος μετατροπής των δεδομένων τροχιάς σε μια δομή δεδομένων προσανατολισμένη σε στήλη, η οποία όχι μόνο υποστηρίζει δεδομένα ποικίλου μήκους και πολυδιάστατων τροχιών, αλλά μπορεί επίσης να εκτελέσει και μια αποτελεσματική επεξεργασία των ερωτημάτων στην κύρια μνήμη. Η δεύτερη είναι πώς να αναπτυχθούν τεχνικές συμπίεσης στη δομή δεδομένων σε στήλη, η οποία μπορεί να υποστηρίξει την επεξεργασία ερωτημάτων σε συμπιεσμένα δεδομένα χωρίς να θυσιάζεται μεγάλη απόδοση.

Το σύστημα αντιμετωπίζει τις παραπάνω προκλήσεις με μια προσεκτικά σχεδιασμένη δομή δεδομένων με επεξεργασία των δεδομένων σε δύο φάσεις, η οποία συνδυάζει τόσο τη δομή δεδομένων σε στήλες όσο και τις τεχνικές συμπίεσης για την αποθήκευση και την ανάλυση του τεράστιου όγκου δεδομένων τροχιάς σε ένα σύστημα βάσης δεδομένων στην εσωτερική μνήμη. Πιο συγκεκριμένα, η φάση κατανομής της τροχιάς θα αξιοποιήσει τις τεχνικές βαθμονόμησης της τροχιάς [24] για τη βαθμονόμηση των δεδομένων τροχιάς και τη μετατροπή των βαθμονομημένων δεδομένων τροχιάς σε δομή δεδομένων

προσανατολισμένη σε στήλη. Στη δεύτερη φάση, ένα στοιχείο κωδικοποίησης της τροχιάς θα συμπιέσει τα δεδομένα τροχιάς και, στη συνέχεια, αυτή η κωδικοποίηση των συμπιεσμένων δεδομένων τροχιάς έχει σαν αποτέλεσμα την αποθήκευση των δεδομένων σε μια προσανατολισμένη σε στήλη δομή και την υποστήριξη μιας αποτελεσματικής επεξεργασίας των ερωτημάτων σε αυτήν.



Σχήμα 5. Αρχιτεκτονική αποθήκευσης στο σύστημα SharkDB *[20]*

Η αρχιτεκτονική της αποθήκευσης απεικονίζεται στο σχήμα 5, το οποίο περιλαμβάνει δύο κύρια μέρη: το στοιχείο επεξεργασίας των ερωτημάτων σε πραγματικό χρόνο και την αποθήκευση των δεδομένων στην εσωτερική μνήμη. Το σύστημα αποθήκευσης των δεδομένων στην εσωτερική μνήμη κωδικοποιεί τις πρωτογενείς τροχιές και τις αποθηκεύει σε μια επανασχεδιασμένη δομή δεδομένων σε στήλη. Το στοιχείο επεξεργασίας του ερωτήματος σε πραγματικό χρόνο μπορεί να υποστηρίξει δύο βασικά ερωτήματα, συμπεριλαμβανομένου του ερωτήματος παραθύρου και του ερωτήματος πλησιέστερου γείτονα σε πραγματικό χρόνο.

Αποθήκευση στην εσωτερική μνήμη: Στο στοιχείο αποθήκευσης δεδομένων στην εσωτερική μνήμη, δημιουργείται μια νέα, βάσει πλαισίου, δομή δεδομένων σε στήλες για την αποθήκευση των δεδομένων τροχιάς. Ως εκ τούτου, προτείνονται τρία μοντέλα αποθήκευσης: αποθήκευση που βασίζεται σε ένα πλαίσιο, αποθήκευση βάσει πλαισίου I/P και αποθήκευση βάσει πλαισίου με επίγνωση κρυφής μνήμης, για τη μετατροπή και τη βαθμονόμηση των πρωτογενών δεδομένων τροχιάς σε αυτήν τη νέα δομή δεδομένων που βασίζεται σε πλαίσιο. Τα δύο τελευταία μοντέλα ενσωματώνουν επίσης έναν αλγόριθμο συμπίεσης τροχιάς για να μειώσουν το αποτύπωμα των δεδομένων στη μνήμη.

Σύστημα επεξεργασίας ερωτημάτων σε πραγματικό χρόνο: Εφαρμόζονται τρεις προσεγγίσεις επεξεργασίας ερωτημάτων για την υποστήριξη της επεξεργασίας των ερωτημάτων σε πραγματικό χρόνο στο σύστημα. Η απλή προσέγγιση που βασίζεται σε ένα πλαίσιο λειτουργεί πάνω στη δομή δεδομένων του πλαισίου και η προσέγγιση βάσει ιεραρχικού πλαισίου δημιουργεί μια δομή δεδομένων πλαισίου πολλαπλών επιπέδων από ένα στοιχείο αποθήκευσης στη μνήμη για την επίτευξη καλύτερης απόδοσης. Επιπλέον, επεκτείνονται προηγούμενες προσεγγίσεις για να επιτραπεί η παράλληλη επεξεργασία των ερωτημάτων.

TrajStore [18]

Λόγω των περιορισμών στην απόδοση των υπαρχουσών δομών δεδομένων, δημιουργήθηκε το TrajStore, που αποτελεί ένα βελτιωμένο σύστημα για την αποτελεσματική ανάκτηση όλων των τροχιών σε μια συγκεκριμένη γεωχωρική/χρονική περιοχή. Αν και υπάρχει μεγάλος όγκος σχετικών εργασιών για την

ευρετηρίαση των τροχιών, οι περισσότερες από τις τρέχουσες προσεγγίσεις πραγματοποιούν τελικά αναζήτηση στο δίσκο ανά τροχιά, η οποία μπορεί να επηρεάσει σημαντικά την απόδοση στις ρυθμίσεις για τους ίδιους λόγους που τα R-Tree δεν έχουν καλή απόδοση.

Σε αντίθεση με τα περισσότερα υπάρχοντα συστήματα που απλώς ευρετηριάζουν γεωχωρικά δεδομένα, το TrajStore είναι ένα σύστημα αποθήκευσης που έχει σχεδιαστεί για να τμηματοποιεί τις τροχιές και να εντοπίζει ταυτόχρονα τα τμήματα τροχιών που βρίσκονται γεωγραφικά και χρονικά το ένα κοντά στο άλλο. Κόβει τις τροχιές σε υποτροχιές που ταιριάζουν σε χωροχρονικές περιοχές και συμπυκνώνει τα δεδομένα για κάθε περιοχή σε ένα μπλοκ (ή μια συλλογή μπλοκ) στο δίσκο. Χρησιμοποιεί ένα προσαρμοστικό πλέγμα πολλαπλών επιπέδων [25] σε αυτά τα μπλοκ για να αναζητήσει δεδομένα στο χώρο και ένα μη μηδενικό (sparse) ευρετήριο στο χρόνο για να απαντήσει σε ιστορικά ερωτήματα. Με αυτόν τον τρόπο, τα περισσότερα ερωτήματα μπορούν να απαντηθούν με την ανάγνωση μόλις λίγων μπλοκ από το δίσκο, ακόμα κι αν αυτά τα μπλοκ περιέχουν δεδομένα από εκατοντάδες ή χιλιάδες τροχιές. Εκτός από αυτή τη βασική διατύπωση, το TrajStore συνεισφέρει τα εξής:

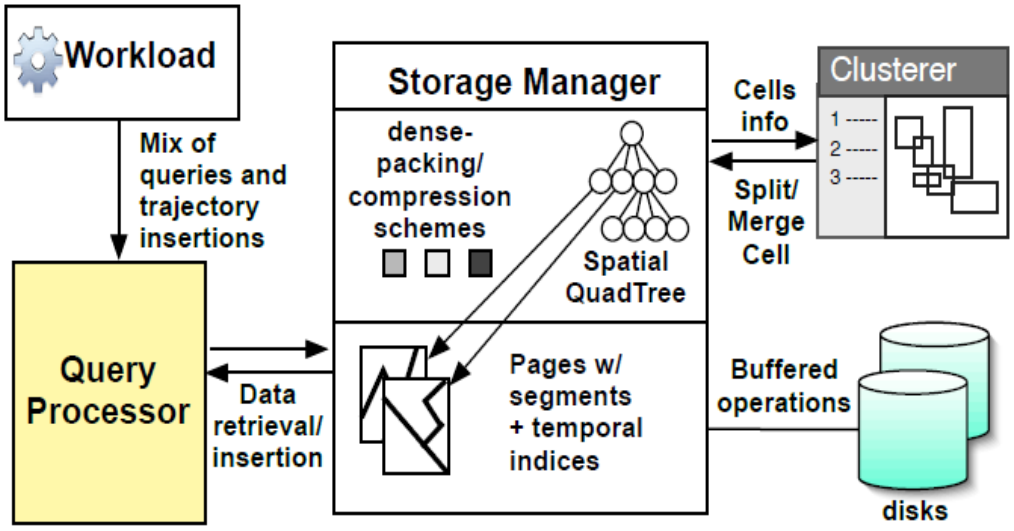
1) Περιγράφει ένα προσαρμοστικό σχήμα αποθήκευσης που επιλέγει το μέγεθος των χωρικών κελιών στο quadtree ευρετήριό με βάση την πυκνότητα των δεδομένων, τα αναμενόμενα (ή παρατηρούμενα) μεγέθη των ερωτημάτων που εκτελούνται στα δεδομένα και το μέγεθος της σελίδας του συστήματος.

2) Περιγράφει πώς το σχήμα μπορεί να προσαρμόσει την ομαδοποίηση των δεδομένων καθώς εισάγονται ή διαγράφονται νέα σημεία ή καθώς αλλάζει ο φόρτος εργασίας και πώς διατηρεί ένα βέλτιστο ευρετήριο ανά πάσα στιγμή με διαχωρισμό / συγχώνευση των κελιών αναδρομικά.

3) Περιγράφει έναν αλγόριθμο συμπίεσης για τη συμπίεση των τροχιών στα κελιά χρησιμοποιώντας ένα, κατά προσέγγιση, σχήμα κωδικοποίησης τροχιάς, το οποίο δέχεται ένα όριο σφάλματος που καθορίζεται από τον χρήστη.

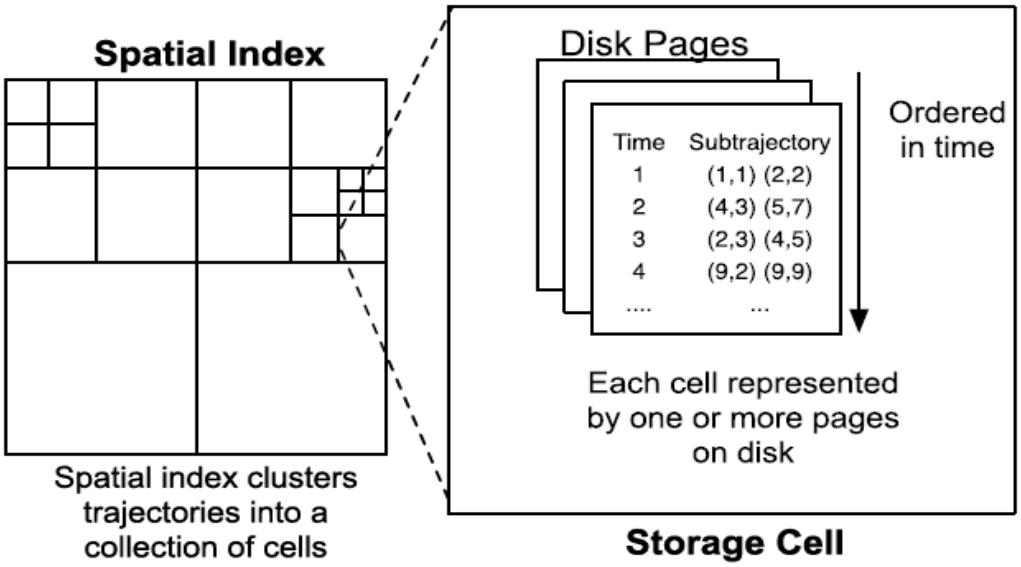
4) Έχει αποδειχθεί ότι το TrajStore μπορεί να ανακτήσει αποτελέσματα από το σύνολο των δεδομένων του πραγματικού κόσμου 8 φορές πιο γρήγορα από τις υπάρχουσες προσεγγίσεις που βασίζονται στην τμηματοποίηση τροχιών και στην αποθήκευση τους σε ένα ομαδοποιημένο R-Tree.

Η αρχιτεκτονική του TrajStore φαίνεται στο σχήμα 6. Ο επεξεργαστής ερωτημάτων λαμβάνει μια ροή ερωτημάτων καθώς και νέες τροχιές για αποθήκευση. Ένα ερώτημα Q(*T, r, t*) πάνω σε έναν πίνακα T, σε μια χωρική περιοχή r (με τη μορφή ορθογωνίου) και σε ένα χρονικό διάστημα t επιστρέφει ένα σύνολο υποτροχιών που βρίσκονται μέσα στο r κατά τη διάρκεια του t. Οι υποτροχιές είναι τμήματα μεγαλύτερων τροχιών κομμένες στο ορθογώνιο r.



***Σχήμα 6. Επισκόπηση συστήματος TrajStore** [18]*

Οι δομές αποθήκευσης του TrajStore βελτιστοποιούνται για χωρικά ερωτήματα σε συγκεκριμένες περιοχές με σχετικά μεγάλα χρονικά όρια, αντί να βρίσκουν μόνο μία ή λίγες τροχιές που διέρχονται από μια περιοχή σε μια ακριβή χρονική στιγμή. Για το λόγο αυτό, η αποθήκευση είναι κατά κύριο λόγο οργανωμένη σύμφωνα με ένα χωρικό ευρετήριο, με χρονικά κατηγορήματα που εφαρμόζονται στα δεδομένα που ανακτώνται από αυτό το χωρικό ευρετήριο, καθώς αναμένεται ότι τα χωρικά κατηγορήματα θα είναι γενικά πιο επιλεκτικά από τα χρονικά κατηγορήματα.



Σχήμα 7. Βασικές δομές αποθήκευσης που αναπαριστούν τα δεδομένα στο δίσκο *[18]*

Τα δεδομένα στο TrajStore αποθηκεύονται από το χωρικό ευρετήριο και το συσταδοποιητή, όπως φαίνεται στο σχήμα 7. Το TrajStore χρησιμοποιεί αρκετούς αλγόριθμους ευρετηρίασης, όλοι εκ των οποίων λειτουργούν τμηματοποιώντας τις τροχιές σε υποτροχιές και στη συνέχεια συσταδοποιώντας τις χωρικά συντεντοπισμένες υποτροχιές σε κελιά στο δίσκο. Ο κύριος νέος χωρικός δείκτης που προτείνεται είναι ένα προσαρμοσμένο quadtree, όπου κάθε κελί στο quadtree αντιστοιχεί σε μια συλλογή σελίδων στο δίσκο, που περιέχουν τα κομμάτια της τροχιάς σε αυτό το κελί. Τα μεταδεδομένα σχετικά με τα όρια και τη θέση στο δίσκο των κελιών χωρούν στη μνήμη.

Κάθε κελί αποτελείται από μία ή περισσότερες σελίδες δίσκου, οι οποίες αποτελούν τη θεμελιώδη μονάδα μεταφοράς μεταξύ του συστήματος αρχείων και του TrajStore. Οι νέες σελίδες εκχωρούνται από μια ελεύθερη λίστα. Διατηρώντας τις σελίδες σχετικά μεγάλες, περιορίζεται το σχετικό κλάσμα του χρόνου που δαπανάται για αναζήτηση μεταξύ των σελίδων κατά την ανάγνωση ενός κελιού από το δίσκο.

Το TrajStore είναι κυρίως βελτιστοποιημένο για αποθήκευση μόνο προσαρτήσεων. Οι τροχιές θεωρείται ότι φτάνουν με τη σειρά εισαγωγής και οι νέες δευτερεύουσες τροχιές απλώς προστίθενται στην πιο πρόσφατη σελίδα που έχει εκχωρηθεί για κάθε κελί του χωρικού ευρετηρίου. Πιστεύεται ότι αυτή είναι μια λογική υπόθεση, καθώς οι τροχιές γενικά συλλέγονται από συσκευές GPS και δεν πρέπει να αλλάζουν ή να χρειάζεται να ενημερωθούν εκ των υστέρων. Για να υποστηρίξει την αφαίρεση δεδομένων, το TrajStore συσχετίζει ένα bit «διαγραφής» με κάθε υποτροχιά.

Καθώς προστίθενται δευτερεύουσες τροχιές σε ένα κελί, ενδέχεται να εκχωρηθούν νέες σελίδες. Συσχετίζεται ένα μη μηδενικό ευρετήριο χρονικής σήμανσης με κάθε κελί, που αποθηκεύει απλώς μια σήμανση της ώρας έναρξης και μια σήμανση της ώρας λήξης για κάθε σελίδα στο κελί,

καθιστώντας εύκολη την εύρεση του υποσυνόλου των σελίδων σε ένα κελί που επικαλύπτει μια δεδομένη ώρα.

Κάθε κελί μπορεί να συμπιεστεί χρησιμοποιώντας μια συλλογή σχημάτων συμπίεσης. Η κύρια ιδέα πίσω από την προσέγγισή συμπίεσης είναι να εξαλειφθεί ο χωρικός πλεονασμός που προκύπτει όταν πολλές τροχιές διασχίζουν περίπου την ίδια διαδρομή μέσα από ένα κελί.

Το TrajStore διατηρεί μια ρύθμιση στη μνήμη για τις σελίδες που διαβάστηκαν ή ενημερώθηκαν πρόσφατα. Οι σελίδες επανεγγράφονται χρησιμοποιώντας μια πολιτική διαχείρισης κρυφής μνήμης LRU. Το TrajStore διατηρεί ένα τυπικό αρχείο καταγραφής λανθασμένων ενημερώσεων σε σελίδες για να διευκολύνει την ανάκτηση σφαλμάτων.

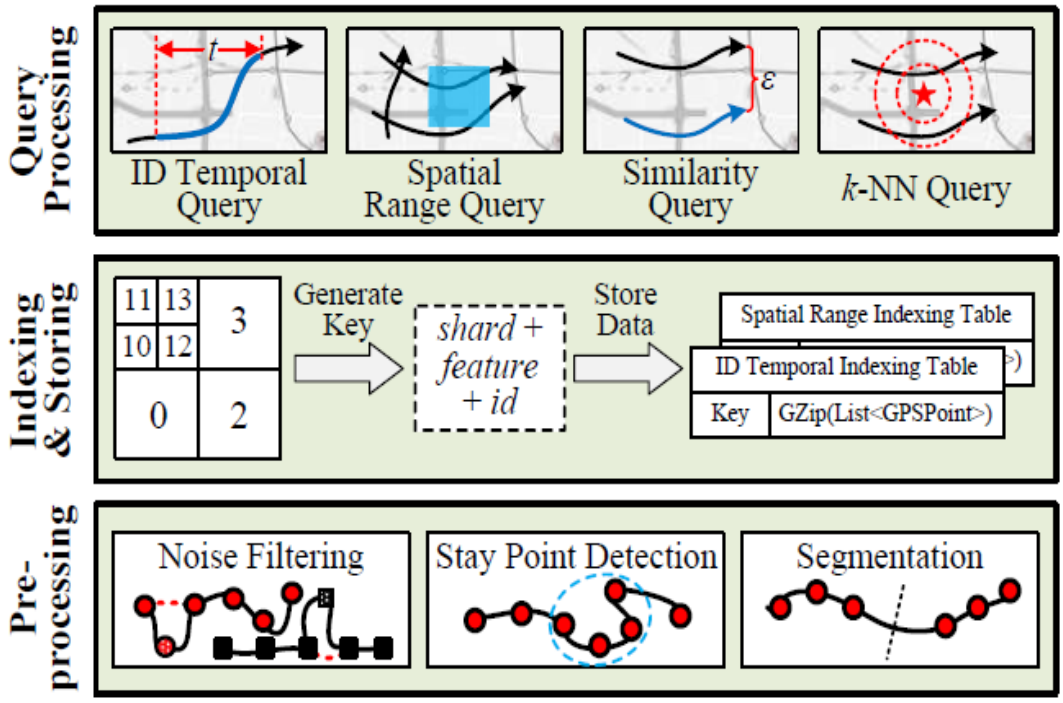
Τέλος, το TrajStore διατηρεί ένα ευρετήριο τροχιάς το οποίο, για κάθε τροχιά, αντιστοιχίζεται σε μια λίστα κελιών που περιέχει τις υποτροχιές αυτής της τροχιάς.

TrajMesa [22]

Το TrajMesa αποτελεί μια ολιστική κατανεμημένη μηχανή αποθήκευσης τροχιών NoSQL, βασισμένη στο σύστημα αποθήκευσης δεδομένων GeoMesa. Το TrajMesa ενσωματώνει τρεις τυπικές διαδικασίες:

(1) Προεπεξεργασία, η οποία εκτελεί τρεις κύριες εργασίες: φιλτράρισμα θορύβου, ανίχνευση σημείου παραμονής και τμηματοποίηση. Η προεπεξεργασία τροχιάς είναι απαραίτητη για πολλές αστικές εφαρμογές, όπως επίσης και για την επιλογή της υποκείμενης μεθόδου αποθήκευσης και τη δημιουργία του ευρετηρίου.

(2) Ευρετηρίαση και αποθήκευση, η οποία δημιουργεί ευρετήρια για τις προεπεξεργασμένες τροχιές και αποθηκεύει τα δεδομένα τροχιάς μέσω του GeoMesa. Συγκεκριμένα, δημιουργούνται δύο διαφορετικά κλειδιά που συνδυάζουν τις χωροχρονικές και άλλες απαραίτητες πληροφορίες μιας τροχιάς. Κάθε κλειδί και τα δεδομένα τροχιάς σχηματίζουν ένα ζεύγος κλειδιού-τιμής, το οποίο στη συνέχεια αποθηκεύεται στον χώρο αποθήκευσης δεδομένων κλειδιού-τιμής του GeoMesa. Με άλλα λόγια, αποθηκεύονται δύο αντίγραφα μιας τροχιάς σε δύο πίνακες με διαφορετικά κλειδιά.



***Σχήμα 8. Επισκόπηση συστήματος TrajMesa** [22]*

(3) Επεξεργασία ερωτημάτων. Με τη βοήθεια ενσωματωμένων ευρετηρίων, το TrajMesa υποστηρίζει αποτελεσματικά τα πιο χρήσιμα ερωτήματα τροχιάς, συμπεριλαμβανομένων των εξής: Χρονικό ερώτημα ID, ερώτημα χωρικού διαστήματος, ερώτημα ομοιότητας και ερώτημα πλησιέστερων γειτόνων k-NN.

Στο TrajMesa εφαρμόζονται δύο μέθοδοι αποθήκευσης των δεδομένων, η μέθοδος κάθετης αποθήκευσης και η μέθοδος οριζόντιας αποθήκευσης.

Μέθοδος Κάθετης αποθήκευσης. Μια βασική ιδέα για την αποθήκευση τροχιών με μία μέθοδο αποθήκευσης κλειδιών-τιμών είναι ότι τα δεδομένα της τροχιάς αποθηκεύονται με κάθε σημείο του GPS σαν μία γραμμή, όπως συμβαίνει με τα περισσότερα συστήματα διαχείρισης τροχιών που βασίζονται στο cloud ([26],[27]). Ονομάζουμε αυτό το σχήμα ως σχήμα κάθετης αποθήκευσης (V-Store).

Για παράδειγμα, σε ένα τέτοιο σχήμα, η τιμή κάθε σημείου περιέχει δύο μέρη:

(1) Χωροχρονικές ιδιότητες, οι οποίες αποτελούνται από το γεωγραφικό πλάτος, το γεωγραφικό μήκος και χρόνο αυτού του σημείου του GPS.

(2) Άλλες ιδιότητες, οι οποίες περιλαμβάνουν το αναγνωριστικό του κινούμενου αντικειμένου, το αναγνωριστικό της τροχιάς και άλλες ενδείξεις ιδιοτήτων.

Το σχήμα κάθετης αποθήκευσης (V-Store) θεωρεί κάθε σημείο GPS ως μια ανεξάρτητη οντότητα, η οποία δεν έχει πλήρη πληροφόρηση μιας τροχιάς. Ως εκ τούτου, δεν είναι κατάλληλο για τα ερωτήματα τροχιάς, ειδικά για τα ερωτήματα ομοιότητας και τα ερωτήματα k-NN. Επιπλέον, ο αριθμός των γραμμών είναι ίσος με τον αριθμό των σημείων GPS, με αποτέλεσμα πολλές καταχωρίσεις κλειδιού-τιμής να είναι απαγορευτικές. Περισσότερες καταχωρήσεις κλειδιού-τιμής χρειάζονται περισσότερο χώρο αποθήκευσης στο δίσκο. Κατά την ανάκτηση του ίδιου αριθμού τροχιών, ενεργοποιούνται περισσότερα ανοίγματα/κλεισίματα του δίσκου, γεγονός που βλάπτει περαιτέρω την αποτελεσματικότητα του ερωτήματος.

Μέθοδος Οριζόντιας αποθήκευσης. Για την αντιμετώπιση των προαναφερθέντων ζητημάτων, προτείνεται ένα νέο σχήμα οριζόντιας αποθήκευσης, το H-Store, για την αποθήκευση κάθε τροχιάς σε μία μόνο σειρά. Η τιμή κάθε καταχώρισης περιέχει τέσσερα μέρη:

(1) Χωροχρονικές ιδιότητες, οι οποίες περιλαμβάνουν το MBR, τον χρόνο έναρξης και λήξης και τις θέσεις έναρξης και λήξης μιας τροχιάς.

(2) Λίστα σημείων GPS. Τα σημεία GPS σε μια τροχιά αρχικά σειριοποιούνται χρησιμοποιώντας τη δομή Kryo και στη συνέχεια συμπιέζονται με το GZip. Όχι μόνο μειώνεται τρομερά το κόστος αποθήκευσης, αλλά βελτιώνεται επίσης η αποτελεσματικότητα της αποθήκευσης και των ερωτημάτων μειώνοντας τα ανοίγματα/κλεισίματα του δίσκου.

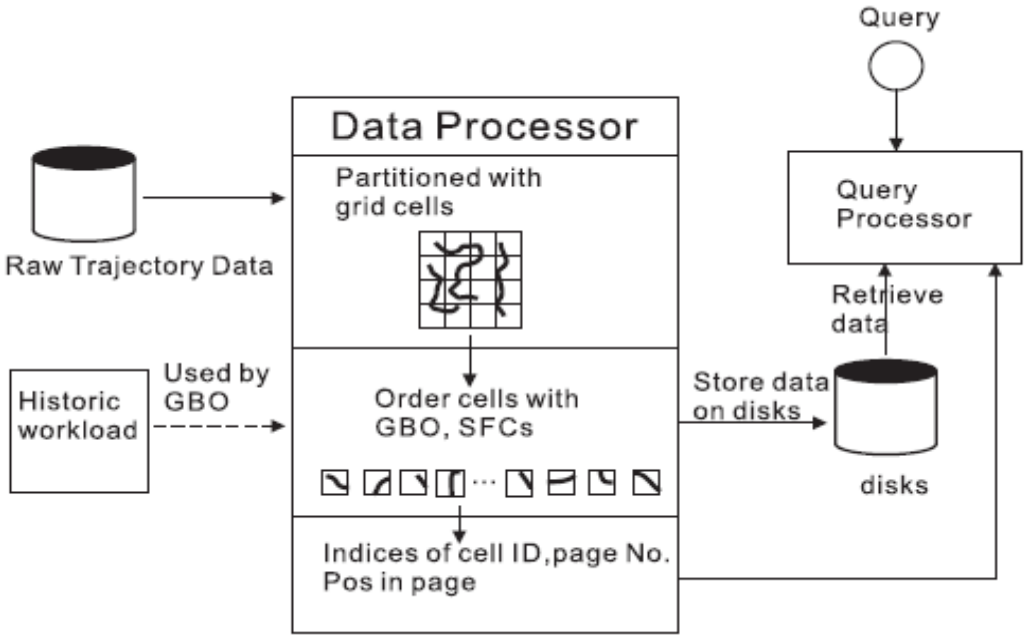
(3) Υπογραφή. Στα περισσότερα σενάρια, μια τροχιά εντοπίζει μόνο ένα πολύ μικρό μέρος του MBR της, δηλαδή, το MBR μιας τροχιάς δεν μπορεί να αντιπροσωπεύει ακριβώς τη θέση της. Για το σκοπό αυτό, σχεδιάζεται η δημιουργία μιας υπογραφής, η οποία παρέχει πιο λεπτομερείς πληροφορίες για τη θέση της τροχιάς.

(4) Άλλες ιδιότητες. Όπως το V-Store, αποθηκεύεται επίσης το αναγνωριστικό του κινούμενου αντικειμένου, το id της τροχιάς και άλλες σχετικές ιδιότητες.

GCOTraj [28]

Το GCOTraj (Grid Cell Ordering Trajectory indexing and storage) αποτελεί ένα σχήμα ευρετηρίασης και αποθήκευσης για μεγάλα σύνολα δεδομένων τροχιάς. Διαχωρίζει το χώρο των δεδομένων τροχιάς σε πολυδιάστατα κελιά πλέγματος και τμηματοποιεί τις τροχιές σε υποτροχιές για να ταιριάζουν στα υποκείμενα κελιά πλέγματος. Οι δευτερεύουσες τροχιές στο ίδιο κελί βρίσκονται μαζί και αποθηκεύονται στο ίδιο μπλοκ στο δίσκο. Τα κελιά του πλέγματος ταξινομούνται με καμπύλες πλήρωσης χώρου (Space Filling Curves, SFC), όπως η καμπύλη Hilbert, η καμπύλη Z και η καμπύλη του κώδικα Gray και άλλες. Εναλλακτικά, μπορούν να ταξινομηθούν με την τεχνική της ταξινόμησης βάσει γραφήματος

(GBO) ανάλογα με τον φόρτο εργασίας. Μια καλή σειρά θα έχει ως αποτέλεσμα λιγότερες αναζητήσεις στο δίσκο κατά τη διάρκεια του χρόνου του ερωτήματος. Εφόσον το GCOTraj ευρετηριάζει τις τροχιές τόσο σε χωρικές όσο και σε χρονικές διαστάσεις, είναι σε θέση να στοχεύει με μεγαλύτερη ακρίβεια τα κελιά του πλέγματος, στο δίσκο, που επικαλύπτουν το εύρος των ερωτημάτων. Αυτό μειώνει τη φόρτωση των περιττών δεδομένων από το δίσκο. Το GCOTraj είναι σε θέση να προσαρμόζεται στην επιλεκτικότητα των ερωτημάτων διαστήματος τόσο όσον αφορά τις χρονικές όσο και τις χωρικές διαστάσεις.



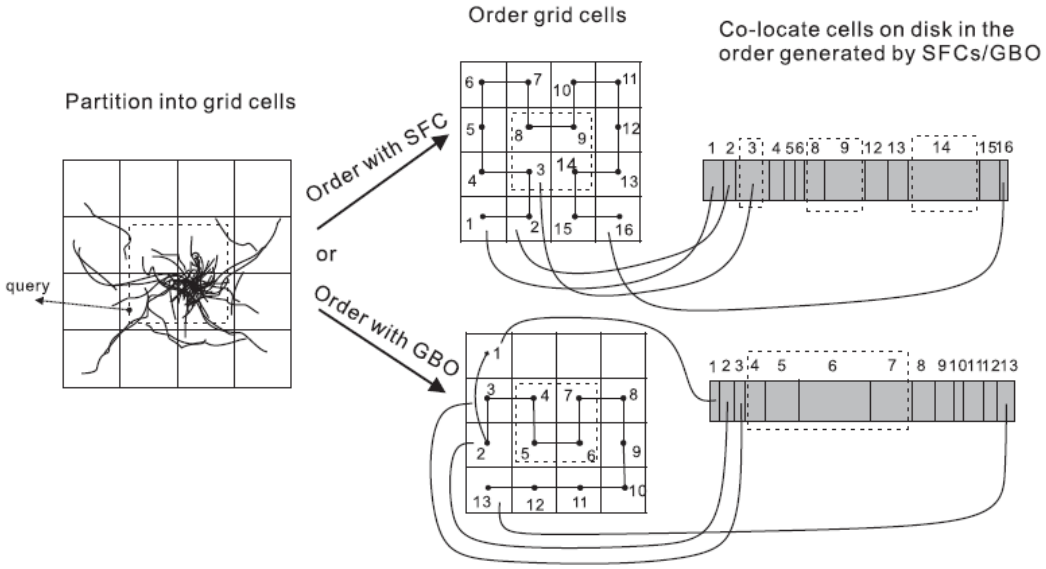
Σχήμα 9. Επισκόπηση συστήματος GCOTraj *[28]*

Το σχήμα 9 δείχνει την αρχιτεκτονική του GCOTraj. Πρώτα, το GCOTraj χωρίζει το χώρο σε κελιά πλέγματος και, στη συνέχεια, τα κελιά ταξινομούνται. Στη συνέχεια, δημιουργείται ένα ευρετήριο που αποθηκεύει το πού βρίσκεται κάθε κελί πλέγματος στο δίσκο. Το ερώτημα του χρήστη αναζητά το ευρετήριο για να προσδιορίσει ποιο σύνολο μπλοκ πρέπει να ανακτηθεί και στη συνέχεια το ανακτά από το δίσκο. Πιο συγκεκριμένα, ο επεξεργαστής δεδομένων του GCOTraj επεξεργάζεται τα υπάρχοντα ακατέργαστα δεδομένα τροχιάς τα οποία μπορούν να εκτείνονται σε μεγάλα χρονικά διαστήματα όπως χρόνια. Κάθε τροχιά Tj είναι ένα διάνυσμα μιας σειράς σημείων δεδομένων (*x, y, t*). Στη συνέχεια χωρίζει τις τροχιές σε κελιά πλέγματος, δηλ. διαιρεί κάθε τροχιά Tj σε υπο-τροχιές όπου κάθε υποτροχιά χωράει σε ένα κελί πλέγματος. Ο επεξεργαστής των δεδομένων ταξινομεί αυτά τα κελιά με SFC όπως η καμπύλη Hilbert, η καμπύλη Z ή η μέθοδος ταξινόμησης βάσει φόρτου εργασίας GBO, για να δημιουργήσει μια μονοδιάστατη ακολουθία των κελιών πλέγματος.

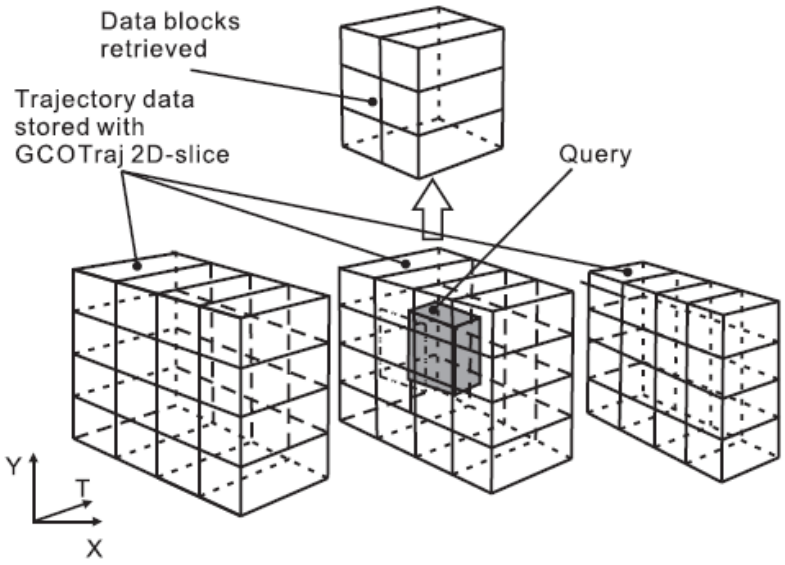
Το GCOTraj, για την αποθήκευση των δεδομένων τροχιάς, χρησιμοποιεί μια προσέγγιση η οποία βασίζεται στην ταξινόμηση των κελιών του πλέγματος, ώστε να επιλύσει το πρόβλημα της μετατροπής των πολλών διαστάσεων των ερωτημάτων στη μία διάσταση της αποθήκευσης.

Οι υπάρχουσες μέθοδοι που χρησιμοποιούν quadtree για την ευρετηρίαση των χωρικών διαστάσεων των τροχιών και για την ομαδοποίησή τους σε δίσκο, έχουν καλή απόδοση σε ερωτήματα διαστήματος σε συγκεκριμένες χωρικές περιοχές. Αλλά τα quadtrees έχουν σχεδιαστεί για να βελτιώνουν τη χωρική τοποθεσία και όχι τη χρονική. Ωστόσο, στην πράξη, τα ερωτήματα για τα δεδομένα τροχιάς είναι μερικές φορές πιο επιλεκτικά στον χρονικό τομέα από τον χωρικό τομέα. Έτσι, χρειάζεται μια λύση που μπορεί, προσαρμοστικά, να ευνοήσει τα χρονικά ή χωρικά ή και τα δύο, ανάλογα με τα μοτίβα των ερωτημάτων. Για την επίλυση του προβλήματος που προκύπτει το σύστημα GCOTraj διαθέτει τρεις παραλλαγές.

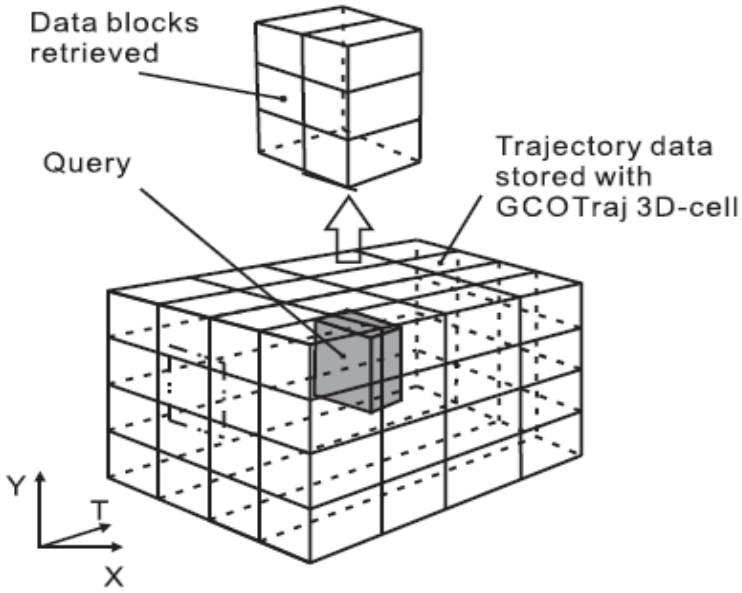
Η πρώτη, που ονομάζεται GCOTrajSP, χωρίζει τα δεδομένα τροχιάς σε δισδιάστατα κελιά πλέγματος μόνο στις χωρικές διαστάσεις X και Y και ταξινομεί αυτά τα κελιά χρησιμοποιώντας SFC ή GBO [29]. Οι υποτροχιές μέσα στο ίδιο κελί τοποθετούνται στο ίδιο μπλοκ στο δίσκο (σχήμα 10). Το δεύτερο, που ονομάζεται 2D-sliced GCOTraj, διαιρεί τον τρισδιάστατο (x, y και χρονικές διαστάσεις) χώρο σε δισδιάστατα κομμάτια. Κάθε δισδιάστατο κομμάτι αποθηκεύει όλες τις τροχιές μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Τα δισδιάστατα κομμάτια ταξινομούνται χρησιμοποιώντας SFC ή GBO. Η τρίτη παραλλαγή, που ονομάζεται 3D GCOTraj, διαχωρίζει τα δεδομένα τροχιάς σε τρισδιάστατα κελιά και στη συνέχεια ταξινομεί αυτά τα κελιά χρησιμοποιώντας SFC ή GBO.



Σχήμα 10. Δομή GCOTraSP [28]



***Σχήμα 11. 2D-sliced GCOTraj** [28]*



***Σχήμα 12. 3D GCOTraj** [28]*

Τα σχήματα 11 και 12 απεικονίζουν την ιδέα πίσω από το 2D-sliced GCOTraj και το 3D GCOTraj, αντίστοιχα. Κατά την υποβολή του ερωτήματος χρησιμοποιώντας το 2D-sliced GCOTraj (Σχήμα 11), εντοπίζονται τα κομμάτια που επικαλύπτουν τον κύβο του ερωτήματος στη χρονική διάσταση. Στη συνέχεια, ανακτώνται τα κελιά μέσα στο δισδιάστατο κομμάτι, τα οποία επικαλύπτουν το ερώτημα χωρικά. Το 3D GCOTraj (Σχήμα 12) χωρίζει τα δεδομένα σε τρισδιάστατα κελιά πλέγματος και ταξινομεί αυτά τα τρισδιάστατα κελιά με SFC ή GBO. Για να απαντήσει σε τρισδιάστατα ερωτήματα (διαστάσεις x, y και χρόνου), το 3D GCOTraj ανακτά τα κελιά που επικαλύπτουν το εύρος των ερωτημάτων και στις τρεις διαστάσεις. Αυτή η προσέγγιση είναι η πιο προσαρμοστική, καθώς μπορεί να προσαρμοστεί σε ερωτήματα που είναι επιλεκτικά χρονικά ή χωρικά ή και τα δύο. Αυτό έχει ως αποτέλεσμα τη φόρτωση λιγότερων δεδομένων που δεν επικαλύπτουν το ερώτημα.

HadoopTrajectory [30]

Ο σκοπός του HadoopTrajectory είναι να αναπτύξει ένα χωροχρονικό σύστημα επεξεργασίας δεδομένων που να είναι εξαιρετικά επεκτάσιμο και εξαιρετικά διαθέσιμο, το οποίο βασίζεται πάνω στο σύστημα Hadoop. Η επέκταση αυτή προσθέτει χωροχρονικούς τύπους και τελεστές στο σύστημα Hadoop. Αυτοί οι τύποι και τελεστές μπορούν να χρησιμοποιηθούν απευθείας σε προγράμματα MapReduce, γεγονός που δίνει στον χρήστη του Hadoop τη δυνατότητα να γράψει προγράμματα ανάλυσης χωροχρονικών δεδομένων. Το επίπεδο αποθήκευσης του Hadoop, το HDFS, επεκτείνεται ως προς τους τύπους για να αναπαραστήσει τα δεδομένα τροχιάς και τις αντίστοιχες λειτουργίες εισόδου και εξόδου. Επεκτείνεται επίσης με διαχωριστές αρχείων και συσκευές ανάγνωσης αρχείων. Αυτό επιτρέπει στο Hadoop να διαβάζει μεγάλα αρχεία τροχιών κινούμενων αντικειμένων, όπως ίχνη GPS οχημάτων, και να τα χωρίζει σε κόμβους εργασίας για κατανεμημένη επεξεργασία. Το επίπεδο αποθήκευσης επεκτείνεται επίσης με χωροχρονικά ευρετήρια που βοηθούν στο φιλτράρισμα των δεδομένων πριν από τον διαχωρισμό τους στους κόμβους εργασίας. Παρέχονται πολλές λειτουργίες πρόσβασης δεδομένων έτσι ώστε το επίπεδο MapReduce να μπορεί να αντιμετωπίσει αυτά τα δεδομένα. Το επίπεδο MapReduce επεκτείνεται με τελεστές επεξεργασίας τροχιάς, για τους διάφορους υπολογισμούς που γίνονται μέσω των ερωτημάτων.

Το σύστημα HadoopTrajectory αντιμετωπίζει αποτελεσματικά δύο προβλήματα που προκύπτουν μέσα από το Hadoop. Το πρώτο πρόβλημα είναι ότι ο Hadoop δεν γνωρίζει τη φύση των χωροχρονικών δεδομένων. Μερικά από αυτά τα δεδομένα συσχετίζονται μεταξύ τους όπως οι τροχιές κινούμενων αντικειμένων. Το πρόβλημα αυτό αντιμετωπίζεται εισάγοντας τη σύνταξη των αρχείων

κινούμενων αντικειμένων στην αρχιτεκτονική του HDFS. Έτσι, είναι σε θέση να κατανοήσει τη δομή αποθήκευσής τους και να σέβεται αυτή τη δομή κατά τον διαχωρισμό των αρχείων. Το δεύτερο πρόβλημα είναι ότι το Hadoop δεν παρέχει δομές ευρετηρίου ικανές να περικόψουν ορισμένα δεδομένα πριν από την εκτέλεση μιας λειτουργίας. Η αντιμετώπιση αυτού του προβλήματος γίνεται εισάγοντας μια τεχνική στο HDFS για την καθολική χωροχρονική ευρετηρίαση και τη διαμέριση. Έτσι, είναι σε θέση να φιλτράρει τα δεδομένα πριν τα στείλει σε κόμβους.

Το HadoopTrajectory περιλαμβάνει τα ακόλουθα στοιχεία:

1. **Ευρετηρίαση**, με τη μορφή μιας χωροχρονικής δομής 3DR-tree, που χρησιμοποιείται ως καθολικό ευρετήριο σε ολόκληρο το σύνολο δεδομένων των κινούμενων αντικειμένων και τις τροχιές τους.

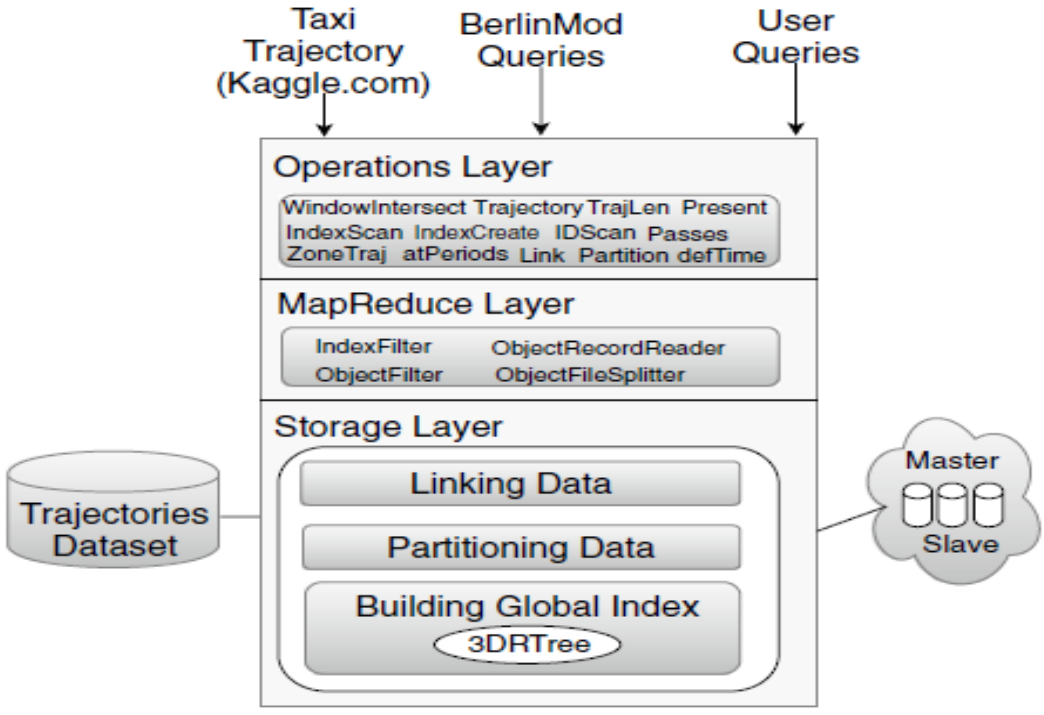
2. **Διαμέριση** των αρχείων μεγάλων δεδομένων των κινούμενων αντικειμένων σε πολλά κομμάτια. Ο διαχωρισμός γίνεται με βάση το μέγεθος της σελίδας του δίσκου, διατηρώντας τη σημασιολογία και τη δομή των κινούμενων αντικειμένων (π.χ., ένα κινούμενο αντικείμενο δεν πρέπει να χωρίζεται σε πολλαπλά τμήματα).

3. **Σύνδεση** κάθε κινούμενου αντικειμένου με τα μεταδεδομένα του που μπορούν να βρίσκονται σε άλλα αρχεία. Βοηθά στην απάντηση ερωτημάτων τόσο σε επίπεδο τροχιάς όσο και σε επίπεδο κινούμενου αντικειμένου. Ένα κινούμενο αντικείμενο μπορεί να έχει πολλαπλές τροχιές, π.χ. ένα αυτοκίνητο που εκτελεί πολλαπλές διαδρομές.

4. **Τελεστές** για την επεξεργασία των τροχιών, συμπεριλαμβανομένου του υπολογισμού των χαρακτηριστικών κίνησης (π.χ. ταχύτητα, κατεύθυνση), δηλώσεις και περιορισμό τροχιάς στο χώρο και το χρόνο. Πολλοί από αυτούς τους τελεστές χρησιμοποιούν το ευρετήριο.

5. **Εργασίες** με τη μορφή της λειτουργίας MapReduce από τον χρήστη, η οποία καλεί τα προαναφερθέντα στοιχεία για την εκτέλεση μεγάλης χωροχρονικής επεξεργασίας των δεδομένων.

Αυτά τα στοιχεία ομαδοποιούνται στα ακόλουθα αρχιτεκτονικά στρώματα, όπως απεικονίζονται στο σχήμα 13. Στη συνέχεια γίνεται μια αναφορά στα επίπεδα και τα στοιχεία τους.



Σχήμα 13. Επισκόπηση του συστήματος HadoopTrajectory *[30]*

• Επίπεδο αποθήκευσης. Επεκτείνεται το HDFS ώστε να μπορεί να διαχειρίζεται χωροχρονικά δεδομένα. Η επέκταση έχει τέσσερα μέρη: τους χωροχρονικούς τύπους, το καθολικό ευρετήριο, τη διαμέριση των δεδομένων του κινούμενου αντικειμένου και τον σύνδεσμο. Ο καθολικός δείκτης είναι ένα χωροχρονικό τρισδιάστατο R-tree που χρησιμοποιείται για την ευρετηρίαση και το φιλτράρισμα των κινούμενων αντικειμένων. Μια εργασία μπορεί να επωφεληθεί από τα πλεονεκτήματα του ευρετηρίου, εάν ορισμένες από τις δηλώσεις της υποστηρίζονται με ευρετήριο. Κατά συνέπεια, μπορεί να γίνει χρήση ενός φίλτρου ευρετηρίου, έτσι ώστε μόνο τα υποψήφια αποτελέσματα να αποστέλλονται στους κόμβους εργασίας. Ο διαχωριστής λειτουργεί σε συνδυασμό με το ευρετήριο για να χωρίσει το αρχείο μεγάλων δεδομένων σε μπλοκ και να συνδέσει τα κινούμενα αντικείμενα μέσα στα αρχεία του μπλοκ με το ευρετήριο. Ο σύνδεσμος χρησιμοποιείται για τη σύνδεση των πληροφοριών τροχιάς του κινούμενου αντικειμένου. Δημιουργούμε ορισμένους τελεστές για τη δημιουργία και τη σάρωση του ευρετηρίου: IndexCreate, IndexScan και MultipleIndexScan. Για την διαμέριση των μεγάλων αρχείων εισόδου, υλοποιούνται το Partition και το IDScan. Τέλος, ο τελεστής σύνδεσης υλοποιείται για τη σύνδεση των τροχιών με άλλα χαρακτηριστικά που μπορεί να υπάρχουν στα αρχεία εισόδου.

• Επίπεδο MapReduce. Το Hadoop MapReduce επεκτείνεται με τις λειτουργίες υποστήριξης που απαιτούνται για την αλληλεπίδραση με τα στοιχεία του επιπέδου αποθήκευσης, π.χ. ευρετήριο, διαμεριστής και σύνδεσμος. Αυτό περιλαμβάνει τις συναρτήσεις: IndexFilter, ObjectFilter, ObjectFileSplitter και ObjectRecordReader. Το IndexFilter σαρώνει το καθολικό ευρετήριο και ανακτά τα αναγνωριστικά των κινούμενων αντικειμένων που τέμνουν ένα δεδομένο χωροχρονικό πλαίσιο. Το ObjectFilter χρησιμοποιείται για ερωτήματα που απαιτούν ένα συγκεκριμένο κινούμενο αντικείμενο με το αναγνωριστικό του. Το ObjectFileSplitter επεκτείνει τη λειτουργία FileSplitter του Hadoop, έτσι ώστε να μπορεί να χωρίσει τα αρχεία εισόδου. Αυτή η συνάρτηση είναι το κύριο συστατικό των τελεστών Partition του επιπέδου αποθήκευσης. Τέλος, το ObjectRecordReader είναι μια επέκταση του RecordReader στο HDFS. Επιτρέπει στο Hadoop να κατανοήσει τις διαφορετικές μορφές/δομές των τροχιών των κινούμενων αντικειμένων.

• Επίπεδο λειτουργίας. Ομαδοποιεί τους χωροχρονικούς τελεστές που μπορούν να ασχοληθούν με τους τύπους επέκτασης.

UlTraMan [17]

Το UlTraMan είναι μια ενοποιημένη πλατφόρμα για τη διαχείριση και ανάλυση των μεγάλων δεδομένων τροχιάς. Το UlTraMan επιτυγχάνει τους σχεδιαστικούς του στόχους, (i) υιοθετώντας έναν ενιαίο μηχανισμό αποθήκευσης και υπολογισμού και (ii) προτείνοντας ένα βελτιωμένο κατανεμημένο υπολογιστικό πρότυπο βασισμένο στο MapReduce με ευέλικτες διεπαφές εφαρμογών.

Ο ενοποιημένος πυρήνας του UlTraMan είναι χτισμένος πάνω στο Apache Spark [40], ένα δημοφιλές κατανεμημένο υπολογιστικό σύστημα. Το Spark επιτρέπει κατανεμημένους υπολογισμούς υψηλής απόδοσης, αλλά, ωστόσο, από μόνο του δεν είναι βέλτιστο για διαχείριση δεδομένων μεγάλης τροχιάς λόγω (i) έλλειψης μηχανισμών ευρετηρίασης, (ii) περιορισμένης και αναποτελεσματικής διατήρησης των δεδομένων του χρόνου εκτέλεσης και (iii) σημαντικής πίεσης που ασκεί στον συλλέκτη σκουπιδιών JVM (Java Virtual Machine)[1]. Για το λόγο αυτό, το UlTraMan χρησιμοποιεί το Chronicle Map [31], έναν ενσωματωμένο χώρο αποθήκευσης κλειδιών-τιμών, στον εσωτερικό διαχειριστή μπλοκ του Spark. Με αυτόν τον τρόπο, το UlTraMan παρέχει αποτελεσματική επεξεργασία και αξιόπιστη διαχείριση των δεδομένων.

Βασισμένη στον ενοποιημένο μηχανισμό αποθήκευσης και υπολογισμού, η συνολική διαδικασία ανάλυσης της τροχιάς μπορεί να δρομολογηθεί στο UlTraMan χωρίς περιττή αντιγραφή και σειριοποίηση των δεδομένων. Το σύστημα επιτρέπει βελτιστοποιήσεις στην αποθήκευση, την

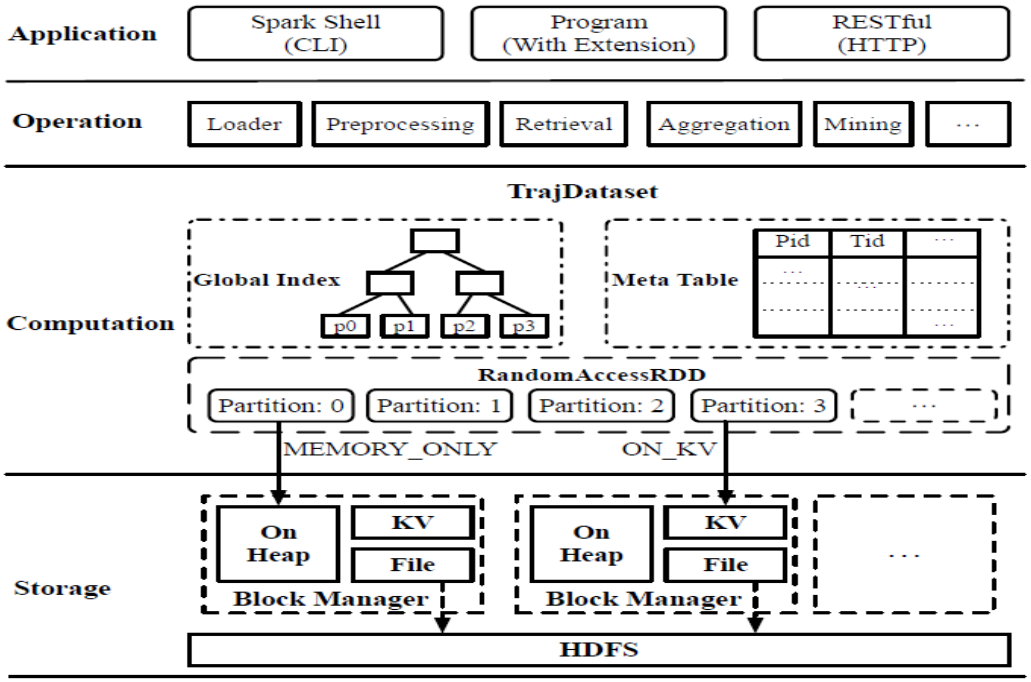
[1] Όσο γίνεται αναφορά σε ένα αντικείμενο, το JVM το θεωρεί ενεργό. Μόλις ένα αντικείμενο δεν αναφέρεται πλέον και ως εκ τούτου δεν είναι προσβάσιμο από τον κώδικα της εφαρμογής, ο συλλέκτης σκουπιδιών το αφαιρεί και ανακτά την αχρησιμοποίητη μνήμη.

επεξεργασία και την ανάλυση. Για παράδειγμα, η οργάνωση δεδομένων και οι δομές ευρετηρίου μπορούν να αλλάξουν σύμφωνα με τις απαιτήσεις ανάλυσης και οι εργασίες υπολογισμού μπορούν να προγραμματιστούν σύμφωνα με την κατανομή των δεδομένων.

Επιπλέον, με τη βοήθεια του ενοποιημένου πυρήνα, βελτιώνεται το MapReduce για να γίνει ένα πιο ισχυρό και ευέλικτο πρότυπο κατανεμημένου υπολογισμού. Το MapReduce [32] και τα Resilient Distributed Datasets (RDD) του Spark [33] υιοθετούν την έννοια του λειτουργικού προγραμματισμού, η οποία διευκολύνει τις διαδοχικές λειτουργίες στα δεδομένα. Αντίθετα, πολλές σημαντικές τεχνικές και βελτιστοποιήσεις πραγματοποιούνται με βάση την τυχαία πρόσβαση σε δεδομένα, όπως οι χάρτες κατακερματισμού και τα ευρετήρια. Λόγω αυτού του γεγονότος ενσωματώνεται στο UlTraMan μια λειτουργία που ονομάζεται TrajDataset. Αυτή η λειτουργία επιτρέπει την τυχαία πρόσβαση τόσο σε τοπικό όσο και σε καθολικό επίπεδο. Σε τοπικό επίπεδο, τα δεδομένα σε κάθε διαμέρισμα μπορούν να προσπελαστούν τυχαία με τη βοήθεια νέων διεπαφών προγραμματισμού που παρέχονται από την ενοποιημένη μηχανή. Σε καθολικό επίπεδο, οι διαμερίσεις δεδομένων διαχειρίζονται ρητά από δύο τύπους γενίκευσης δεδομένων: (i) τα καθολικά ευρετήρια που διατηρούνται σε έναν κόμβο προγράμματος οδήγησης για την οργάνωση των χαρακτηριστικών των μικρών διαμερίσεων και (ii) τους μεταπίνακες που διανέμονται σε εκτελεστές για τη διαχείριση μεγάλων χαρακτηριστικών.

Με βάση την ενοποιημένη μηχανή και τη λειτουργία TrajDataset, το UlTraMan είναι σε θέση να χρησιμεύσει ως μια αποτελεσματική και ευέλικτη πλατφόρμα για ένα μεγάλο αριθμό τεχνικών διαχείρισης και ανάλυσης δεδομένων τροχιάς.

Φυσικά, το UlTraMan υιοθετεί την αρχιτεκτονική που αποτελείται από έναν κύριο κόμβο και πολλούς επιμέρους κόμβους εργασίας. Ο κύριος κόμβος είναι υπεύθυνος για τον προγραμματισμό των εργασιών, ενώ η αποθήκευση των δεδομένων και οι υπολογισμοί κατανέμονται στους κόμβους εργασίας. Εννοιολογικά, όπως απεικονίζεται στο σχήμα 14, η αρχιτεκτονική του συστήματος περιέχει τέσσερα επίπεδα, δηλαδή αποθήκευσης, υπολογισμών, λειτουργίας και εφαρμογής. Μεταξύ αυτών των επιπέδων, τα επίπεδα αποθήκευσης και υπολογισμών αποτελούν τον υποκείμενο ενοποιημένο πυρήνα του UlTraMan, το επίπεδο λειτουργίας προσφέρει επαναχρησιμοποιήσιμα στοιχεία στους προγραμματιστές και το επίπεδο εφαρμογής παρέχει διεπαφές στους τελικούς χρήστες. Παρακάτω περιγράφεται κάθε επίπεδο, λεπτομερώς.



***Σχήμα 14. Επισκόπηση του συστήματος UlTraMan** [17]*

Επίπεδο αποθήκευσης. Η διαχείριση όλων των δεδομένων καθώς και των ευρετηρίων γίνεται στο επίπεδο αποθήκευσης μέσω της επέκτασης της διαχείρισης των μπλοκ. Όταν ένα RDD αποθηκεύεται προσωρινά, ο διαχειριστής μπλοκ σε κάθε εκτελεστή αποθηκεύει τις διαμερίσεις των δεδομένων του σε συστοιχίες σωρού ή σε εκτός σωρού παραδείγματα Chronicle Map, σύμφωνα με το επίπεδο αποθήκευσης που έχει εκχωρηθεί από τους χρήστες. Με βάση αυτόν το μηχανισμό αποθήκευσης, τα δεδομένα σε σωρό και εκτός σωρού μπορούν να προσπελαστούν τυχαία, επιτρέποντας βελτιστοποιήσεις στα ανώτερα επίπεδα.

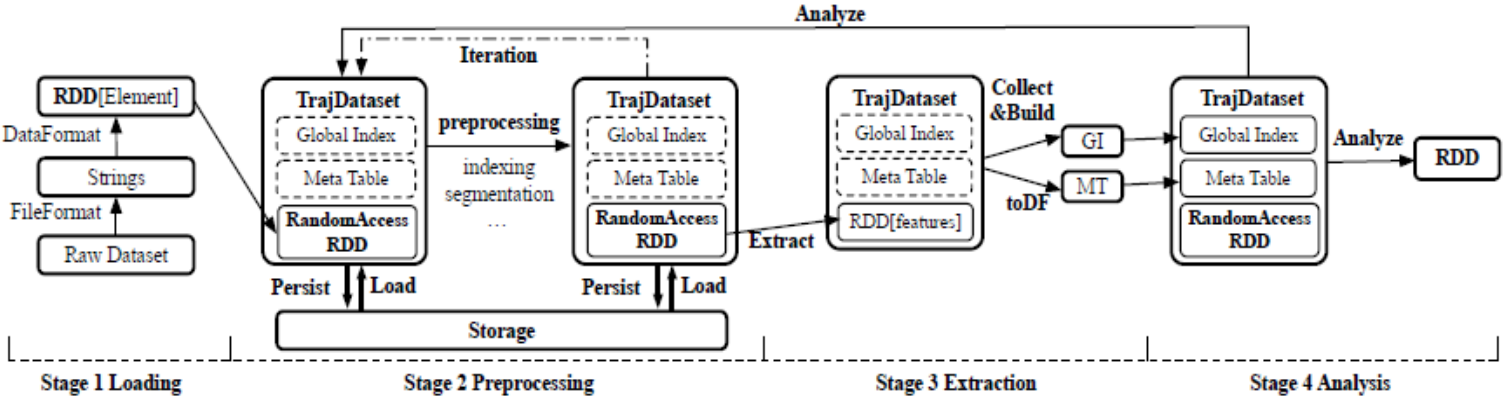
Υπολογιστικό Επίπεδο. Το επίπεδο υπολογισμών υποστηρίζει το κατανεμημένο υπολογισμό χρησιμοποιώντας τη δομή του TrajDataset. Όντας συμβατή με το MapReduce και τα RDD, η δομή TrajDataset αποτελείται από έναν εκτεταμένο τύπο RDD που ονομάζεται RandomAccessRDD και δύο καθολικές δομές: καθολικό ευρετήριο και μεταπίνακας. Τοπικά, το RandomAccessRDD εκμεταλλεύεται το επίπεδο αποθήκευσης για να ενεργοποιήσει την τυχαία πρόσβαση σε προσωρινά δεδομένα. Σε καθολικό επίπεδο, όλες οι διαμερίσεις δεδομένων γενικεύονται σε συγκεκριμένα χαρακτηριστικά σε καθολικά ευρετήρια και μεταπίνακες, έτσι ώστε οι κατανεμημένες εργασίες να μπορούν να προγραμματιστούν σε συγκεκριμένες διαμερίσεις.

Επίπεδο Λειτουργίας. Το UlTraMan προσφέρει διεπαφές προγραμματισμού στο επίπεδο λειτουργίας. Η διαχείριση και η επεξεργασία των δεδομένων τροχιάς γίνεται μέσω λειτουργιών στο TrajDataset, συμπεριλαμβανομένης της φόρτωσης, του φιλτραρίσματος, της χαρτογράφησης, της αναδιαμέρισης κ.λπ. Προηγμένες τεχνικές επεξεργασίας και ανάλυσης δεδομένων μπορούν να υποστηριχθούν με βάση τις διεπαφές. Για την επαλήθευση της ευελιξίας του συστήματος, έχουμε εφαρμόσει πολλές προσθήκες στο UlTraMan, συμπεριλαμβανομένου ενός προγράμματος φόρτωσης αρχείων csv, τμηματοποίησης, ερωτημάτων διαστήματος και συσταδοποίησης.

Επίπεδο Εφαρμογής. Για να λειτουργήσει ως πλατφόρμα, το UlTraMan υποστηρίζει πολλαπλές μεθόδους αλληλεπίδρασης όπως το επίπεδο εφαρμογής. Πιο απλά και διαδραστικά, οι χρήστες μπορούν να εκτελέσουν εργασίες διαχείρισης και ανάλυσης δεδομένων μέσω του Spark. Οι προχωρημένοι προγραμματιστές και χρήστες μπορούν επίσης να υποβάλλουν εργασίες μέσω προγραμμάτων μαζί με προσαρμοσμένα πρόσθετα. Τέλος, το UlTraMan έρχεται με έναν διακομιστή HTTP για να απαντά σε αιτήματα ιστού και να υποστηρίζει την οπτικοποίηση της εφαρμογής.

Συνοπτικά, η αρχιτεκτονική του UlTraMan προσφέρει τρεις κύριες καινοτομίες: (i) εξοπλίζει το μηχανισμό υπολογισμών με μια αξιόπιστη ικανότητα αποθήκευσης για την υποστήριξη της ενοποιημένης επεξεργασίας και διαχείρισης δεδομένων. (ii) επιτρέπει ένα νέο υπολογιστικό μοντέλο για να υποστηρίζει την επεκτάσιμη καθολική-τοπική τυχαία πρόσβαση και (iii) υποστηρίζει σύνθετες αναλύσεις δεδομένων τροχιάς με επεκτάσιμα και επαναχρησιμοποιήσιμα πρόσθετα και τεχνικές.

Με τη βοήθεια της ευέλικτης αρχιτεκτονικής του, το UlTraMan είναι προσαρμόσιμο τόσο όσον αφορά τα πρόσθετα του συστήματος όσο και τις διαδικασίες επεξεργασίας. Η διαδικασία επεξεργασίας των δεδομένων για μια τυπική εργασία ανάλυσης στο UlTraMan περιλαμβάνει τέσσερα στάδια, όπως απεικονίζεται στο σχήμα 15.



Σχήμα 15. Διαδικασία λειτουργίας UlTraMan [17]

Στάδιο 1: Φόρτωση. Στο πρώτο στάδιο, το UlTraMan φορτώνει ένα μη επεξεργασμένο σύνολο δεδομένων τροχιάς και εξάγει τα στοιχεία της τροχιάς στο επίπεδο αποθήκευσης. Το πρωτογενές σύνολο δεδομένων αποθηκεύεται συνήθως σε HDFS, έτσι ώστε να μπορεί να φορτωθεί παράλληλα. Παρέχεται ένα προσαρμοσμένο πρόγραμμα φόρτωσης δεδομένων για την υποστήριξη των διαφορετικών μορφών αρχείων (π.χ. csv ή xml) και μορφών δεδομένων (π.χ. σημείων ή τμημάτων).

Στάδιο 2: Προεπεξεργασία. Το δεύτερο στάδιο περιλαμβάνει ευέλικτες διαδικασίες που εφαρμόζονται πριν από την αποθήκευση και την ανάλυση των δεδομένων τροχιάς, όπως ο μετασχηματισμός της μορφής και η τμηματοποίηση. Επιπλέον, ορισμένες εργασίες διαχείρισης δεδομένων (όπως η κατασκευή ευρετηρίου) εκτελούνται επίσης σε αυτό το στάδιο. Έχοντας υποστεί σωστή επεξεργασία, ένα σύνολο δεδομένων διατηρείται στο UlTraMan, προκειμένου να υποστηρίζεται η αποτελεσματική ανάλυση και η γρήγορη ανάκτηση της αποτυχίας.

Στάδιο 3: Εξαγωγή. Η εξαγωγή είναι ένα ειδικό στάδιο που χρησιμεύει για τη διευκόλυνση της ανάλυσης ενός TrajDataset. Εφόσον το TrajDataset επιτρέπει τον καθολικό προγραμματισμό με βάση τα καθολικά ευρετήρια και τους μεταπίνακες, πρέπει να εξαχθούν και να δημιουργηθούν οι καθολικές πληροφορίες πριν από την ανάλυση. Για παράδειγμα, για να δημιουργηθεί ένα καθολικό R-tree, πρέπει να εξαχθούν και να συλλεχθούν τα χαρακτηριστικά των αναγνωριστικών της διαμέρισης και η οριοθέτησή της. Στη συνέχεια, ένα καθολικό R-tree χτίζεται πάνω στα συλλεγμένα χαρακτηριστικά.

Στάδιο 4: Ανάλυση. Δεδομένου ότι υπάρχει ένα ανοιχτό σύνολο σεναρίων ανάλυσης για δεδομένα τροχιάς, το UlTraMan έχει σχεδιαστεί για να λειτουργεί ως πλατφόρμα που μπορεί να υποστηρίξει τα περισσότερα (αν όχι όλα) από αυτά. Για να ολοκληρωθεί μια πολύπλοκη εργασία, τα στάδια μπορεί να χρειαστεί να εφαρμοστούν επαναληπτικά. Για παράδειγμα, για να πραγματοποιηθεί η εξόρυξη προτύπων ταυτόχρονης κίνησης, τα στάδια προεπεξεργασίας και ανάλυσης διεξάγονται δύο φορές.

DITA [19]

Το DITA αποτελεί ένα κατανεμημένο σύστημα, που λειτουργεί στην εσωτερική μνήμη και περιλαμβάνει απλές διεπαφές για την επεξεργασία των δεδομένων καθώς και εύκολη χρήση της SQL. Γεφυρώνει το χάσμα μεταξύ της περιορισμένης διαθεσιμότητας τεχνικών ανάλυσης τροχιών μεγάλης κλίμακας και της επείγουσας ανάγκης για αποτελεσματική και επεκτάσιμη ανάλυση των δεδομένων τροχιάς στον πραγματικό κόσμο. Ο τρόπος λειτουργίας του συστήματος DITA, συνοπτικά, είναι ο εξής: Πρώτον, για μια τροχιά T επιλέγονται ορισμένα "αντιπροσωπευτικά σημεία" ως άξονες περιστροφής και οι άξονες αυτοί χρησιμοποιούνται για να υπολογίσουμε ένα κατώτερο όριο της απόστασης μεταξύ της τροχιάς που

αντιπροσωπεύεται από αυτούς τους άξονες και μιας άλλης τροχιάς Q. Εάν ένα τέτοιο κατώτερο όριο είναι ήδη μεγαλύτερο από ένα κατώφλι, τότε το T και το Q δεν μπορούν να είναι όμοια.

Προτείνεται μια δομή ευρετηρίασης με τη μορφή δέντρου, για την ευρετηρίαση των αξόνων και τον σχεδιασμό αποτελεσματικού καθολικού και τοπικού ευρετηρίου, όπου το καθολικό ευρετήριο βρίσκει τις σχετικές διαμερίσεις δεδομένων που περιέχουν πιθανά αποτελέσματα και το τοπικό ευρετήριο υπολογίζει τα αποτελέσματα σε κάθε διαμέριση. Στο σύστημα DITA, υλοποιούνται αποτελεσματικοί αλγόριθμοι επαλήθευσης μέσω φίλτρων για τον υπολογισμό των αποτελεσμάτων. Στο βήμα φιλτραρίσματος χρησιμοποιείται ένα φίλτρο για να περικόψει μεγάλο αριθμό ανόμοιων ζευγών και να πάρει ένα σύνολο υποψηφίων αποτελεσμάτων και το βήμα επαλήθευσης χρησιμοποιεί αποτελεσματικές τεχνικές για την επαλήθευση των υποψηφίων αποτελεσμάτων. Τέλος, στο DITA, προτείνεται ένα σταθμισμένο μοντέλο κόστους, όπου χρησιμοποιείται ένας μηχανισμός με προσανατολισμό γραφήματος για τον συντονισμό της κατανεμημένης ένωσης και ένας μηχανισμός εξισορρόπησης φορτίου για να αποτραπούν οι λαθεμένες εκτιμήσεις. Όλες οι τεχνικές μπορούν επίσης να υποστηρίξουν τις περισσότερες από τις υπάρχουσες συναρτήσεις ομοιότητας τροχιάς.

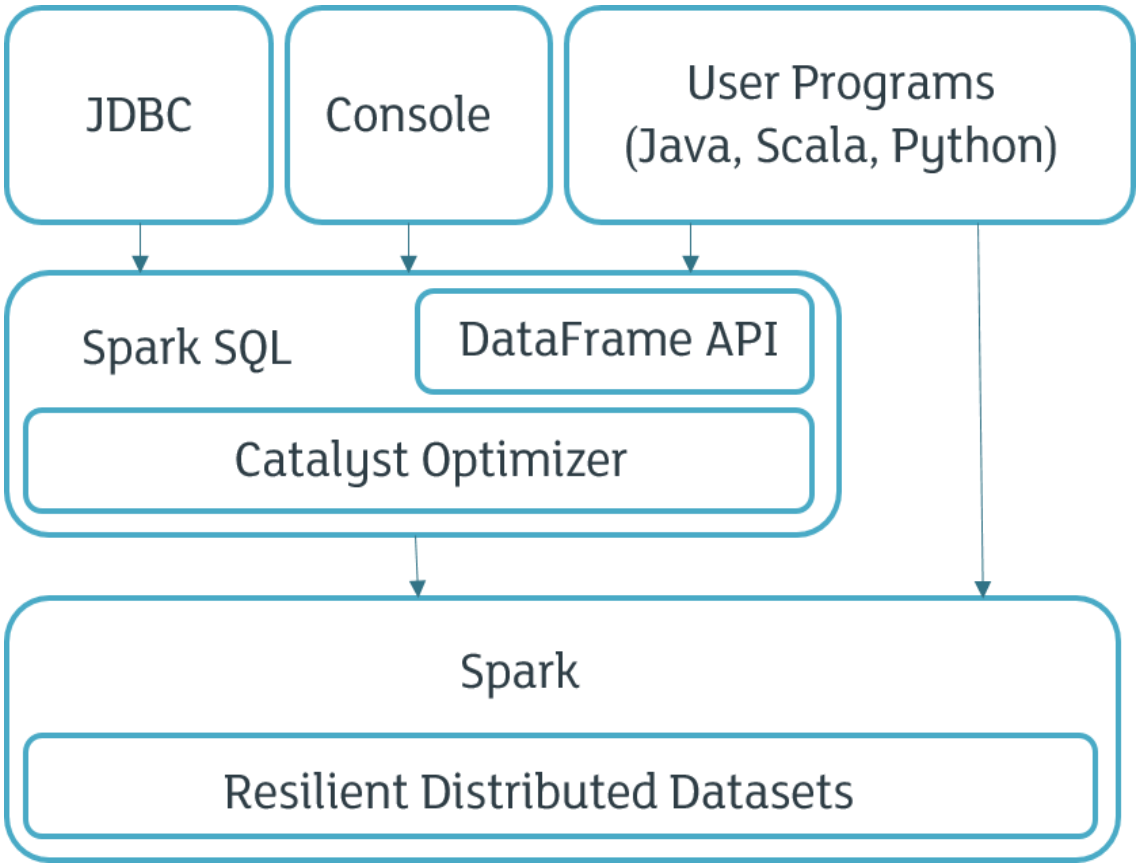
Συνοπτικά οι συνεισφορές του DITA, είναι οι εξής:

1) Αποτελεί ένα πλήρες κατανεμημένο σύστημα ανάλυσης δεδομένων τροχιάς, στην εσωτερική μνήμη, το οποίο είναι μια προέκταση του SPARK SQL. Επιτρέπει τη δημιουργία ευρετηρίου σε RDD και διαθέτει τη γλώσσα SQL καθώς και οθόνη για την ανάλυση της τροχιάς.
2) Επιλέγονται αντιπροσωπευτικά σημεία ως άξονες για τον καλύτερο υπολογισμό των αποτελεσμάτων και σχεδιάζεται μια δομή με τη μορφή δέντρου για την ευρετηρίαση των ανόμοιων τροχιών. Επιπλέον υποστηρίζονται οι περισσότερες από τις συναρτήσεις ομοιότητας τροχιάς.
3) Χρησιμοποιεί την τεχνική φιλτράρισμα-επαλήθευση. Στο βήμα του φιλτραρίσματος, χρησιμοποιούνται σημεία περιστροφής για την εκτίμηση της ομοιότητας μεταξύ δύο τροχιών και την περικοπή των ανόμοιων ζευγών. Στο βήμα της επαλήθευσης, εφαρμόζονται αποτελεσματικές τεχνικές επαλήθευσης.
4) Εφαρμόζονται τεχνικές βελτιστοποίησης με βάση το κόστος για τη μείωση της μετάδοσης μεταξύ των κόμβων εργασίας και την εξισορρόπηση του φόρτου εργασίας.
5) Πραγματοποιείται μια ολοκληρωμένη αξιολόγηση σε σύνολα δεδομένων που αφορούν τον πραγματικό κόσμο. Τα αποτελέσματα δείχνουν ότι το DITA ξεπερνά κατά πολύ τις υπάρχουσες προσεγγίσεις των κατανεμημένων συστημάτων για την εύρεση της τροχιάς και την ένωση των τροχιών.

Το DITA, χρησιμοποιεί την αρχιτεκτονική της δομής του συστήματος SparkSQL, όπως φαίνεται στο σχήμα 16, και επεκτείνει τη σύνταξη της SQL ώστε να μπορεί να υποστηρίξει τις διαδικασίες αναζήτησης όμοιας τροχιάς και ένωσης όμοιων τροχιών.

DataFrame. Επιπρόσθετα με την επέκταση της σύνταξης της SQL, οι χρήστες μπορούν να εκτελούν αυτές τις διαδικασίες μέσω του DataFrame χρησιμοποιώντας μια γλώσσα παρόμοια με την R. Επίσης, η εφαρμογή της DataFrame του Spark, έχει επεκταθεί, για να υποστηριχθεί η αναζήτηση ομοιότητας τροχιάς και η ένωση.

Ευρετήριο. Το DITA επεκτείνει το Spark SQL, ώστε να μπορεί να υποστηρίζει την κατασκευή ευρετηρίου για την αναζήτηση ομοιότητας τροχιάς και την ένωση. Οι χρήστες μπορούν, με τη χρήση ενός ερωτήματος να δημιουργήσουν ένα ευρετήριο με τη μορφή δέντρου (συμπεριλαμβανομένου τόσο του καθολικού όσο και του τοπικού ευρετηρίου).



Σχήμα 16. Δομή DITA

Επεξεργασία ερωτήματος. Με δεδομένο ένα ερώτημα SQL ή ένα αίτημα στην εφαρμογή DataFrame, το σύστημα το μετατρέπει πρώτα σε ένα λογικό σχέδιο και στη συνέχεια το βελτιστοποιεί με βελτιστοποιήσεις που βασίζονται σε κανόνες (π.χ. pushdown κατηγόρημα, συνεχή αναδίπλωση). Στη συνέχεια, το σύστημα δημιουργεί το πιο αποτελεσματικό φυσικό σχέδιο εφαρμόζοντας τόσο τις βελτιστοποιήσεις που βασίζονται στο κόστος όσο και τις εσωτερικές βελτιστοποιήσεις του Spark SQL. Το φυσικό σχέδιο εκτελείται στο Spark για τη δημιουργία των αποτελεσμάτων.

Βελτιστοποίηση ερωτημάτων. Το σύστημα DITA επεκτείνει το εργαλείο βελτιστοποίησης Catalyst του Spark SQL και εισάγει μια ενότητα βελτιστοποίησης βάσει κόστους (CBO) για τη βελτιστοποίηση των ερωτημάτων ομοιότητας τροχιάς. Η ενότητα CBO αξιοποιεί το καθολικό και το τοπικό ευρετήριο για τη βελτιστοποίηση σύνθετων ερωτημάτων SQL.

Distributed MobilityDB [34]

Το MobilityDB [35] είναι ένα σύστημα διαχείρισης βάσεων δεδομένων για κινούμενα αντικείμενα, ειδικά σχεδιασμένο για τη διαχείριση των χωροχρονικών δεδομένων τροχιάς. Αποτελεί επέκταση των PostgreSQL και PostGIS, ώστε να περιλαμβάνει χρονικούς και χωροχρονικούς τύπους βάσεων δεδομένων. Για παράδειγμα, το χρονικό σημείο (*tgeompoint*) χρησιμοποιείται για να αναπαραστήσει την κίνηση στο χρόνο ενός γεωμετρικά σημειακού αντικειμένου, π.χ. ενός οχήματος. Για την αναπαράσταση της ταχύτητάς του με την πάροδο του χρόνου, χρησιμοποιείται η χρονική μεταβλητή (*tfloat*). Εκτός από τους χρονικούς τύπους που αναφέραμε, το MobilityDB παρέχει κάποιους επιπλέον τύπους χρόνου (κυρίως βάσει της της ώρας) για την αναπαράσταση της χρονικής διάστασης, όπως η

χρονική σήμανση (*timestamp*), ένα σύνολο χρονικής σήμανσης (*timestampset*), η περίοδος (*period*) και ένα σύνολο περιόδου (*periodset*). Οι τύποι υποστηρίζονται από μεθόδους πρόσβασης ευρετηρίου, συμπεριλαμβανομένων των GiST (που είναι R-tree) και SP-GiST (που είναι Oct-tree). Το MobilityDB υλοποιεί περισσότερες από 300 λειτουργίες, όπως απόσταση, χωροχρονικές ενώσεις και χρονικά συγκεντρωτικά στοιχεία. Η διεπαφή για τα ερωτήματα είναι SQL. Επί της ουσίας, περιλαμβάνει ένα πρόγραμμα σχεδιασμού ερωτημάτων και ένα εργαλείο βελτιστοποίησης. Τα δυαδικά αρχεία, ο πηγαίος κώδικας και τα εγχειρίδια είναι όλα ανοιχτού κώδικα.

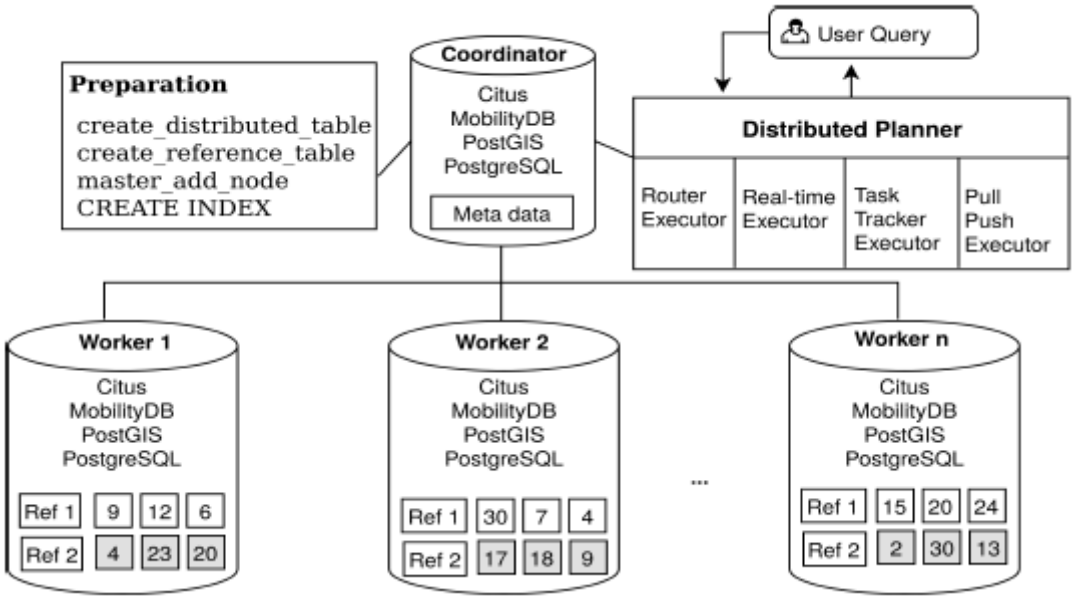
Το MobilityDB αντιμετωπίζει ορισμένες προκλήσεις που προκύπτουν από τα μεγάλα δεδομένα τροχιάς. Τέτοιες προκλήσεις έχουν να κάνουν: α) με την κατανομή των δεδομένων και β) με την κατανομή των ερωτημάτων.

Για την περίπτωση των δεδομένων, υπάρχουν πολλές προκλήσεις λόγω της φύσης των μεγάλων συνόλων δεδομένων. Ορισμένα από αυτά περιέχουν τροχιές που καλύπτουν το μεγαλύτερο μέρος του χώρου και του χρόνου. Για τη διαμέριση αυτών των δεδομένων, η τροχιά μπορεί να εμπίπτει σε μια μεγάλη ομάδα διαμερίσεων, η οποία απαιτεί είτε αντιγραφή της τροχιάς είτε διαχωρισμό της σε κομμάτια. Η πρώτη λύση είναι πιο απλή στην επεξεργασία ερωτημάτων, αλλά θα αύξανε το μέγεθος των δεδομένων και το κόστος δικτύωσης. Η τελευταία, από την άλλη, είναι πιο αποτελεσματική όσον αφορά την αποθήκευση και τη δικτύωση, αλλά θα απαιτούσε την ανακατασκευή των τροχιών κατά το χρόνο εκτέλεσης. Μια άλλη πρόκληση είναι η ασυμμετρία των δεδομένων στο χώρο, στο χρόνο ή και στα δύο. Η μέθοδος διαμέρισης πρέπει να προσαρμόζεται στα χαρακτηριστικά του συνόλου δεδομένων [36] (π.χ. την κατανομή του) και την ανάλυση των διαστάσεών του [37].

Για την περίπτωση των ερωτημάτων, ο στόχος είναι να παραμείνει η κατανομή φανερή στους χρήστες. Έτσι, οι χρήστες θα πρέπει να περιμένουν να γράφουν κανονικά τα ερωτήματα σε SQL όπως στο μη κατανεμημένο περιβάλλον. Ο σχεδιαστής ερωτημάτων πρέπει να επεκταθεί ώστε να κατανοήσει τη σημασιολογία των χωροχρονικών τύπων και λειτουργιών, όπως επίσης πρέπει να επεκταθεί και με μεθόδους για τη διανομή τους.

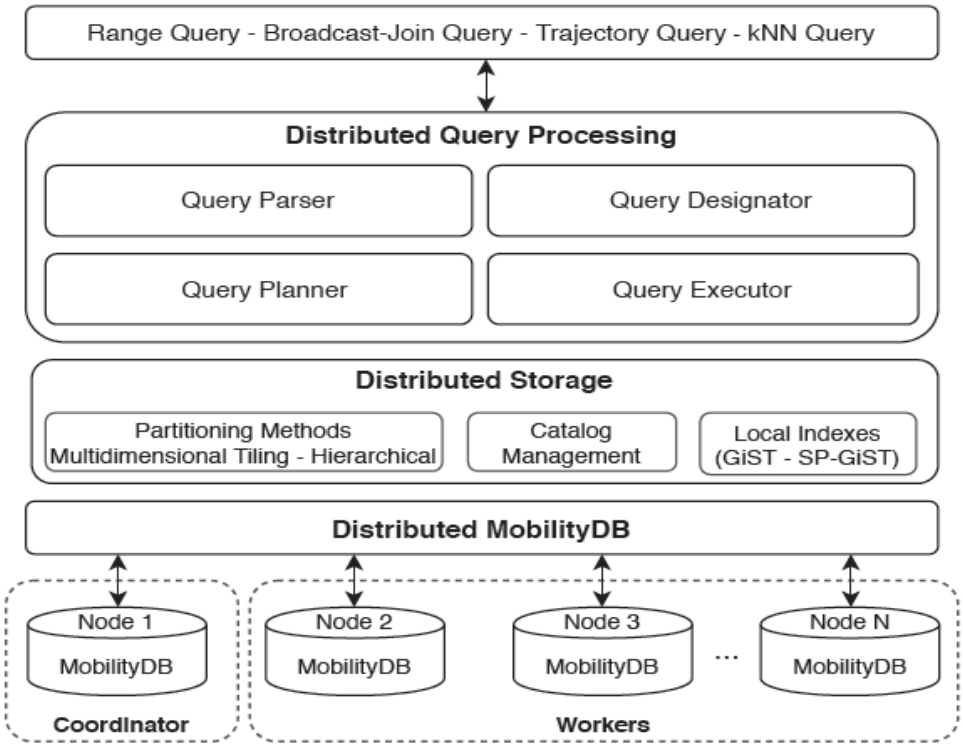
Για το λόγο αυτό η έκδοση του MobilityDB, την οποία θα περιγράψουμε, αποτελεί μια κατανεμημένη έκδοση του αρχικού συστήματος.

Στο σχήμα 17 απεικονίζεται η δομή της κατανεμημένης έκδοσης του MobilityDB. Αποτελείται από ένα σύμπλεγμα οντοτήτων MobilityDB (κόμβοι), όπου μία από αυτές λειτουργεί ως συντονιστής. Μια οντότητα MobilityDB είναι μια βάση δεδομένων PostgreSQL, που έχει εγκατεστημένες τόσο τις επεκτάσεις PostGIS όσο και τις επεκτάσεις MobilityDB. Ένα φυσικό μηχάνημα μπορεί επομένως να φιλοξενήσει πολλαπλές παρουσίες MobilityDB. Ο συντονιστής έχει μια πρόσθετη επέκταση που ονομάζεται διαχειριστής κατανομής (distribution manager). Οι άλλοι κόμβοι, που ονομάζονται κόμβοι εργασίας, είναι απλές οντότητες MobilityDB. Στην πραγματικότητα δεν γνωρίζουν ότι αποτελούν μέρος ενός κατανεμημένου συμπλέγματος. Αυτό σημαίνει ότι εκτός από τον ρόλο τους ως κόμβου εργασίας στο κατανεμημένο σύμπλεγμα, μπορεί να εξυπηρετούν ανεξάρτητα και άλλους πελάτες βάσεων δεδομένων. Όλη η λογική της κατανομής εκτελείται επομένως από τον διαχειριστή κατανομής στον κόμβο συντονιστή.



Σχήμα 17. Επισκόπηση του συστήματος MobilityDB *[34]*

Ο συντονιστής διατηρεί έναν κατάλογο μεταδεδομένων σχετικά με τη δομή του συμπλέγματος και τις λογικές και φυσικές διαμερίσεις δεδομένων. Η φυσική αποθήκευση των δεδομένων πραγματοποιείται στους κόμβους εργασίας. Έτσι, ένας κόμβος εργασίας θα αποθηκεύει σαν πίνακες βάσης δεδομένων, τμήματα των μεγάλων πινάκων, αντίγραφα διαμερίσεων που είναι αποθηκευμένα σε άλλους κόμβους εργασίας και αντίγραφα μικρότερων πινάκων αναφοράς.



Σχήμα 18. Διαχειριστής κατανομής *[35]*

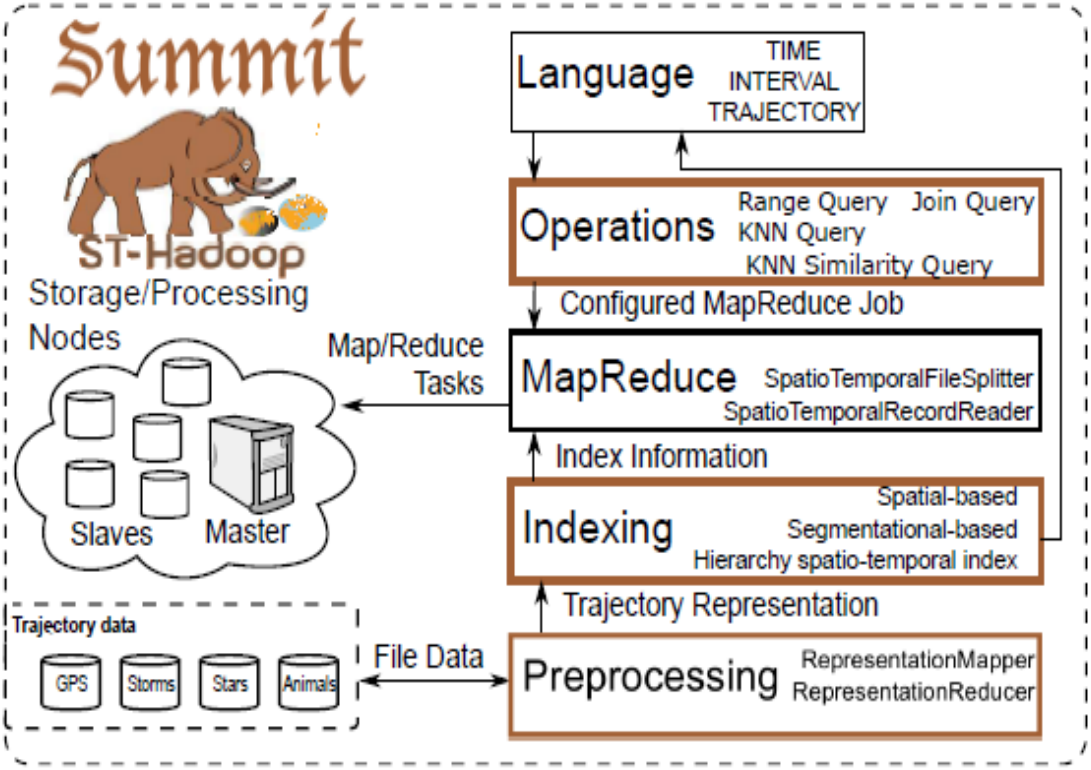
Στο σχήμα 18 απεικονίζονται τα στοιχεία του διαχειριστή κατανομής, τα οποία περιγράφονται εν συντομία ως εξής:

Κατανεμημένη αποθήκευση. Η αποθήκευση οργανώνεται σύμφωνα με μία από τις προτεινόμενες μεθόδους κατανομής των δεδομένων τροχιάς. Για να χωρίσει έναν πίνακα, ο διαχειριστής κατανομής δημιουργεί το ίδιο σχήμα πίνακα στον συντονιστή και σε όλους τους κόμβους εργασίας. Ο συντονιστής παραμένει κενός και τα δεδομένα κατανέμονται στους πίνακες των κόμβων εργασίας. Τα μεταδεδομένα για κάθε τμήμα αποθηκεύονται στον κατάλογο μεταδεδομένων. Επιπλέον, είναι δυνατή η δημιουργία ενός τοπικού ευρετηρίου (R-tree, Oct-tree ή B-tree), όπου κάθε κόμβος εργασίας ευρετηριάζει μόνο τα αντικείμενά του. Όλα αυτά πραγματοποιούνται από συναρτήσεις SQL που ορίζονται στο διαχειριστή κατανομής. Σχετικά με τον κατάλογο, χρησιμοποιείται από το διαχειριστή κατανομής για να αποθηκεύσει τα μεταδεδομένα σχετικά με τους κόμβους του συμπλέγματος, τους κατανεμημένους/αντεγραμμένους πίνακες, τις διαμερίσεις και τις δηλώσεις φιλτραρίσματος. Αυτές οι πληροφορίες βοηθούν τον προγραμματιστή να διανείμει και να βελτιστοποιήσει τα ερωτήματα των χρηστών. Στους κόμβους του συμπλέγματος, αποθηκεύονται οι πληροφορίες για τις ικανότητες του κάθε κόμβου. Για τους κατανεμημένους/αντεγραμμένους πίνακες, αποθηκεύονται πληροφορίες για τη μέθοδο των διαμερίσεων και στατιστικά στοιχεία για κάθε πίνακα, όπως την έκταση και τον αριθμό των διαμερίσεων. Όσον αφορά τις πληροφορίες των διαμερίσεων, αποθηκεύονται δεδομένα όπως ο αριθμός του κόμβου και το χωροχρονικό πλαίσιο. Τέλος, αποθηκεύονται οι δηλώσεις φιλτραρίσματος που θα χρησιμοποιηθούν από το χωροχρονικό φίλτρο.

Κατανεμημένη επεξεργασία ερωτημάτων. Είναι μια ακολουθία τεσσάρων κύριων στοιχείων: (1) Ένας αναλυτής ερωτήματος για τον προσδιορισμό στοιχείων του ερωτήματος, (2) ένας προσδιοριστής ερωτήματος που αναλύει το ερώτημα και επιλέγει τους κόμβους που θα κληθούν κατά την εκτέλεση του ερωτήματος, (3) ένα πρόγραμμα σχεδιασμού ερωτημάτων για τη δημιουργία ενός κατανεμημένου σχεδίου ερωτημάτων, σύμφωνα με τις δηλώσεις του ερωτήματος και (4) πολλαπλοί εκτελεστές ερωτημάτων για την παρακολούθηση της εκτέλεσης του κατανεμημένου σχεδίου στους κόμβους, την αντιμετώπιση αστοχιών των κόμβων και τη συλλογή των αποτελεσμάτων.

Summit [38]

Το Summit είναι μια πλήρης, ανοιχτού κώδικα, βιβλιοθήκη τροχιών, η οποία χρησιμοποιεί το σύστημα MapReduce και συμπεριλαμβάνεται στο σύστημα ST-Hadoop [39]. Η εκμετάλλευση της αποτελεσματικότητας του Hadoop, σε συνδυασμό με την ευελιξία του συστήματος MapReduce για την επεξεργασία των τροχιών έθεσε πολλές προκλήσεις. Μερικές από τις πιο σημαντικές προκλήσεις της επεξεργασίας δεδομένων τροχιάς είναι η αδυναμία του Hadoop να διατηρήσει τη χωροχρονική τοποθεσία, την απόδοση εξισορρόπησης φορτίου και την ικανότητα υποστήριξης διαφόρων λειτουργιών τροχιάς. Το Summit ξεπερνά όλες αυτές τις προκλήσεις με χωροχρονικά φορτία και διαμερίσεις των δεδομένων τροχιάς. Το Summit είναι εξοπλισμένο με θεμελιώδεις λειτουργίες όπως ερωτήματα διαστήματος, ερωτήματα πλησιέστερων γειτόνων (kNN), ερωτήματα ομοιότητας και ερωτήματα ενώσεων. Το Summit είναι, ουσιαστικά, μια επέκταση του συστήματος Hadoop που ενσωματώνει τη χωροχρονική τοποθεσία της τροχιάς στον βασικό κώδικα τριών επιπέδων μέσα στο ST-Hadoop, δηλαδή τα επίπεδα προεπεξεργασίας, ευρετηρίασης και λειτουργίας. Το βασικό σημείο πίσω από το κέρδος της απόδοσης του Summit είναι η ιδέα της ευρετηρίασης, όπου τα δεδομένα φορτώνονται χωροχρονικά και διαιρούνται ως τμήματα τροχιάς στους κόμβους υπολογισμού.



Σχήμα 19. Επισκόπηση συστήματος Summit *[38]*

Στο σχήμα 19 απεικονίζεται η αρχιτεκτονική του συστήματος Summit. Διαθέτει ενσωματωμένη υποστήριξη για δεδομένα τροχιάς και τις λειτουργίες τους. Ένα σύμπλεγμα Summit περιέχει έναν κύριο κόμβο που χωρίζει μια εργασία map-reduce σε μικρότερες εργασίες, που εκτελούνται από επιμέρους κόμβους εργασίας. Το Summit υιοθετεί την πολυεπίπεδη λογική του ST-Hadoop που περιγράφεται παρακάτω:

Επίπεδο γλώσσας: Αυτό το επίπεδο παρέχει μια απλή γλώσσα υψηλού επιπέδου που μοιάζει με SQL, η οποία υποστηρίζει τους τύπους δεδομένων τροχιάς.

Επίπεδο προεπεξεργασίας: Αυτό το επίπεδο είναι υπεύθυνο για την ανακατασκευή της ακατέργαστης αναπαράστασης αντικειμένων σε τμήματα τροχιάς, όπου το καθένα περιέχει μια συνεχή ακολουθία χωροχρονικών σημείων.

Επίπεδο ευρετηρίου: Το Summit χρησιμοποιεί μια δομή ευρετηρίου δύο επιπέδων καθολικής και τοπικής ευρετηρίασης. Το καθολικό ευρετήριο χωρίζει τα δεδομένα στους υπολογιστικούς κόμβους, ενώ το τοπικό ευρετήριο οργανώνει τα δεδομένα μέσα σε κάθε κόμβο. Ο χώρος και ο χρόνος των τροχιών λαμβάνονται υπόψη σε κάθε επίπεδο.

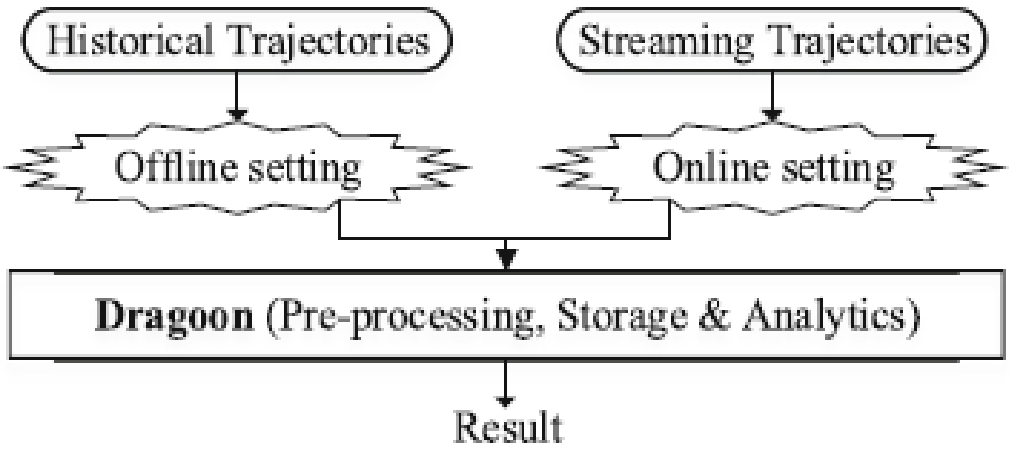
Επίπεδο MapReduce: Το πρωταρχικό καθήκον αυτού του επιπέδου είναι να εκμεταλλευτεί τα καθολικά και τα τοπικά ευρετήρια, αντίστοιχα, για την περικοπή δεδομένων.

Επίπεδο λειτουργιών: Αυτό το επίπεδο ενσωματώνει την υλοποίηση τριών βασικών λειτουργιών για τα δεδομένα τροχιάς, συγκεκριμένα, ερωτήματα διαστήματος, ερωτήματα kNN και ερωτήματα ένωσης. Σε αυτό το επίπεδο μπορούν να προστεθούν, στο μέλλον, περισσότερες λειτουργίες.

Dragoon [40]

Το Dragoon είναι ένα νέο υβριδικό και αποτελεσματικό σύστημα διαχείρισης μεγάλων δεδομένων τροχιάς για ανάλυση εκτός σύνδεσης (offline) και σε σύνδεση (online). Το Dragoon έχει σχεδιαστεί για να είναι μια ολιστική λύση, όπως απεικονίζεται στο σχήμα 20, η οποία υποστηρίζει την πλήρη διαδικασία

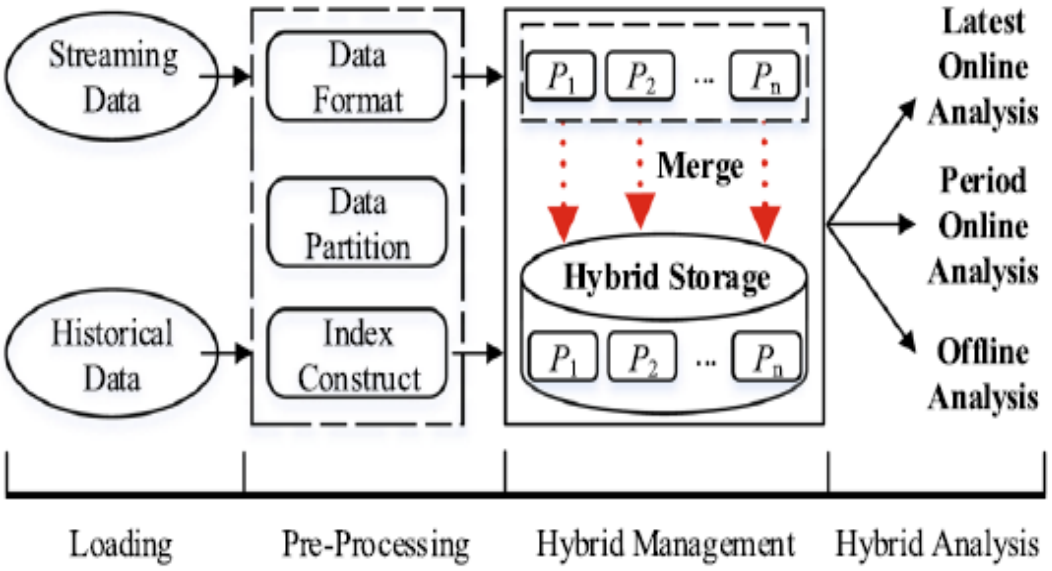
προεπεξεργασίας, διαχείρισης και ανάλυσης δεδομένων τροχιάς, τόσο παρελθοντικών όσο και ροών δεδομένων τροχιάς, σε ένα ενιαίο σύστημα. Υπάρχουν δύο προσεγγίσεις για την ανάπτυξη ενός τέτοιου συστήματος.



Σχήμα 20. Διαδικασία ανάλυσης υβριδικών δεδομένων τροχιάς *[40]*

Η πρώτη προσέγγιση είναι να επεκτείνει ένα υπάρχον μεγάλο σύνολο ή ένα εκτός σύνδεσης σύστημα (π.χ. Spark [33] για να βελτιώσει τις δυνατότητες που έχει για την επεξεργασία σε πραγματικό χρόνο των τροχιών συνεχόμενης ροής. Η δεύτερη προσέγγιση είναι να επεκταθεί ένα υπάρχον διαδικτυακό ή προσανατολισμένο στη ροή σύστημα (π.χ. Flink [9], εξοπλίζοντάς το με τη δυνατότητα διαχείρισης και επεξεργασίας μεγάλης κλίμακας δεδομένων παρελθοντικής τροχιάς. Το σύστημα Dragoon υιοθετεί την πρώτη προσέγγιση. Συγκεκριμένα, επιλέγεται η επέκταση της πλατφόρμας Spark για να κατασκευαστεί το σύστημα Dragoon, καθώς το Spark είναι ένα ευρέως χρησιμοποιούμενο και αποτελεσματικό, εκτός σύνδεσης, κατανεμημένο σύστημα επεξεργασίας. Επιπλέον υπάρχει το Spark Streaming, το οποίο επεκτείνει επίσης το Spark για την επεξεργασία δεδομένων συνεχόμενης ροής. Ωστόσο, μπορεί να χρησιμοποιηθεί μόνο για την επεξεργασία μιας τροχιάς συνεχόμενης ροής με έναν ευθύ τρόπο, δηλαδή, απαιτεί μια στρατηγική που κόβει τις συνεχόμενες ροές των τροχιών σε πολλά RDD δεδομένων και διαχειρίζεται αυτά τα RDD χωριστά. Αντίθετα, το Dragoon αποτελεί ένα ενιαίο σύστημα για τη διαχείριση παρελθοντικών τροχιών και τροχιών συνεχόμενης ροής με υβριδικό τρόπο, ενισχύοντας το επίπεδο αποθήκευσης του Spark, βάσει του οποίου επιτρέπονται καλύτερες αναλύσεις για τις υβριδικές τροχιές.

Στο σχήμα 21 παρουσιάζεται μια επισκόπηση του Dragoon, συμπεριλαμβανομένης της φόρτωσης, της προεπεξεργασίας, της υβριδικής διαχείρισης και της υβριδικής ανάλυσης.



Σχήμα 21. Επισκόπηση του συστήματος Dragoon *[40]*

Φόρτωση. Στο πρώτο στάδιο, το Dragoon απορροφά τα δεδομένα της παρελθοντικής τροχιάς από μια πηγή δεδομένων στατικής τροχιάς ή φορτώνει συνεχώς τα πιο πρόσφατα δεδομένα τροχιάς από μια πηγή συνεχόμενης ροής τροχιάς.

Προεπεξεργασία. Η προεπεξεργασία περιλαμβάνει το μετασχηματισμό της μορφής, το διαμερισμό των δεδομένων και την κατασκευή ευρετηρίου. Πρώτον, τα πρωτογενή δεδομένα τροχιάς αντιπροσωπεύονται ως μια ακολουθία εγγραφών GPS (*id, l, t*), όπου το id υποδηλώνει το αναγνωριστικό του κινούμενου αντικειμένου, το l αντιπροσωπεύει μια τοποθεσία που περιέχει γεωγραφικό πλάτος και μήκος και t είναι ο χρόνος που παρατηρήθηκε η τοποθεσία . Στη συνέχεια, τόσο οι ιστορικές τροχιές όσο και οι τροχιές συνεχόμενης ροής μπορούν να χωριστούν σε πολλές διαμερίσεις δεδομένων σύμφωνα με συγκεκριμένους κανόνες διαμέρισης. Όπως φαίνεται στο σχήμα 17, τα δεδομένα τροχιάς κατανέμονται σε n διαμερίσεις δεδομένων Pi ($1 \le i \le n$). Τέλος, η κατασκευή ευρετηρίου περιλαμβάνει την κατασκευή τοπικού ευρετηρίου σε κάθε διαμέριση δεδομένων και την κατασκευή καθολικού ευρετηρίου σε όλες τις διαμερίσεις των δεδομένων.

Υβριδική διαχείριση. Για τη διαχείριση των δεδομένων παρελθοντικής τροχιάς και συνεχόμενης ροής δεδομένων, σχεδιάζεται ένας υβριδικός χώρος αποθήκευσης, ο οποίος είναι το βασικό μέρος του Dragoon. Η υβριδική αποθήκευση βασίζεται φυσικά στο Chronicle Map. Για να αποθηκευτούν τα δεδομένα υβριδικής τροχιάς, πρέπει να συγχωνευτούν η ροή και οι παρελθοντικές τροχιές. Η διαδικασία συγχώνευσης δεν είναι μια απλή λειτουργία ένωσης όπως εφαρμόζεται στο Spark, αλλά πρέπει να μπορεί να υποστηρίζει ενημερώσεις δεδομένων στο φυσικό επίπεδο του Chronicle Map. Για να υποστηρίξει ενημερώσεις δεδομένων, έχει σχεδιαστεί το μοντέλο mRDD. Επιπλέον, τα τοπικά ευρετήρια και το καθολικό ευρετήριο αποθηκεύονται επίσης στο Chronicle Map και πρέπει να ενημερωθούν μετά την επεξεργασία της ενημέρωσης δεδομένων.

Υβριδική ανάλυση. Η υβριδική ανάλυση περιλαμβάνει ανάλυση εκτός σύνδεσης για δεδομένα παρελθοντικής τροχιάς και διαδικτυακή ανάλυση για δεδομένα τροχιάς συνεχόμενης ροής. Συγκεκριμένα, το Dragoon υποστηρίζει δύο τύπους διαδικτυακών αναλύσεων, όπως απεικονίζονται στο σχήμα 20. Ο πρώτος τύπος διαδικτυακής ανάλυσης, που ονομάζεται τελευταία διαδικτυακή ανάλυση, εστιάζει στις πιο πρόσφατες τιμές δεδομένων τοποθεσίας των παρατηρούμενων κινούμενων αντικειμένων. Ο δεύτερος τύπος διαδικτυακής ανάλυσης, που ονομάζεται διαδικτυακή ανάλυση περιόδου, εκτελείται τόσο στα πιο πρόσφατα δεδομένα τοποθεσίας όσο και στα προηγούμενα

παρελθοντικά δεδομένα τροχιάς των κινούμενων αντικειμένων (δηλαδή τα δεδομένα που σχετίζονται με μια ορισμένη πρόσφατη χρονική περίοδο).

Υβριδική Αποθήκευση. Τα δεδομένα τροχιάς συνεχόμενης ροής συλλέγονται συνεχώς. Αντί να χρησιμοποιεί τη στρατηγική mini-batch που διαιρεί τα εισερχόμενα δεδομένα σε μικρότερες ομάδες και τα διαχειρίζεται ξεχωριστά, το σύστημά Dragoon προσθέτει τα εισερχόμενα δεδομένα στις προηγούμενες διαμερίσεις δεδομένων, γεγονός που επιφέρει μια ενημέρωση στον υποκείμενο χώρο αποθήκευσης για κάθε εισερχόμενο σημείο τροχιάς. Ωστόσο, τα αρχικά RDD του Spark είναι αμετάβλητα και οποιαδήποτε ενημέρωση σε ένα RDD θα δημιουργήσει ένα νέο RDD, με αποτέλεσμα υψηλό κόστος αποθήκευσης λόγω περιττών αντιγράφων των δεδομένων. Ως εκ τούτου, ο τρόπος υποστήριξης συχνών ενημερώσεων RDD των δεδομένων συνεχόμενης ροής είναι το κλειδί για την υλοποίηση της υβριδικής αποθήκευσης. Ενόψει αυτού, στο Dragoon υλοποιείται ένα μεταβλητό μοντέλο RDD (mRDD) που έχει σχεδιαστεί για να είναι συμβατό με το αρχικό μοντέλο RDD του Spark. Το μοντέλο mRDD περιλαμβάνει τρία μέρη, δηλαδή τα, RDD Share, RDD Update και RDD Mirror, προκειμένου να υποστηρίζεται η υβριδική αποθήκευση τόσο για παρελθοντικά δεδομένα όσο και για δεδομένα τροχιάς συνεχόμενης ροής. Για να γίνεται η διάκριση μεταξύ τους, τα mRDD χρησιμοποιούνται για να δηλωθεί το προτεινόμενο μοντέλο mRDD του Dragoon, ενώ χρησιμοποιούνται τα RDD για να αντιπροσωπευθεί το αρχικό μοντέλο RDD του Spark.

TrajSpark [41]

Το TrajSpark (Trajectory on Spark) είναι ένα επεκτάσιμο σύστημα που υποστηρίζει ερωτήματα χαμηλής καθυστέρησης σε δεδομένα μεγάλης τροχιάς, στην εσωτερική μνήμη. Το TrajSpark προτείνει μια νέα μέθοδο που ονομάζεται IndexTRDD για τη διαχείριση των τροχιών ως ένα σύνολο τμημάτων τροχιάς. Για να επιταχύνει την επεξεργασία των ερωτημάτων, εισάγει τον καθολικό και τοπικό μηχανισμό ευρετηρίασης που ενσωματώνει έναν τοπικό κατακερματισμό του ευρετηρίου σε κάθε διαμέριση των δεδομένων και δημιουργεί ένα καθολικό ευρετήριο πάνω σε αυτές τις διαμερίσεις. Επιπλέον, το TrajSpark παρακολουθεί την αλλαγή της κατανομής των δεδομένων χρησιμοποιώντας ένα μοντέλο χρονικής αποσύνθεσης για να υποστηρίζει συνεχώς την αποτελεσματική διαχείριση των καθημερινά αυξανόμενων δεδομένων μεγάλης τροχιάς. Επιπλέον, στο TrajSpark εκτελούνται τρεις τύποι ερωτημάτων πάνω στα δεδομένα τροχιάς.

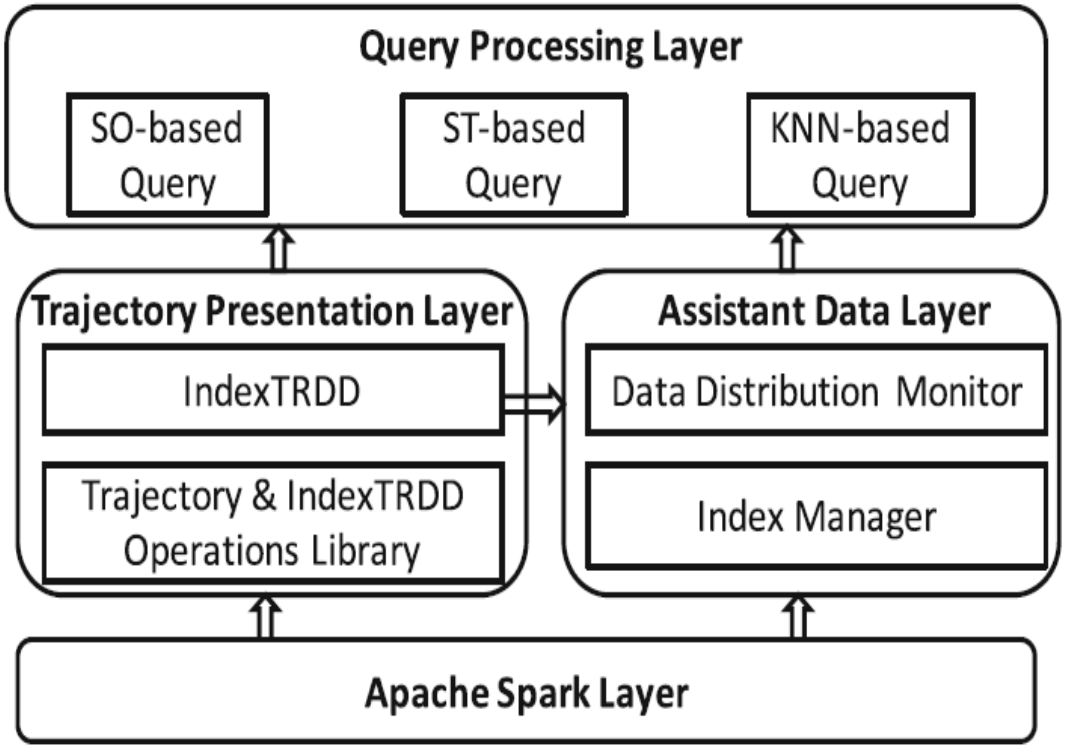
Η αρχιτεκτονική του TrajSpark, όπως φαίνεται στο σχήμα 22, αποτελείται από τέσσερα επίπεδα: (1) Επίπεδο Apache Spark, όπου οι τακτικές λειτουργίες και οι μηχανισμοί ανοχής σφαλμάτων υποστηρίζονται από το Apache Spark, (2) Επίπεδο Παρουσίασης Τροχιάς, όπου μια νέα μέθοδος που ονομάζεται IndexTRDD έχει σχεδιαστεί για να υποστηρίζει ευρετήρια πάνω στα δεδομένα τροχιάς. (3) Βοηθητικό Επίπεδο Δεδομένων, το οποίο παρακολουθεί την αλλαγή της κατανομής των δεδομένων και καθοδηγεί τη διαμέριση των προσεχών δεδομένων. Διατηρείται ένα καθολικό ευρετήριο που ευρετηριάζει τις διαμερίσεις του IndexTRDD. (4) Επίπεδο επεξεργασίας ερωτημάτων, το οποίο επεξεργάζεται τα ερωτήματα τροχιάς με αποτελεσματικό τρόπο χρησιμοποιώντας τα ευρετήρια.

– Επίπεδο Apache Spark: Αυτό το επίπεδο κληρονομείται απευθείας από το Apache Spark.

– Επίπεδο παρουσίασης τροχιάς: Σε αυτό το επίπεδο, τα τμήματα τροχιάς που είναι χωροχρονικά κοντά θα ομαδοποιηθούν στην ίδια διαμέριση δεδομένων. Σε κάθε διαμέριση, τα τμήματα που ανήκουν στο ίδιο κινούμενο αντικείμενο απεικονίζονται σε μορφή αποδοτικού χώρου. Μια νέα μέθοδος που ονομάζεται IndexTRDD προτείνεται για την οργάνωση όλων αυτών των τμημάτων και παρέχεται μια πλούσια βιβλιοθήκη λειτουργιών για τον χειρισμό των τροχιών και του IndexTRDD.

– Βοηθητικό Επίπεδο Δεδομένων: Σε αυτό το επίπεδο διατηρούνται μερικά στατιστικά στοιχεία για την καταγραφή της αλλαγής της κατανομής των δεδομένων. Αυτό το επίπεδο διατηρεί επίσης ένα καθολικό ευρετήριο που ευρετηριάζει όλα τα διαμερίσματα του IndexTRDD.

– Επίπεδο Επεξεργασίας Ερωτημάτων: Εισάγουμε την υλοποίηση τριών τυπικών ερωτημάτων τροχιάς σε αυτό το επίπεδο.



Σχήμα 22. Επισκόπηση συστήματος TrajSpark *[41]*

JUST-Traj [42]

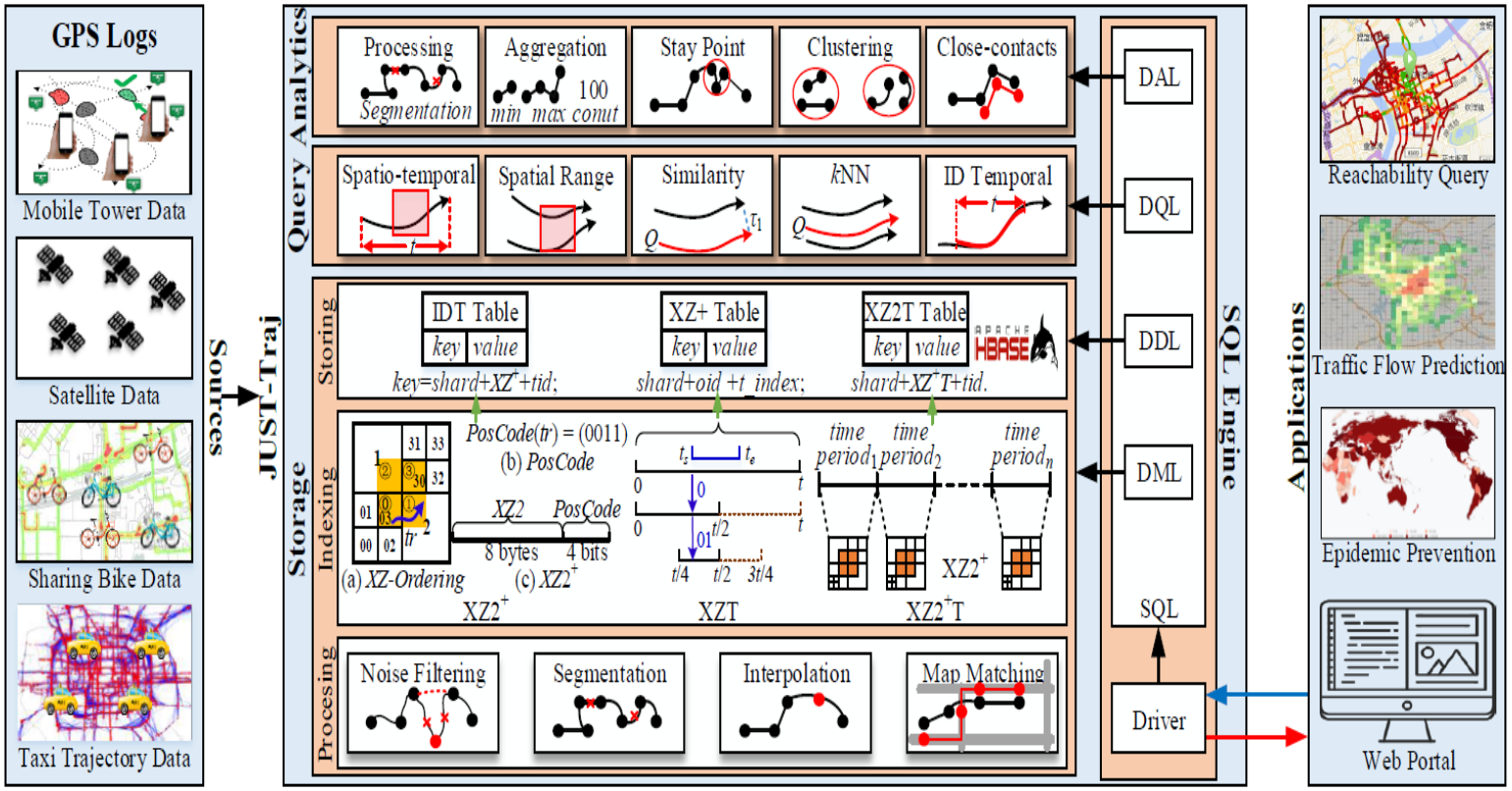
Το JUST-Traj είναι ένα κατανεμημένο και ολιστικό σύστημα διαχείρισης δεδομένων τροχιάς, που βασίζεται στο JUST και στο TrajMesa. Το JUST παρέχει μια ενοποιημένη πλατφόρμα που βασίζεται στο Spark και ένα χώρο αποθήκευσης δεδομένων NoSQL. Μπορεί να εκτελέσει ερωτήματα χωροχρονικών δεδομένων και αναλύσεις μέσω μιας βολικής μηχανής SQL. Το TrajMesa παρέχει τρία χωροχρονικά ευρετήρια για την αποτελεσματική αποθήκευση και αναζήτηση των τροχιών σε ένα χώρο αποθήκευσης δεδομένων NoSQL (δηλαδή, HBase).

Τα πλεονεκτήματα του συστήματος συνοψίζονται ως εξής:

• Το JUST-Traj είναι ένα κατανεμημένο και ολιστικό σύστημα με αποτελεσματική διαχείριση δεδομένων τεράστιας τροχιάς.

• Το JUST-Traj παρέχει μια πλήρη μηχανή SQL για εύκολη λειτουργία (δηλαδή αποθήκευση, αναζήτηση, ανάλυση) μεγάλων δεδομένων τροχιάς.

Το σχήμα 23 δίνει μια επισκόπηση του JUST-Traj, το οποίο περιέχει τέσσερα βασικά επίπεδα: (1) αποθήκευσης, το JUST-Traj αποθηκεύει τις τροχιές σε μια βάση δεδομένων NoSQL με τρία βήματα, δηλαδή, προεπεξεργασία, ευρετηρίαση και αποθήκευση. (2) ερωτημάτων, το JUST-Traj παρέχει πολλά χρήσιμα χωροχρονικά ερωτήματα για τροχιές. (3) ανάλυσης στοιχείων, το JUST-Traj παρέχει πολλές χρήσιμες λειτουργίες ανάλυσης για αστικές εφαρμογές, π.χ. επεξεργασία, συνάθροιση, ανίχνευση σημείων παραμονής, ομαδοποίηση και παρακολούθηση στενών επαφών. (4) Μηχανή SQL, υλοποιείται μια πλήρης μηχανή SQL με πολλές προκαθορισμένες λειτουργίες out-of-the-box, βάσει των οποίων όλες οι λειτουργίες (δηλαδή αποθήκευση, ερώτημα και αναλυτικά στοιχεία) μπορούν να εκτελεστούν μέσω μιας δήλωσης ερωτήματος παρόμοιας με την SQL.



***Σχήμα 23. Επισκόπηση συστήματος JUST-Traj** [42]*

Όπως φαίνεται στο σχήμα 23 (Storing), το JUST-Traj προ-επεξεργάζεται και ευρετηριάζει τις πρωτογενείς τροχιές στο Spark, και στη συνέχεια αποθηκεύει τις μεγάλες τροχιές στο HBase.

Είναι απαραίτητο να γίνει μια προεπεξεργασία της τροχιάς, καθώς ο θόρυβος δεδομένων και ο ρυθμός δειγματοληψίας των πρωτογενών αρχείων καταγραφής μιας συσκευής GPS μπορεί να επηρεάσει την ακρίβεια και την απόδοση των εφαρμογών. Το JUST-Traj υποστηρίζει τέσσερις λειτουργίες που χρησιμοποιούνται συχνά για την επεξεργασία τροχιών, δηλαδή το φιλτράρισμα θορύβου, την τμηματοποίηση, την παρεμβολή και την αντιστοίχιση χαρτών.

Το φιλτράρισμα θορύβου φιλτράρει τα μη φυσιολογικά αρχεία καταγραφής του GPS, π.χ., ένα σημείο απομακρύνεται σημαντικά από μια τροχιά, ως το αναπόφευκτο σφάλμα πολλών τερματικών GPS. Εάν δεν καταργηθούν τα σημεία σφάλματος σε μια τροχιά, οι εφαρμογές ενδέχεται να αντιμετωπίσουν προβλήματα στις εργασίες ανάλυσης δεδομένων.

Η τμηματοποίηση σπάει μια τροχιά σε πολλά τμήματα. Ένα τερματικό θα μπορούσε να δημιουργήσει μεγάλο αριθμό σημείων GPS χωρίς διακοπή, αλλά μόνο ένα μέρος των σημείων θα χρησιμοποιηθεί για την αναζήτηση και την ανάλυση. Έτσι, η τμηματοποίηση θα μείωνε την υπολογιστική πολυπλοκότητα κατά την εκτέλεση εργασιών ανάλυσης δεδομένων.

Η παρεμβολή εισάγει νέα σημεία σε μια τροχιά, καθώς τα τερματικά GPS μπορεί να αγνοήσουν ορισμένα σημαντικά αρχεία καταγραφής (π.χ. η μπαταρία είναι χαμηλή).

Η αντιστοίχιση της χαρτογράφησης προβάλλει μια πρωτογενή τροχιά στο οδικό δίκτυο. Επομένως, είναι απαραίτητο για πολλές εφαρμογές που βασίζονται στο οδικό δίκτυο, π.χ., πρόβλεψη ροής κυκλοφορίας και ερώτημα προσβασιμότητας.

Στο επίπεδο της αποθήκευσης, αποθηκεύονται οι καθαρισμένες και ευρετηριασμένες τροχιές σε μια βάση δεδομένων NoSQL χρησιμοποιώντας τη μορφή ζευγών κλειδιών-τιμών. Αρχικά,

δημιουργούνται διαφορετικά είδη κλειδιών που περιέχουν τις χωροχρονικές πληροφορίες μιας τροχιάς, όπως φαίνεται στο σχήμα 23 (Storing)(Storing). Στη συνέχεια, συμπιέζεται η τιμή μιας τροχιάς σε μία στήλη, η οποία μειώνει το μέγεθος αποθήκευσης και την επιβάρυνση των ανοιγμάτων/κλεισιμάτων του δίσκου. Μετά από αυτό, αποθηκεύονται τα ζεύγη κλειδιού-τιμής μιας τροχιάς στους πίνακες ευρετηρίασης για μεταγενέστερα ερωτήματα.

CloudTP [26]

Το CloudTP αποτελεί ένα ευέλικτο σύστημα προεπεξεργασίας δεδομένων τροχιάς, το οποίο βασίζεται στην τεχνολογία του cloud. Όπως φαίνεται στο σχήμα 24, το CloudTP λαμβάνει ακατέργαστα αρχεία καταγραφής GPS ως είσοδο και δημιουργεί οργανωμένες καθαρές τροχιές για τους χρήστες. Για να βελτιώσει την αποτελεσματικότητα της διαχείρισης ακατέργαστων αρχείων καταγραφής GPS μεγάλης κλίμακας, αξιοποιεί την πλατφόρμα παράλληλων υπολογισμών (δηλαδή το Spark) για να επιταχύνει την εκτέλεση εργασιών και το χώρο αποθήκευσης στο cloud για να βοηθήσει στη δημιουργία ευρετηρίου. Όταν ολοκληρωθεί η προεπεξεργασία, τα δεδομένα τροχιάς οργανώνονται στον αποθηκευτικό χώρο cloud με χρονικά αναγνωριστικά και χωροχρονικά ευρετήρια. Το CloudTP παρέχει επίσης μια διεπαφή για την ανάκτηση των αποτελεσμάτων (π.χ. τροχιές αντιστοιχισμένες με χάρτες), καθώς και στατιστικών δεδομένων και απεικονίσεις για την υποστήριξη των προαναφερθέντων εφαρμογών. Καλύπτει τις ακόλουθες επιμέρους εργασίες στην προεπεξεργασία των δεδομένων τροχιάς:

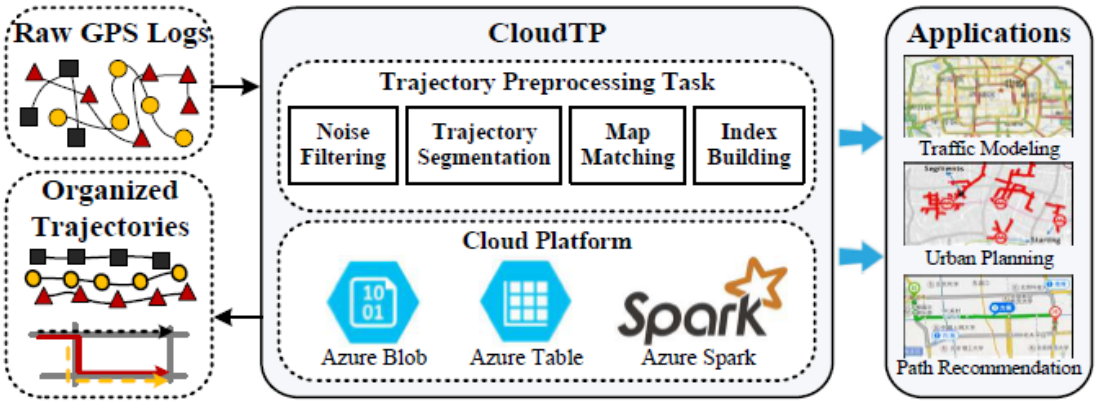
Φιλτράρισμα θορύβου, μια διαδικασία η οποία φιλτράρει μη φυσιολογικά σημεία GPS, π.χ., οι θέσεις των μετρήσεων απομακρύνονται σημαντικά από το διάστημα, καθώς αυτές οι μετρήσεις σφάλματος GPS ενδέχεται να προκαλέσουν προβλήματα στις μεταγενέστερες εργασίες εξόρυξης των δεδομένων και τη μοντελοποίηση.

Τμηματοποίηση τροχιάς, η οποία όχι μόνο ανιχνεύει τη λανθάνουσα πληροφορία, π.χ. σημεία παραμονής [13], αλλά μειώνει επίσης την υπολογιστική πολυπλοκότητα της τροχιάς.

Αντιστοίχιση χάρτη, η οποία μετατρέπει μια ακολουθία σημείων GPS σε ένα σύνολο τμημάτων δρόμου. Είναι ένα ουσιαστικό βήμα σε εφαρμογές που σχετίζονται με το οδικό δίκτυο.

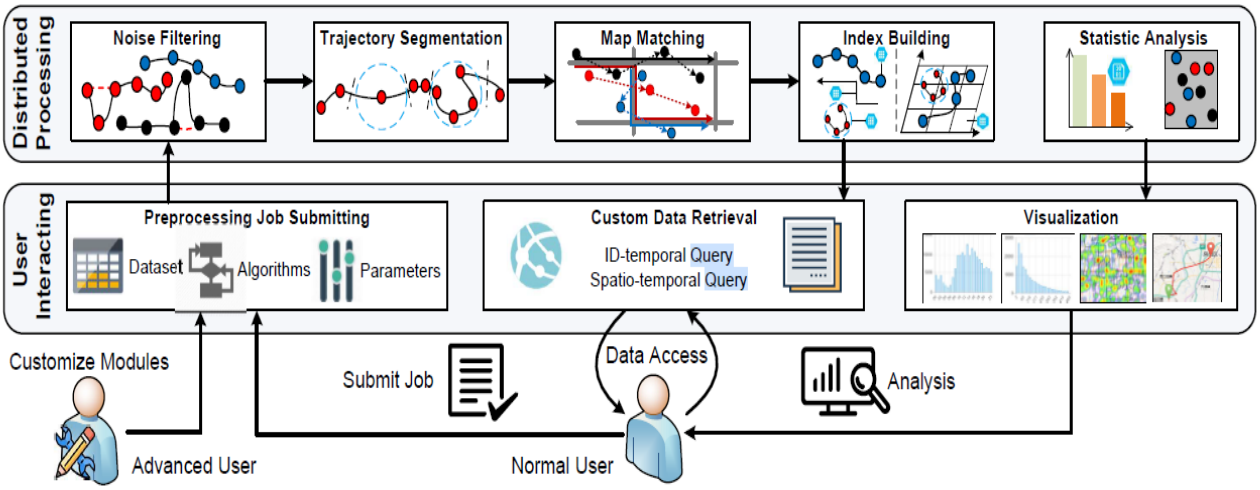
Δημιουργία ευρετηρίου, η οποία δημιουργεί ευρετήριο με το χρονικό αναγνωριστικό και χωροχρονικό ευρετήριο πάνω στα προεπεξεργασμένα δεδομένα και τα οργανώνει στο χώρο αποθήκευσης cloud για την αποτελεσματικότητα των ερωτημάτων.

Η ευελιξία του προτεινόμενου συστήματος είναι διπλή: 1) Η υποστήριξη πολλαπλών τρόπων μετακίνησης. Όλες οι παράμετροι που σχετίζονται με τις λειτουργίες ταξιδιού μπορούν να διαμορφωθούν στο CloudTP. 2) Συνδεόμενες μονάδες επεξεργασίας. Οι προχωρημένοι χρήστες μπορούν να προσαρμόσουν τους αλγόριθμους επεξεργασίας.



Σχήμα 24. Επισκόπηση CloudTP [26]

Συγκεκριμένα, το σύστημα CloudTP περιέχει δύο επίπεδα, όπως απεικονίζεται και στο σχήμα 25. Το κατανεμημένο επίπεδο επεξεργασίας λαμβάνει ακατέργαστα αρχεία καταγραφής GPS, τα επεξεργάζεται αποτελεσματικά, ευρετηριάζει και αποθηκεύει τα δεδομένα με κατανεμημένο τρόπο. Το επίπεδο αλληλεπίδρασης με τον χρήστη παρέχει τα αιτήματα εργασίας των χρηστών και παρέχει μια διεπαφή για την προεπεξεργασμένη ανάκτηση των δεδομένων και τη στατιστική ανάλυση. Οι προχωρημένοι χρήστες μπορούν επίσης να σχεδιάσουν προσαρμοσμένους αλγόριθμους για την αντικατάσταση κατά την υποβολή των αιτημάτων εργασίας.



Σχήμα 25. Επισκόπηση συστήματος CloudTP *[26]*

Το σύστημα CloudTP χρησιμοποιεί σαν σύστημα αποθήκευσης το Azure Storage, το οποίο παρέχει αξιόπιστες, επεκτάσιμες υπηρεσίες αποθήκευσης, συμπεριλαμβανομένων των δομών Blob Storage και Table Storage. Το Blob Storage αποθηκεύει μη δομημένα δεδομένα αντικειμένων, τα οποία είναι χρήσιμα για την αποθήκευση ακατέργαστων αρχείων καταγραφής GPS που ανεβαίνουν από χρήστες και των αποτελεσμάτων της στατιστικής ανάλυσης. Το Table Storage αποθηκεύει δομημένα σύνολα δεδομένων. Είναι ένας χώρος αποθήκευσης δεδομένων κλειδιού NoSQL και κάθε οντότητα σε έναν πίνακα προσδιορίζεται μοναδικά από κλειδιά δύο επιπέδων, το PartitionKey (PK) και το RowKey (RK). Οι οντότητες με διαδοχικά κλειδιά βελτιώνουν την απόδοση των ερωτημάτων διαστήματος, η οποία είναι κατάλληλη για την κατασκευή του χρονικού ευρετηρίου σε προεπεξεργασμένες τροχιές.

VIPTRA [43]

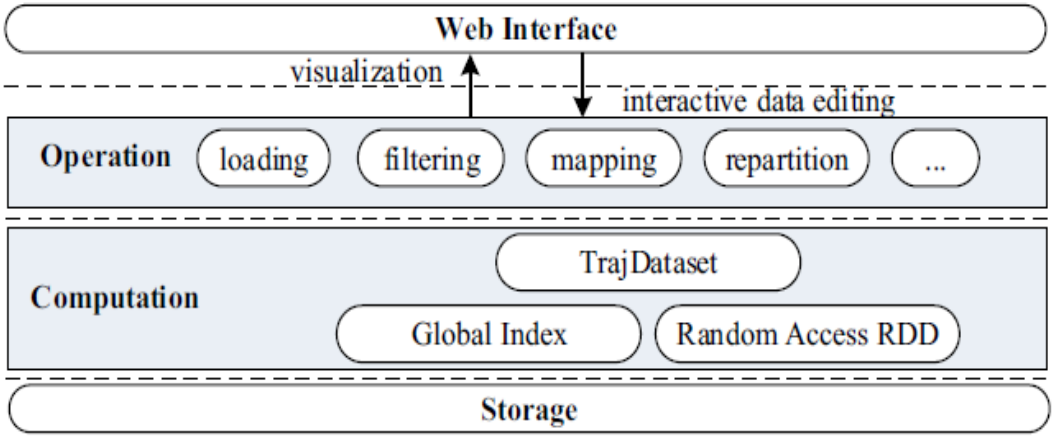
Το VIPTRA αποτελεί ένα αποτελεσματικό σύστημα για την υποστήριξη της οπτικοποίησης και της διαδραστικής επεξεργασίας των μεγάλων δεδομένων τροχιάς. Το VIPTRA αναπτύσσεται στο UlTraMan, το οποίο, όπως έχουμε παραθέσει προηγουμένως, είναι μια κατανεμημένη πλατφόρμα επεξεργασίας μεγάλων δεδομένων στην εσωτερική μνήμη και υιοθετεί μια μέθοδο ευρετηρίασης δύο επιπέδων για περαιτέρω βελτίωση της αποτελεσματικότητας. Με αυτόν τον τρόπο εκμεταλλεύεται την ικανότητά του για υψηλή απόδοση.

Στο σύστημα VIPTRA πραγματοποιείται η σχεδίαση και η υλοποίηση μια σειράς ευρετηρίων κατανεμημένων τροχιών και δύο θεμελιωδών ερωτημάτων τροχιάς, συμπεριλαμβανομένου του ερωτήματος διαστήματος και του ερωτήματος k-πλησιέστερου γείτονα (kNN). Συνοψίζοντας, οι βασικές συνεισφορές του συστήματος είναι οι εξής:

Αποτελεί ένα αποτελεσματικό σύστημα για οπτικοποίηση μεγάλης τροχιάς και την επεξεργασία των ερωτημάτων.

Εφαρμόζονται αποτελεσματικοί αλγόριθμοι αναζήτησης διαστήματος και kNN καθώς και φιλικές προς το χρήστη οπτικές διεπαφές.

Φυσικά, το VIPTRA υιοθετεί μια αρχιτεκτονική προγράμματος περιήγησης-διακομιστή, η οποία αποτελείται από μια διεπαφή βασισμένη στον ιστό, ένα επίπεδο λειτουργίας, ένα επίπεδο υπολογισμού και ένα στρώμα αποθήκευσης, όπως απεικονίζεται στο σχήμα 26. Στο πίσω μέρος του VIPTRA, η πλατφόρμα UlTraMan αναπτύσσεται με έναν κύριο κόμβο και πολλούς επιμέρους κόμβους εργασίας. Ο κύριος κόμβος είναι υπεύθυνος για τον προγραμματισμό των εργασιών, ενώ η αποθήκευση των δεδομένων και ο υπολογισμός τους κατανέμονται στους κόμβους εργασίας.



Σχήμα 26. Επισκόπηση συστήματος VIPTRA [43]

Η διαδικτυακή διεπαφή του VIPTRA μπορεί να υποστηρίξει διάφορες συσκευές και πλατφόρμες όπως Η/Υ, έξυπνα τηλέφωνα κ.λπ. Οι χρήστες μπορούν να υποβάλουν ένα ερώτημα τροχιάς με καθορισμένες παραμέτρους μέσω της διεπαφής και τα αποτελέσματα που επιστρέφονται οπτικοποιούνται σε έναν ενσωματωμένο ψηφιακό χάρτη.

Το επίπεδο λειτουργίας προσφέρει διάφορες διεπαφές προγραμματισμού. Τα δεδομένα τροχιάς υποβάλλονται σε επεξεργασία μέσω λειτουργιών του TrajDataset, όπως φόρτωση, φιλτράρισμα, αντιστοίχιση, αναδιαμέριση κ.λπ. Με βάση αυτά, προηγμένες τεχνικές επεξεργασίας και ανάλυσης δεδομένων μπορούν να επικοινωνήσουν με τη διεπαφή ιστού.

Χρησιμοποιώντας την αφηρημένη δομή του TrajDataset, το επίπεδο υπολογισμού υποστηρίζει το παράδειγμα του κατανεμημένου υπολογισμού. Το TrajDataset, ένα βελτιωμένο κατανεμημένο υπολογιστικό παράδειγμα που βασίζεται στο MapReduce και στο Resilient Distributed Dataset (RDD), εκμεταλλεύεται το επίπεδο αποθήκευσης για να επιτρέψει την τυχαία πρόσβαση στα προσωρινά αποθηκευμένα δεδομένα. Επομένως, ευρετήρια και βελτιστοποιήσεις μπορούν να εφαρμοστούν σε ερωτήματα και στην επεξεργασία των δεδομένων.

Το επίπεδο αποθήκευσης, αποθηκεύει το σύνολο των δεδομένων τόσο στη μνήμη όσο και στο κατανεμημένο σύστημα αρχείων (π.χ. HDFS) για να υποστηρίζει την αποτελεσματική τυχαία πρόσβαση στα δεδομένα και την αξιόπιστη ανοχή των σφαλμάτων.

PRADASE [44]

Το σύστημα PRADASE (Επεξεργασία ερωτημάτων για μαζικά δεδομένα τροχιάς με βάση το MapReduce), βασίζεται σε έναν χώρο αποθήκευσης τύπου GFS που υποστηρίζει μόνο την προσάρτηση δεδομένων. Ο κύριος κόμβος είναι υπεύθυνος για το κλειδί στους επιμέρους κόμβους, για την αντιστοίχιση και αναζήτηση (μέσω κατακερματισμού). Έτσι τα αντίστοιχα δεδομένα μπορούν να επιστραφούν αυτόματα εάν δοθεί το κλειδί. Ο μηχανισμός εκτέλεσης είναι χτισμένος σε μοντέλο MapReduce, το οποίο μπορεί να απλοποιήσει σημαντικά πολλές έντονες υπολογιστικές εργασίες δεδομένων. Είναι κατάλληλο για κατανεμημένα συστήματα μεγάλης κλίμακας όπως το GFS και το Hadoop, καθώς οι αλληλεπιδράσεις

μεταξύ δεδομένων σε διαφορετικούς κόμβους συμβαίνουν μόνο στην αυτόματη ομαδοποίηση, η οποία είναι προγραμματισμένη και βελτιστοποιημένη.

Η διαδικασία εκτέλεσης είναι: 1) Διαχωρισμός δεδομένων εισόδου σε πολλά κομμάτια, 2) εκτέλεση δευτερεύουσας εργασίας σε κάθε τμήμα δεδομένων, 3) ομαδοποίηση όλων των ενδιάμεσων τιμών με το ίδιο κλειδί, 4)μείωση κάθε ομάδας.

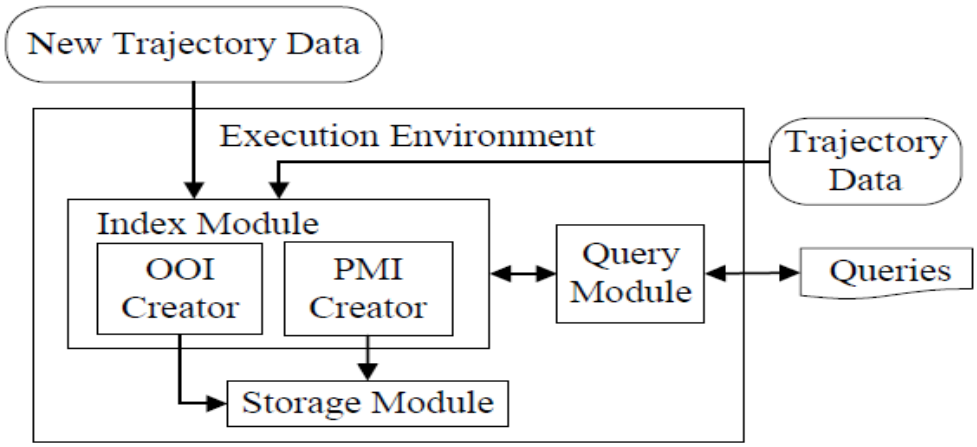
Ένα σημαντικό χαρακτηριστικό του συστήματος PRADASE είναι η ύπαρξη δύο επεκτάσιμων μεθόδων ευρετηρίασης που μπορούν να διευκολύνουν την επεξεργασία των ερωτημάτων αποτελεσματικά και να αποσυνδέσουν τη χωρική και τη χρονική διάσταση. Το πρώτο είναι το ευρετήριο πολλαπλών επιπέδων βάσει διαμέρισης, PMI (Partition based Multilevel Index), που προτείνεται για την αντιμετώπιση ερωτημάτων χωροχρονικού εύρους. Το δεύτερο είναι το ανεστραμμένο ευρετήριο αντικειμένου, OII (Object Inverted Index) που μπορεί να διευκολύνει τα ερωτήματα που βασίζονται σε τροχιά.

Το σύστημα PRADASE για τη διαδικασία αναζήτησης δεδομένων μαζικής ιστορικής τροχιάς αποτελείται από το τμήμα του ευρετηρίου και το τμήμα του ερωτήματος, η οποία φαίνεται στο σχήμα 27. Όταν τα δεδομένα τροχιάς εισάγονται στο σύστημα, το τμήμα του ευρετηρίου χρησιμοποιεί τον δημιουργό PMI και τον δημιουργό OII για να δημιουργήσει δύο τύπους ευρετηρίων αντίστοιχα. Μετά τη δημιουργία των ευρετηρίων, όλα τα δεδομένα μεταφέρονται στη μονάδα αποθήκευσης. Όταν υποβάλλονται τα ερωτήματα, τα αποτελέσματα θα επιστραφούν γρήγορα με το κάλεσμα του τμήματος ευρετηρίου.

1. Μονάδα αποθήκευσης: Τα δεδομένα ιστορικής τροχιάς είναι πάντα σε μεγάλη ποσότητα και παραμένουν στατικά. Ένα μόνο μηχάνημα μπορεί να μην παρέχει αρκετή ικανότητα αποθήκευσης και επεξεργασίας για τον μεγάλο όγκο δεδομένων ιστορικής τροχιάς. Έτσι, ένα κατανεμημένο σύμπλεγμα ή μια υποδομή υπολογιστικού νέφους προτιμάται να χρησιμοποιείται για την αποθήκευση και επεξεργασία του μεγάλου αριθμού ιστορικών τροχιών.

2. Τμήμα ευρετηρίου: Στο τμήμα ευρετηρίου, τα δεδομένα ιστορικής τροχιάς χωρίζονται σε διαφορετικές διαμερίσεις. Όλα τα δεδομένα σε μια διαμέριση διατηρούνται σε ένα κομμάτι και αποθηκεύονται ταυτόχρονα σε περισσότερους από έναν κόμβους δεδομένων ως επαναλήψεις. Δύο διαφορετικά ευρετήρια χρησιμοποιούνται στο τμήμα ευρετηρίασης προκειμένου να βελτιστοποιηθεί το ερώτημα διαστήματος και το ερώτημα βάσει τροχιάς.

3. Τμήμα ερωτημάτων: Στο τμήμα ερωτημάτων, γίνεται αποτελεσματική επεξεργασία δύο ειδών ερωτημάτων. Λαμβάνοντας ένα ερώτημα χωροχρονικού διαστήματος, η χωρική έκταση του διαστήματος ελέγχεται έναντι της διαμέρισης του PMI. Για το ερώτημα που βασίζεται σε τροχιά, με δεδομένο ένα αναγνωριστικό του αντικειμένου, η αντίστοιχη διεύθυνση αποθήκευσης μπορεί να εντοπιστεί απευθείας.



Σχήμα 27. Επισκόπηση συστήματος PRADASE και διαδικασίας εκτέλεσης *[44]*

Συγκεκριμένα, στο επίπεδο της αποθήκευσης το PRADASE έχει δημιουργηθεί με τη λογική του χώρου αποθήκευσης τύπου GFS με έναν κύριο και πολλαπλών δεδομένων διακομιστή. Τα δεδομένα ομαδοποιούνται με κλειδί και οργανώνονται σε κομμάτια. Κάθε κομμάτι έχει διπλασιασμούς σε πολλούς κόμβους, έτσι ώστε η αποτυχία ενός σημείου να μην επηρεάζει τη διαθεσιμότητα του συστήματος. Η αποτυχία του κύριου κόμβου μπορεί να ανακτηθεί από τις πληροφορίες των επιμέρους κόμβων δεδομένων. Το PRADASE αποτελείται από πολλούς κόμβους. Ολόκληρο το σύνολο δεδομένων χωρίζεται σε πολλά μέρη και κάθε μέρος εκχωρείται σε ένα κομμάτι για αποθήκευση.

4. Τεχνικές Διαχείρισης και Επεξεργασίας Μεγάλων Δεδομένων Τροχιάς

Διαμέριση (Partitioning)

Η διαμέριση δεδομένων είναι μια θεμελιώδης λειτουργία σε κατανεμημένα συστήματα για τη διαχείριση και επεξεργασία μεγάλων δεδομένων σε συμπλέγματα υπολογιστών. Όταν ο όγκος των δεδομένων γίνεται πολύ μεγάλος για τον χειρισμό από έναν μεμονωμένο κόμβο, τα δεδομένα συνήθως διαμερίζονται και κάθε ομάδα διαμερίσεων αποθηκεύεται σε διαφορετικό μπλοκ δεδομένων. Έτσι, όταν πραγματοποιείται ένα ερώτημα, αυτά τα μπλοκ δεδομένων δίνονται σε διαφορετικούς υπολογιστικούς κόμβους, όπου καθένας από αυτούς εκτελεί μέρος του ερωτήματος στο μπλοκ δεδομένων του. Ο σκοπός της διαμέρισης δεδομένων είναι, δηλαδή, είτε η επεξεργασία ερωτημάτων σε συστήματα βάσεων δεδομένων είτε ο εντατικός υπολογισμός δεδομένων στα συστήματα ανάλυσης μεγάλων δεδομένων.

Οι τεχνικές διαμέρισης για μεγάλα δεδομένα κινητικότητας και κατ' επέκταση για μεγάλα δεδομένα τροχιάς, έχουν τα ακόλουθα διακριτικά χαρακτηριστικά: (α) λειτουργούν σε ένα δείγμα δεδομένων για να παράγουν διαμερίσεις για το σύνολο των δεδομένων, (β) πρέπει να αντιμετωπίσουν ασύμμετρες[2] (skewed) κατανομές δεδομένων, (γ) θα πρέπει να είναι προσαρμοστικές τόσο σε σχέση με την αλλαγή των κατανομών δεδομένων όσο και με τις αλλαγές στον φόρτο εργασίας του ερωτήματος, (δ) πρέπει να εξισορροπούν το φόρτο εργασίας στους διαθέσιμους κόμβους, κάτι που περιπλέκεται περαιτέρω από την αντιγραφή αντικειμένων σε κοντινά διαμερίσματα.

Επιπρόσθετα, η τεχνική διαμέρισης θα πρέπει να επιλέγεται ανάλογα με τον τύπο των δεδομένων, την πολυπλοκότητά τους και τον τύπο επεξεργασίας. Διαφορετικά ερωτήματα λειτουργούν καλύτερα κάτω από διαφορετικές διαμερίσεις, επομένως για να επιλέξουμε μια τεχνική διαμερίσεων, θα πρέπει να ληφθούν υπόψη τα πιο κοινά και χρονοβόρα ερωτήματα που πραγματοποιούνται.

Ορισμένα συστήματα για τη διαχείριση δεδομένων τροχιάς υιοθετούν τεχνικές διαμέρισης που χρησιμοποιούνται και για τη διαχείριση χωρικών ή χωροχρονικών δεδομένων. Συνοπτικά θα αναφέρουμε ορισμένες τεχνικές:

Διαμέριση πλέγματος (Grid Partitiong)

Αυτή είναι μια τυπική τεχνική διαμερίσματος χώρου που χωρίζει τον υποκείμενο χώρο σε μη επικαλυπτόμενα κελιά. Θα πρέπει να σημειωθεί ότι μερικές φορές το βήμα κατάτμησης ακολουθείται από αντιγραφή αντικειμένων σε γειτονικά κελιά. Αυτό συμβαίνει συνήθως για συνενώσεις απόστασης πάνω σε σημειακά δεδομένα ή χωρικές συνενώσεις πάνω σε δεδομένα με έκταση. Επίσης, στην τελευταία περίπτωση, μια εναλλακτική δυνατότητα είναι η εκτέλεση αποκοπής αντικειμένων, διαχωρίζοντας έτσι ένα αντικείμενο σε πολλά μέρη και εκχωρώντας κάθε τμήμα σε διαφορετικό κελί.

Διαμέριση STR

[2] Η τεχνική που ακολουθείται για τη δημιουργία των διαμερίσεων μπορεί να οδηγήσει σε ανομοιόμορφη κατανομή φορτίου μεταξύ των διαμερίσεων. Ορισμένες διαμερίσεις εξυπηρετούν μεγαλύτερο αριθμό ερωτημάτων από άλλες. Μια τέτοιου είδους «άδικη» διαμέριση χαρακτηρίζεται ως ασύμμετρη (skewed). Σε μία ακραία περίπτωση, ολόκληρο το φορτίο μπορεί να καταλήξει σε μία μόνο διαμέριση, π.χ. 4 στις 5 διαμερίσεις είναι σε αδράνεια. Μία διαμέριση με, δυσανάλογα, υψηλό φορτίο είναι γνωστή ως hotspot. Αυτός ο τρόπος διαμέρισης είναι λιγότερο αποτελεσματικός και οδηγεί σε ανομοιόμορφη κατανομή φορτίου μεταξύ των κόμβων.

Ένα ευρέως χρησιμοποιούμενο σχήμα διαμερίσεων που υιοθετείται από πολλά πρωτότυπα είναι το Sort-Tile-Recursive (STR) [45], το οποίο θεωρείται ένα από τα καλύτερα σχήματα διαμερίσεων για χωρικά δεδομένα.

Σε αυτή την τεχνική, εκτελείται τυχαία δειγματοληψία πάνω στα δεδομένα εισόδου και στη συνέχεια εκτελείται η πρώτη επανάληψη του STR για την δημιουργία ορίων διαμερίσεων. Προφανώς, αυτές οι διαμερίσεις ενδέχεται να μην καλύπτουν ολόκληρο τον χώρο δεδομένων, καθώς έχουν κατασκευαστεί με βάση μόνο ένα δείγμα, επομένως πρέπει να επεκταθούν ώστε να καλύπτουν ολόκληρο τον χώρο δεδομένων.

Διαμέριση Quadtree

Άλλα πρωτότυπα υποστηρίζουν τη διαμέριση που βασίζεται σε Quadtree σε δείγμα δεδομένων ως εναλλακτική τεχνική. Και πάλι, ο στόχος είναι η παραγωγή διαμερίσεων που λαμβάνουν υπόψη την (προκύπτουσα) κατανομή δεδομένων και χειρίζονται τις ασύμμετρες χωρικές κατανομές.

Διαμέριση δέντρου K-d

Μια άλλη προσέγγιση για τον χειρισμό της ασυμμετρίας των δεδομένων εισόδου είναι η χρήση ενός δέντρου k-d, στο οποίο τα φύλλα αντιστοιχούν σε διαμερίσεις δεδομένων στο κατανεμημένο σύστημα αρχείων.

Διαμέριση με βάση τις καμπύλες διάσχισης χώρου

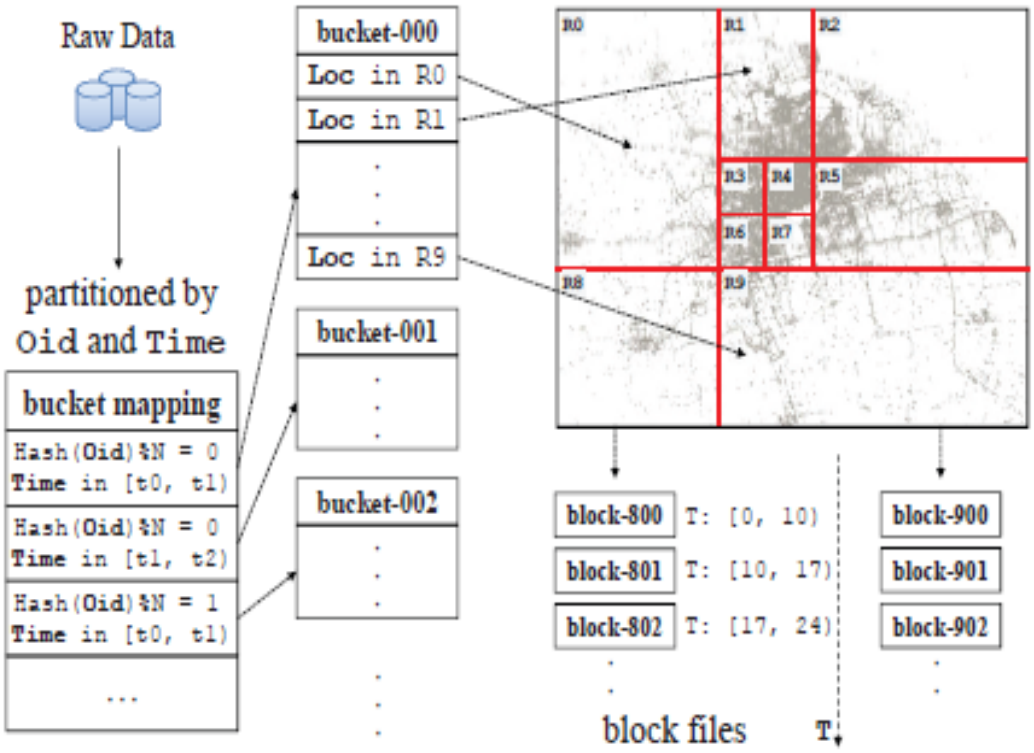
Σε αυτήν την προσέγγιση, τα δισδιάστατα δεδομένα αντιστοιχίζονται σε μονοδιάστατες τιμές χρησιμοποιώντας μια καμπύλη διάσχισης χώρου (όπως καμπύλη Z-order ή Hilbert), και στη συνέχεια δημιουργούνται διαμερίσεις ομαδοποιώντας τις μονοδιάστατες τιμές σε διαστήματα.

Ωστόσο, χρησιμοποιούνται και άλλες εξειδικευμένες τεχνικές διαχωρισμού, όπως θα δούμε στα παρακάτω συστήματα.

CloST

Για την περιγραφή του τρόπου με τον οποίο τα δεδομένα διαμερίζονται, αποθηκεύονται και ευρετηριάζονται στο CloST, υπάρχει η απαίτηση οι δομές αποθήκευσης να μπορούν να αντιμετωπίσουν τόσο ερωτήματα ενός αντικειμένου όσο και ερωτήματα όλων των αντικειμένων. Χρησιμοποιούνται οι συναρτήσεις Q(*I, S, T*) και Q(*S, T*) για την εκτέλεση ερωτήματος ενός απλού αντικειμένου και ενός ερωτήματος ολόκληρου αντικειμένου αντίστοιχα, όπου το S υποδηλώνει ένα χωρικό διάστημα, το T υποδηλώνει μια χρονική περιοχή και το I υποδηλώνει ένα αναγνωριστικό αντικειμένου (*Oid*).

Για να καταστεί δυνατή η αποτελεσματική επεξεργασία τόσο του Q1 όσο και του Q2, το CloST διαχωρίζει τα δεδομένα ιεραρχικά χρησιμοποιώντας όλα τα βασικά χαρακτηριστικά. Όπως απεικονίζεται στο σχήμα 28, τα δεδομένα αρχικά χωρίζονται σε έναν αριθμό διαμερίσεων επιπέδου-1 σύμφωνα με τις τιμές κατακερματισμού του Oid και τις μεγάλες περιοχές του χρόνου. Στη συνέχεια, κάθε διαμέριση επιπέδου-1 χωρίζεται σε έναν αριθμό διαμερίσεων επιπέδου 2 σύμφωνα με ένα χωρικό ευρετήριο. Τέλος, κάθε διαμέριση επιπέδου-2 χωρίζεται περαιτέρω σε μια ακολουθία διαμερίσεων επιπέδου-3 σύμφωνα με μικρότερα διαστήματα χρόνου. Οι διαμερίσεις επιπέδου-3 περιέχουν πραγματικές εγγραφές και οι διαμερίσεις υψηλότερου επιπέδου χρησιμεύουν ως ευρετήρια για τις διαμερίσεις επιπέδου-3. Όλες οι εγγραφές και τα ευρετήρια αποθηκεύονται ως κανονικά αρχεία στο HDFS. Στη συνέχεια, γίνεται αναφορά σε μια διαμέριση επιπέδου-1 ως bucket, σε μια διαμέριση επιπέδου-2 ως περιοχή, σε μια διαμέριση επιπέδου-3 ως μπλοκ και στο αντίστοιχο αρχείο ως αρχείο μπλοκ.



Σχήμα 28. Ιεραρχική διαμέριση CIoST [23]

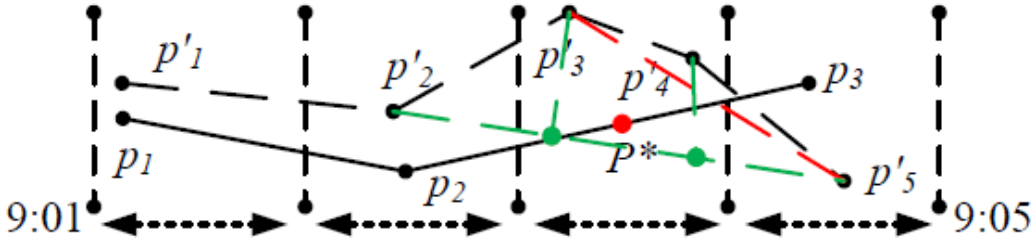
Η διαμέριση επιπέδου-1 σπάει ένα πολύ μεγάλο σύνολο δεδομένων σε σχετικά μικρά bucket. Ως αποτέλεσμα, κάθε bucket μπορεί να εκτελέσει ανεξάρτητα επεξεργασία ερωτημάτων, φόρτωση δεδομένων και βελτιστοποίηση αποθήκευσης. Η ουσία της διαμέρισης επιπέδου-2 και επιπέδου-3 είναι πρώτα η χωρική διαμέριση και μετά η χρονική διαμέριση. Χρησιμοποιείται ένα quadtree για τη διαίρεση ενός bucket σε περιοχές και στη συνέχεια χρησιμοποιείται 1-D διαμέριση διαστήματος για τη διαίρεση μιας περιοχής σε μπλοκ.

SharkDB

Σε αντίθεση με τις περισσότερες υπάρχουσες μεθόδους ευρετηρίασης τροχιών που διατηρούν διαδοχικά δείγματα της ίδιας τροχιάς στην ίδια σελίδα του δίσκου, στο SharkDB, η βάση δεδομένων διαμερίζεται σε πλαίσια στα οποία οι θέσεις όλων των κινούμενων αντικειμένων αποθηκεύονται ταυτόχρονα μαζί και ευθυγραμμίζονται στην κύρια μνήμη. Έχει βρεθεί ότι αυτός ο αποθηκευτικός χώρος κατά στήλη είναι εκπληκτικά κατάλληλος για υπολογισμούς στη μνήμη, καθώς τα περισσότερα πλαίσια μπορούν να αποθηκευτούν σε εξαιρετικά συμπιεσμένη μορφή, η οποία είναι ζωτικής σημασίας για την αύξηση της απόδοσης της μνήμης και τη μείωση της απώλειας της κρυφής μνήμης της CPU.

Αναλυτικότερα, δημιουργείται μια ακολουθία πλαισίων, απευθείας, από δεδομένα τροχιάς. Σε κάθε πλαίσιο εκχωρείται ένα συγκεκριμένο χρονικό διάστημα και, στη συνέχεια, κατανέμεται κάθε σημείο του δείγματος της τροχιάς στο σχετικό πλαίσιο με βάση τον καταγεγραμμένο χρόνο του. Επομένως, ένα πλαίσιο περιέχει όλα τα σημεία του δείγματος της τροχιάς που καταγράφονται στο χρονικό διάστημα του πλαισίου. Ένα τέτοιο χρονικό διάστημα ονομάζεται ρυθμός πλαισίου. Για

παράδειγμα, όπως φαίνεται στο σχήμα 29, εάν οριστεί σαν χρονικό διάστημα το 1 λεπτό (δηλαδή ο ρυθμός πλαισίου είναι 60 δευτερόλεπτα), η χρονική περίοδος από 9:01 έως 9:05 χωρίζεται σε 4 πλαίσια.



Σχήμα 29. Στιγμιότυπο τροχιάς *[46]*

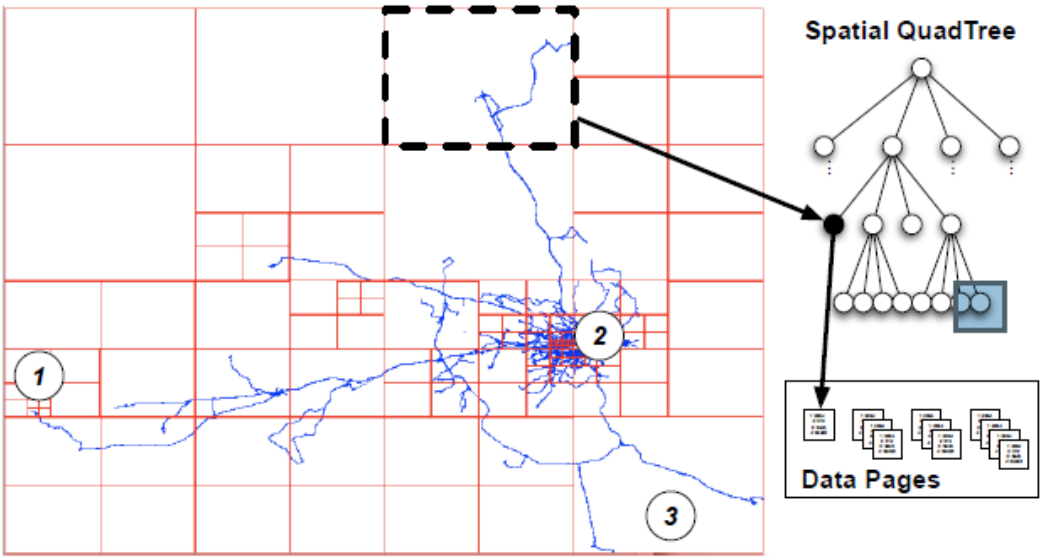
Για να είναι απλό και τακτοποιημένο, κάθε πλαίσιο περιέχει το πολύ ένα σημείο δείγματος για κάθε τροχιά. Για να γίνει αυτή η δομή πλαισίου συνεχής στο χρονικό διάστημα, κάθε πλαίσιο πρέπει να συγχρονίζεται αυστηρά κατά την ίδια χρονική περίοδο (δηλαδή με τον ίδιο ρυθμό). Κανονικά, ο ρυθμός πλαισίου θα είναι ίδιος με τον ρυθμό δειγματοληψίας της τροχιάς. Ωστόσο, μια τροχιά σε ένα σύνολο δεδομένων πρωτογενούς τροχιάς μπορεί να έχει διαφορετικούς ρυθμούς δειγματοληψίας σε σύγκριση με άλλες τροχιές λόγω του ασταθούς σήματος GPS. Επομένως, αυτή η κατάσταση καθιστά ολόκληρο το σύνολο των δεδομένων τροχιάς ετερογενές, γεγονός που μπορεί να επηρεάσει τη δομή του πλαισίου. Για να αποφευχθεί αυτό το ζήτημα, εάν υπάρχουν περισσότερα από ένα σημεία δείγματος που ανήκουν στην ίδια τροχιά, ευθυγραμμίζονται στο ίδιο πλαίσιο. Στη συνέχεια υπολογίζεται η απόσταση SED [47] κάθε σημείου δείγματος, διατηρείται το σημείο δείγματος με τη μεγαλύτερη απόσταση SED και αφαιρούνται τα υπόλοιπα σημεία δείγματος στο πλαίσιο. Αυτό συμβαίνει επειδή το σημείο δείγματος με τη μεγαλύτερη απόσταση SED περιέχει περισσότερες πληροφορίες από τα άλλα σημεία. Στο σχήμα 29, τα P'_3 και P'_4 είναι ευθυγραμμισμένα στο Πλαίσιο 3. Σύμφωνα με το SED, το ευθύγραμμο τμήμα μεταξύ P'_2 και P'_5 χωρίζεται σε 3 τμήματα ίσου μήκους. Εφόσον η απόσταση SED μεταξύ του P'_3 στο αντίστοιχο σημείο του τμήματός του είναι μεγαλύτερη από εκείνη του P'_4, το P'_3 διατηρείται και το P'_4 αφαιρείται. Επιπλέον, εάν δεν υπάρχει δείγμα σημείου μιας τροχιάς που βρίσκεται σε ένα πλαίσιο, χρησιμοποιείται η μέθοδος γραμμικής παρεμβολής για να προστεθεί ένα νέο σημείο δείγματος της τροχιάς σε αυτό το πλαίσιο. Στο σχήμα 29, εισάγεται το P* στο Πλαίσιο 3.

TrajStore

Στο σύστημα TrajStore, όπως φαίνεται και στο σχήμα 30, η διαδικασία της διαμέρισης των δεδομένων τροχιάς λειτουργεί χρησιμοποιώντας ένα προσαρμοσμένο σχήμα πλέγματος που διαμερίζει το χώρο σε έναν αριθμό κελιών πλέγματος και χωρίζει ή συγχωνεύει προσαρμοστικά τα κελιά καθώς προστίθενται νέα δεδομένα ή αλλάζει το μέγεθος του ερωτήματος.

Η δυναμική αυτή προσέγγιση χωρίζει τις περιοχές προκειμένου να ελαχιστοποιήσει τον κενό χώρο στα κελιά που περιέχουν τμήματα της τροχιάς. Στην αριστερή πλευρά του σχήματος 30 (σημείο 1), για παράδειγμα, τα κελιά χωρίζονται αναδρομικά για να απομονωθούν μερικά τμήματα. Τα κελιά χωρίζονται επίσης σε πυκνές περιοχές (σημείο 2), όπου η πρόσβαση σε οποιοδήποτε σημείο στο διάστημα αυτό είναι μια δαπανηρή διαδικασία. Ωστόσο, ο διαχωρισμός των κελιών μπορεί επίσης να οδηγήσει σε μη βέλτιστες διαμερίσεις, για παράδειγμα για αραιές περιοχές που περιέχουν λίγα μόνο τμήματα που θα πρέπει να αποθηκευτούν μαζί προκειμένου να ελαχιστοποιηθεί η πρόσβαση στο δίσκο (σημείο 3) ή για μεγαλύτερα μεγέθη ερωτημάτων.

Τα αποτελέσματά δείχνουν ότι η προσέγγισή αυτή είναι σημαντικά καλύτερη από τις υπάρχουσες προσεγγίσεις, αποφεύγοντας δαπανηρά ανοίγματα/κλεισίματα του δίσκου και συμπυκνώνοντας τα σχετικά δεδομένα στο δίσκο.



Σχήμα 30. Διαμέριση στο σύστημα TrajStore [18]

TrajMesa

Στο TrajMesa τα δεδομένα τροχιάς, αφού υποστούν ένα φιλτράρισμα θορύβου ώστε να περικοπούν τα μη φυσιολογικά αρχεία καταγραφής δεδομένων, σπάνε σε πολλά τμήματα, μέσα από τη λειτουργία της τμηματοποίησης. Ένα τερματικό μπορεί να δημιουργήσει μεγάλο αριθμό σημείων από ένα GPS χωρίς διακοπή, αλλά μόνο ένα μέρος των σημείων θα χρησιμοποιηθεί για την αναζήτηση και την ανάλυση. Έτσι, η τμηματοποίηση μπορεί να μειώσει την υπολογιστική πολυπλοκότητα κατά την εκτέλεση των εργασιών ανάλυσης δεδομένων.

HadoopTrajectory

Το HadoopTrajectory υποστηρίζει τόσο διαχωρισμό ανά κινούμενο αντικείμενο (έτσι ώστε να υπάρχει ολόκληρη η τροχιά στο ίδιο διαμέρισμα), όσο και χωροχρονικές διαμερίσεις. Η διαδικασία της διαμέρισης πραγματοποιείται με τη βοήθεια ενός τελεστή, που ονομάζεται διαχωριστής. Πιο αναλυτικά, ο στόχος της διαμέρισης είναι να χωρίσει τα μεγάλα αρχεία εισόδου σε πολλά μικρότερα, έτσι ώστε να είναι πιο αποτελεσματική η διαχείρισή τους. Επιπλέον, ο διαχωριστής θα δημιουργήσει ένα κύριο αρχείο (master file) που αντιστοιχίζει τα αναγνωριστικά τροχιάς στα διαμερίσματα/αρχεία που περιέχουν τα δεδομένα τους (δηλαδή, έναν χάρτη κατακερματισμού).

Η διαμέριση γίνεται ανεξάρτητα από την ευρετηρίαση. Υπάρχουν δύο βασικά ζητήματα για τη διαμέριση δεδομένων τροχιών. Το πρώτο είναι να αποφεύγεται η εξέταση πολλών τροχιών ώστε να λάβουμε τις επιθυμητές (δηλαδή να ευνοούνται μικρά χωρίσματα). Το δεύτερο είναι να αποφευχθεί η κατανομή των τμημάτων τροχιάς σε πολλά αρχεία. Δηλαδή, μια μεμονωμένη τροχιά κινούμενου αντικειμένου αποτελείται από μια λίστα τμημάτων, καθένα από τα οποία περιγράφει ένα μέρος της κίνησης. Αυτή η απαίτηση σε αντίθεση με την προηγούμενη μπορεί να οδηγήσει σε μεγαλύτερα διαμερίσματα. Ωστόσο, είναι σημαντικό να αποφευχθεί η σάρωση πολλών μπλοκ για την ανάκτηση των πληροφοριών σχετικά με ένα συγκεκριμένο κινούμενο αντικείμενο.

UlTraMan

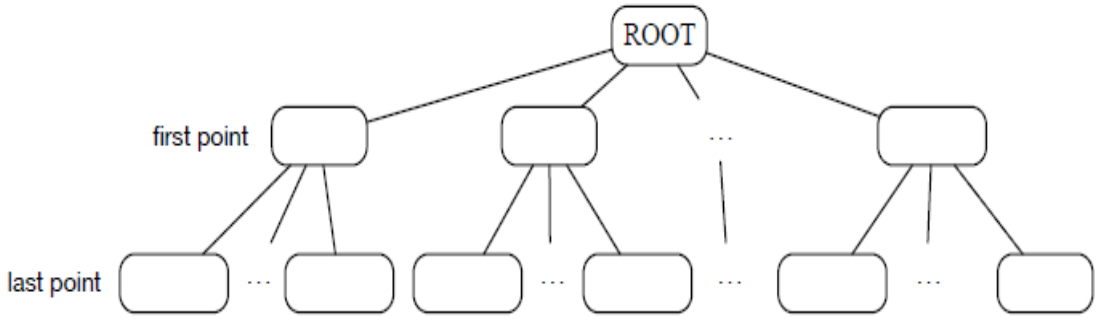
Στο σύστημα UlTraMan η διαμέριση των δεδομένων πραγματοποιείται σε τοπικό και καθολικό επίπεδο, μέσω της λειτουργίας της δομής TrajDataset. Σε τοπικό επίπεδο, οι διαμερίσεις των δεδομένων είναι αυτοδιαχειριζόμενες με ευέλικτα τοπικά ευρετήρια. Σε καθολικό επίπεδο η στρατηγική διαμέρισης

ελέγχεται από τον χρήστη και ενεργοποιούνται δύο κλίμακες γενίκευσης δεδομένων, δηλαδή το καθολικό ευρετήριο και ο μεταπίνακας.

Η διαμέριση των δεδομένων είναι κρίσιμη για τον αποτελεσματικό συνολικό προγραμματισμό. Για παράδειγμα, εάν οι τροχιές χωρίζονται σύμφωνα με τις χωρικές τους πληροφορίες, θα επιταχυνθεί ένα ερώτημα χωρικής περιοχής, καθώς τα διαμερίσματα προς αναζήτηση μπορούν να περικοπούν σε μεγάλο βαθμό λόγω διαστήματος. Αντίθετα, εάν οι τροχιές διαχωρίζονται σύμφωνα με τις πληροφορίες χρόνου τους, ένα συνολικό χρονοδιάγραμμα μπορεί να είναι άχρηστο για ένα ερώτημα διαστήματος, επειδή τα περισσότερα διαμερίσματα είναι υποψήφια προς αναζήτηση. Η δομή TrajDataset παρέχει μια λειτουργία επαναδιαμέρισης για την υποστήριξη διαφορετικών στρατηγικών διαμέρισης και αρκετοί διαμεριστές υλοποιούνται στο UlTraMan, συμπεριλαμβανομένου του STRPpartitioner. [48]

DITA

Όπως φαίνεται στο σχήμα 31, πρώτα ομαδοποιούνται όλες οι τροχιές σύμφωνα με το πρώτο τους σημείο σε *n* ασύνδετα σύνολα. Στη συνέχεια, οι τροχιές σε κάθε σύνολο ομαδοποιούνται περαιτέρω σε υποσύνολα, σύμφωνα με το τελευταίο σημείο αυτών των τροχιών. Κάθε υποσύνολο λαμβάνεται ως διαμέριση και επομένως δημιουργούνται *n*n* διαμερίσεις. Με αυτόν τον τρόπο, παρόμοιες τροχιές είναι πιο πιθανό να βρίσκονται στην ίδια διαμέριση και κάθε διαμέριση έχει περίπου τον ίδιο αριθμό τροχιών. Υιοθετείται η μέθοδος διαμέρισης Sort-Tile-Recursive (STR) [49] για τη διαίρεση των σημείων των τροχιών, όπου μια τέτοια μέθοδος διαμέρισης μπορεί να εγγυηθεί ότι κάθε διαμέριση έχει περίπου τον ίδιο αριθμό σημείων, ακόμη και για εξαιρετικά ασύμμετρα δεδομένα. Επιπλέον, αυτή η τεχνική διαμερίσματος στοχεύει να ομαδοποιήσει τροχιές με παρόμοιες αρχικές θέσεις και παρόμοιες θέσεις τερματισμού.



Σχήμα 31. Παράδειγμα διαμέρισης DITA *[19]*

Distributed MobilityDB

Η τεχνική που χρησιμοποιείται από το σύστημα MobilityDB, στηρίζεται σε δύο βασικές προσεγγίσεις διαμέρισης των δεδομένων: α) την ιεραρχική και β) την πολυδιάστατη. Στην ιεραρχική διαμέριση, πρώτα διαιρείται το χωρικό τμήμα και μετά η χρονική περιοχή ή το αντίστροφο. Μάλιστα, η αντίστροφη διαδικασία αποδείχθηκε πιο αποτελεσματική στη βιβλιογραφία, π.χ., ([50],[51]). Στην πολυδιάστατη διαμέριση χωρίζεται ο n-διάστατος χώρος σε κελιά πλέγματος και τα αντικείμενα δεδομένων τοποθετούνται σε αυτά.

Αναλυτικότερα, στην ιεραρχική μέθοδο, χρησιμοποιούνται δύο επίπεδα διαμέρισης: ένα χρονικό ακολουθούμενο από ένα χωρικό. Στη χρονική διαμέριση, η χρονική περιοχή του συνόλου δεδομένων χωρίζεται σε ισομεγέθη διαστήματα. Οι τροχιές αντιγράφονται σε όλες τις χρονικές διαμερίσεις που επικαλύπτουν τη χρονική τους έκταση. Μέσα σε κάθε προσωρινή διαμέριση, χρησιμοποιείται ένα Quadtree για τη χωρική κατανομή των δεδομένων. Είναι επιθυμητό η διαμέριση που προκύπτει από τα δύο βήματα να είναι αρκετά λεπτή, ώστε τα μέρη να είναι μικρότερα από το

επιθυμητό μέγεθος διαμέρισης, δηλ. ≤ d/w. Πέρα από αυτή την απαίτηση, ο αριθμός των χρονικών διαμερίσεων και οι παράμετροι του Quadtree δεν έχουν μεγάλη σημασία. Μετά από αυτό, τα μέρη σε κάθε χρονική διαμέριση ταξινομούνται με βάση την τεχνική z-order και κατανέμονται στις τελικές διαμερίσεις με ισορροπημένο τρόπο. Αυτό επαναλαμβάνεται για κάθε χρονική διαμέριση. Το αποτέλεσμα είναι ότι μία διαμέριση δεδομένων μεταγλωττίζεται ανά κόμβο εργασίας, όπου όλες οι διαμερίσεις έχουν παρόμοια μεγέθη.

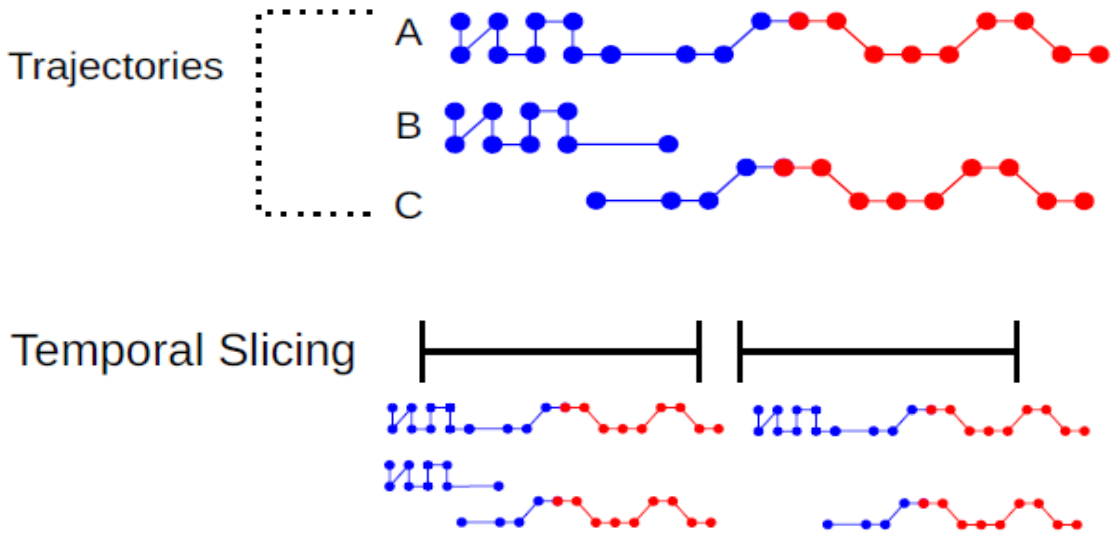
Η ιεραρχική διαμέριση λειτουργεί καλά όταν τα δεδομένα έχουν ορισμένα πρότυπα χρονικής πυκνότητας. Για τη γενικότερη περίπτωση, όπου δεν μπορεί να υποτεθεί ένα τέτοιο μοτίβο, η πολυδιάστατη διαμέριση μπορεί να προσαρμοστεί περισσότερο στη χωροχρονική κατανομή των δεδομένων. Λαμβάνοντας υπόψη ένα σύνολο δεδομένων τροχιών τρισδιάστατων ή τεσσάρων διαστάσεων, κατασκευάζονται αντίστοιχα τρισδιάστατα ή τετραδιάστατα κελιά πλέγματος για να καλύπτουν ολόκληρη την έκταση των δεδομένων. Και πάλι είναι επιθυμητό το μέγεθος κάθε κελιού να είναι μικρότερο από το μέγεθος της διαμέρισης. Μία τροχιά μπορεί να επικαλύπτει πολλά κελιά πλέγματος. Σε αντίθεση με την ιεραρχική διαμέριση που θα αντιγράψει την τροχιά σε κάθε επικαλυπτόμενη διαμέριση, εδώ, οι τροχιές χωρίζονται στα όρια των κελιών. Έτσι, εάν μια τροχιά επικαλύπτει τρία κελιά, θα χωριστεί σε τρία τμήματα και κάθε κελί θα αποθηκεύει μόνο το αντίστοιχο τμήμα του.

Summit

Το Summit οργανώνει τα αρχεία εισόδου στο HDFS με τρόπο που διατηρεί τη γεωμετρική τοπολογία των τροχιών. Συγκεκριμένα, τα δεδομένα φορτώνονται χωροχρονικά και κατανέμονται σε υπολογιστικούς κόμβους. Κάθε διαμέριση έχει την πλήρη ακολουθία των τροχιών που επικαλύπτονται με τα χωροχρονικά του όρια.

Η τεχνική διαμέρισης που ακολουθείται από το σύστημα Summit στηρίζεται στο γεγονός ότι χειρίζεται το δείγμα για να δημιουργήσει ευρετήρια δύο επιπέδων, τόσο χρονικά όσο και χωρικά. Το Summit εφαρμόζει τη χρονική διαμέριση που υπάρχει ήδη στο ST-Hadoop για να χωρίσει τη χρονική διάσταση των τροχιών είτε σε ίσο πλάτος είτε σε ίσο βάθος [39].

Ως προς το χρονικό κομμάτι, πραγματοποιείται ένας τεμαχισμός, που αντιγράφει τις τροχιές εάν αλληλεπικαλύπτονται μεταξύ των χρονικών τμημάτων, διατηρώντας παράλληλα ένα μη επικαλυπτόμενο διαχωρισμό. Όπως φαίνεται στο σχήμα 32, η διάρκεια ζωής της τροχιάς A επικαλύπτει τόσο την πρώτη όσο και τη δεύτερη τομή, επομένως, ολόκληρη η τροχιά θα αντιγραφεί μεταξύ αυτών των δύο χρονικών τμημάτων.



Σχήμα 32. Χρονικός τεμαχισμός Summit *[38]*

Dragoon

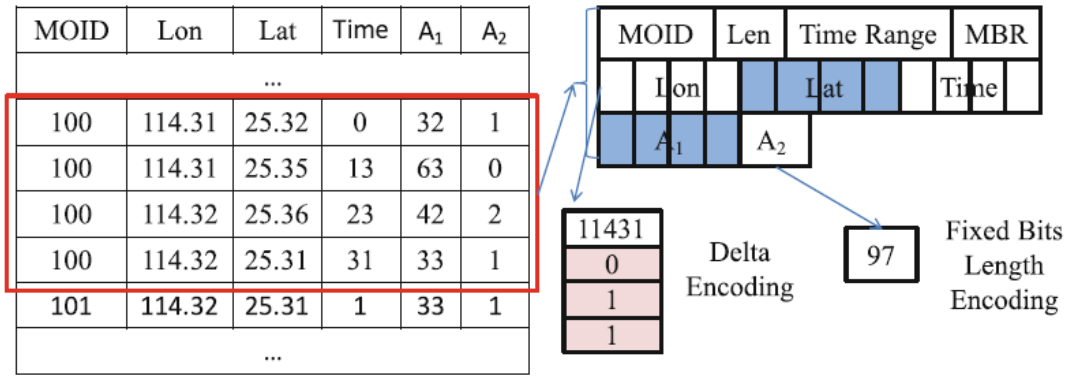
Λόγω της υβριδικής του σχεδίασης, το Dragoon δημιουργεί διαμερίσεις τόσο σε επίπεδο δεδομένων εκτός σύνδεσης, όσο και σε επίπεδο δεδομένων σε σύνδεση. Το σύστημα διαθέτει προσαρμοσμένους διαμεριστές δεδομένων, οι οποίοι χρησιμοποιούν διαφορετικά χαρακτηριστικά (π.χ. το αναγνωριστικό, τις χωρικές ή χρονικές πληροφορίες) των τροχιών για να χωρίσουν τα φορτωμένα δεδομένα τροχιάς λαμβάνοντας υπόψη τα διαφορετικά χαρακτηριστικά των τροχιών και την ισορροπία των δεδομένων του συστήματος. Με άλλα λόγια, τροχιές με τα ίδια ή παρόμοια χαρακτηριστικά ταυτότητας/χωρικής τοποθεσίας/χρόνου αποστέλλονται στην ίδια διαμέριση δεδομένων για παράλληλη επεξεργασία. Τέτοιοι διαμεριστές είναι: ο διαμεριστής αναγνωριστικού (IDPartitioner), ο διαμεριστής πλέγματος (GridPartitioner), ο STRPartitioner και ο διαμεριστής χρόνου (TimePartitioner).

Παρόλο που τόσο η εντός σύνδεσης όσο και η εκτός σύνδεσης προεπεξεργασία της τροχιάς χρειάζονται τη διαδικασία της διαμέρισης των δεδομένων που βασίζεται στο αναγνωριστικό, τη χωρική θέση ή τις χρονικές πληροφορίες των τροχιών, υπάρχουν αρκετές διαφορές μεταξύ της προεπεξεργασίας σε σύνδεση και εκτός σύνδεσης. Η διαμέριση των δεδομένων τροχιάς εκτός σύνδεσης λαμβάνει ολόκληρο το σύνολο δεδομένων της ιστορικής τροχιάς ως είσοδο και εκτελεί τη διαμέριση των δεδομένων μόνο μία φορά, ενώ τα δεδομένα τροχιάς συνεχόμενης ροής διαμερίζονται από διαμεριστές πραγματικού χρόνου, που λαμβάνουν τα σημεία των δεδομένων τροχιάς των κινούμενων αντικειμένων που παρατηρούνται κάθε φορά ως εισροές και εκτελούν την διαμέριση των δεδομένων κάθε φορά σε πραγματικό χρόνο.

TrajSpark

Οι πρωτογενείς τροχιές που παράγονται από τις πηγές δεδομένων αποθηκεύονται συνήθως ως αρχεία καταγραφής GPS και κάθε εγγραφή καταγραφής αντιστοιχεί σε ένα σημείο της τροχιάς. Ένα τέτοιο σχήμα σημείων μπορεί να προβληθεί ως πίνακας με την ακόλουθη μορφή: (*MOID, Location, Time, A1, · · · , An*), όπου MOID είναι το αναγνωριστικό ενός κινούμενου αντικειμένου (MO), Time και Location είναι οι χρονικές και χωρικές πληροφορίες. Τα υπόλοιπα χαρακτηριστικά διαφέρουν ανάλογα με τις πηγές δεδομένων.

Για να μετατραπούν τα ακατέργαστα δεδομένα σε τμήματα τροχιάς, το TrajSpark, πρώτα, χωρίζει τα σημεία, που είναι χωροχρονικά κοντά, στην ίδια διαμέριση. Κατόπιν, τα σημεία του ίδιου MO ταξινομούνται για να σχηματίσουν ένα τμήμα τροχιάς και το τμήμα μετατρέπεται σε μια μορφή αποδοτικής χρήσης χώρου, όπως φαίνεται στο σχήμα 33.



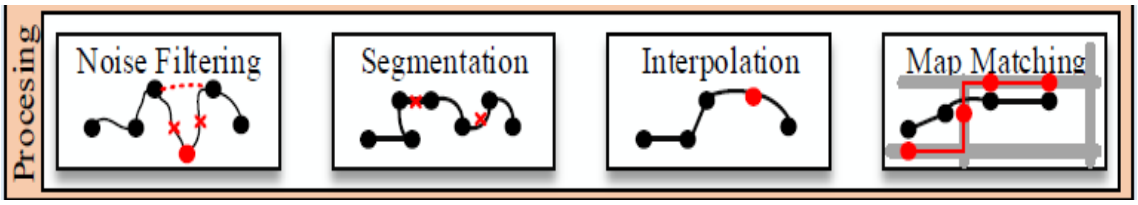
Σχήμα 33. Παρουσίαση μιας τροχιάς στο σύστημα TrajSpark. Αριστερά: ακατέργαστα δεδομένα τροχιάς. Δεξιά: Μορφή αποθήκευσης της τροχιάς [41]

Στη συνέχεια, τα δεδομένα σε κάθε διαμέριση, ύστερα από μια διαδικασία συμπίεσης και περικοπών, αλλάζουν σε ένα σύνολο συμπιεσμένων αντικειμένων τροχιάς και ολόκληρο το σύνολο δεδομένων μετατρέπεται σε ένα RDD τέτοιων αντικειμένων, το οποίο ονομάζεται TRDD.

Πιο αναλυτικά, κατά τη φάση της διαμέρισης, το TrajSpark φορτώνει το πρωτογενές σύνολο δεδομένων από το δίσκο στη μνήμη ως RDD σημείων τροχιάς. Αυτό το RDD πρέπει να επαναδιαμεριστεί εκ νέου. Το TrajSpark ορίζει ένα νέο διαμεριστή με το όνομα STPartitioner που περιέχει ένα χωρικό ευρετήριο quadtree ή k-d tree. Το χωρικό ευρετήριο μπορεί να προσδιοριστεί από την κατανομή των δεδομένων και διασφαλίζει ότι κάθε κόμβος εργασίας περιέχει την ίδια ποσότητα δεδομένων. Το STPartitioner χρησιμοποιεί τα όρια αυτών των κόμβων εργασίας σαν σημεία διαμέρισης. Στη συνέχεια, τα σημεία τροχιάς που βρίσκονται στο ίδιο όριο ομαδοποιούνται. Τέλος, λόγω του περιορισμού του μεγέθους της διαμέρισης, το TrajSpark διαχωρίζει τα σημεία που ανήκουν στο ίδιο όριο σε μερικές διαμερίσεις δεδομένων σύμφωνα με το αναγνωριστικό του κινούμενου αντικειμένου ή το χαρακτηριστικό του χρόνου και διασφαλίζει ότι κάθε διαμέριση πληροί τους ακόλουθους περιορισμούς: i) της τοπικότητας των δεδομένων, δηλαδή τα σημεία των τροχιών που πλησιάζουν χωροχρονικά μεταξύ τους, πρέπει να αναθέτονται στην ίδια διαμέριση, ii) της εξισορρόπησης του φορτίου, δηλαδή όλες οι διαμερίσεις να έχουν το ίδιο μέγεθος και iii) του μεγέθους της διαμέρισης, ώστε να αποφεύγεται η υπερχείλιση της μνήμης.

JUST-Traj

Στο JUST-Traj τα δεδομένα τροχιάς, όπως ακριβώς συμβαίνει και στο TrajMesa, αφού υποστούν ένα φιλτράρισμα θορύβου ώστε να περικοπούν τα μη φυσιολογικά αρχεία καταγραφής δεδομένων, σπάνε σε πολλά τμήματα, μέσα από τη λειτουργία της τμηματοποίησης, όπως παρουσιάζεται και στο σχήμα 34. Ένα τερματικό μπορεί να δημιουργήσει μεγάλο αριθμό σημείων από ένα GPS χωρίς διακοπή, αλλά μόνο ένα μέρος των σημείων θα χρησιμοποιηθεί για την αναζήτηση και την ανάλυση. Με αυτόν τον τρόπο, η διαδικασία της τμηματοποίησης, στο σύστημα JUST-Traj, μπορεί να μειώσει την υπολογιστική πολυπλοκότητα κατά την εκτέλεση των εργασιών ανάλυσης δεδομένων.



Σχήμα 34. Επεξεργασία των δεδομένων στο JUST-Traj [42]

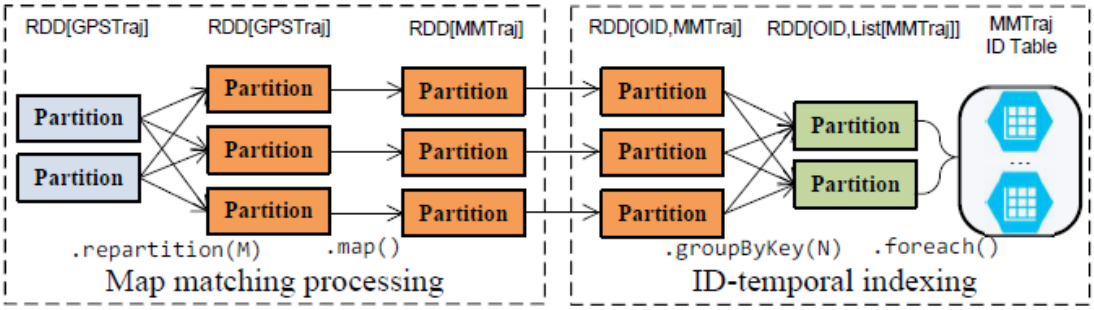
CloudTP

Στο σύστημα CloudTP, στο βήμα της τμηματοποίησης, τα καθαρά (από θορύβους) αρχεία καταγραφής GPS θα τμηματοποιηθούν σε πολλές μικρότερες τροχιές με βάση ορισμένα κριτήρια.

Κατά την εκτέλεση της τμηματοποίησης, επειδή κάθε καθαρό αρχείο καταγραφής GPS μπορεί να παράγει πολλές τροχιές μετά την τμηματοποίηση, για την περαιτέρω διευκόλυνση της επεξεργασίας, συλλέγονται οι τμηματοποιημένες τροχιές σε κάθε αρχείο καταγραφής για να σχηματιστεί ένα σετ RDD[GPSTraj] χρησιμοποιώντας τη μέθοδο flatMap. Παρέχονται δύο αλγόριθμοι που χρησιμοποιούνται συνήθως: μια μέθοδος με βάση το σημείο παραμονής [52], η οποία τμηματοποιεί μια τροχιά σύμφωνα με τα σημεία παραμονής και μια μέθοδος με βάση το χρονικό διάστημα [13], η οποία τμηματοποιεί μια τροχιά εάν το χρονικό διάστημα δύο διαδοχικών σημείων είναι μεγαλύτερο από ένα όριο. Οι παράμετροι, π.χ., μέγιστος χρόνος παραμονής (MST), μέγιστη απόσταση παραμονής (MSD) και μέγιστο χρονικό διάστημα, μπορούν να ρυθμιστούν για διαφορετικούς σκοπούς.

Τέλος, πραγματοποιείται διαμέριση των δεδομένων και κατά τη φάση της αντιστοίχισης στο χάρτη, όπως φαίνεται και στο σχήμα 35. Σε αυτή τη φάση, καλείται η συνάρτηση *repartition*(*M*), όπου M είναι ο αριθμός των διαμερίσεων μετά τον παραπάνω μετασχηματισμό, για ανακατεύθυνση και δημιουργία περισσότερων διαμερίσεων πριν εκτελεστεί ο αλγόριθμος της αντιστοίχισης χάρτη. Οι λόγοι για την αναδιαμέριση βασίζονται σε δύο παρατηρήσεις. Από τη μία πλευρά, μετά την τμηματοποίηση της τροχιάς, υπάρχουν ορισμένα αντικείμενα που κινούνται πιο συχνά από άλλα, γεγονός που οδηγεί σε

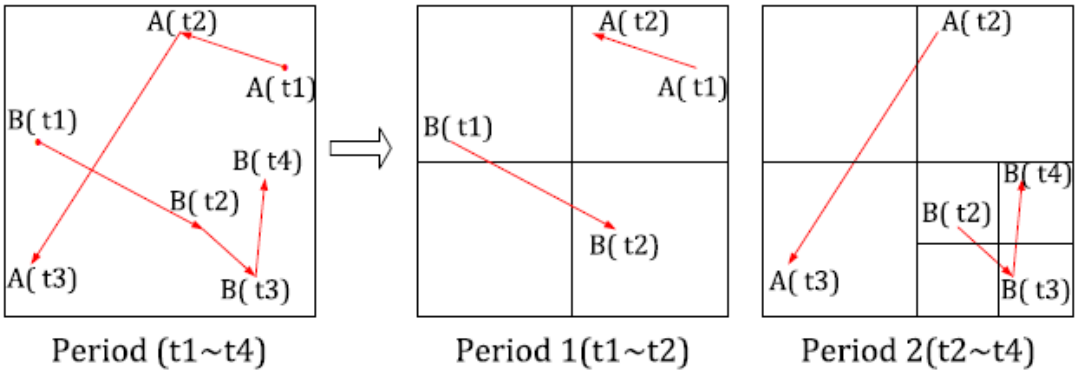
ανισόρροπη κατανομή των εργασιών μεταξύ των διαμερίσεων. Από την άλλη πλευρά, καθώς υπάρχουν πολύ περισσότερες τμηματοποιημένες τροχιές από τα κινούμενα αντικείμενα, χρειάζεται να αυξηθεί ο αριθμός των διαμερίσεων για να βελτιωθεί η παράλληλη επεξεργασία.



Σχήμα 35. Αντιστοίχιση χάρτη και δημιουργία ευρετηρίου στο σύστημα CloudTP [26]

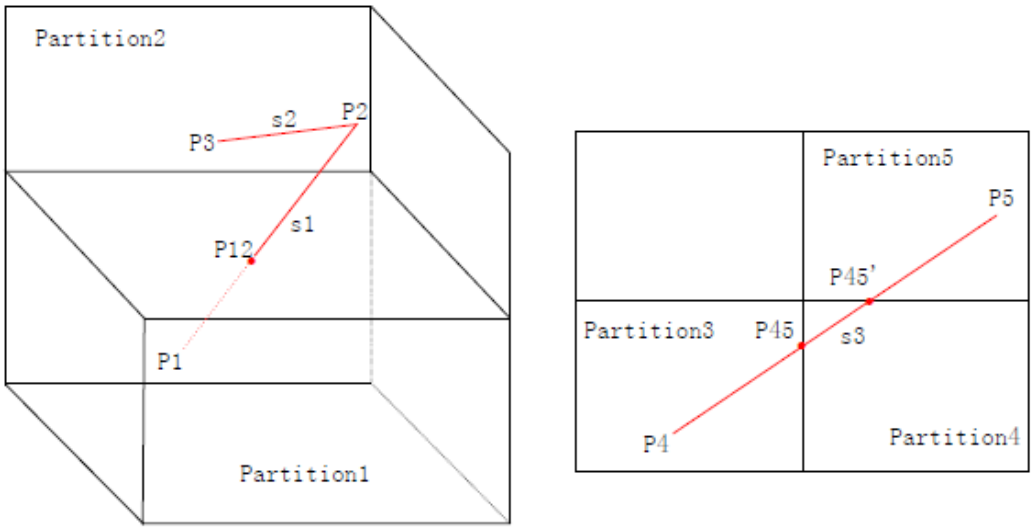
PRADASE

Στο σύστημα PRADASE προτείνεται μια υβριδική μέθοδος διαμέρισης που χρησιμοποιεί τη στρατηγική της στατικής κατανομής για κάθε χρονική περίοδο, προκειμένου να διασφαλιστεί η καλύτερη ισορροπία φορτίου σε δεδομένα τροχιάς μεγάλης ασυμμετρίας, καθώς ο αριθμός των κινούμενων αντικειμένων θα αλλάζει με την αύξηση του χρόνου και αυτό οδηγεί σε αλλαγή της κατανομής των δεδομένων ανάλογα. Στο PRADASE, μια οθόνη τμημάτων έχει σχεδιαστεί για να καταγράφει το μέγεθος όλων των τμημάτων. Εάν το σύστημα διαπιστώσει ότι το μέγεθος των δεδομένων για κάθε κομμάτι δεν είναι ισορροπημένο, πράγμα που σημαίνει ότι η στρατηγική διαμερίσεων δεν είναι κατάλληλη, θα εκπαιδευτεί μια νέα στρατηγική διαμέρισης. Τα ιστορικά δεδομένα που έχουν ήδη εισαχθεί στο σύστημα διατηρούνται αμετάβλητα, ενώ τα νέα ενημερωμένα δεδομένα θα ακολουθούν τη νέα αρχή διαμέρισης. Ως αποτέλεσμα, η διαμέριση είναι δυναμική από την ακόλουθη χρονική διάσταση, αλλά είναι στατική σε κάθε χρονική περίοδο. Στο σχήμα 36 παρουσιάζεται ένα παράδειγμα της παραπάνω διαδικασίας.



Σχήμα 36. Στατική διαμέριση για κάθε χρονική περίοδο [44]

Μετά την διαμέριση του χρονικού ή χωροχρονικού χώρου, όλα τα δεδομένα τροχιάς εκχωρούνται σε τουλάχιστον μια διαμέριση σύμφωνα με τις χωροχρονικές πληροφορίες. Συνοψίζοντας, εάν ένα τμήμα γραμμής καλύπτεται πλήρως από μια διαμέριση, το τμήμα γραμμής εκχωρείται σε αυτή τη διαμέριση. Εάν ένα τμήμα γραμμής εκτείνεται σε περισσότερες από μια διαμερίσεις, ένα τέτοιο τμήμα γραμμής χωρίζεται σε πολλά τμήματα υπογραμμών. Κάθε τμήμα υπογραμμής καλύπτεται πλήρως από μια διαμέριση και εκχωρείται σε αυτή τη διαμέριση. Τα νέα τελικά σημεία τέτοιων τμημάτων υπογραμμών είναι οι τομές με το αρχικό τμήμα γραμμής και τα όρια της διαμέρισης. Ο αντίστοιχος χρόνος για κάθε τελικό σημείο προσδιορίζεται με παρεμβολή.



Σχήμα 37. Παραδείγματα διαίρεσης τμημάτων [44]

Το σχήμα 37 δείχνει δύο παραδείγματα αυτής της διαδικασίας. Η χωροχρονική διαίρεση απεικονίζεται αριστερά. Το γραμμικό τμήμα $s_2(P_2, P_3)$ μετατρέπεται σε (Partition2, $s_2(P_2, P_3)$). Το γραμμικό τμήμα $s_1(P_1, P_2)$ χωρίζεται σε δύο υποτμήματα και αντιστοιχίζεται σε (Partition1, (P_1, P_{12})) και (Partition2, (P_{12}, P_2)). Η χωρική διαίρεση απεικονίζεται δεξιά. Το ευθύγραμμο τμήμα $s_3(P_4, P_5)$ χωρίζεται σε τρία υποτμήματα και αντιστοιχίζεται σε (Partition3, (P_4, P_{45})), (Partition4, $(P_{45}, P_{45'})$) και (Partition5, $(P_{45'}, P_5)$).

Ευρετηρίαση (Indexing)

Αποτελεί σωστή επιλογή τεχνικής για τον γρήγορο εντοπισμό δεδομένων από μεγάλο όγκο πολύπλοκων συνόλων δεδομένων. Οι πληροφορίες μπορούν να ομαδοποιούνται ή να δημιουργούνται συνολικά αρχεία πληροφοριών [53]. Το μειονέκτημα του ευρετηρίου είναι το υψηλό κόστος ανάκτησης. [54]

Τα ευρετήρια χρησιμοποιούνται για την αποτελεσματική αναζήτηση δεδομένων με βάση ένα πεδίο διαφορετικό από το πρωτεύον κλειδί. Η βασική ιδέα πίσω από την κατανεμημένη ευρετηρίαση, η οποία υιοθετείται από τα περισσότερα υπάρχοντα πρωτότυπα και συστήματα, είναι η χρήση ενός σχήματος ευρετηρίασης δύο επιπέδων. Πρώτον, σε τοπικό (κόμβο) επίπεδο, χρησιμοποιούνται παραδοσιακές δομές ευρετηρίασης για την κινητικότητα. Στη συνέχεια, σε καθολικό επίπεδο, διατηρούνται περιληπτικές πληροφορίες που περιγράφουν τα τοπικά ευρετήρια, προκειμένου να είναι σε θέση να προσδιορίσει το υποσύνολο των κόμβων που αποθηκεύουν πιθανά αποτελέσματα ερωτημάτων.

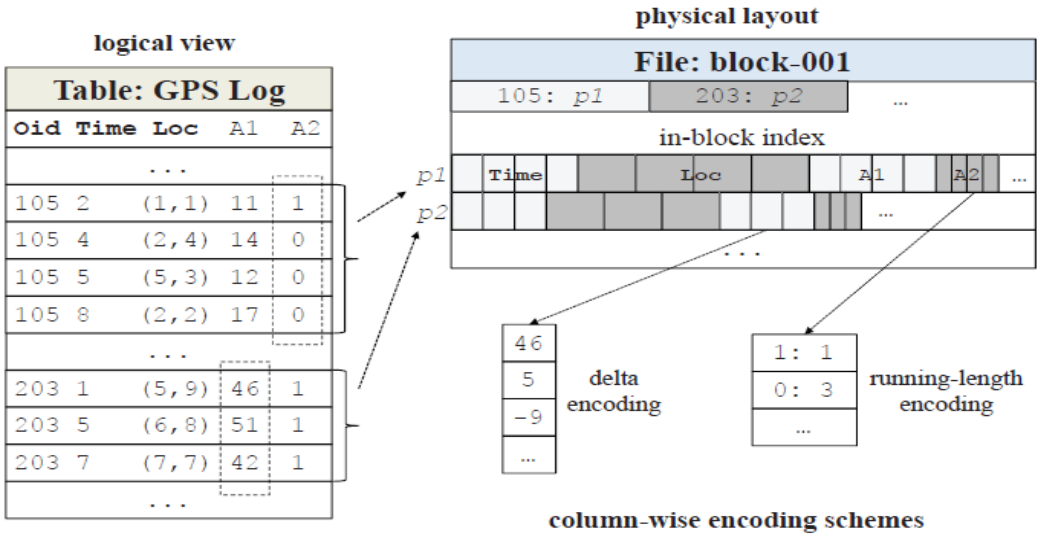
Τοπική ευρετηρίαση. Σε αυτό το επίπεδο, κάθε διαμέριση διατηρεί το δικό της τοπικό ευρετήριο. Το ερώτημα λειτουργεί με τη λογική του διασκορπισμού και της συγκέντρωσης, δηλαδή απαιτεί τη συλλογή αποτελεσμάτων από κάθε διαμέριση και στη συνέχεια τη συγχώνευσή τους. Σε τοπικό επίπεδο, οι δομές ευρετηρίου όπως τα R-tree, Quadtrees και Grid αρχεία χρησιμοποιούνται συνήθως σε χωρικό επίπεδο. Στην περίπτωση των χωροχρονικών δεδομένων, αρκετές περιπτώσεις ευρετηριάζουν πρώτα τη χρονική διάσταση και στη συνέχεια τις χωρικές διαστάσεις. Για παράδειγμα, το ST-Hadoop χρησιμοποιεί μια χρονική ιεραρχία χωρικών ευρετηρίων σε πολλαπλά επίπεδα χρονικής ανάλυσης.

Καθολική ευρετηρίαση. Σε καθολικό επίπεδο, η πιο συνηθισμένη προσέγγιση είναι η συλλογή συνοπτικών πληροφοριών από τα τοπικά ευρετήρια των κόμβων, προκειμένου να δημιουργηθεί ένα

καθολικό ευρετήριο που χρησιμοποιείται για την κατεύθυνση ερωτημάτων σε κόμβους. Ουσιαστικά, αυτή η περίληψη είναι πληροφορίες ορίων διαμερίσματος, όπως τα MBR ή MBR που περιγράφουν τα τοπικά δεδομένα σε κάθε κόμβο ([55],[48]). Τα καθολικά ευρετήρια αυξάνουν τον λανθάνοντα χρόνο εγγραφής αλλά παρέχουν ταχύτερες αναγνώσεις.

CloST

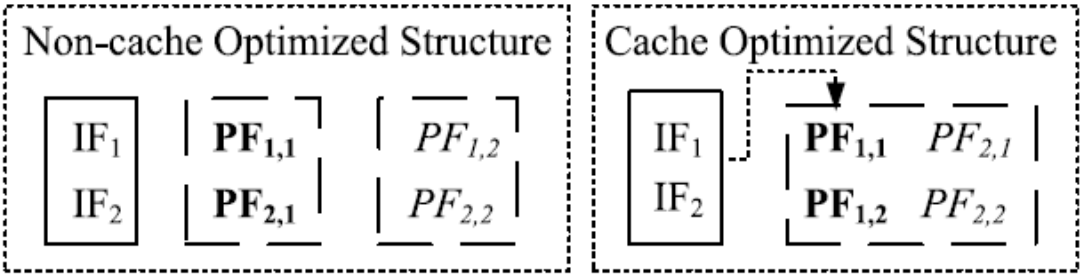
Όπως αναφέρθηκε και στο κομμάτι της διαμέρισης, κατά το οποίο τα δεδομένα διαμερίζονται σε αρχεία μπλοκ, κατά τη διάρκεια της διαδικασίας χωρικά ευρετήρια χρησιμοποιούνται ανάμεσα στα διαφορετικά επίπεδα της διαμέρισης. Επιπλέον, οι διαμερίσεις υψηλότερου επιπέδου, χρησιμεύουν σαν ευρετήρια για τις διαμερίσεις του τρίτου επιπέδου. Κατά τη δημιουργία του αρχείου μπλοκ (σχήμα 38), δηλαδή κατά τη δημιουργία της διαμέρισης επιπέδου 3, ένα ευρετήριο εντός του μπλοκ τοποθετείται στην αρχή του αρχείου για να αντιστοιχίσει ένα αναγνωριστικό του αντικειμένου στην αντίστοιχη θέση του τμήματος. Επομένως δεν χρειάζεται να αποθηκευτεί το αναγνωριστικό του αντικειμένου σε κάθε τμήμα, καθώς μπορεί να αναζητηθεί από το ευρετήριο του μπλοκ, με αποτέλεσμα να εξοικονομείται χώρος.



Σχήμα 38. Διάταξη αρχείου μπλοκ στο σύστημα CloST [23]

SharkDB

Τα ευρετήρια στο σύστημα SharkDB δημιουργούνται κατά τη διαδικασία της αποθήκευσης των δεδομένων, η οποία βασίζεται στα πλαίσια αποθήκευσης. Συγκεκριμένα, στην αποθήκευση που βασίζεται στο πλαίσιο I/P και για να αυξηθεί η απόδοση της επεξεργασίας των δεδομένων, απαιτείται να γεμίσει η κρυφή μνήμη του επεξεργαστή με όσο το δυνατόν περισσότερα χρήσιμα δεδομένα, για κάθε πρόσβαση στη μνήμα, ώστε να μειωθεί ο συνολικός αριθμός προσβάσεων από τον επεξεργαστή. Για να εξασφαλιστεί αυτή η προϋπόθεση, το SharkDB χρησιμοποιεί μια υβριδική δομή δεδομένων για την αποθήκευση των P-πλαισίων σε μια ομάδα πλαισίων με την οποία σχετίζονται. Η υβριδική δομή δεδομένων βασίζεται σε έναν δισδιάστατο πίνακα, ο οποίος δημιουργείται ως πίνακας [x][y], όπου το x ισούται με τον αριθμό των σημείων του I-πλαισίου σε αυτήν την ομάδα πλαισίων και το y ισούται με τον αριθμό των στηλών του P-πλαισίου σε αυτήν την ομάδα. Έτσι, η τιμή του x στον πίνακα είναι ο δείκτης των σημείων του I-πλαισίου, ενώ η τιμή του y υποδεικνύει τον αριθμό στήλης της ομάδας P-πλαισίου. Το αντικείμενο στον πίνακα [x][y] είναι το σημείο του P-πλαισίου. Για παράδειγμα, ο πίνακας[1][2] είναι το σημείο $PF_{1,2}$ του P-πλαισίου, το οποίο φαίνεται στο σχήμα 39, το 1 δείχνει το πρώτο σημείο του I-πλαισίου και το 2 είναι για τη δεύτερη στήλη του P-πλαισίου.



Σχήμα 39. Παράδειγμα δομής I/P-πλαισίου κρυφής μνήμης *[46]*

TrajStore

Όπως αναφέρθηκε και στην ενότητα της διαμέρισης, στο σύστημα TrajStore έχει αναπτυχθεί μία νέα, τεχνική χωρικής ευρετηρίασης πολλαπλών επιπέδων, η οποία διαχωρίζει και συγχωνεύει αναδρομικά τα κελιά προκειμένου να ελαχιστοποιηθεί ο αριθμός των μεταφορών του δίσκου. Η βασική ιδέα πίσω από αυτή την τεχνική είναι να τμηματοποιηθεί ο χώρος σε μια σειρά από ορθογώνια βέλτιστου μεγέθους για την ανάκτηση μεγάλου αριθμού σχετιζόμενων χωρικά υποτροχιών. Η προσέγγισή αυτή προσαρμόζει δυναμικά το ευρετήριο καθώς φορτώνονται νέα δεδομένα ή καθώς εξελίσσεται ο φόρτος εργασίας του ερωτήματος. Καθώς το σύστημα λαμβάνει συνεχώς δεδομένα τροχιάς από ένα σύνολο κινούμενων αντικειμένων, σε οποιαδήποτε χρονική στιγμή, οι πληροφορίες για κάθε κελί διατηρούνται σε ένα δυναμικό χωρικό ευρετήριο (Quadtree) που αντιπροσωπεύει τη χωρική διαμόρφωση των κελιών του ευρετηρίου και του οποίου τα κελιά δείχνουν σε μια σειρά σελίδων που αποθηκεύουν τα δεδομένα, όπως φαίνεται στο σχήμα 30.

Για την κατασκευή του ευρετηρίου, το σύστημα, αρχικά, ξεκινάει με ένα σύνολο τεσσάρων κενών κελιών που καλύπτουν ολόκληρη την περιοχή ενδιαφέροντος. Στη συνέχεια διαχωρίζει οποιαδήποτε εισερχόμενη τροχιά σύμφωνα με το χωρικό ευρετήριο, δημιουργώντας λίστες υποτροχιών, καθεμία από τις οποίες περιέχεται χωρικά σε ένα κελί. Ωστόσο, πριν γίνει εγγραφή μιας υποτροχιάς στις σελίδες που αντιστοιχούν στο κελί που περικλείει μέσα στο δίσκο, διαχωρίζεται η υποτροχιά μία επιπλέον φορά, δημιουργώντας επιπλέον υποτροχιές για κάθε ένα από τα (εικονικά) υπο-τεταρτημόρια του συγκεκριμένου κελιού.

TrajMesa

Στο TrajMesa έχουν σχεδιαστεί δύο είδη ευρετηρίων, ένα χρονικό ευρετήριο και ένα χωρικό ευρετήριο, για τη διαχείριση των δεδομένων τροχιάς:

Το χρονικό ευρετήριο που ακολουθεί τη μέθοδο XZT περιλαμβάνει μία μεταβλητή για την επίτευξη της ισορροπίας του φορτίου (*shard*), το αναγνωριστικό του κινούμενου αντικειμένου (*oid*), το αναγνωριστικό της τροχιάς (*tid*) και μία μεταβλητή που μετατρέπει χρονικά διαστήματα σε μονοδιάστατα κλειδιά.

$$shard + oid + \overbrace{BinNum(2\ \text{bytes}) + ElementCode(8\ \text{bytes})}^{XZT} + tid$$

Σχήμα 40. Σύνθεση κλειδιού για το αναγνωριστικό του χρονικού πίνακα ευρετηρίασης *[22]*

Καθώς δεν υπάρχει όριο για τη χρονική διάσταση, χωρίζεται η χρονική γραμμή σε χωριστές χρονικές περιόδους (*bins*) (μία ημέρα, μία εβδομάδα, ένας μήνας ή ένα έτος). Για κάθε bin, κωδικοποιούνται οι χρονικές περιοχές των οποίων η ώρα έναρξης βρίσκεται σε αυτήν. Ειδικά, όπως φαίνεται στο σχήμα 40, το XZT αποτελείται από δύο μέρη:

(1) BinNum. Υποδεικνύει σε ποιο χώρο αποθήκευσης της χρονικής περιόδου ανήκει ένα χρονικό εύρος.

(2) ElementCode. Αντιπροσωπεύει τη μετατόπιση ενός χρονικού εύρους στο bin του χρόνου του.

Το χωρικό ευρετήριο, που ακολουθεί τη μέθοδο XZ2, περιλαμβάνει: μία μεταβλητή για την επίτευξη της ισορροπίας του φορτίου (*shard*), το αναγνωριστικό της τροχιάς (*tid*) και μία μεταβλητή που προέρχεται από τις χωρικές πληροφορίες της τροχιάς, όπως φαίνεται στο σχήμα 41. Η μεταβλητή XZ2 [56] επεκτείνεται από το Z2 [57] και βρίσκει μια περιοχή της τροχιάς που αντιπροσωπεύει μη-σημειακά δεδομένα, όπως για παράδειγμα είναι οι γραμμές. Η μεταβλητή PosCode χρησιμοποιείται για να υποδείξει τη θέση των τροχιών με μεγαλύτερη ακρίβεια.

$$shard + \overbrace{PosCode(4\text{ bits}) + XZ2(8\text{ bytes})}^{XZ2^{+}} + tid$$

Σχήμα 41. Σύνθεση κλειδιού για τον χωρικό πίνακα ευρετηρίασης *[22]*

GCOTraj

Στο σύστημα GCOTraj, η ευρετηρίαση των δεδομένων πραγματοποιείται με δύο τρόπους, με βάση τη διαμέριση των δεδομένων και με βάση τη διαμέριση του χώρου.

Ευρετηρίαση με βάση τη διαμέριση των δεδομένων

Το GCOTraj διαχωρίζει τις τροχιές τόσο χωρικά όσο και χρονικά με μη επικαλυπτόμενα κελιά πλέγματος και διατάσσει τα κελιά πλέγματος με SFC ή GBO. Η διάταξη των κελιών του πλέγματος κρατά τις χωρικές/χρονικές κοντινές τροχιές επίσης κοντά στο δίσκο και αυτό, με τη σειρά του, μειώνει περαιτέρω τις αναζητήσεις του δίσκου.

Ευρετηρίαση με βάση τη διαμέριση του χώρου

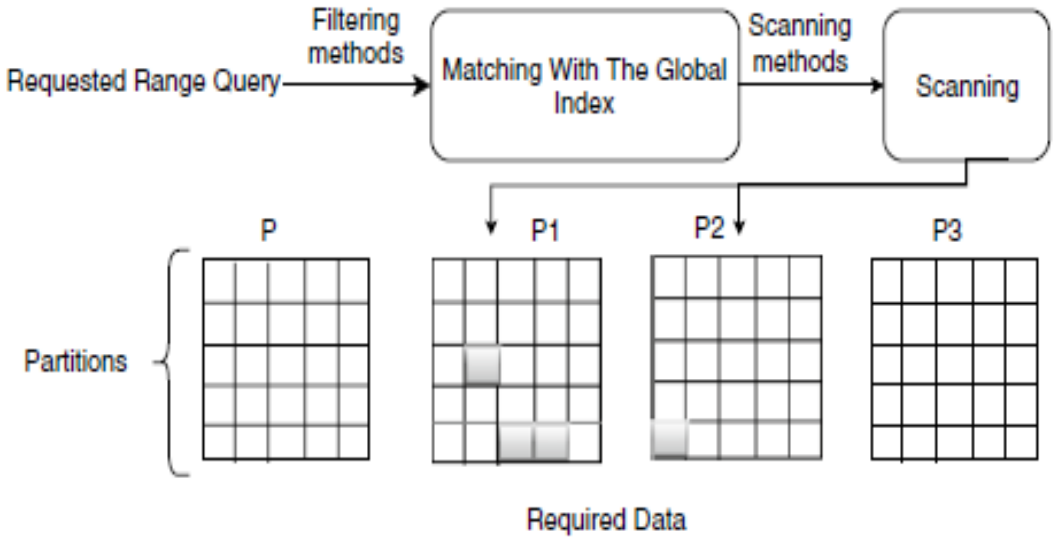
Το GCOTraj είτε χωρίζει το χώρο των δεδομένων της τροχιάς σε τμήματα με βάση ολόκληρο το χρονικό διάστημα των δεδομένων είτε χωρίζει τη χρονική διάσταση και τις χωρικές διαστάσεις μαζί σε ένα τρισδιάστατο κελί πλέγματος. Το GCOTraj διαμερίζει το χώρο των δεδομένων σε κελιά πλέγματος και ταξινομεί τα κελιά του πλέγματος χρησιμοποιώντας SFC ή GBO. Το GCOTraj χρησιμοποιεί μια δομή ευρετηρίου ενός επιπέδου που αντιστοιχίζει τμήματα της τροχιάς σε αντίστοιχα κελιά πλέγματος. Αυτό κάνει την αναζήτηση ευρετηρίου πιο γρήγορη. Η διάταξη των κελιών πλέγματος διατηρεί τη χωρική τοποθεσία των κελιών στο δίσκο, επομένως μειώνει τις αναζητήσεις δίσκου κατά την ανάκτηση συνεχών μπλοκ δεδομένων στο δίσκο σε μία ανάγνωση. Το GCOTraj ευρετηριάζει τις χωρικές και χρονικές διαστάσεις μαζί, επομένως είναι πολύ προσαρμοστικό και ευέλικτο σε διάφορα μοτίβα ερωτημάτων και μεταφέρει πολύ λιγότερα περιττά δεδομένα από το δίσκο.

HadoopTrajectory

Το HadoopTrajectory διαθέτει ένα καθολικό ευρετήριο που εφαρμόζεται στο περιβάλλον του HDFS. Το καθολικό ευρετήριο εφαρμόζεται στα αρχεία μεγάλων δεδομένων των κινούμενων αντικειμένων. Το καθολικό ευρετήριο συνδέεται απευθείας με το μπλοκ των αρχείων και αναγνωρίζει τις διαμερίσεις τους και τα μεταδεδομένα τους (δηλαδή, όλες τις πληροφορίες που περιγράφουν την τροχιά, όπως η ταχύτητα, ο τύπος κ.λπ.). Όλες οι πληροφορίες σχετικά με το καθολικό ευρετήριο αποθηκεύονται στο κύριο αρχείο. Χρησιμοποιείται για την οργάνωση των δεδομένων στους άλλους κόμβους για πιο αποτελεσματική ενημέρωση και ταχύτερη πρόσβαση. Τα δεδομένα χωρίζονται σε πολλές διαμερίσεις και αποθηκεύονται στους κόμβους εργασίας. Το καθολικό ευρετήριο χρησιμεύει ως κύριο μέσο για τον μοναδικό προσδιορισμό των διαμερίσεων στους κόμβους εργασίας.

Το HadoopTrajectory χρησιμοποιεί δύο δομές ευρετηρίου, τις οποίες ο χρήστης μπορεί να επιλέξει, το Grid και το R-tree. Το Grid είναι μια δομή διαμερίσεων χώρου που χωρίζει την τρισδιάστατη έκταση των δεδομένων εισόδου σε κύβους, καθένας από τους οποίους αντιστοιχίζεται σε ένα ή σε πολλά αρχεία. Αυτή η δομή επιτρέπει μια τροχιά να εκτείνεται σε πολλαπλά κελιά πλέγματος. Το 3DR-tree είναι μια δομή διαμέρισης των δεδομένων που αποθηκεύει τα τρισδιάστατα όρια των τροχιών. Και τα δύο ευρετήρια υποστηρίζουν ερωτήματα που βασίζονται σε χωροχρονικά παράθυρα. Όλες οι πληροφορίες σχετικά με το ευρετήριο αποθηκεύονται στην κύρια μνήμη του κύριου κόμβου. Το ευρετήριο αντιστοιχίζει τα πλαίσια οριοθέτησης απευθείας στα διαμερισμένα αρχεία, έτσι ώστε τα μεγάλα αρχεία

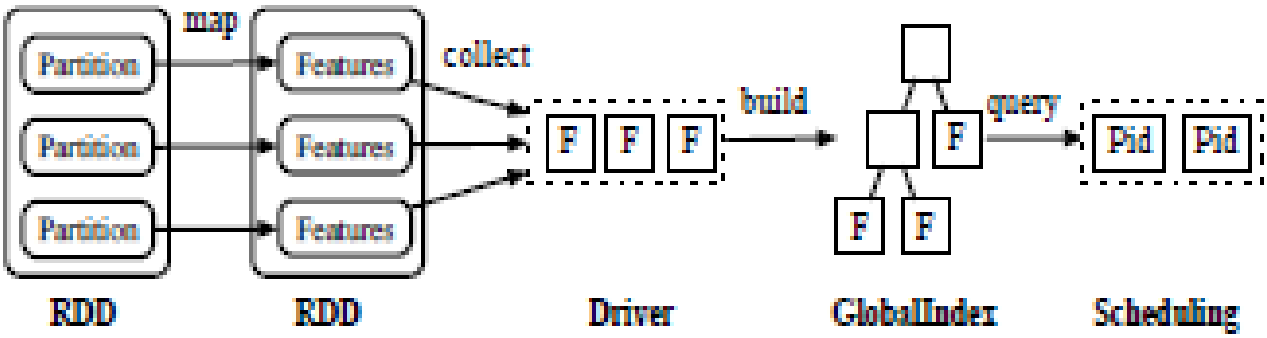
εισόδου να μην χρησιμοποιούνται πλέον. Στο HadoopTrajectory επιτρέπονται δύο επίπεδα λεπτομέρειας στη δημιουργία ενός ευρετηρίου: η τροχιά και το κινούμενο αντικείμενο. Στο επίπεδο της τροχιάς, ένα τρισδιάστατο πλαίσιο ανά τροχιά αποθηκεύεται στο ευρετήριο, ενώ στο επίπεδο του κινούμενου αντικειμένου, τα τρισδιάστατα πλαίσια όλων των τροχιών που ανήκουν στο ίδιο κινούμενο αντικείμενο ενώνονται σε ένα ενιαίο τρισδιάστατο πλαίσιο και αυτό το πλαίσιο αποθηκεύεται στο ευρετήριο. Στο σχήμα 42 απεικονίζεται η δημιουργία ενός καθολικού ευρετηρίου στο σύστημα HadoopTrajectory.



Σχήμα 42. Περιγραφή καθολικού ευρετηρίου στο HadoopTrajectory *[30]*

UlTraMan

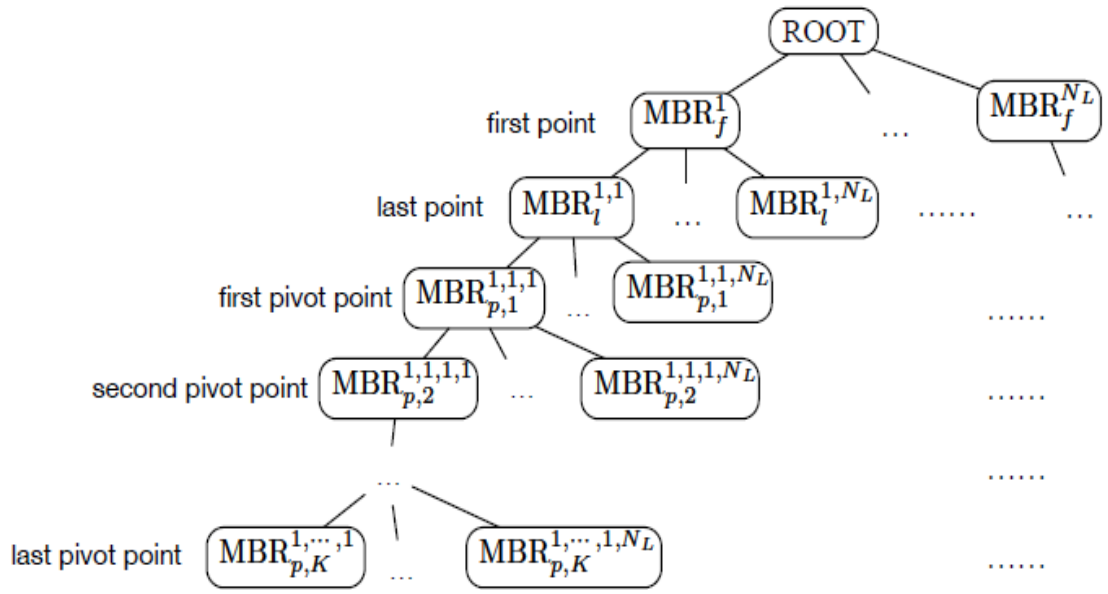
Το σύστημα UlTraMan υποστηρίζει τη δημιουργία ευρετηρίων τόσο σε καθολικό, όσο και σε τοπικό επίπεδο. Σε τοπικό επίπεδο, το UlTraMan παρέχει έναν ευέλικτο μηχανισμό για τη δημιουργία του ευρετηρίου σε κάθε δομή εκτελεστή. Με αυτόν τον τρόπο εκμεταλλεύεται πλήρως τις δυνατότητες τυχαίας προσπέλασης της μνήμης. Για την υλοποίηση του μηχανισμού δημιουργίας του ευρετηρίου, εισάγεται ένας διαχειριστής ευρετηρίου στο Spark για να συνεργαστεί με τον διαχειριστή των μπλοκ. Αναλαμβάνει τη διαδικασία κατασκευής ευρετηρίου, διατήρησης του ευρετηρίου και πρόσβασης των δεδομένων βάσει ευρετηρίου σύμφωνα με το επίπεδο αποθήκευσης του συνόλου των δεδομένων. Σε καθολικό επίπεδο ακολουθείται μια διαδικασία τριών βημάτων για τη δημιουργία καθολικών ευρετηρίων. Πρώτον, διεξάγεται ένα ερώτημα σε κάθε διαμέριση δεδομένων για τη δημιουργία χαρακτηριστικών. Τα χαρακτηριστικά μπορεί να είναι χωρικά πλαίσια οριοθέτησης, χρονικά διαστήματα, εύρη αναγνωριστικών ή κάτι άλλο που έχουν ορίσει οι χρήστες. Στη συνέχεια, όλα τα χαρακτηριστικά συλλέγονται στο πρόγραμμα οδήγησης του UlTraMan και δημιουργείται ένα καθολικό ευρετήριο με τα χαρακτηριστικά ως κλειδιά και τα αντίστοιχα αναγνωριστικά διαμερίσεων ως τιμές. Επίσης, το ενσωματωμένο καθολικό ευρετήριο αποθηκεύεται στη δομή TrajDataset για καθολική δρομολόγηση.



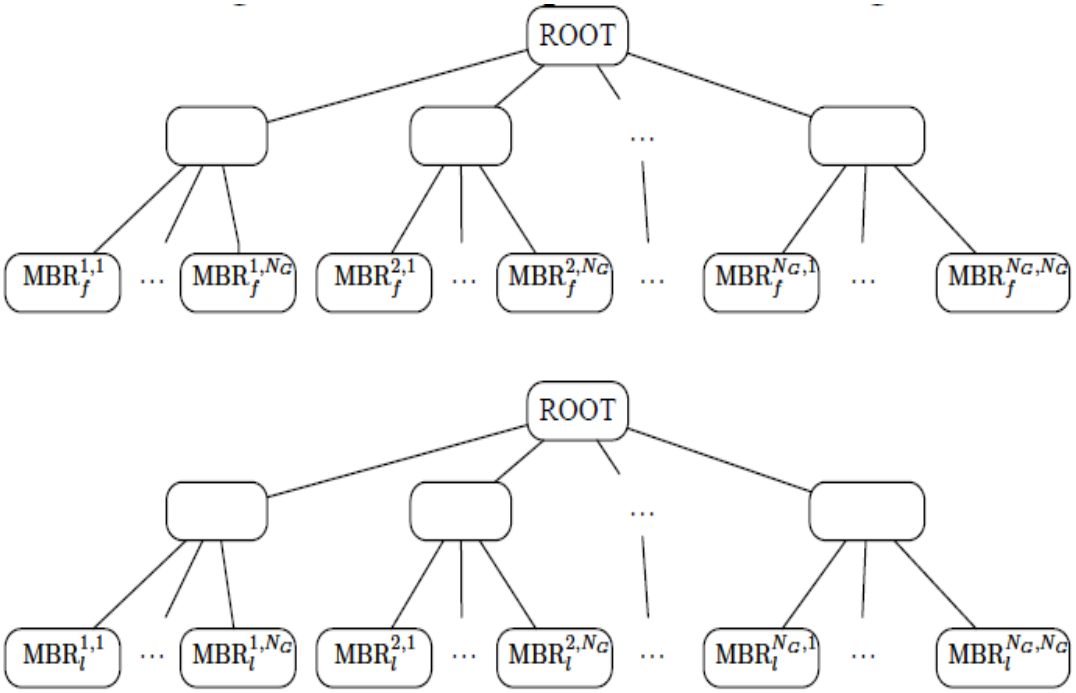
Σχήμα 43. Καθολικό ευρετήριο UlTraMan [17]

DITA

Κατά τη δημιουργία ενός ευρετηρίου σε μια τεράστια συλλογή τροχιών, αφού χωριστούν τα δεδομένα σε πολλαπλές διαμερίσεις σε διαφορετικά μηχανήματα για κατανεμημένους υπολογισμούς, στη συνέχεια, το DITA χρησιμοποιεί δύο επίπεδα ευρετηρίων: (1) το καθολικό ευρετήριο που βρίσκει σχετικές διαμερίσεις που μπορεί να περιέχουν τροχιές παρόμοιες με την τροχιά ενός ερωτήματος, το οποίο αποτελείται από δύο R-tree, ένα κατασκευασμένο στα MBR των πρώτων σημείων και ένα άλλο κατασκευασμένο στα MBR των τελευταίων σημείων και (2) το τοπικό ευρετήριο που βρίσκει υποψήφιες τροχιές σε κάθε σχετική διαμέριση τοπικά και έχει δημιουργηθεί πάνω στα σημεία pivot της τροχιάς. Δηλαδή, γενικεύοντας, χρησιμοποιούνται ευρετήρια με τη μορφή δέντρου, τα οποία βασίζονται και δημιουργούνται χρησιμοποιώντας το αρχικό σημείο, το τελικό σημείο και το σημείο pivot σε κάθε τροχιά (συμπεριλαμβανομένης της καθολικής ευρετηρίασης και της τοπικής ευρετηρίασης), όπως φαίνεται στα σχήματα 44 και 45.



Σχήμα 44. Τοπικό ευρετήριο DITA [19]



***Σχήμα 45. Καθολικό ευρετήριο DITA** [19]*

Distributed MobilityDB

Στο MobiltyDb, το γεγονός ότι οι διαμερίσεις είναι κανονικοί πίνακες βάσης δεδομένων στους κόμβους εργασίας, έχει σαν αποτέλεσμα κάθε κόμβος να μπορεί να δημιουργήσει τα δικά του τοπικά ευρετήρια των αντικειμένων στη διαμέρισή του. Το MobilityDB υποστηρίζει τρεις τύπους ευρετηρίου για χωροχρονικά χαρακτηριστικά: γενικευμένο δέντρο αναζήτησης (GiST), χωρικό διαμερισμό GiST (SP-GiST) και ένα R-tree για συγκρίσεις ισότητας. Ο διαχειριστής κατανομής, διανέμει τη δήλωση CREATE INDEX SQL στον κατανεμημένο πίνακα στους διάφορους κόμβους εργασίας, έτσι ώστε καθένας από αυτούς να δημιουργήσει ένα τοπικό ευρετήριο.

Summit

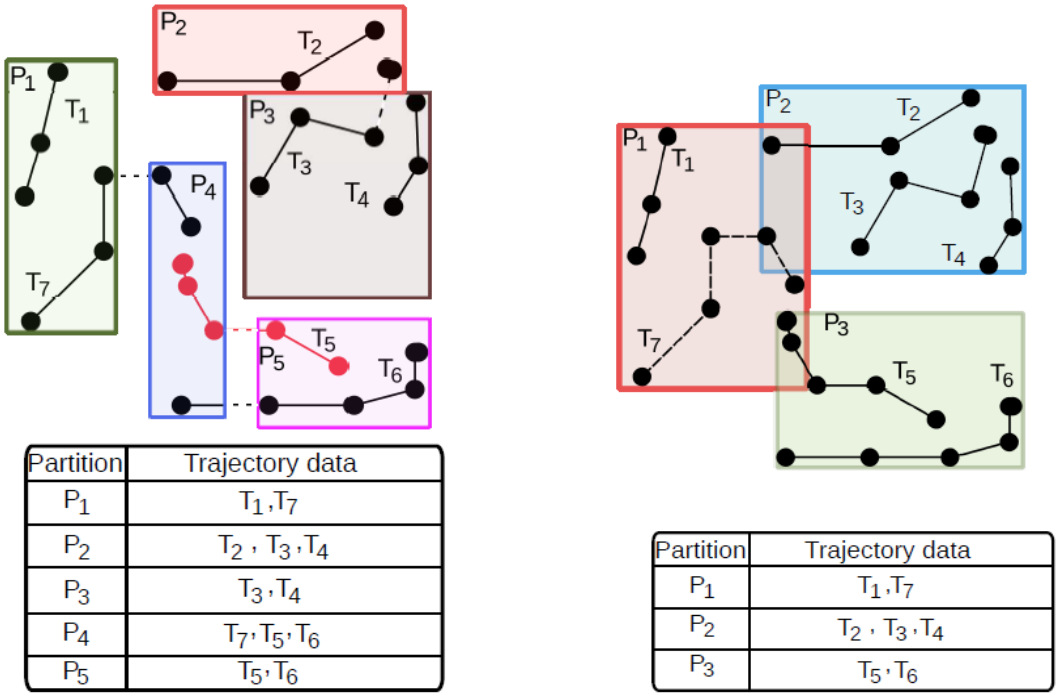
Το Summit χρησιμοποιεί μια δομή ευρετηρίου δύο επιπέδων καθολικής και τοπικής ευρετηρίασης.

Το ευρετήριο αποθηκεύεται στον κύριο κόμβο ως βοηθητικό αρχείο. Το καθολικό ευρετήριο χωρίζει τα δεδομένα στους κόμβους υπολογισμού, ενώ το τοπικό ευρετήριο οργανώνει τα δεδομένα μέσα σε κάθε κόμβο. Ο χώρος και ο χρόνος των τροχιών λαμβάνονται υπόψη σε κάθε επίπεδο.

Συγκεκριμένα, όσον αφορά το χωρικό επίπεδο ευρετηρίασης, το Summit είναι εξοπλισμένο με δύο προσεγγίσεις για κάθε χρονική τομή που προκύπτει από το βήμα του χρονικού τεμαχισμού, οι οποίες βασίζονται στον χώρο ή στα τμήματα. Το σχήμα 46 απεικονίζει τη λογική σχεδίαση και των δύο μεθόδων, όπου τα ορθογώνια αντιπροσωπεύουν τα όρια των διαμερίσεων HDFS ενώ οι τελείες και οι γραμμές απεικονίζουν τις πληροφορίες της τροχιάς.

(α) Βάσει χώρου: Αυτή η προσέγγιση διατηρεί τη χωροχρονική εγγύτητα μεταξύ των υποτροχιών. Τα όρια της διαμέρισης HDFS χωρίζουν τις τροχιές όπως φαίνεται στο σχήμα 46.

(β) Βάσει τμήματος: Πρόκειται για μια διαμέριση δεδομένων που εγγυάται ότι ολόκληρη η τροχιά αποθηκεύεται σε ένα μόνο μπλοκ HDFS, όπως φαίνεται στο σχήμα 46. Τα ελάχιστα όρια του ορθογωνίου αυτού του ευρετηρίου ενδέχεται να επικαλύπτονται. Όταν μια τροχιά τέμνεται με περισσότερα από ένα ορθογώνια, πρόκειται να αντιγραφεί μεταξύ των διαμερίσεων.



Partition	Trajectory data
P_1	T_1, T_7
P_2	T_2, T_3, T_4
P_3	T_3, T_4
P_4	T_7, T_5, T_6
P_5	T_5, T_6

Partition	Trajectory data
P_1	T_1, T_7
P_2	T_2, T_3, T_4
P_3	T_5, T_6

Σχήμα 46. Χωρική ευρετηρίαση Summit [38]

Dragoon

Η κατασκευή ευρετηρίου στο σύστημα Dragoon, περιλαμβάνει τη δημιουργία τοπικού ευρετηρίου σε κάθε διαμέριση δεδομένων και τη δημιουργία ενός καθολικού ευρετηρίου που αφορά το σύνολο των διαμερίσεων. Τα τοπικά ευρετήρια και το καθολικό ευρετήριο αποθηκεύονται, επίσης, στη δομή Chronicle Map και πρέπει να ενημερωθούν μετά τη διαδικασία ανανέωσης των δεδομένων.

Στην περίπτωση της ανάλυσης δεδομένων εκτός σύνδεσης, μετά τη διαμέριση των δεδομένων, δημιουργούνται τοπικά ευρετήρια και ένα καθολικό ευρετήριο, παρόμοιο με τον τρόπο που γίνεται στο UlTraMan. Συγκεκριμένα, το Dragoon δημιουργεί ένα τοπικό ευρετήριο σε κάθε διαμέριση δεδομένων και, στη συνέχεια, δημιουργεί ένα καθολικό ευρετήριο με βάση τα χαρακτηριστικά όλων των διαμερίσεων. Για παράδειγμα, για να δημιουργήσει ένα καθολικό R-tree, το Dragoon χρειάζεται να συλλέξει τα αναγνωριστικά και τα ελάχιστα όρια (MBR) από κάθε διαμέριση δεδομένων. Η κατασκευή ευρετηρίου στην ανάλυση εκτός σύνδεσης συνήθως εκτελείται μόνο μία φορά και τα (τοπικά και καθολικά) ευρετήρια γενικά δεν χρειάζεται να ενημερωθούν.

Στην περίπτωση της ανάλυσης δεδομένων εντός σύνδεσης, η κατασκευή των ευρετηρίων είναι παρόμοια με εκείνη των δεδομένων εκτός σύνδεσης και, επιπλέον, όχι μόνο τα δεδομένα τροχιάς στις διαμερίσεις των δεδομένων αλλά και τα τοπικά και καθολικά ευρετήρια πρέπει να ενημερωθούν κατά το στάδιο της διαδικασίας συγχώνευσης.

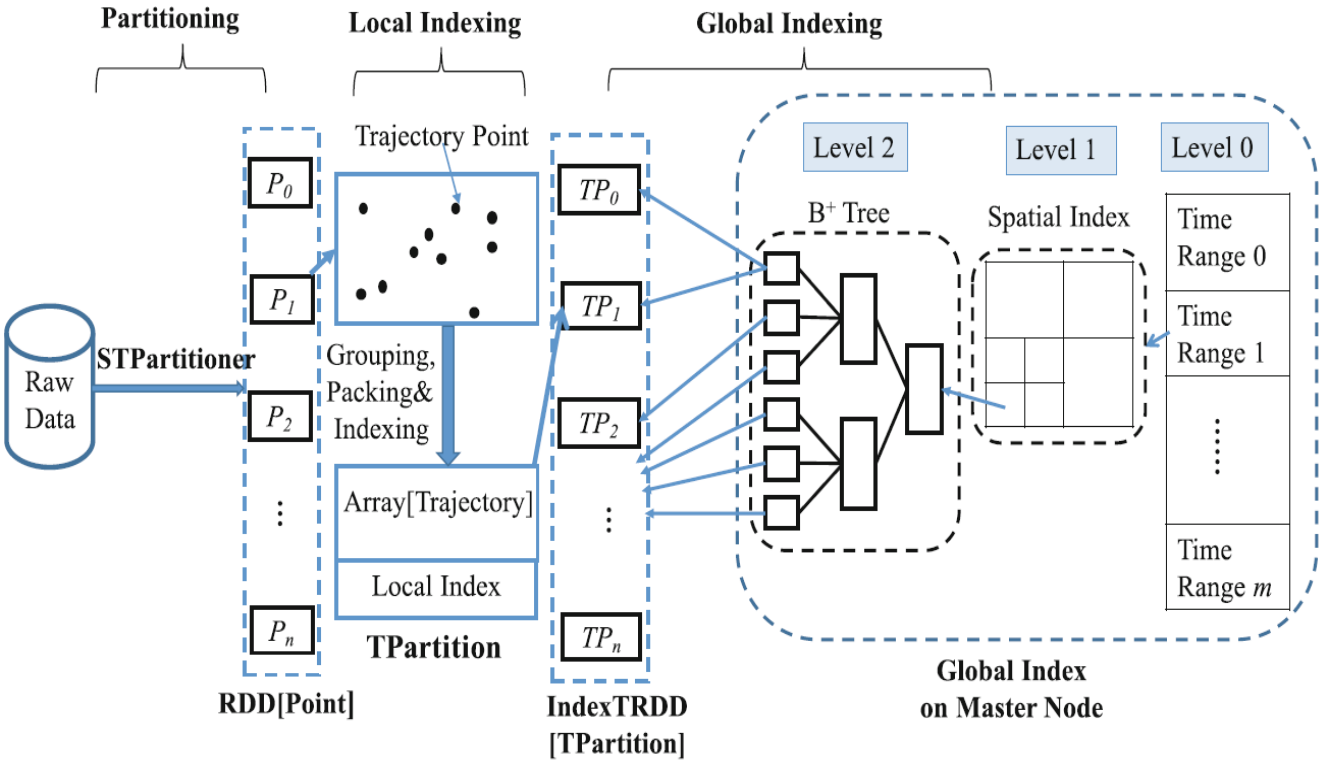
TrajSpark

Το σύστημα TrajSpark χρησιμοποιεί τη δομή IndexTRDD που αλλάζει τη δομή αποθήκευσης του TRDD ενσωματώνοντας ένα τοπικό ευρετήριο κατακερματισμού σε κάθε διαμέριση, το οποίο αντιστοιχίζει το αναγνωριστικό του κινούμενου αντικειμένου κάθε τροχιάς με τα ευρετήριά του και λαμβάνει έναν

υπολογισμό για να διατηρήσει την τροχιά ενός κινούμενου αντικειμένου. Επιπλέον, κατασκευάζεται ένα καθολικό ευρετήριο πάνω στις διαμερίσεις των δεδομένων του IndexTRDD για να περικοπούν οι διαμερίσεις που δεν είναι χρήσιμες. Πιο αναλυτικά:

Διαδικασία τοπικής ευρετηρίασης. Μετά τη φάση της διαμέρισης, το σύνολο των δεδομένων εξακολουθεί να είναι ένα RDD σημείων τροχιάς. Σε αυτό το σημείο, το RDD μετατρέπεται σε TRDD ομαδοποιώντας και πακετάροντας πρώτα τέτοια σημεία. Στη συνέχεια, το σύστημα προσθέτει ένα τοπικό ευρετήριο στην κορυφή κάθε διαμέρισης που αντιστοιχίζει το αναγνωριστικό του κινούμενου αντικειμένου (MOID) κάθε τροχιάς στους δείκτες του. Αυτή η συνδυασμένη δομή δεδομένων του πίνακα ευρετηρίου και τροχιάς καλείται TPpartition. Έτσι, ολόκληρο το σύνολο δεδομένων μετατρέπεται σε ένα RDD από TPpartitions, όπου το RDD είναι IndexTRDD. Τέλος, πραγματοποιείται συλλογή του αναγνωριστικού (κάθε διαμέριση του RDD έχει ένα μοναδικό αναγνωριστικό), των χωρικών και χρονικών περιοχών κάθε διαμέρισης δεδομένων (ένα αντικείμενο TPpartition) για την κατασκευή του καθολικού ευρετηρίου.

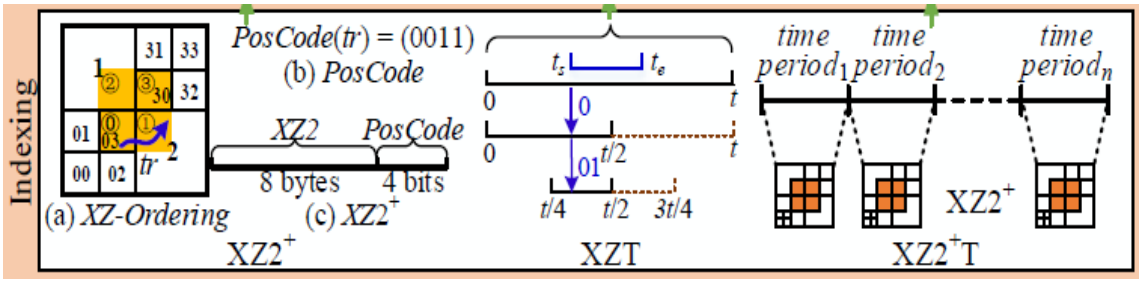
Διαδικασία καθολικής ευρετηρίασης. Σε αυτό το σημείο, δημιουργείται το καθολικό ευρετήριο (gIndex) πάνω σε όλες τις διαμερίσεις. Όπως φαίνεται στο σχήμα 47, το gIndex είναι ένα υβριδικό ευρετήριο τριών επιπέδων. Αρχικά, τα δεδομένα διαιρούνται με τα χρονικά διαστήματα επιπέδου-0 σύμφωνα με το χρονικό τους χαρακτηριστικό. Κάθε χρονικό διάστημα αντιστοιχεί σε ένα χωρικό ευρετήριο επιπέδου-1 που χρησιμοποιείται από το STPartitioner. Για την ευρετηρίαση των διαμερίσεων που ανήκουν στο ίδιο χωρικό όριο, χρησιμοποιείται ένα B^+-Tree επιπέδου-2. Όταν το TrajSpark αρχικοποιείται με την πρώτη παρτίδα δεδομένων, το ευρετήριο επιπέδου-0 περιέχει μόνο μία τιμή (τη χρονική σήμανση έναρξης αυτής της παρτίδας δεδομένων) και το χωρικός ευρετήριο επιπέδου-1 είναι το ίδιο που χρησιμοποιείται στον STPartitioner. Οι χωρικές και χρονικές πληροφορίες που συλλέγονται από όλες τις διαμερίσεις χρησιμοποιούνται για την κατασκευή των ευρετηρίων επιπέδου-2. Κάθε χωρική περιοχή στο ευρετήριο επιπέδου-1 αντιστοιχεί με ένα ευρετήριο επιπέδου-2. Το TrajSpark διατηρεί το gIndex στη μνήμη του κύριου κόμβου και το ενημερώνει όταν φτάνουν νέες διαμερίσεις δεδομένων.



Σχήμα 47. Διαδικασία ευρετηρίασης πρωτογενών δεδομένων στο TrajSpark [41]

JUST-Traj

Το JUST-Traj παρέχει τρεις μεθόδους χωροχρονικής ευρετηρίασης για τις τροχιές, όπως φαίνεται στο σχήμα 48. (1) XZ2⁺, το οποίο είναι ένα ευρετήριο για αποτελεσματική αναζήτηση της τροχιάς από ένα χωρικό διάστημα. Χρησιμοποιεί έναν χώρο ευρετηρίου XZ-Ordering [56] για να αναπαραστήσει το ελάχιστο οριοθετημένο ορθογώνιο μιας τροχιάς και έναν κωδικό θέσης για να περιγράψει το σχήμα της τροχιάς. (2) XZT. Μια τροχιά έχει ένα χρονικό διάστημα από το σημείο έναρξης έως το σημείο λήξης. Καταγράφεται το χρονικό διάστημα μιας τροχιάς με το ευρετήριο του χρονικού διαστήματος (XZT) που προτείνεται στο TrajMesa [58], το οποίο βοηθά το JUST-Traj να αναζητήσει τροχιές εντός ενός δεδομένου χρονικού διαστήματος. (3) XZ2⁺T, το οποίο είναι ένα χωροχρονικό ευρετήριο για την αναζήτηση τροχιών από ένα δεδομένο χωροχρονικό διάστημα. Αρχικά διαχωρίζεται η χρονική διάσταση σε πολλαπλές ασύνδετες χρονικές περιόδους και, στη συνέχεια, κατασκευάζεται ένα μεμονωμένο ευρετήριο XZ2⁺ σε κάθε χρονική περίοδο.

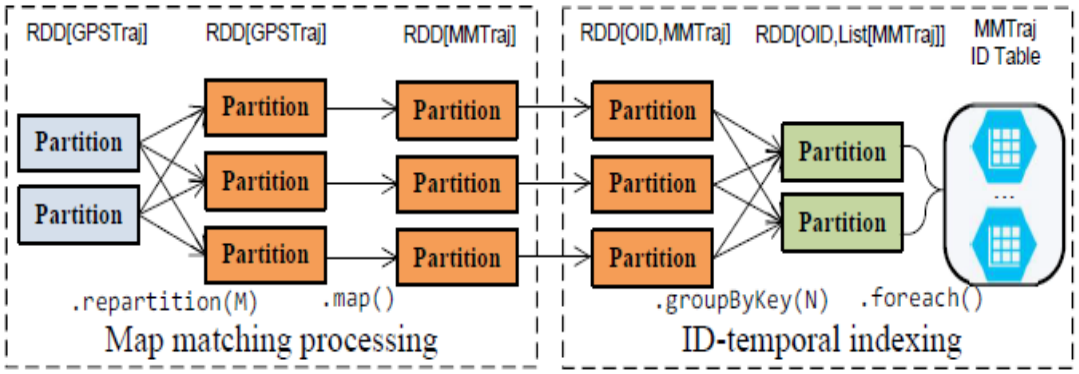


Σχήμα 48. Διαδικασία ευρετηρίασης στο JUST-Traj [42]

CloudTP

Ευρετηρίαση με χρονικό αναγνωριστικό. Οι τμηματοποιημένες τροχιές, τα σημεία παραμονής και οι τροχιές που αντιστοιχίζονται σε χάρτη αποθηκεύονται σε αντίστοιχους πίνακες που ονομάζονται με το αναγνωριστικό του αντικειμένου. Για παράδειγμα, στο δεξί μέρος του σχήματος 49, εκχωρείται πρώτα ένα κλειδί σε κάθε τροχιά που αντιστοιχεί στο χάρτη, το οποίο είναι το αναγνωριστικό του αντικειμένου στο οποίο ανήκει. Στη συνέχεια καλείται το groupByKey(N) για να ομαδοποιήσει τις τροχιές σύμφωνα με τα κλειδιά, γεγονός που μπορεί να μειώσει τον αριθμό των διαμερίσεων σε N. Τέλος, κάθε ομάδα τροχιών εισάγεται στον αντίστοιχο πίνακα (που προσδιορίζεται από το αναγνωριστικό του αντικειμένου) κατά σύνολα χρησιμοποιώντας την εντολή foreach, η οποία αποφεύγει το υψηλό κόστος ανοίγματος/κλεισίματος των συχνών μικρών εγγραφών. Για να απαντήσουμε στο ερώτημα χρονικού αναγνωριστικού, βρίσκουμε πρώτα τον πίνακα με το αναγνωριστικό ονόματος του αντικειμένου και, στη συνέχεια, φιλτράρουμε το χρονικό διάστημα με βάση τα κλειδιά.

Χωροχρονική Ευρετηρίαση. Τα σημεία GPS στις τμηματικές τροχιές και τα σημεία παραμονής, αποθηκεύονται στους αντίστοιχους πίνακες που ονομάζονται ανά διάστημα. Διαχωρίζεται ολόκληρος ο χώρος σε ομοιόμορφα πλέγματα και εκχωρείται ένα κλειδί σε κάθε χωρικό δεδομένο, το οποίο είναι το αναγνωριστικό του πλέγματος στο οποίο ανήκει. Στη συνέχεια, οι εγγραφές που ομαδοποιούνται κατά πλέγματα αποθηκεύονται σε αντίστοιχους πίνακες πλέγματος, όπως το ευρετήριο με χρονικό αναγνωριστικό.



Σχήμα 49. Αντιστοίχιση χάρτη και δημιουργία ευρετηρίου στο σύστημα CloudTP *[26]*

PRADASE

Στο σύστημα PRADASE χρησιμοποιείται ένα υβριδικό ευρετήριο, το οποίο συνδυάζει δύο μεθόδους ευρετηρίασης: υιοθετείται το ευρετήριο πολλαπλών επιπέδων βάσει διαμερίσεων (PMI) για ερωτήματα χωροχρονικού διαστήματος και το ευρετήριο αντεστραμμένου αντικειμένου (OII) για τα ερωτήματα που βασίζονται σε τροχιά.

Ευρετήριο πολλαπλών επιπέδων βάσει διαμέρισης. Κάθε διαμέριση μπορεί να θεωρηθεί ως ένας χώρος που περιέχει σχεδόν τον ίδιο όγκο δεδομένων τροχιάς. Ταυτόχρονα, κάθε διαμέριση περιέχει μόνο τμήματα τροχιάς που βρίσκονται πλήρως μέσα στη διαμέριση και όλα τα δεδομένα που καλύπτουν αυτή τη διαμέριση διατηρούνται σε ένα κομμάτι. Με τον περιορισμό ότι κάθε κομμάτι δεδομένων περιέχει μόνο τμήματα τροχιάς που ανήκουν στην ίδια διαμέριση, το σύστημα PRADASE εκμεταλλεύεται αυτή τη συνθήκη για να δημιουργήσει έναν κατακερματισμό ευρετηρίου ώστε να επιταχύνει το ερώτημα διαστήματος. Δεδομένου οποιουδήποτε χωροχρονικού διαστήματος, δημιουργούνται όλες οι υποψήφιες διαμερίσεις τις οποίες καλύπτει το διάστημα του ερωτήματος. Τα αποτελέσματα του ερωτήματος πρέπει να περιέχονται σε αυτές τις υποψήφιες διαμερίσεις και, στη συνέχεια, το σύστημα εντοπίζει απευθείας τη διεύθυνση όπου αποθηκεύεται το τμήμα των δεδομένων κατακερματίζοντας το αναγνωριστικό της διαμέρισης. Το σύστημα σαρώνει μόνο μερικά υποψήφια κομμάτια αντί για όλα τα κομμάτια των δεδομένων.

Για περαιτέρω επεκτάσεις, το σύστημα μπορεί να δημιουργήσει ένα ευρετήριο πολλαπλών επιπέδων για κάθε κομμάτι στον τοπικό κόμβο όπου αποθηκεύεται αυτό το κομμάτι.

Αντεστραμμένο Ευρετήριο Αντικειμένου. Προκειμένου να επιταχυνθεί η αναζήτηση με βάση την τροχιά ορισμένων κινούμενων αντικειμένων, σχεδιάστηκε το ευρετήριο για κάθε κινούμενο αντικείμενο που ονομάζεται ευρετήριο ανεστραμμένου αντικειμένου. Συλλέγονται όλες οι ιστορικές τροχιές κάθε αντικειμένου και αποθηκεύονται αυτά τα δεδομένα μαζί. Χρησιμοποιείται μια τριπλέτα <OID, Pl, T> για να παρουσιαστεί το μοντέλο δεδομένων του OII, όπου OID είναι το αναγνωριστικό του αντικειμένου, Pl είναι η τρέχουσα θέση του αντικειμένου συμπεριλαμβανομένης της χωρικής θέσης και της αντίστοιχης χρονικής σφραγίδας, T είναι η ιστορική τροχιά αυτού του αντικειμένου που αναπαρίσταται ως μοντέλο γραμμικών τμημάτων. Με δεδομένο λοιπόν οποιοδήποτε αναγνωριστικό αντικειμένου, η τελευταία γνωστή τοποθεσία και όλες οι αντίστοιχες τροχιές θα επιστραφούν γρήγορα.

Η κατανομή των αναγνωριστικών αντικειμένου είναι διακριτή και τα διαφορετικά αντικείμενα δεν είναι σταθερά, επομένως όλα τα ιστορικά δεδομένα των αντικειμένων μπορούν να αντιστοιχιστούν ομοιόμορφα σε διαφορετικούς κόμβους μέσω κατακερματισμού του αναγνωριστικού του αντικειμένου. Έχει σχεδιαστεί, λοιπόν, ένα ευρετήριο κατακερματισμού δύο επιπέδων για ερωτήματα με αντικειμενοστραφή προσανατολισμό. Χρησιμοποιώντας το πρώτο επίπεδο επιστρέφεται ο κόμβος δεδομένων στον οποίο αποθηκεύεται το κομμάτι δεδομένων και, στη συνέχεια, τα δεδομένα τροχιάς μπορούν να βρεθούν με το κάλεσμα του δεύτερου ευρετηρίου.

Επεξεργασία ερωτημάτων για μεγάλα δεδομένα τροχιάς

Παρακάτω θα αναφερθούμε στα είδη των ερωτημάτων, καθώς και στις τεχνικές που χρησιμοποιούνται από τα συστήματα που αναφέραμε προηγουμένως, για την επεξεργασία των ερωτημάτων αυτών.

CloST

Για ένα ερώτημα Q(*I, S, T*) ενός αντικειμένου, το CloST χρησιμοποιεί τον πίνακα μεταδεδομένων και τα ευρετήρια εντός μπλοκ για τον εντοπισμό των εγγραφών. Αρχικά ανακαλύπτει τα σύνολα όσον αφορά τα I και T σύμφωνα με τις παραμέτρους διαμέρισης του πρώτου επιπέδου. Στη συνέχεια, εντοπίζει τις περιοχές που τέμνουν το S στα παραπάνω σύνολα σύμφωνα με τα ευρετήρια των διαμερίσεων του δεύτερου επιπέδου. Σε αυτές τις περιοχές, ανακαλύπτονται περαιτέρω τα μπλοκ των οποίων το χρονικό διάστημα τέμνει το T σύμφωνα με τα ευρετήρια διαμερίσεων του τρίτου επιπέδου. Για κάθε εμπλεκόμενο αρχείο μπλοκ, διαβάζει το ευρετήριο εντός του μπλοκ, παίρνει τη θέση του τμήματος που αντιστοιχεί στο I, αναζητά τη θέση και διαβάζει ολόκληρο το τμήμα στη μνήμη. Όλες τις εγγραφές συγκεντρώνονται αφού αποσυμπιεστούν τα δεδομένα του τμήματος και εκτελείται το τελικό φιλτράρισμα χρησιμοποιώντας τα S και T. Για ένα ερώτημα πολλών αντικειμένων Q(*S, T*), το CloST εκμεταλλεύεται το MapReduce για να εκτελέσει το ερώτημα παράλληλα. Αρχικά, ανακαλύπτονται όλα τα αρχεία μπλοκ των οποίων το χωροχρονικό εύρος τέμνει τα S και T. Στη συνέχεια, ξεκινάει μια εργασία αντιστοίχισης για κάθε εμπλεκόμενο αρχείο μπλοκ. Κάθε εργασία αντιστοίχισης σαρώνει διαδοχικά το αντίστοιχο αρχείο μπλοκ, αποσυμπιέζει τα δεδομένα και συγκεντρώνει τις εγγραφές. Στη συνέχεια φιλτράρει κάθε εγγραφή κατά S και T και τελικά βγάζει το αποτέλεσμα.

SharkDB

Υπάρχουν δύο βασικοί τύποι ερωτημάτων, το ερώτημα παραθύρου και το ερώτημα πλησιέστερου γείτονα top-k (kNN). Το ερώτημα παραθύρου, θα βρει όλες τις τροχιές στο σύνολο των δεδομένων που είναι ενεργές κατά τη διάρκεια μιας δεδομένης περιόδου και διατρέχουν μια δεδομένη περιοχή. Το ερώτημα kNN, θα βρει τις top-k τροχιές στο σύνολο των δεδομένων τροχιάς που είναι κοντά σε ένα δεδομένο σημείο και ενεργές κατά τη διάρκεια μιας δεδομένης χρονικής περιόδου.

TrajStore

Στο TrajStore έχει αναπτυχθεί ένα σύστημα αποθήκευσης που είναι βελτιστοποιημένο για χωροχρονικά ερωτήματα σε μεγάλα σύνολα δεδομένων τροχιάς. Για να ανακτήσει τα δεδομένα τροχιάς για ένα δεδομένο ερώτημα, το TrajStore εκτελεί μια αναζήτηση στο χωρικό ευρετήριο για να προσδιοριστούν τα κελιά που τέμνουν το ερώτημα στο χωρικό επίπεδο. Εξετάζεται το χρονικό διάστημα όλων των σελίδων που σχετίζονται με τα κελιά που τέμνουν το ερώτημα και ανακτώνται αυτές οι σελίδες που περιέχουν τα δεδομένα στο χρονικό διάστημα που καθορίζεται από το ερώτημα. Τέλος, ανακατασκευάζονται οι τροχιές συγχωνεύοντας όλες τις δευτερεύουσες τροχιές με το ίδιο αναγνωριστικό τροχιάς και κόβεται το προκύπτον σύνολο τροχιών ακολουθώντας τα ακριβή όρια του ερωτήματος ώστε να επιστραφούν τα αποτελέσματα.

TrajMesa

Το TrajMesa περιλαμβάνει τέσσερα είδη ερωτημάτων, το χρονικό ερώτημα αναγνωριστικού, το ερώτημα χωρικού διαστήματος, το ερώτημα ομοιότητας και το ερώτημα πλησιέστερων γειτόνων (knn).

Χρονικό ερώτημα αναγνωριστικού. Για την εκτέλεση του ερωτήματος το TrajMesa ενεργοποιεί παράλληλες λειτουργίες σάρωσης στον υποκείμενο χώρο αποθήκευσης των δεδομένων χρησιμοποιώντας τα παράθυρα ερωτημάτων, με δεδομένο το ερώτημα, το χρονικό διάστημα και το αναγνωριστικό του κινούμενου αντικειμένου. Όταν ολοκληρωθούν όλες οι λειτουργίες σάρωσης, καταργούνται οι τροχιές που δεν είναι κατάλληλες.

Ερώτημα χωρικού διαστήματος. Το ερώτημα χωρικού διαστήματος βασίζεται στον πίνακα ευρετηρίασης χωρικού διαστήματος. Η εκτέλεσή του είναι παρόμοια με το χρονικό ερώτημα αναγνωριστικού, αλλά χρησιμοποιείται ο πίνακας ευρετηρίασης του χωρικού διαστήματος σε αυτήν την περίπτωση. Τέλος, οι τροχιές βελτιώνονται με βάση το δεδομένο χωρικό διάστημα.

Ερώτημα ομοιότητας. Το ερώτημα ομοιότητας θεωρεί το ερώτημα χωρικού διαστήματος ως δομικό στοιχείο.

Ερώτημα πλησιέστερων γειτόνων (knn). Η κύρια ιδέα του ερωτήματος k-NN είναι η επαναληπτική επέκταση του ερωτήματος χωρικού διαστήματος, μέχρι να βρεθούν οι περισσότερες k παρόμοιες τροχιές.

GCOTraj

Κάθε παραλλαγή του GCOTraj προσεγγίζει με διαφορετικό τρόπο την επεξεργασία των ερωτημάτων διαστήματος.

Το ερώτημα στα δεδομένα που αποθηκεύονται από το GCOTrajSP ξεκινά με τον υπολογισμό του αριθμού των κελιών που επικαλύπτονται με το διάστημα των ερωτημάτων. Στη συνέχεια, το ευρετήριο αναζητείται για να ανακτηθούν τα μπλοκ δεδομένων από το δίσκο. Τα συνεχή μπλοκ δεδομένων μεταφέρονται σε μία αναζήτηση.

Η αναζήτηση δεδομένων που αποθηκεύονται από το 2D-sliced GCOTraj περιλαμβάνει πρώτα την αναζήτηση του ευρετηρίου για να προσδιοριστεί ποιο κομμάτι ή ποια κομμάτια επικαλύπτουν το ερώτημα προσωρινά και, στη συνέχεια, ποια κελιά πλέγματος σε κάθε κομμάτι επικαλύπτουν το ερώτημα χωρικά. Χρησιμοποιώντας τα δεδομένα ευρετηρίου, ανακτώνται τα αντίστοιχα μπλοκ κελιών πλέγματος. Στη συνέχεια, τα δεδομένα που ανακτώνται φιλτράρονται για να επιστρέψουν τα ακριβή αποτελέσματα.

Για να απαντήσει σε ένα ερώτημα εύρους, το 3D GCOTraj επεξεργάζεται ποια κελιά πλέγματος επικαλύπτονται με τον κύβο του ερωτήματος και αναζητά το ευρετήριο για τη θέση στο δίσκο για κάθε κελί πλέγματος. Χρησιμοποιώντας τη μετατόπιση διεύθυνσης και το μέγεθος του μπλοκ δεδομένων για κάθε κελί πλέγματος, η προσέγγιση ανακτά τις σελίδες των δεδομένων από το δίσκο. Για τις διαδοχικές σελίδες στο δίσκο (σελίδες δίσκου για συνεχόμενα κελιά πλέγματος), το 3D GCOTraj μεταφέρει το μπλοκ των δεδομένων σε μία ανάγνωση για να αποθηκεύσει τις αναζητήσεις του δίσκου.

HadoopTrajectory

Στο HadoopTrajectory, σε επίπεδο επεξεργασίας, υπάρχει ένας αριθμός τελεστών για τα κινούμενα αντικείμενα που αποτελούν από μόνοι τους εργασίες MapReduce και μπορούν να χρησιμοποιηθούν στον κώδικα MapReduce του κάθε χρήστη, ώστε να εκτελέσουν πολύπλοκες εργασίες, με πιο αξιοσημείωτες τα ερωτήματα διαστήματος για τις τροχιές. Οι κυριότεροι τελεστές είναι:

Ο τελεστής Passes είναι μια εργασία MapReduce που χρησιμοποιείται για την εύρεση όλων των κινούμενων αντικειμένων που έχουν περάσει ένα συγκεκριμένο χωρικό σημείο.

Ο τελεστής Trajectory χρησιμοποιείται για την ανάκτηση της συγκεκριμένης τροχιάς κινούμενου αντικειμένου με βάση το αναγνωριστικό. Ανακτά όλα τα τμήματα μιας τροχιάς κινούμενου αντικειμένου.

Ο τελεστής Length χρησιμοποιείται για τον υπολογισμό της διανυόμενης απόστασης της τροχιάς του κινούμενου αντικειμένου. Είναι η διαδρομή που ακολουθεί ένα κινούμενο αντικείμενο μέσα στο χώρο σε συνάρτηση με το χρόνο.

Ο τελεστής ZoneTraj χρησιμοποιείται για την ανάκτηση του συνόλου των κινούμενων αντικειμένων σε μια συγκεκριμένη περιοχή που περιβάλλει ένα σημείο ενδιαφέροντος. Μοιάζει πολύ με τον τελεστή Passes εκτός από το ότι το όρισμα του ερωτήματος είναι ένας κύκλος που ορίζεται από τον χρήστη.

UlTraMan

Στο UlTraMan, οι υποστηριζόμενοι τελεστές ερωτημάτων περιλαμβάνουν ερωτήματα διαστήματος, ερωτήματα KNN, ερωτήματα συνάθροισης. Επιπλέον, υποστηρίζεται επίσης η εξόρυξη προτύπων ταυτόχρονης κίνησης σε δεδομένα τροχιάς, καταδεικνύοντας τις δυνατότητες ανάλυσης τροχιών του UlTraMan. Παραθέτουμε τα ερωτήματα που υποστηρίζονται από το UlTraMan:

Ερώτημα αναγνωριστικού. Το ερώτημα αναγνωριστικού είναι ένα απλό ερώτημα δεδομένων τροχιάς. Υπάρχουν τρεις τύποι αναγνωριστικών σε ένα σύνολο δεδομένων τροχιάς: το αναγνωριστικό στοιχείου (π.χ. αναγνωριστικό σημείου), το αναγνωριστικό τροχιάς και το αναγνωριστικό κινούμενου αντικειμένου. Δεδομένου ότι ένα κινούμενο αντικείμενο μπορεί να παράγει πολλές τροχιές και μια τροχιά μπορεί να περιέχει πολλά στοιχεία, τα ερωτήματα σε διαφορετικούς τύπους αναγνωριστικών είναι συνήθως διαφορετικά.

Ερώτημα διαστήματος. Στο UlTraMan, το ερώτημα διαστήματος βρίσκει δεδομένα τροχιάς σε σχέση με ένα χωρικό ή χωροχρονικό διάστημα. Το UlTraMan μεγιστοποιεί τα πλεονεκτήματα των πολυδιάστατων ευρετηρίων (π.χ. το R-tree) μέσω της υποστήριξής του για διαμερίσεις και καθολική-τοπική ευρετηρίαση. Η χρήση του STRPartitioner [48] βοηθάει στην επίτευξη της μέγιστης διαμέρισης, καθώς πακετάρει τα δεδομένα σε διαμερίσεις με τον ίδιο τρόπο που γίνεται η δημιουργία των κόμβων σε ένα R-tree. Με την εφαρμογή της τεχνικής διαμερίσεων, μπορεί να κατασκευαστεί ένα καθολικό R-tree που επιτρέπει το αποτελεσματικό συνολικό φιλτράρισμα. Επιπλέον, μπορούν να κατασκευαστούν τοπικά R-trees μέσα σε κάθε διαμέριση, έτσι ώστε να επιταχύνονται τα τοπικά ερωτήματα διαστήματος στις διαμερίσεις που προκύπτουν από το καθολικό φιλτράρισμα.

Ερώτημα knn. Στο UlTraMan, προτείνεται και υλοποιείται μια παραλλαγή R-tree που επιτρέπει αποτελεσματικά ερωτήματα kNN. Σε αυτό το R-δέντρο, κάθε κόμβος του δέντρου διατηρεί έναν αριθμό διαφορετικών τροχιών στις καλυπτόμενες διαμερίσεις του. Ως αποτέλεσμα, ένα ερώτημα τροχιάς kNN στο UlTraMan επεξεργάζεται ως εξής:

Διαμέριση (προαιρετικά). Μια καθολική διαμέριση σύμφωνα με τη χωρική μπορεί να βελτιώσει την ικανότητα καθολικής περικοπής και την αποτελεσματικότητα του ερωτήματος.

Ευρετηρίαση και Εξαγωγή. Πρώτον, τα τοπικά R-δέντρα δημιουργούνται στις διαμερίσεις των δεδομένων. Στη συνέχεια, εξάγονται τα αναγνωριστικά της τροχιάς και τα αναγνωριστικά των διαμερίσεων (tid, pid) ώστε να δημιουργηθεί ένας μεταπίνακας. Στη συνέχεια, τα πλαίσια οριοθέτησης και τα αναγνωριστικά των διαμερίσεων εξάγονται ως χαρακτηριστικά για την κατασκευή ενός καθολικού R-δέντρου. Για κάθε κόμβο δέντρου του καθολικού R-δέντρου, βρίσκονται οι διαμερίσεις που καλύπτονται και υπολογίζεται ο αριθμός των τροχιών.

Τέλος, πραγματοποιούνται δύο διαδοχικά καθολικά φιλτραρίσματα και τα τοπικά ερωτήματα knn. Τα τοπικά knn ερωτήματα πραγματοποιούνται στις τροχιές του τελικού αποτελέσματος, που προέκυψε από το 2ο καθολικό φιλτράρισμα. Τα αποτελέσματα ταξινομούνται συνολικά με βάση τις αποστάσεις τους και επιστρέφονται οι top-k τροχιές. Η υποστήριξη για το ερώτημα kNN δείχνει την ικανότητα του UlTraMan να υποστηρίζει ευρετήρια που υπάρχουν ήδη ή που προκύπτουν στην πορεία, καθώς και αλγόριθμους για την ανάλυση δεδομένων τροχιάς.

Ερώτημα συνάθροισης. Το UlTraMan υποστηρίζει αποτελεσματικές αναλύσεις συνάθροισης (π.χ. μέγιστο, ελάχιστο, αριθμός και μέσος όρος) μέσω της κατασκευής του μεταπίνακα.

Εξόρυξη προτύπων ταυτόχρονης κίνησης. Η εξόρυξη προτύπων ταυτόχρονης κίνησης συνιστά μια προηγμένη λειτουργία εξόρυξης για δεδομένα τροχιάς και έχει διερευνηθεί σε αρκετές μελέτες ([59],[60],[61],[62]). Αναλυτικά η διαδικασία εξόρυξης προτύπων ταυτόχρονης κίνησης στο UlTraMan έχει ως εξής.

Προεπεξεργασία: Μετασχηματισμός μορφής. Τα δεδομένα της τροχιάς θα πρέπει να μετατραπούν σε μια απαιτούμενη μορφή, όπως χωρικές συντεταγμένες μετρούμενες σε μέτρα (αντί για γεωγραφικό πλάτος και μήκος).

Προεπεξεργασία: Συγχρονισμός. Μετά τη μετατροπή της μορφής, συγχρονίζουμε τις τροχιές με μια καθολική ακολουθία χρονικής σήμανσης. Κατά τη διάρκεια της διαδικασίας, ο μεταπίνακας χρησιμοποιείται για να ληφθεί η συνολική χρονική περίοδος και ένας συγκεκριμένος διαμεριστής

χρησιμοποιείται για την αναδιαμέριση του συνόλου δεδομένων μέσω διαστημάτων χρονικών σημάνσεων.

Ανάλυση: Συσταδοποίηση. Για τις συν-κινήσεις, πρέπει να συσταδοποιήσουμε τα δεδομένα σε κάθε χρονική σήμανση. Στο UlTraMan, ο αλγόριθμος συσταδοποίησης (π.χ. DBSCAN [63] μπορεί να επιταχυνθεί από ένα R-tree που έχει κατασκευαστεί εκ των προτέρων.

Εξόρυξη: Μοτίβο ταυτόχρονης κίνησης. Ο υπάρχων αλγόριθμος κατανεμημένης εξόρυξης [64], συμπεριλαμβανομένης της διαμέρισης τύπου αστέρα και της απαρίθμησης εκ των προτέρων, μπορεί να εφαρμοστεί στο UlTraMan με τα API που παρέχονται από το TrajDataset και το RDD.

DITA

Στο επίπεδο αλγοριθμικής/επεξεργασίας, το DITA υιοθετεί το παράδειγμα φιλτραρίσματος και επαλήθευσης, προκειμένου να επεξεργάζεται αποτελεσματικά την αναζήτηση ομοιότητας και τις ενώσεις ομοιότητας. Η περικοπή μπορεί να επιτευχθεί με συγκεκριμένες συνθήκες που μπορούν να ελεγχθούν στα καθολικά και τοπικά ευρετήρια.

Αναζήτηση ομοιότητας τροχιάς. Στο DITA, τα ερωτήματα αναζήτησης ομοιότητας τροχιάς υποβάλλονται σε επεξεργασία σε τρία βήματα: (1) ο κύριος κόμβος, χρησιμοποιεί το καθολικό ευρετήριο για να υπολογίσει τις σχετικές διαμερίσεις που περιέχουν τροχιές παρόμοιες με το ερώτημα και στέλνει το ερώτημα στους αντίστοιχους κόμβους εργασίας αυτών των διαμερίσεων, (2) σε κάθε διαμέριση, οι κόμβοι εργασίας χρησιμοποιούν πρώτα το τοπικό ευρετήριο για να δημιουργήσουν υποψήφιες τροχιές του ερωτήματος και στη συνέχεια δημιουργούν τα τοπικά αποτελέσματα επαληθεύοντας εάν είναι πραγματικά παρόμοια με το ερώτημα και (3) ο κύριος κόμβος συλλέγει τα αποτελέσματα και τα επιστρέφει στον χρήστη.

Ένωση ομοιότητας τροχιάς. Στο επίπεδο της διαμέρισης, κατά την ένωση δύο πινάκων, το σύστημα DITA δημιουργεί αρχικά ευρετήρια για αυτούς επειδή δεν απαιτεί πολύ μεγάλο κόστος και μπορεί επίσης να χρησιμοποιηθεί ξανά για μελλοντικούς υπολογισμούς. Επομένως, σε αυτήν την ενότητα, υποθέτουμε ότι οι δύο πίνακες έχουν ήδη ευρετηριαστεί. Κατά τη διαμέριση, το DITA θα διασφαλίσει ότι το μέγεθος της διαμέρισης δεν μπορεί να υπερβαίνει το μισό της μνήμης του κόμβου εργασίας, έτσι ώστε οποιεσδήποτε δύο διαμερίσεις να μπορούν να διατηρηθούν ταυτόχρονα στη μνήμη. Στη συνέχεια πραγματοποιείται η καθολική και η τοπική ένωση των ζευγών των διαμερίσεων όπου πραγματοποιείται και η περικοπή των αποτελεσμάτων που δεν είναι χρήσιμα.

Distributed MobilityDB

Το κατανεμημένο MobilityDB υποστηρίζει πολλούς χωροχρονικούς τύπους ερωτημάτων, αλλά όχι τη γενική περίπτωση των κατανεμημένων χωροχρονικών ενώσεων (π.χ. αυτοσύνδεση σε θέσεις κινούμενων αντικειμένων) που απαιτούν ανακατανομή των δεδομένων. Ο διαχειριστής κατανομής λαμβάνει το ερώτημα του χρήστη, το περνά στον αναλυτή, εμπλουτίζει το δέντρο ανάλυσης με ετικέτες από τον κατάλογο διανομής, προσδιορίζει τους κόμβους εργασίας που πρέπει να κληθούν, παράγει το κατανεμημένο σχέδιο, παρακολουθεί την εκτέλεση και παράγει το τελικό αποτέλεσμα. Επί του παρόντος, μπορούν να κατανεμηθούν οι ακόλουθες κατηγορίες ερωτημάτων.

Ερώτημα διαστήματος. Υπολογίζει τις τροχιές που επικαλύπτουν ένα δεδομένο εύρος ερωτημάτων. Το εύρος μπορεί να είναι χωρικό, χρονικό ή χωροχρονικό.

Ερώτημα ένωσης μετάδοσης. Ενώνει έναν κατανεμημένο πίνακα με έναν ή περισσότερους πίνακες αναφοράς με βάση τη χωρική, τη χρονική ή τη χωροχρονική τομή τους.

Ερώτημα τροχιάς. Είναι ένα ερώτημα που ανακτά ένα αντικείμενο τροχιάς από το αναγνωριστικό του ή από ένα άλλο φίλτρο χαρακτηριστικών.

kNN-Query. Ανακτά τις k τροχιές που είχαν τη μικρότερη μέση απόσταση με μια δεδομένη τροχιά.

Αναλυτής Ερωτημάτων. Ο τυπικός αναλυτής ερωτημάτων της PostgreSQL αναλύει και μετατρέπει το ερώτημα του χρήστη σε αναλυτική δομή δέντρου. Ωστόσο, δεν γνωρίζει για τις πληροφορίες καταλόγου, που αποθηκεύονται, για τους κατανεμημένους πίνακες. Επομένως, εμπλουτίζεται το δέντρο ανάλυσης με ετικέτες που θα βοηθήσουν τον σχεδιαστή να δημιουργήσει το κατανεμημένο σχέδιο.

Σχεδιαστής Ερωτημάτων. Ελέγχει ποιοι κόμβοι εργασίας (δηλαδή, διαμερίσεις δεδομένων) μπορούν να συνεισφέρουν στο αποτέλεσμα. Αυτό γίνεται με την επιθεώρηση των δηλώσεων του ερωτήματος και την έκταση των διαμερίσεων των δεδομένων που είναι αποθηκευμένα στον κατάλογο. Υποστηρίζεται επίσης το χωρικό και το χρονικό φιλτράρισμα με βάση τα ορίσματα συνάρτησης, καθώς οι περισσότερες από αυτές τις συναρτήσεις έχουν διαφορετικά ορίσματα. Με άλλα λόγια, τόσο οι γεωμετρικοί τελεστές του MobilityDB όσο και οι τοπολογικές συναρτήσεις μπορούν να χρησιμοποιηθούν στο ερώτημα από τον χρήστη. Για παράδειγμα, ο τελεστής && χρησιμοποιείται για να φιλτράρει τροχιές που επικαλύπτουν ένα χωρικό, χρονικό ή χωροχρονικό αντικείμενο.

Προγραμματιστής Ερωτημάτων. Το πρόγραμμα σχεδιασμού ερωτημάτων ξεκινά από τον χρήστη SQL και το εμπλουτισμένο δέντρο ανάλυσης του. Στη συνέχεια δημιουργεί μια δήλωση SQL ανά κόμβο εργασίας. Οι κόμβοι εκτελούν παράλληλα. Ο συντονιστής συλλέγει τα αποτελέσματα των κόμβων και κατασκευάζει τα συνολικά αποτελέσματα του ερωτήματος. Για τους τύπους ερωτημάτων που υποστηρίζονται στο MobilityDB, ο σχεδιαστής πρέπει να αναλύσει και να διανείμει τόσο τους όρους SELECT όσο και τους όρους WHERE του ερωτήματος του χρήστη. Για τον όρο WHERE, ο χειρισμός ισοδυναμεί με την επανεγγραφή του ονόματος του πίνακα, στο όνομα της διαμέρισης, στον καθορισμένο κόμβο εργασίας.

Η κατανομή του όρου SELECT εξαρτάται από τη μέθοδο διαμέρισης, είτε οι μεμονωμένες τροχιές χωρίζονται σε διαμερίσεις είτε όχι. Στην περίπτωση της ιεραρχικής διαμέρισης, οι τροχιές δεν χωρίζονται. Αντιγράφονται σε όλα τα επικαλυπτόμενα τμήματα. Ο ίδιος όρος SELECT στη συνέχεια αποστέλλεται στους κόμβους εργασίας και ο ρόλος του συντονιστή είναι να αφαιρέσει τα διπλά αποτελέσματα. Για τα χρονικά αθροίσματα, μια συνάρτηση συνδυασμού προστίθεται στο δέντρο μετα-επεξεργασίας, που θα εκτελεστεί στον κόμβο συντονιστή, στα αποτελέσματα που συλλέγονται από τους επιμέρους κόμβους εργασίας.

Εκτελεστής Ερωτήματος. Ο εκτελεστής είναι η ενότητα που παρακολουθεί την εκτέλεση του ερωτήματος στους κόμβους εργασίας και την παραγωγή των τελικών αποτελεσμάτων του ερωτήματος στον συντονιστή. Όπως είπαμε προηγουμένως, το ερώτημα μπορεί να χρειαστεί να εκτελεστεί σε μία ή δύο φάσεις. Επιλέγεται ως επιλογή σχεδίασης η υλοποίηση ενός εκτελεστή ανά τύπο ερωτήματος. Δηλαδή, επί του παρόντος υπάρχουν τέσσερις εκτελεστές που αντιστοιχούν στους τέσσερις υποστηριζόμενους τύπους ερωτημάτων: διαστήματος, ένωσης μετάδοσης, τροχιάς και kNN. Αυτή η επιλογή επιτρέπει περισσότερες βελτιστοποιήσεις για συγκεκριμένα ερωτήματα. Ωστόσο, όλοι μοιράζονται κοινές λειτουργίες όπως η αποστολή ερωτημάτων σε μεμονωμένους κόμβους εργασίας, η συλλογή των αποτελεσμάτων των κόμβων και η δημιουργία του τελικού αποτελέσματος.

Summit

Στο Summit μπορούν να εκτελεστούν ορισμένα βασικα ερωτήματα, δηλαδή ερωτήματα διαστήματος, πλησιέστερου γείτονα και ομοιότητας. Άλλες χωροχρονικές λειτουργίες που αφορούν τροχιές π.χ., αντίστροφα ερωτήματα kNN, συνάθροισης και διαδρομής, μπορούν να πραγματοποιηθούν ακολουθώντας παρόμοιες προσεγγίσεις.

Ερώτημα διαστήματος για τροχιές (Trajectory Range Query): Με δεδομένη μια τρισδιάστατη δήλωση ερωτήματος, αυτό το ερώτημα ανακτά όλες τις τροχιές που επικαλύπτονται με την περιοχή του ερωτήματος τόσο στον χώρο όσο και στον χρόνο. Ανεξάρτητα από τον τύπο της διαμέρισης για την απάντηση στο ερώτημα, χρησιμοποιείται ένας αλγόριθμος που εκτελείται σε τρία βήματα και συγκεκριμένα, χρονικό φιλτράρισμα, χωρική αναζήτηση και χωροχρονική βελτίωση. Στη φάση της βελτίωσης, απαιτείται μια επιπλέον επεξεργασία για την αφαίρεση των διπλότυπων από την απάντηση του ερωτήματος, καθώς οι τροχιές ενδέχεται να αντιγραφούν μεταξύ των διαμερίσεων.

Ερώτημα πλησιέστερου γείτονα (Trajectory kNN): Το Summit υποστηρίζει τις ακόλουθες δύο παραλλαγές της λειτουργίας kNN: (1) kNN με βάση το σημείο. Με δεδομένη τη δήλωση ενός ερωτήματος που αποτελείται από το σημείο του ερωτήματος P(*x,y*) , και το χρονικό διάστημα [*t1, t2*], μπορούν να βρεθούν οι k πλησιέστερες τροχιές στο σημείο του ερωτήματος κατά τη διάρκεια του δεδομένου χρονικού διαστήματος. (2) kNN με βάση την τροχιά. Δεδομένης μιας τροχιάς Tr j που αποτελείται από μια ακολουθία χωροχρονικών σημείων, μπορούν να βρεθούν οι πλησιέστερες τροχιές σε όλα τα σημεία της τροχιάς για κάθε χρονική στιγμή σύμφωνα με κάποια αθροιστική συνάρτηση, όπως η Min ή η Max. Για να απαντηθούν τέτοιου είδους ερωτήματα, το Summit χρησιμοποιεί έναν αλγόριθμο που αποτελείται από τρεις φάσεις, δηλαδή, διαμέριση, τοπικό υπολογισμό και καθολικό υπολογισμό. Στη φάση της διαμέρισης, το Summit αποφασίζει ποια τεχνική διαμερίσεων θα χρησιμοποιηθεί. Μόλις τα δεδομένα χωριστούν, το Summit ενεργοποιεί έναν αλγόριθμο τοπικού υπολογισμού για να βρει ένα υποψήφιο σύνολο από τα επικαλυπτόμενα τμήματα. Μετά την εκτέλεση του τοπικού υπολογισμού, κάθε υπολογιστικός κόμβος στο σύμπλεγμα Summit θα έχει το δικό του σύνολο υποψηφίων. Η φάση καθολικού υπολογισμού υλοποιείται στο Summit ως συνάρτηση μείωσης, η οποία εκτελείται σε ένα μόνο μηχάνημα για τον υπολογισμό του τελικού αποτελέσματος. Σε αυτή τη φάση εφαρμόζεται διπλότυπη εξάλειψη.

Ερώτημα ομοιότητας τροχιάς (Trajectory Similarity Query): Ο στόχος αυτού του ερωτήματος είναι να βρει παρόμοιες τροχιές με μια δεδομένη, με βάση κάποια καθορισμένη συνάρτηση ομοιότητας. Αυτό είναι ένα πολύ χρήσιμο ερώτημα για πολλές εφαρμογές, όπως οι μεταφορές και τα ερωτήματα εξόρυξης προτύπων εκ των προτέρων. Το Summit περνάει από δύο φάσεις για να βρει παρόμοιες τροχιές, δηλαδή, τις φάσεις διαμέρισης και υπολογισμού. Η φάση της διαμέρισης κατηγοριοποιεί τα δεδομένα με βάση το μοντέλο τμηματοποίησης. Η φάση υπολογισμού εκτελείται σε δύο εργασίες MapReduce για τοπικούς και καθολικούς υπολογισμούς, που πραγματοποιούνται η μία πίσω από την άλλη. Η αφαίρεση διπλότυπων πραγματοποιείται στη φάση μείωσης πριν από την έναρξη του καθολικού υπολογισμού. Στο Summit, εφαρμόζεται η πιο ισχυρή και ευρέως διαδεδομένη συνάρτηση ομοιότητας, δηλαδή η Dynamic Time Warping [65], όπου εφαρμόζονται χωροχρονικά κατώφλια.

Dragoon

Στο Dragoon, οι υποστηριζόμενοι τελεστές ερωτημάτων περιλαμβάνουν ερωτήματα διαστήματος και ερωτήματα KNN, τόσο για τις περιπτώσεις εκτός σύνδεσης όσο και για τις περιπτώσεις σε σύνδεση. Επιπλέον, υποστηρίζεται επίσης η εξόρυξη προτύπων ταυτόχρονης κίνησης σε δεδομένα τροχιάς, καταδεικνύοντας τις δυνατότητες ανάλυσης τροχιών του Dragoon. Παραθέτουμε τα ερωτήματα που υποστηρίζονται από το Dragoon:

Ερωτήματα αναγνωριστικού και διαδικτυακά ερωτήματα αναγνωριστικού. Τα ερωτήματα αναγνωριστικού και τα διαδικτυακά ερωτήματα αναγνωριστικού στοχεύουν στην εύρεση μιας συγκεκριμένης τροχιάς σύμφωνα με τις πληροφορίες της ταυτότητάς της, οι οποίες είναι χρήσιμες σε περιπτώσεις παρακολούθησης της τροχιάς.

Για το ερώτημα αναγνωριστικού εκτός σύνδεσης, ο IdPartitioner διαμερίζει πρώτα ολόκληρο το σύνολο δεδομένων των στατικών τροχιών (STD) σε πολλαπλές διαμερίσεις δεδομένων σύμφωνα με τα αναγνωριστικά των κινούμενων αντικειμένων για να επιτρέψει την επακόλουθη παράλληλη επεξεργασία.

Για το διαδικτυακό ερώτημα αναγνωριστικού, ο πραγματικού χρόνου IdPartitioner χρησιμοποιείται για την αντιστοίχιση των νεοαφιχθέντων σημείων τροχιάς, τη στιγμή t, σε διάφορες διαμερίσεις δεδομένων σύμφωνα με τα αναγνωριστικά των τροχιών συνεχόμενης ροής. Στη συνέχεια τα συγχωνεύει με τις παρελθοντικές τοποθεσίες σε κάθε διαμέριση. Το διαδικτυακό ερώτημα αναγνωριστικού είναι ένα παράδειγμα της πιο πρόσφατης διαδικτυακής ανάλυσης, καθώς επιστρέφει την τρέχουσα θέση μιας τροχιάς συνεχόμενης ροής της οποίας το αναγνωριστικό ισούται με το δεδομένο αναγνωριστικό κάθε φορά. Έτσι, το διαδικτυακό ερώτημα αναγνωριστικού διεξάγεται στις τρέχουσες τοποθεσίες όλων των τροχιών συνεχόμενης ροής σε κάθε χρονικό σημείο.

Ερωτήματα διαστήματος και διαδικτυακά ερωτήματα διαστήματος. Τα ερωτήματα διαστήματος και τα διαδικτυακά ερωτήματα διαστήματος στοχεύουν στην εύρεση τροχιών σε μια συγκεκριμένη

περιοχή χωρικής εμβέλειας και είναι χρήσιμα σε εφαρμογές όπως η παρακολούθηση της κυκλοφορίας και η ανίχνευση hotspot [13].

Για το ερώτημα διαστήματος εκτός σύνδεσης, υιοθετείται ο GridPartitioner για το διαμερισμό των STD με βάση τις χωρικές πληροφορίες των τροχιών, όπου ένα τοπικό R-tree είναι σχεδιασμένο σε κάθε διαμέριση δεδομένων και ένα καθολικό ευρετήριο πλέγματος βασίζεται σε ολόκληρο το STD των δεδομένων τροχιάς. Παρόμοια με το ερώτημα αναγνωριστικού, το καθολικό ευρετήριο πλέγματος μπορεί να φιλτράρει εκείνες τις διαμερίσεις δεδομένων που δεν τέμνονται με την περιοχή αναζήτησης και, στη συνέχεια, τα τοπικά R-tree μπορούν να χρησιμοποιηθούν για την υποστήριξη ερωτημάτων διαστήματος στα υποψήφια τμήματα, ώστε να ληφθεί το τελικό αποτέλεσμα.

Για το διαδικτυακό ερώτημα διαστήματος, ο διαδικτυακός GridPartitioner χρησιμοποιείται πρώτα για να χωρίσει τα νέα εισερχόμενα σημεία από τις τροχιές συνεχόμενης ροής και στη συνέχεια να συγχωνεύσει τα εισερχόμενα σημεία με τα παρελθοντικά δεδομένα των αντίστοιχων διαμερίσεων των δεδομένων. Το διαδικτυακό ερώτημα διαστήματος είναι επίσης ένα παράδειγμα της πιο πρόσφατης διαδικτυακής ανάλυσης. Επιστρέφει τις τροχιές συνεχόμενης ροής των οποίων οι τρέχουσες θέσεις περιέχονται στην περιοχή αναζήτησης και το ερώτημα εκτελείται στις πιο πρόσφατες θέσεις όλων των τροχιών συνεχόμενης ροής. Επιπλέον, τα αντίστοιχα τοπικά και καθολικά ευρετήρια ενημερώνονται μετά τη συγχώνευση των δεδομένων.

Ερωτήματα kNN και διαδικτυακά ερωτήματα kNN. Τα ερωτήματα πλησιέστερου γείτονα kNN και τα αντίστοιχα διαδικτυακά kNN στοχεύουν να βρουν τις k πλησιέστερες τροχιές για μια καθορισμένη χωρική τοποθεσία. Τα ερωτήματα αυτά είναι χρήσιμα σε υπηρεσίες που βασίζονται σε τοποθεσία, όπως η ταξινόμηση τροχιών και η κοινή χρήση διαδρομής [66].

Για το ερώτημα kNN εκτός σύνδεσης, υιοθετείται ο STRPpartitioner [48] για ομοιόμορφη κατανομή των δεδομένων τροχιάς σύμφωνα με τις χωρικές πληροφορίες του. Εδώ, η παραλλαγή R-tree [17] χρησιμοποιείται για τη βελτίωση της αποδοτικότητας του ερωτήματος kNN, όπου κάθε κόμβος του R-tree διατηρεί τόσο το ελάχιστο οριοθετημένο ορθογώνιο (MBR) όσο και έναν αριθμό διακριτών τροχιών που περιέχονται στο MBR του. Κατά τη διάρκεια του ερωτήματος kNN, το καθολικό φιλτράρισμα χρησιμοποιείται για τη λήψη υποψήφιων διαμερίσεων των δεδομένων με μετρήσεις τροχιών όχι μικρότερες από k και, στη συνέχεια, τα τοπικά ερωτήματα kNN εκτελούνται στα υποψήφια τμήματα, ξεχωριστά. Τέλος, τα τοπικά αποτελέσματα από τις υποψήφιες διαμερίσεις ταξινομούνται με αύξουσα σειρά των αποστάσεων τους, στη θέση του ερωτήματος και, στη συνέχεια, επιστρέφονται οι καθολικές τροχιές top-k.

Για το διαδικτυακό ερώτημα kNN, χρησιμοποιούνται τα MBR στο καθολικό ευρετήριο του R-tree αντί για ένα STRPpartitioner, για να χωριστούν οι πιο πρόσφατες τοποθεσίες από τις τροχιές συνεχόμενης ροής. Αυτό συμβαίνει επειδή ο STRPpartitioner θα μπορούσε να προκαλέσει πρόσθετο κόστος για τη συγχώνευση των δεδομένων. Στη συνέχεια, συγχωνεύονται τα δεδομένα και ενημερώνονται τα τοπικά και καθολικά ευρετήρια. Παρόμοια με τα διαδικτυακά ερωτήματα ταυτότητας και διαστήματος, το διαδικτυακό ερώτημα kNN είναι ένα παράδειγμα της πιο πρόσφατης διαδικτυακής ανάλυσης. Το διαδικτυακό ερώτημα kNN επιστρέφει τις k τροχιές των οποίων οι τρέχουσες θέσεις είναι οι πλησιέστερες στη θέση του ερωτήματος κάθε φορά. Έτσι, εκτελείται στις πιο πρόσφατες τοποθεσίες όλων των τροχιών συνεχόμενης ροής.

Εξόρυξη προτύπων ταυτόχρονης κίνησης. Η εξόρυξη προτύπων ταυτόχρονης κίνησης στοχεύει στην ανακάλυψη συν-κινούμενων αντικειμένων που ικανοποιούν συγκεκριμένους χωροχρονικούς περιορισμούς [67], συμπεριλαμβανομένης της εγγύτητας, της σημαντικότητας, της διάρκειας, της διαδοχικότητας και της σύνδεσης. Εδώ, το Dragoon εστιάζει στην ανίχνευση σε πραγματικό χρόνο των προτύπων ταυτόχρονης κίνησης σε τροχιές συνεχόμενης ροής, που είναι χρήσιμο σε πολλές εφαρμογές, όπως η πρόβλεψη μελλοντικών κινήσεων. Η ανίχνευση προτύπων ταυτόχρονης κίνησης σε πραγματικό χρόνο είναι ένα παράδειγμα περιοδικής διαδικτυακής ανάλυσης και, επομένως, πρέπει να ληφθούν υπόψη οι πιο πρόσφατες τοποθεσίες καθώς και οι παρελθοντικές τοποθεσίες κάθε τροχιάς συνεχόμενης ροής.

TrajSpark

Τα τυπικά ερωτήματα τροχιάς περιλαμβάνουν το ερώτημα που βασίζεται σε ένα αντικείμενο ([68],[44],[23]), το ερώτημα που βασίζεται σε χωροχρονικό διάστημα ([68],[69],[20]) και το ερώτημα που βασίζεται στον k πλησιέστερο γείτονα ([68],[70],[20]).

Ένα ερώτημα απλού αντικειμένου ανακτά την τροχιά ενός δεδομένου κινούμενου αντικειμένου λαμβάνοντας δύο παραμέτρους: το αναγνωριστικό του κινούμενου αντικειμένου το χρονικό περιορισμό. Το ερώτημα αυτό αποτελεί και ένα είδος φιλτραρίσματος, καθώς χρησιμοποιεί το τοπικό ευρετήριο.

Ερώτημα που βασίζεται σε χωροχρονικό διάστημα. Ένα ερώτημα που βασίζεται σε χωροχρονικό διάστημα ανακτά τις τροχιές μέσα σε ένα χωροχρονικό διάστημα. Λαμβάνει δύο παραμέτρους: το χρονικό και το χωρικό. Χρησιμοποιώντας τα ευρετήρια, το TrajSpark μπορεί επίσης να επιτύχει αποδοτικότερο φιλτράρισμα των διαμερίσεων σε σχέση με το Spark. Αρχικά, το TrajSpark διασχίζει το καθολικό ευρετήριο για να φιλτράρει τα διαμερίσματα που τέμνονται με το δεδομένο χωροχρονικό διάστημα. Στη συνέχεια, για κάθε διαμέριση, το TrajSpark φιλτράρει υποψηφίους των οποίων το χωρικό πλαίσιο οριοθέτησης και το χρονικό εύρος τέμνονται με τον δεδομένο χωροχρονικό περιορισμό. Επιπλέον, βρίσκει μια υποτροχιά που οριοθετείται από τον χωροχρονικό περιορισμό για κάθε υποψήφιο.

Ερώτημα που βασίζεται στον k πλησιέστερο γείτονα. Υπάρχουν πολλές παραλλαγές του ερωτήματος που βασίζεται στο kNN και εστιάζουμε στην εύρεση των top-k τροχιών που μοιάζουν περισσότερο με την τροχιά της αναφοράς. Το TrajSpark λαμβάνει πρώτα το MBR και το χρονικό διάστημα της τροχιάς. Αυτό συμβαίνει επειδή τα υποψήφια αποτελέσματα είναι χωροχρονικά κοντά στην αναφορά, επομένως η χρήση των MBR και του χρονικού περιορισμού διευκολύνει την περικοπή του υποψήφιου. Στη συνέχεια, το TrajSpark φιλτράρει τις υποψήφιες διαμερίσεις χρησιμοποιώντας το καθολικό ευρετήριο. Μετά από αυτό, οι υποτροχιές περικόπτονται περαιτέρω και συγχωνεύονται σε πλήρεις τροχιές. Αυτές οι τροχιές είναι οι υποψήφιες του τελικού αποτελέσματος.

JUST-Traj

Το JUST-Traj περιλαμβάνει τα εξής ερωτήματα για τη διαχείριση των δεδομένων:

Χρονικό ερώτημα αναγνωριστικού (ID). Ανακτά τις τροχιές με ένα αναγνωριστικό αντικειμένου και ένα χρονικό διάστημα, το οποίο θα μπορούσε να βοηθήσει τους διαχειριστές να γνωρίζουν τη λεπτομερή τροχιά ενός συγκεκριμένου οχήματος σε ένα δεδομένο χρονικό διάστημα.

Ερώτημα χωρικού διαστήματος. Βρίσκει τροχιές από τη σχέση τους με ένα δεδομένο χωρικό διάστημα.

Χωροχρονικό ερώτημα. Αναζητά τροχιές σε ένα χωροχρονικό διάστημα.

Άλλα ερωτήματα. Το JUST-Traj υποστηρίζει μια ποικιλία ειδικών ερωτημάτων για τροχιές, π.χ., το ερώτημα ομοιότητας βρίσκει τροχιές παρόμοιες με μια δεδομένη τροχιά και το ερώτημα kNN βρίσκει παρόμοιες τροχιές top-k.

VIPTRA

Το σύστημα VIPTRA, για εύκολη οπτική ανάλυση, προσφέρει δύο βασικούς τύπους ερωτημάτων στα δεδομένα τροχιάς, δηλαδή το ερώτημα διαστήματος και το ερώτημα k-πλησιέστερου γείτονα (kNN).

Το ερώτημα διαστήματος επιστρέφει τις τροχιές που τέμνονται με την καθορισμένη χωροχρονική περιοχή. Το VIPTRA ενισχύει το ερώτημα διαστήματος περικόπτοντας πρώτα τις διαμερίσεις με το καθολικό ευρετήριο και, στη συνέχεια, δημιουργώντας ερωτήματα στα κατάλληλα τοπικά ευρετήρια. Οι χρήστες μπορούν να δημιουργήσουν ένα ερώτημα διαστήματος για να οπτικοποιήσουν τις τροχιές που περιέχονται σε μια καθορισμένη χωροχρονική περιοχή.

Το ερώτημα kNN βρίσκει τις k πλησιέστερες τροχιές σε ένα χωροχρονικό σημείο του ερωτήματος. Το VIPTRA καταγράφει τον αριθμό των τροχιών εντός των διαμερίσεων στο καθολικό ευρετήριο, βάσει του οποίου ο αλγόριθμος των ερωτημάτων kNN υιοθετεί ένα πλαίσιο δύο φάσεων. Πρώτα πραγματοποιεί τοπικά ερωτήματα στις πλησιέστερες διαμερίσεις που περιέχουν περισσότερες από k τροχιές συνολικά. Τα αποτελέσματα που λαμβάνονται μέσω της πρώτης φάσης χρησιμοποιούνται

για να ληφθεί το ανώτερο όριο της απόστασης για την περικοπή. Στη δεύτερη φάση, εκτελούνται τοπικά ερωτήματα στις υπόλοιπες διαμερίσεις για να βρεθεί το τελικό αποτέλεσμα.

PRADASE

Στο σύστημα PRADASE μπορεί να διενεργηθεί η εκτέλεση ερωτημάτων διαστήματος και ερωτημάτων που βασίζονται στην τροχιά, τα αποτελέσματα των οποίων επιταχύνονται με τη χρήση των ευρετηρίων PMI και OII, όπως παρουσιάζεται παρακάτω.

Για ερωτήματα διαστήματος, το ευρετήριο PMI μπορεί να επιταχύνει την υποβολή των ερωτημάτων. Στη φάση της αντιστοίχισης, κάθε κόμβος δεδομένων υιοθετεί τον αναλυτή ερωτημάτων για να δημιουργήσει μια λίστα υποψήφιων διαμερίσεων που αποτελείται από όλες τις πιθανές διαμερίσεις που είναι αποθηκευμένες σε τοπικά υποερωτήματα. Κάθε κόμβος δεδομένων διαβάζει τα δεδομένα και εκτελεί δευτερεύοντα ερωτήματα, ενώ στη συνέχεια εξάγει τα αποτελέσματα για κάθε δευτερεύον ερώτημα με τη μορφή <*objectID, {segment*}>.

Μετά την ομαδοποίηση ανά αναγνωριστικό αντικειμένου, η φάση της μείωσης συλλέγει τα υποαποτελέσματα με το ίδιο αναγνωριστικό αντικειμένου, στη συνέχεια εκτελεί τη λειτουργία ένωσης για όλα τα υποτμήματα κάθε αντικειμένου και τέλος επιστρέφει τα αποτελέσματα <*objectID, {trajectory segments*}>. Το ευρετήριο OII μπορεί να επιταχύνει τα ερωτήματα που βασίζονται σε τροχιά. Με δεδομένο οποιοδήποτε αναγνωριστικό αντικειμένου, το σύστημα μπορεί να εντοπίσει τη διεύθυνση όπου αποθηκεύονται τα δεδομένα και να διαβάσει όλες τις ιστορικές τροχιές αυτού του αντικειμένου.

5. Ταξινόμηση Συστημάτων

Σε αυτή την ενότητα παραθέτουμε μια ταξινόμηση των παραπάνω συστημάτων, σε πίνακες, ανάλογα με ορισμένα χαρακτηριστικά που θεωρούμε ότι είναι άξια αναφοράς. Ταυτόχρονα, μέσα από τους πίνακες, πραγματοποιείται μια περιληπτική περιγραφή και σύγκριση των δυνατοτήτων των παραπάνω συστημάτων.

Περιληπτική Περιγραφή Των Συστημάτων Μέσω Πινάκων

Στον πίνακα 1, τα χαρακτηριστικά που χρησιμοποιούνται για την ταξινόμηση είναι τα εξής:

i. το σύστημα επεξεργασίας δεδομένων πάνω στο οποίο βασίζεται το πρωτότυπο,
ii. ο τύπος δεδομένων για τον οποίο έχει αναπτυχθεί το κάθε πρωτότυπο,
iii. το είδος των δεδομένων που μπορεί να επεξεργαστεί το πρωτότυπο, δηλαδή αν σετ δεδομένων που χρησιμοποιείται αφορά ιστορικά δεδομένα ή δεδομένα πραγματικού χρόνου
iv. η μνήμη που χρησιμοποιεί το κάθε πρωτότυπο για να εκτελέσει την επεξεργασία των δεδομένων
v. η γλώσσα πάνω στην οποία μπορεί να γίνει η διατύπωση των ερωτημάτων

Πίνακας 1. Συγκριτική επισκόπηση των πρωτοτύπων ως προς τα χαρακτηριστικά τους

Πρωτότυπο	Σύστημα	Τύπος δεδομένων	Είδος δεδομένων	Μνήμη	Γλώσσα ερωτημάτων
CloST	Hadoop	χωροχρονικά	historic	δίσκος	Hadoop API
SharkDB	-	τροχιάς	real-time	εσωτερική μνήμη	-
TrajStore	-	τροχιάς	real-time	δίσκος	-
TrajMesa	Spark	τροχιάς	historic	δίσκος	SQL
GCOTraj	-	τροχιάς	historic	δίσκος	-
HadoopTrajectory	Hadoop	τροχιάς	historic	δίσκος	-
UltraMan	Spark	τροχιάς	historic	εσωτερική μνήμη/δίσκος	Spark SQL
DITA	Spark	τροχιάς	real-time	εσωτερική μνήμη	SQL
MobilityDB	PostgreSQL, PostGIS	τροχιάς	historic	δίσκος	SQL
Summit	Hadoop	τροχιάς	historic	δίσκος	SQL (pigeon)
Dragoon	Spark	τροχιάς	real-time/historic	εσωτερική μνήμη	SQL
TrajSpark	Spark	τροχιάς	real-time	εσωτερική μνήμη	-
JUST-Traj	Spark, Hbase	τροχιάς	real-time	δίσκος	SQL
CloudTP	Spark	τροχιάς	historic	εσωτερική μνήμη	-
VIPTRA	-	τροχιάς	historic	εσωτερική μνήμη	-
PRADASE	Hadoop	τροχιάς	historic	δίσκος	-

Τα περισσότερα πρωτότυπα που αναλύθηκαν στην παρούσα εργασία χρησιμοποιούν δεδομένα τροχιάς. Μόνο το πρωτότυπο CloST χρησιμοποιεί χωροχρονικά δεδομένα. Παρατηρούμε ότι δύο είναι τα κύρια συστήματα πάνω στα οποία βασίζονται τα πρωτότυπα που περιγράψαμε, το Hadoop και το Spark. Επίσης, το MobilityDB χρησιμοποιεί το PostgreSQL και το JUST-Traj χρησιμοποιεί, πέρα από το Spark, το HBase.

Τα περισσότερα πρωτότυπα χρησιμοποιούν το δίσκο του υπολογιστή, πάνω στον οποίο είναι εγκατεστημένα, για την επεξεργασία των δεδομένων, με αποτέλεσμα να περιορίζεται η απόδοσή τους. Ωστόσο υπάρχουν και συστήματα που επεξεργάζονται τα σετ δεδομένων στην εσωτερική μνήμη, δίνοντας έτσι περισσότερες δυνατότητες σε αυτά τα συστήματα. Το πρωτότυπο UltraMan βλέπουμε ότι χρησιμοποιεί τόσο τον δίσκο όσο και την εσωτερική μνήμη.

Τέλος, η γλώσσα SQL είναι κυρίαρχη, σαν γλώσσα για τη σύνταξη των ερωτημάτων, στα περισσότερα πρωτότυπα, με το CloST να χρησιμοποιεί ένα Haddop API. Υπήρχαν όμως και συστήματα στα οποία δεν αναφερόταν ποια γλώσσα χρησιμοποιούν για την επεξεργασία των ερωτημάτων.

Στον πίνακα 2, τα χαρακτηριστικά που χρησιμοποιούνται είναι:

i. η ικανότητα που έχει το πρωτότυπο να αντιμετωπίζει την ασυμμετρία που παρουσιάζουν τα δεδομένα (π.χ. όταν ένας μεγάλος αριθμός δεδομένων είναι συγκεντρωμένος σε μία περιοχή σε σχέση με τις υπόλοιπες),
ii. η προσαρμοστικότητα που εμφανίζει το κάθε πρωτότυπο, δηλαδή η ικανότητα που έχει το σύστημα να προσαρμόζεται στις διαφορετικές απαιτήσεις των δεδομένων τροχιάς, όπως για παράδειγμα το GCOTraj το οποίο προσαρμόζεται σε ερωτήματα που είναι επιλεκτικά ως προς το χρόνο ή/και το χώρο,
iii. η επεκτασιμότητα που μπορεί να έχει το πρωτότυπο, δηλαδή η ικανότητά του να αυξάνει ή να μειώνει την απόδοση και το κόστος ως απόκριση στις απαιτήσεις της κάθε εφαρμογής και επεξεργασίας,
iv. η δυνατότητα του πρωτοτύπου να επεξεργάζεται δεδομένα που ανανεώνονται σε τακτά χρονικά διαστήματα,
v. η δυνατότητα που έχει το πρωτότυπο να πραγματοποιεί παράλληλη επεξεργασία των δεδομένων σε πολλά τερματικά,
vi. η δυνατότητα που έχει το πρωτότυπο να προχωρήσει σε κατανομή του σετ των δεδομένων, δηλαδή αν χρησιμοποιεί κάποιο σύμπλεγμα τερματικών.

Πίνακας 2. Συγκριτική επισκόπηση των πρωτοτύπων ως προς τα χαρακτηριστικά τους

Πρωτότυπο	Ασυμμετρία δεδομένων	Προσαρμοστικότητα	Επεκτασιμότητα	Ανανέωση δεδομένων	Παράλληλη επεξεργασία	Κατανεμημένο
CloST	-	-	✓	-	✓	✓
SharkDB	-	-	✓	✓	✓	✓
TrajStore	-	✓	✓	✓	x	x
TrajMesa	✓	-	✓	x	✓	✓
GCOTraj	✓	✓	✓	x	x	x
HadoopTrajectory	-	-	✓	✓	✓	✓
UltraMan	-	-	✓	x	✓	✓
DITA	-	-	✓	✓	✓	✓
MobilityDB	✓	✓	✓	x	✓	✓
Summit	-	-	✓	-	-	✓
Dragoon	✓	✓	✓	✓	✓	✓
TrajSpark	-	✓	✓	✓	-	✓
JUST-Traj	✓	-	✓	✓	-	✓
CloudTP	-	-	✓	-	✓	✓
VIPTRA	-	-	✓	x	✓	✓
PRADASE	-	✓	✓	✓	✓	✓

Παρατηρούμε ότι λίγα συστήματα έχουν την ικανότητα να αντιμετωπίσουν την ασυμμετρία που προέρχεται από τα μεγάλα σε των δεδομένων τροχιάς, καθώς αποτελεί μια πολύπλοκη διαδικασία. Επίσης, ως προς την προσαρμοστικότητα, πάλι είναι λίγα τα συστήματα που υποστηρίζουν αυτή τη δυνατότητα. Αντιθέτως, όλα τα συστήματα που περιγράψαμε, υποστηρίζουν τη δυνατότητα της επεκτασιμότητας, με αποτέλεσμα να μπορούν να αυξομειώνουν την απόδοσή τους ανάλογα με τις απαιτήσεις της επεξεργασίας.

Επιπλέον, βλέπουμε ότι τα μισά, τουλάχιστον (για ορισμένα πρωτότυπα δεν αναφέρεται με σαφήνεια), πρωτότυπα έχουν τη δυνατότητα να επεξεργάζονται δεδομένα που ανανεώνονται διαρκώς, σε αντίθεση με τα υπόλοιπα που υποστηρίζουν μόνο την επεξεργασία στατικών δεδομένων.

Ως προς το χαρακτηριστικό της παράλληλης επεξεργασίας, παρατηρούμε ότι τα περισσότερα πρωτότυπα, έχουν την παραπάνω δυνατότητα. Επομένως μπορούν να πραγματοποιούν ταυτόχρονα υπολογισμούς στους διάφορους κόμβους πάνω στους οποίους λειτουργούν.

Τέλος, γίνεται εμφανές ότι η πλειοψηφία των πρωτοτύπων που χρησιμοποιήθηκαν στην εργασία είναι κατανεμημένα, εκτός από το TrajStore και το GCOTraj. Σαν αποτέλεσμα, χρησιμοποιούν συμπλέγματα που διαθέτουν έναν αριθμό κόμβων εργασίας για τη διανομή και επεξεργασία των δεδομένων.

Πίνακας 3. Συγκριτική επισκόπηση των πρωτοτύπων ως προς τα χαρακτηριστικά τους

Πρωτότυπο	Τρόπος αποθήκευσης	Συμπίεση	Περικοπή/Φιλτράρισμα
CloST	Αρχείο μπλοκ όπου οι εγγραφές αποθηκεύονται σε στήλες	δύο φορές, πρώτα σε επίπεδο στήλης και στη συνέχεια σε επίπεδο τμήματος, κωδικοποίηση δέλτα, κωδικοποίηση μήκους λειτουργίας	φιλτράρισμα σε χωρικό και χρονικό διάστημα
SharkDB	σε στήλη	κωδικοποίηση δελτα	φιλτράρισμα σε χωρικό και χρονικό διάστημα, μέσα στα πλαίσια
TrajStore	σε πλέγμα	2 σχήματα συμπίεσης, συμπίεση δελτα και σχήμα συμπίεσης με απώλειες	-
TrajMesa	Αποθήκευση βασισμένη σε ένα κάθετο και ένα οριζόντιο σχήμα	Gzip	3 τύποι περικοπών, MBR, SIG_LB, SEP_LB
GCOTraj	αποθήκευση μέσω 3 διαφορετικών προσεγγίσεων	-	φιλτράρισμα κατά τη διάρκεια της επεξεργασίας των ερωτημάτων
HadoopTrajectory	HDFS	-	φιλτράρισμα μέσω ενός 3DR-tree που χρησιμοποιείται σαν ευρετήριο
UltraMan	Αποθήκευση βασισμένη σε κλειδιά-τιμές, μέσω του ChronicleMap	-	φιλτράρισμα μέσω τελεστών
DITA	-	συμπίεση βάσει κελιών	περικοπή μέσω των καθολικών και τοπικών ευρετηρίων
MobilityDB	Αποθήκευση σε κατάλογο μεταδεδομένων,μετά τη διαδικασία της διαμέρισης	συμπίεση στο επίπεδο της αποθήκευσης	φιλτράρισμα μέσω του καταλόγου
Summit	HDFS	-	χρονικό φιλτράρισμα, μέσω των ερωτημάτων διαστήματος
Dragoon	Υβριδική αποθήκευση μέσω ενός μοντέλου RDD	-	φιλτράρισμα μέσω των ευρετηρίων
TrajSpark	IndexTRDD	κωδικοποίηση δελτα , κωδικοποίηση σταθερού μήκους bit, gzip	φιλτράρισμα μέσω των ευρετηρίων
JUST-Traj	NoSQL βάση δεδομένων με ζεύγη κλειδιών τιμών	συμπίεση της τιμής κάθε τροχιάς σε μια στήλη	φιλτράρισμα θορύβου των μη κανονικών αρχείων καταγραφής
CloudTP	Κάθετη αποθήκευση στο Azure cloud	-	φιλτράρισμα σημείων θορύβου
VIPTRA	Αποθήκευση των δεδομένων στη μνήμη και στο HDFS	-	φιλτράρισμα
PRADASE	Αποθήκευση τύπου GFS	-	-

Στον πίνακα 3, τα χαρακτηριστικά που χρησιμοποιούνται για την ταξινόμηση είναι:

i. ο τρόπος αποθήκευσης των δεδομένων που εξυπηρετεί το κάθε πρωτότυπο,
ii. η συμπίεση που πραγματοποιεί το πρωτότυπο στα δεδομένα για τον περιορισμό του μεγέθους τους και τη βελτίωση της εμφάνισης των ερωτημάτων και τέλος
iii. η δυνατότητα που έχει το πρωτότυπο να πραγματοποιεί περικοπή ή φιλτράρισμα των δεδομένων για την καλύτερη επεξεργασία τους.

Είναι εμφανές ότι υπάρχουν πολλοί διαφορετικοί τρόποι αποθήκευσης των δεδομένων και κάθε πρωτότυπο χρησιμοποιεί εκείνον που το εξυπηρετεί καλύτερα για την μετέπειτα επεξεργασία των δεδομένων τροχιάς. Ορισμένα συστήματα χρησιμοποιούν το HDFS για την αποθήκευση των δεδομένων ενώ το CloudTP πραγματοποιεί αποθήκευση των σετ δεδομένων στο cloud.

Από την άλλη πλευρά, δεν πραγματοποιούν όλα τα πρωτότυπα συμπίεση των δεδομένων. Σε εκείνα τα πρωτότυπα στα οποία υπάρχει η δυνατότητα συμπίεσης, παρατηρούμε ότι μία τεχνική που χρησιμοποιείται αρκετά είναι η κωδικοποίηση δέλτα.

Τέλος, τα περισσότερα πρωτότυπα πραγματοποιούν κάποιο φιλτράρισμα ή περικοπή των δεδομένων πριν την επεξεργασία τους ή κατά τη διάρκεια αυτής, με συνέπεια να περιορίζεται ο χρόνος επεξεργασίας τους και να βελτιώνεται η απόδοση των αποτελεσμάτων.

Πίνακας 4. Συγκριτική επισκόπηση ως προς τον τρόπο ευρετηρίασης

Πρωτότυπο	**Είδη Ευρετηρίων**
CloST	καθολικό
SharkDB	-
TrajStore	τοπικό
TrajMesa	-
GCOTraj	-
HadoopTrajectory	καθολικό
UltraMan	τοπικό/καθολικό
DITA	τοπικό/καθολικό
MobilityDB	τοπικό
Summit	τοπικό/καθολικό
Dragoon	τοπικό/καθολικό
TrajSpark	τοπικό/καθολικό
JUST-Traj	-
CloudTP	-
VIPTRA	τοπικό/καθολικό
PRADASE	τοπικό (μόνο σε περίπτωση επεκτάσεων του συστήματος)

Στον πίνακα 4, πραγματοποιείται η ταξινόμηση των πρωτοτύπων σύμφωνα με το είδος των ευρετηρίων που χρησιμοποιούν. Παρατηρούμε ότι τα είδη των ευρετηρίων που χρησιμοποιούνται χωρίζονται σε τοπικά και καθολικά, έννοιες οι οποίες αναπτύχθηκαν στο κεφάλαιο 4, στην ενότητα της ευρετηρίασης. Αρκετά συστήματα χρησιμοποιούν και τα δύο είδη ευρετηρίων για την περαιτέρω

επεξεργασία των δεδομένων, ενώ υπάρχουν συστήματα στα οποία δεν υπήρχε σαφής αναφορά για το αν η ευρετηρίαση γίνεται σε τοπικό ή καθολικό επίπεδο.

Πίνακας 5. Επισκόπηση των τεχνικών επεξεργασίας των πρωτοτύπων

Πρωτότυπο	Διαμέριση	Τρόπος Ευρετηρίασης	Είδη ερωτημάτων
CloST	Ιεραρχική διαμέριση	Ιεραρχικό ευρετήριο: στο επίπεδο 1 συνδυασμός αναγνωριστικού αντικειμένου και χρόνου, στο επίπεδο 2 χωρικό quadtree, στο επίπεδο 3, 1-D χρονικό ευρετήριο	ερωτήματα διαστήματος
SharkDB	Διαμέριση σε πλαίσια στα οποία οι θέσεις όλων των κινούμενων αντικειμένων ταυτόχρονα αποθηκεύονται μαζί	grid/quadtree, δομή πλαισίου	ερωτήματα διαστήματος, knn
TrajStore	πλέγμα	quadtree	ερωτήματα διαστήματος
TrajMesa	τμηματοποίηση των τροχιών σε πολλά τμήματα	δύο είδη ευρετηρίων, ένα χρονικό ευρετήριο και ένα χωρικό ευρετήριο με βάση τις μεθόδους XZT και XZ2, καμπύλες Z-order	ερωτήματα διαστήματος, knn, ερωτήματα ομοιότητας
GCOTraj	3 μέθοδοι διαμέρισης μέσω των προσεγγίσεων (GCOTrajSP, 2-sliced GCOTraj, 3D GCOTraj)	Ευρετηρίαση βάσει της διαμέρισης των δεδομένων και βάσει της διαμέρισης του χώρου/ grid	ερωτήματα διαστήματος
HadoopTrajectory	ομαδοποίηση βάσει MBR και διαμέριση	3DR-tree/3D grid	ερωτήματα διαστήματος
UltraMan	τελεστές αναδιαμέρισης που εκτελούν διαφορετικές τεχνικές διαμέρισης(π.χ. STR)	RDD τυχαίας προσπέλασης στην εσωτερική μνήμη, ευρετήρια με κλειδιά-τιμές, R-tree	ερωτήματα διαστήματος, knn, άθροισης, ερωτήματα προτύπων ταυτόχρονης κίνησης
DITA	Χρήση του αλγορίθμου STR και ομαδοποίηση των τροχιών με βάση το πρώτο και το τελευταίο σημείο	χρήση σημείων pivot, 2 R-tree σε καθολικό επίπεδο και σε τοπικό επίπεδο, ένα ευρετήριο σε επιλεγμένα σημεία	Αναζήτηση ομοιότητας, ένωση ομοιότητας
MobilityDB	Ιεραρχική και πολυδιάστατη διαμέριση	R-tree/GiST/SP-GiST	ερωτήματα διαστήματος, knn
Summit	ιεραρχική διαμέριση	ευρετηρίαση δύο επιπέδων, χρονική και χωρική	ερωτήματα διαστήματος, knn, ερωτήματα ομοιότητας
Dragoon	διαμέριση εντός και εκτός σύνδεσης μέσω τελεστών (IdPartitioner, STRPartitioner, GridPartitioner, TimePartitioner)	R-tree/grid	ερωτήματα διαστήματος, knn, ερωτήματα προτύπων ταυτόχρονης κίνησης
TrajSpark	Διαμέριση μέσω του τελεστή STPartitioner	κατανεμημένη πολυεπίπεδη υβριδική δομή ευρετηρίου, τρία επίπεδα, όπου το πρώτο επίπεδο είναι το χρονικό ευρετήριο, το δεύτερο επίπεδο είναι το χωρικό ευρετήριο και το τελευταίο ευρετήριο είναι ένα παραδοσιακό B+δέντρο. quadtree/k-d tree	ερωτήματα διαστήματος, knn
JUST-Traj	τμηματοποίηση των τροχιών σε πολλά τμήματα	3 χωροχρονικά ευρετήρια, τα XZ2+, XZT, XZ2+T	ερωτήματα διαστήματος, knn, ερωτήματα ομοιότητας
CloudTP	τμηματοποίηση των τροχιών σε μικρότερες	ευρετήριο χρονικού αναγνωριστικού και χωροχρονικό ευρετήριο	ερωτήματα διαστήματος
VIPTRA	διαμέριση μέσω της δομής TrajDataset του UltraMan	ευρετηρίαση δύο επιπέδων	ερωτήματα διαστήματος και knn
PRADASE	2 στρατηγικές διαμέρισης (βάσει της χωρικής έκτασης και βάσει του αναγνωριστικού αντικειμένου)	2 ευρετήρια (ευρετήριο πολλαπλών επιπέδων βάσει διαμέρισης και ανεστραμμένο ευρετήριο αντικειμένου)	ερωτήματα χωροχρονικού διαστήματος και τροχιών

Στον πίνακα 5, πραγματοποιείται μια ταξινόμηση των πρωτοτύπων σύμφωνα με τις τεχνικές επεξεργασίας που αναπτύσσονται στο καθένα.

Παρατηρούμε ότι ως προς τη διαμέριση, οι τεχνικές που χρησιμοποιούνται από τα διάφορα πρωτότυπα ποικίλουν. Ορισμένα χρησιμοποιούν τελεστές διαμέρισης, όπως το UltraMan, το Dragoon και το TrajSpark. Άλλα πρωτότυπα χρησιμοποιούν την ιεραρχική διαμέριση, όπως το CloST και το MobilityDB.

Ως προς τον τρόπο δημιουργίας των ευρετηρίων, είναι εμφανές ότι τα περισσότερα συστήματα χρησιμοποιούν πλέγματα ή δέντρα για την ευρετηρίαση των δεδομένων.

Τέλος, όσον αφορά τα είδη των ερωτημάτων, τα περισσότερα πρωτότυπα περιλαμβάνουν ερωτήματα διαστήματος και πλησιέστερων γειτόνων (k-nn). Συστήματα όπως το TrajMesa, το DITA και το JUST-Traj περιλαμβάνουν ερωτήματα ομοιότητας, ενώ το Dragoon και το UltraMan υποστηρίζουν ερωτήματα προτύπων ταυτόχρονης κίνησης.

6. Συμπεράσματα και μελλοντική έρευνα

Σε αυτή την εργασία έγινε μια προσπάθεια συγκέντρωσης ενός ικανοποιητικού αριθμού εργαλείων διαχείρισης και επεξεργασίας μεγάλων δεδομένων τροχιάς. Πραγματοποιήθηκε μια παράθεση των συστημάτων, με ανάλυση του τρόπου λειτουργίας τους, του τρόπου με τον οποίο αποθηκεύουν τα δεδομένα, καθώς και των τεχνικών που χρησιμοποιούν για την επεξεργασία των δεδομένων. Παράλληλα με την ανάλυση των παραπάνω συστημάτων, πραγματοποιήθηκαν συγκρίσεις των συστημάτων με βάση τις αναφορές της βιβλιογραφίας, τόσο ως προς τα χαρακτηριστικά τους όσο και ως προς τις τεχνικές που χρησιμοποιούν για την επεξεργασία των δεδομένων.

Πέρα όμως από τη σύγκριση μέσω της βιβλιογραφίας, χρειάζεται να υπάρξει, μελλοντικά, μια σύγκριση των παραπάνω εργαλείων με τη χρήση μεγάλων σετ δεδομένων, ώστε να μελετηθεί και στην πράξη η δυναμική του κάθε συστήματος ξεχωριστά και, ταυτόχρονα, σε σχέση με άλλα παρόμοια συστήματα.

Βιβλιογραφία

[1] X. Xie, B. Mei, J. Chen, X. Du, and C. S. Jensen, “Elite: an elastic infrastructure for big spatiotemporal trajectories,” *The VLDB Journal*, vol. 25, no. 4, pp. 473–493, Aug. 2016, doi: 10.1007/s00778-016-0425-6.

[2] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, “Urban Computing: Concepts, Methodologies, and Applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, Art. no. 3, Oct. 2014, doi: 10.1145/2629592.

[3] Y. Wang, Y. Zheng, and Y. Xue, “Travel time estimation of a path using sparse trajectories,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York New York USA, Aug. 2014, pp. 25–34. doi: 10.1145/2623330.2623656.

[4] Z. Li *et al.*, “MoveMine: Mining moving object data for discovery of animal movement patterns,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 4, Art. no. 4, Jul. 2011, doi: 10.1145/1989734.1989741.

[5] J. Gudmundsson, P. Laube, and T. Wolle, “Computational Movement Analysis,” in *Springer Handbook of Geographic Information*, W. Kresse and D. M. Danko, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 423–438. doi: 10.1007/978-3-540-72680-7_22.

[6] H. Inc, “Apache Hadoop Basics.” 2013.

[7] “Apache Spark™ - Unified Engine for large-scale data analytics.” https://spark.apache.org/ (accessed Oct. 09, 2022).

[8] “Apache Storm.” https://storm.apache.org/ (accessed Oct. 09, 2022).

[9] “Apache Flink: Stateful Computations over Data Streams.” https://flink.apache.org/ (accessed Oct. 09, 2022).

[10] “Apache Kafka,” *Apache Kafka*. https://kafka.apache.org/ (accessed Oct. 09, 2022).

[11] M. M. DATA, “National Association of City Transportation Officials.” 2019. [Online]. Available: https://nacto.org/wp-content/uploads/2019/05/NACTO_IMLA_Managing-Mobility-Data.pdf.

[12] “ACM SIGSPATIAL Cup 2016.” 2016. [Online]. Available: http://sigspatial2016.sigspatial.org/giscup2016/.

[13] Y. Zheng, “Trajectory Data Mining: An Overview,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, May 2015, doi: 10.1145/2743025.

[14] J. D. Mazimpaka and S. Timpf, “Trajectory data mining: A review of methods and applications,” *JOSIS*, no. 13, pp. 61–99, Dec. 2016, doi: 10.5311/JOSIS.2016.13.263.

[15] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong, “A Survey on Trajectory Data Management, Analytics, and Learning,” *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–36, Mar. 2022, doi: 10.1145/3440207.

[16] C. Doulkeridis, A. Vlachou, N. Pelekis, and Y. Theodoridis, “A Survey on Big Data Processing Frameworks for Mobility Analytics,” *SIGMOD Rec.*, vol. 50, no. 2, pp. 18–29, Aug. 2021, doi: 10.1145/3484622.3484626.

[17] X. Ding, L. Chen, Y. Gao, C. S. Jensen, and H. Bao, “UlTraMan: a unified platform for big trajectory data management and analytics,” *Proc. VLDB Endow.*, vol. 11, no. 7, pp. 787–799, Mar. 2018, doi: 10.14778/3192965.3192970.

[18] P. Cudre-Mauroux, E. Wu, and S. Madden, “TrajStore: An adaptive storage system for very large trajectory data sets,” in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, Long Beach, CA, USA, 2010, pp. 109–120. doi: 10.1109/ICDE.2010.5447829.

[19] Z. Shang, G. Li, and Z. Bao, “DITA: Distributed In-Memory Trajectory Analytics,” in *Proceedings of the 2018 International Conference on Management of Data*, Houston TX USA, May 2018, pp. 725–740. doi: 10.1145/3183713.3183743.

[20] H. Wang, K. Zheng, J. Xu, B. Zheng, X. Zhou, and S. Sadiq, “SharkDB: An In-Memory Column-Oriented Trajectory Storage,” in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, Shanghai China, Nov. 2014, pp. 1409–1418. doi: 10.1145/2661829.2661878.

[21] S. Wang, Z. Bao, J. S. Culpepper, Z. Xie, Q. Liu, and X. Qin, “Torch: A Search Engine for Trajectory Data,” in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Ann Arbor MI USA, Jun. 2018, pp. 535–544. doi: 10.1145/3209978.3209989.

[22] R. Li *et al.*, “TrajMesa: A Distributed NoSQL Storage Engine for Big Trajectory Data,” in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, Dallas, TX, USA, Apr. 2020, pp. 2002–2005. doi: 10.1109/ICDE48307.2020.00224.

[23] H. Tan, W. Luo, and L. M. Ni, “CloST: a hadoop-based storage system for big spatio-temporal data analytics,” in *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM ’12*, Maui, Hawaii, USA, 2012, p. 2139. doi: 10.1145/2396761.2398589.

[24] H. Su, K. Zheng, H. Wang, J. Huang, and X. Zhou, “Calibrating trajectory data for similarity-based analysis,” in *Proceedings of the 2013 international conference on Management of data - SIGMOD ’13*, New York, New York, USA, 2013, p. 833. doi: 10.1145/2463676.2465303.

[25] K.-Y. Whang and R. Krishnamurthy, “The Multilevel Grid File - A Dynamic Hierarchical Multidimensional,” in *Proceedings of the Second International Symposium on Database Systems for Advanced Applications*, Tokyo, Japan, Apr. 1991, pp. 449–459.

[26] S. Ruan, R. Li, J. Bao, T. He, and Y. Zheng, “CloudTP: A Cloud-Based Flexible Trajectory Preprocessing Framework,” in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, Paris, Apr. 2018, pp. 1601–1604. doi: 10.1109/ICDE.2018.00186.

[27] J. Bao, R. Li, X. Yi, and Y. Zheng, “Managing massive trajectories on the cloud,” in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame California, Oct. 2016, pp. 1–10. doi: 10.1145/2996913.2996916.

[28] S. Yang, Z. He, and Y.-P. P. Chen, “GCOTraj: A storage approach for historical trajectory data sets using grid cells ordering,” *Information Sciences*, vol. 459, pp. 1–19, Aug. 2018, doi: 10.1016/j.ins.2018.04.087.

[29] S. Yang, Z. He, and Y.-P. P. Chen, “Workload-Based Ordering of Multi-Dimensional Data,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 831–844, Mar. 2016, doi: 10.1109/TKDE.2015.2496252.

[30] M. Bakli, M. Sakr, and T. H. A. Soliman, “HadoopTrajectory: a Hadoop spatiotemporal data processing extension,” *J Geogr Syst*, vol. 21, no. 2, pp. 211–235, Jun. 2019, doi: 10.1007/s10109-019-00292-4.

[31] “Chronicle Map.” Chronicle Software : Open Source, Sep. 15, 2022. Accessed: Sep. 15, 2022. [Online]. Available: https://github.com/OpenHFT/Chronicle-Map

[32] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, doi: 10.1145/1327452.1327492.

[33] M. Zaharia *et al.*, “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing,” *NSDI’12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, p. 14.

[34] M. Bakli, M. Sakr, and E. Zimanyi, “Distributed Mobility Data Management in MobilityDB,” in *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, Versailles, France, Jun. 2020, pp. 238–239. doi: 10.1109/MDM48529.2020.00052.

[35] E. Zimányi, M. Sakr, and A. Lesuisse, “MobilityDB: A Mobility Database Based on PostgreSQL and PostGIS,” *ACM Trans. Database Syst.*, vol. 45, no. 4, pp. 1–42, Dec. 2020, doi: 10.1145/3406534.

[36] A. Belussi, S. Migliorini, and A. Eldawy, “Detecting skewness of big spatial data in SpatialHadoop,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle Washington, Nov. 2018, pp. 432–435. doi: 10.1145/3274895.3274923.

[37] S. Migliorini, A. Belussi, E. Quintarelli, and D. Carra, “A Context-based Approach for Partitioning Big Data,” *Proceedings of the 23rd International Conference on Extending Database Technology (EDBT)*. OpenProceedings.org, 2020. doi: 10.5441/002/EDBT.2020.50.

[38] L. Alarabi, “SIGSPATIAL: G: Summit: A Scalable System for Massive Trajectory Data Management,” p. 5.

[39] L. Alarabi, M. F. Mokbel, and M. Musleh, “ST-Hadoop: A MapReduce Framework for Spatio-Temporal Data,” in *Advances in Spatial and Temporal Databases*, vol. 10411, M. Gertz, M. Renz, X. Zhou, E. Hoel, W.-S. Ku, A. Voisard, C. Zhang, H. Chen, L. Tang, Y. Huang, C.-T. Lu, and S. Ravada, Eds. Cham: Springer International Publishing, 2017, pp. 84–104. doi: 10.1007/978-3-319-64367-0_5.

[40] Z. Fang, L. Chen, Y. Gao, L. Pan, and C. S. Jensen, “Dragoon: a hybrid and efficient big trajectory management system for offline and online analytics,” *The VLDB Journal*, vol. 30, no. 2, pp. 287–310, Mar. 2021, doi: 10.1007/s00778-021-00652-x.

[41] Z. Zhang, C. Jin, J. Mao, X. Yang, and A. Zhou, “TrajSpark: A Scalable and Efficient In-Memory Management System for Big Trajectory Data,” in *Web and Big Data*, vol. 10366, L. Chen, C. S. Jensen, C. Shahabi, X. Yang, and X. Lian, Eds. Cham: Springer International Publishing, 2017, pp. 11–26. doi: 10.1007/978-3-319-63579-8_2.

[42] H. He, R. Li, J. Bao, T. Li, and Y. Zheng, “JUST-Traj: A Distributed and Holistic Trajectory Data Management System,” in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, Beijing China, Nov. 2021, pp. 403–406. doi: 10.1145/3474717.3483990.

[43] X. Ding, R. Chen, L. Chen, Y. Gao, and C. S. Jensen, “VIPTRA: Visualization and Interactive Processing on Big Trajectory Data,” in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, Aalborg, Denmark, Jun. 2018, pp. 290–291. doi: 10.1109/MDM.2018.00055.

[44] Q. Ma, B. Yang, W. Qian, and A. Zhou, “Query processing of massive trajectory data based on mapreduce,” in *Proceeding of the first international workshop on Cloud data management - CloudDB ’09*, Hong Kong, China, 2009, p. 9. doi: 10.1145/1651263.1651266.

[45] P. Raj, “A Detailed Analysis of NoSQL and NewSQL Databases for Bigdata Analytics and Distributed Computing,” in *Advances in Computers*, vol. 109, Elsevier, 2018, pp. 1–48. doi: 10.1016/bs.adcom.2018.01.002.

[46] B. Zheng, H. Wang, K. Zheng, H. Su, K. Liu, and S. Shang, "SharkDB: an in-memory column-oriented storage for trajectory analysis," *World Wide Web*, vol. 21, no. 2, pp. 455–485, Mar. 2018, doi: 10.1007/s11280-017-0466-9.

[47] N. Meratnia and R. A. de By, "Spatiotemporal Compression Techniques for Moving Point Objects," in *Advances in Database Technology - EDBT 2004*, vol. 2992, E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, and E. Ferrari, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 765–782. doi: 10.1007/978-3-540-24741-8_44.

[48] D. Xie, F. Li, B. Yao, G. Li, L. Zhou, and M. Guo, "Simba: Efficient In-Memory Spatial Analytics," in *Proceedings of the 2016 International Conference on Management of Data*, San Francisco California USA, Jun. 2016, pp. 1071–1085. doi: 10.1145/2882903.2915237.

[49] S. T. Leutenegger, M. A. Lopez, and J. Edgington, "STR: a simple and efficient algorithm for R-tree packing," in *Proceedings 13th International Conference on Data Engineering*, Birmingham, UK, 1997, pp. 497–506. doi: 10.1109/ICDE.1997.582015.

[50] L. Alarabi, "Summit: a scalable system for massive trajectory data management," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Seattle Washington, Nov. 2018, pp. 612–613. doi: 10.1145/3274895.3282795.

[51] P. Tampakis, C. Doulkeridis, N. Pelekis, and Y. Theodoridis, "Distributed Subtrajectory Join on Massive Datasets," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 2, pp. 1–29, Jun. 2020, doi: 10.1145/3373642.

[52] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proceedings of the 18th international conference on World wide web - WWW '09*, Madrid, Spain, 2009, p. 791. doi: 10.1145/1526709.1526816.

[53] A. Gani, A. Siddiqa, S. Shamshirband, and F. Hanum, "A survey on indexing techniques for big data: taxonomy and performance evaluation," *Knowl Inf Syst*, vol. 46, no. 2, pp. 241–284, Feb. 2016, doi: 10.1007/s10115-015-0830-y.

[54] K. Funaki, T. Hochin, H. Nomiya, H. Nakanishi, and M. Kojima, "Parallel Indexing Scheme for Data Intensive Applications," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, Kokura Kita-ku, Japan, Aug. 2014, pp. 630–635. doi: 10.1109/IIAI-AAI.2014.133.

[55] D. Xie *et al.*, "Simba: spatial in-memory big data analysis," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame California, Oct. 2016, pp. 1–4. doi: 10.1145/2996913.2996935.

[56] C. BÖxhm, G. Klump, and H.-P. Kriegel, "XZ-Ordering: A Space-Filling Curve for Objects with Spatial Extension," in *Advances in Spatial Databases*, vol. 1651, R. H. Güting, D. Papadias, and F. Lochovsky, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 75–90. doi: 10.1007/3-540-48482-5_7.

[57] H. Sagan, *Space-Filling Curves*, Photocomposed copy prepared from the author's TeX files. New York, NY: Springer Science+Business Media, LLC, 1994. doi: 10.1007/978-1-4612-0871-6.

[58] R. Li *et al.*, "TrajMesa: A Distributed NoSQL-Based Trajectory Data Management System," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2022, doi: 10.1109/TKDE.2021.3079880.

[59] J. Gudmundsson and M. van Kreveld, "Computing longest duration flocks in trajectory data," in *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems - GIS '06*, Arlington, Virginia, USA, 2006, p. 35. doi: 10.1145/1183471.1183479.

[60] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, “Discovery of Convoys in Trajectory Databases.” arXiv, Feb. 04, 2010. Accessed: Oct. 09, 2022. [Online]. Available: http://arxiv.org/abs/1002.0963

[61] X. Li, V. Ceikute, C. S. Jensen, and K.-L. Tan, “Effective Online Group Discovery in Trajectory Databases,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2752–2766, Dec. 2013, doi: 10.1109/TKDE.2012.193.

[62] Z. Li, B. Ding, J. Han, and R. Kays, “Swarm: mining relaxed temporal moving object clusters,” *Proc. VLDB Endow.*, vol. 3, no. 1–2, pp. 723–734, Sep. 2010, doi: 10.14778/1920841.1920934.

[63] M. Ester, H.-P. Kriegel, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in *Proceedings of the second international conference on knowledge discovery & data mining*, Portland, OR, Dec. 1996.

[64] Q. Fan, D. Zhang, H. Wu, and K.-L. Tan, “A general and parallel platform for mining co-movement patterns over large-scale trajectories,” *Proc. VLDB Endow.*, vol. 10, no. 4, pp. 313–324, Nov. 2016, doi: 10.14778/3025111.3025114.

[65] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, “Querying and mining of time series data: experimental comparison of representations and distance measures,” *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1542–1552, Aug. 2008, doi: 10.14778/1454159.1454226.

[66] Y. Zheng and X. Zhou, Eds., *Computing with Spatial Trajectories*. New York, NY: Springer New York, 2011. doi: 10.1007/978-1-4614-1629-6.

[67] L. Chen, Y. Gao, Z. Fang, X. Miao, C. S. Jensen, and C. Guo, “Real-time distributed co-movement pattern detection on streaming trajectories,” *Proc. VLDB Endow.*, vol. 12, no. 10, pp. 1208–1220, Jun. 2019, doi: 10.14778/3339490.3339502.

[68] R. Lange, F. Dürr, and K. Rothermel, “Scalable processing of trajectory-based queries in space-partitioned moving objects databases,” in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems - GIS ’08*, Irvine, California, 2008, p. 10. doi: 10.1145/1463434.1463474.

[69] M. Yuan *et al.*, “OceanST: a distributed analytic system for large-scale spatiotemporal mobile broadband data,” *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1561–1564, Aug. 2014, doi: 10.14778/2733004.2733030.

[70] S. Nishimura, S. Das, D. Agrawal, and A. E. Abbadi, “MD-HBase: A Scalable Multi-dimensional Data Infrastructure for Location Aware Services,” in *2011 IEEE 12th International Conference on Mobile Data Management*, Lulea, Sweden, Jun. 2011, pp. 7–16. doi: 10.1109/MDM.2011.41.