



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Απόρριψη φόρτου από ρεύματα τροχιάς
κινούμενων αντικειμένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΕΡΑΦΕΙΜ ΠΑΠΑΔΙΑ

Επιβλέπων: Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΓΝΩΣΕΩΝ ΚΑΙ ΔΕΔΟΜΕΝΩΝ
Αθήνα, Ιούλιος 2015



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων

Απόρριψη φόρτου από ρεύματα τροχιάς κινούμενων αντικειμένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΕΡΑΦΕΙΜ ΠΑΠΑΔΙΑ

Επιβλέπων: Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 29η Ιουλίου 2015.

(Υπογραφή)

.....
Ιωάννης Βασιλείου
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Νεκτάριος Κοζύρης
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....
Ιωάννης Θεοδορίδης
Καθηγητής Παν. Πειραιώς

Αθήνα, Ιούλιος 2015

(Υπογραφή)

.....

ΣΕΡΑΦΕΙΜ ΠΑΠΑΔΙΑΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

© 2015 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων

Copyright ©–All rights reserved ΣΕΡΑΦΕΙΜ ΠΑΠΑΔΙΑΣ, 2015.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Ευχαριστίες

Καθόλη την πορεία μου στο πολυτεχνείο ήρθα σε επαφή με ενδιαφέροντα άτομα, ανθρώπους που με ενέπνευσαν να προσπαθώ διαρκώς να γίνομαι καλύτερος και πάντα να πετυχαίνω το κάτι παραπάνω. Φυσικά, ό,τι έχω καταφέρει έως τώρα δεν είναι αποκλειστικά δικό μου επίτευγμα, αλλά είχα βοήθεια και ερεθίσματα από όλους όσους πρόκειται να αναφέρω παρακάτω.

Η παράγραφος αυτή πραγματικά δεν θα μπορούσε να μην είναι αφιερωμένη αποκλειστικά στον κ.Κώστα Πατρούμπα. Σε τέτοιες περιπτώσεις το “ευχαριστώ” δεν είναι αρκετό. Δεν πιστεύω στη δύναμη των κλισέ εκφράσεων, αλλά η σχέση που ανέπτυξα με τον συγκεκριμένο άνθρωπο αποτελεί “σχέση ζωής”. Αρχικά, ως αναφέρω πως κατέχει στο έπακρο το αντικείμενο του, τα ρεύματα δεδομένων, και έχει την υπομονή, την επιμονή αλλά και την θέληση να εξηγήσει και να προσπαθεί να δώσει στον άλλον να καταλάβει όσες φορές και αν χρειαστεί. Το ήθος και η ανιδιοτέλειά του πραγματικά με εντυπωσίασαν. Επιπρόσθετα, από την πρώτη μέρα πίστεψε σε μένα περισσότερο απόσο πίστευα εγώ ο ίδιος στον εαυτό μου. Αυτό αποτέλεσε καταλύτη για την ψυχосύνθεσή μου και για την τελική έκβαση αυτή της διπλωματικής εργασίας. Αυτή η παράγραφος περιέχει “τοποθέτηση προϊόντος”, γι’ αυτό όποιος διαβάσει αυτά τα λόγια, προτείνω ανεπιφύλαχτα τη συνεργασία με τον κ.Πατρούμπα, η οποία θα είναι άψογη, άρτια, αποτελεσματική, παραγωγική και πραγματικά μόνο καλά πράγματα μπορεί να απορροφήσει κανείς από αυτήν. Ειλικρινά, θεωρώ πως έκανα έναν πολύ καλό φίλο εκτός των άλλων.

Ακόμα θα ήθελα να ευχαριστήσω τον καθηγητή κ.Ιωάννη Βασιλείου για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων. Επίσης, ευχαριστώ για την βοήθειά τους σε κάθε επίπεδο και νιώθω τυχερός που γνώρισα τους φίλους μου, Χρήστο, Κώστα, Κώστα, Γιώργη, Γιώργο, Βαγγέλη και όποιον άλλον πιθανώς ξεχνάω. Προφανώς, θα ήθελα να ευχαριστήσω την οικογένειά μου για την καθοδήγηση και την ηθική στήριξη που μου προσέφερε όλα αυτά τα χρόνια. Πραγματικά, το ότι με ανέχονται ακόμα αποτελεί άθλο.

Χωρίς όλους τους αναφερθέντες, τόσο το ταξίδι για την Ιθάκη θα ήταν υποδεέστερο, όσο και ο προορισμός θα απείχε παρασάγγας από τον τελικό.

Η διπλωματική εργασία αυτή είναι αφιερωμένη στον παππού μου, τον “κούκλο”, ο οποίος δεν είναι πια κοντά μου.

Περίληψη

Τα τελευταία χρόνια, παρατηρείται ραγδαία αύξηση του όγκου των δεδομένων που παράγονται και πρέπει να εξυπηρετήσουν διάφορες εφαρμογές. Μία κατηγορία είναι οι εφαρμογές παρακολούθησης κινούμενων αντικειμένων, στις οποίες το σύστημα οφείλει να επεξεργάζεται τον όγκο των δεδομένων που του δίνεται και να απαντάει έγκαιρα και αξιόπιστα σε διαφόρων τύπων ερωτήματα.

Οι πληροφορίες για την κίνηση των αντικειμένων καταφθάνουν στο σύστημα με τη μορφή ρευμάτων δεδομένων, για την επεξεργασία των οποίων, διατυπώνονται ερωτήματα διαρκείας που παραμένουν ενεργά για μεγάλο χρονικό διάστημα. Τα κινούμενα αντικείμενα θεωρούνται σημειακά και η θέση τους μεταβάλλεται δυναμικά. Κάθε αντικείμενο αποστέλλει ανά τακτά χρονικά διαστήματα την καινούργια θέση του σε έναν κεντρικό επεξεργαστή που καλείται να δίνει απάντηση στα ερωτήματα διαρκείας. Όμως, υπάρχει ενδεχόμενο το στίγμα των αντικειμένων να καταγράφεται με πολύ μεγάλη συχνότητα (λ.χ. κάθε 10 δευτερόλεπτα), οπότε ο όγκος της πληροφορίας γίνεται υπερβολικά μεγάλος και να καθυστερεί η επεξεργασία ποικίλων ερωτημάτων διαρκείας, τα οποία πρέπει να παρέχουν συνεχώς επίκαιρες απαντήσεις. Για το λόγο αυτό, και με δεδομένο ότι υπάρχουν πεπερασμένη μνήμη και επεξεργαστική ισχύς, μία ιδέα είναι το σύστημα να επεξεργάζεται ένα μόνο μέρος της εισόδου έτσι ώστε να δίνονται έγκαιρα απαντήσεις στα τρέχοντα ερωτήματα, με αντίκτυπο βέβαια στην ποιότητα των αποτελεσμάτων.

Αντικείμενο της διπλωματικής εργασίας είναι η ανάπτυξη και υλοποίηση αλγορίθμου απόρριψης φόρτου, που αφορά την επιλογή με σημασιολογικό τρόπο, ποιές και πόσες από τις διαθέσιμες τροχιές θα εξετάζει πλήρως το σύστημα με σκοπό να ικανοποιεί τις απαιτήσεις. Από την άλλη μεριά, ο αλγόριθμος που προτείνουμε είναι προσαρμοστικός, δηλαδή ο αριθμός τροχιών που εξετάζονται σε κάθε κύκλο εκτέλεσης αναπροσαρμόζεται ανάλογα με τον φόρτο του συστήματος έτσι ώστε να αξιοποιούνται πλήρως οι διαθέσιμοι πόροι του. Με εφαρμογή του ευριστικού αλγορίθμου που κατασκευάστηκε, πραγματοποιήθηκαν πειράματα σε συνθετικά δεδομένα πάνω στο χάρτη της Αττικής, από τα οποία προέκυψαν θετικά αποτελέσματα. Επίσης, επιβεβαιώθηκαν οι αναμενόμενες επιδόσεις τους σχετικά με τους χρόνους εκτέλεσης και την ακρίβεια των απαντήσεων.

Λέξεις Κλειδιά

Ρεύματα δεδομένων, Κινούμενα Αντικείμενα, Τροχιά, Απόρριψη Φόρτου, Παράθυρα, Συνόψεις.

Abstract

In recent years, there is a rapid increase in the volume of data generated. These data are required to get managed and edited by various applications. In applications that involve tracking of moving objects the system has to process a vast amount of incoming data and answer different types of questions, online and precisely.

The data derived from the moving objects form a data stream. For the processing of such streams we define continuous queries which remain active for a long period of time. Moving objects are abstracted as points in the 2D plane and their position is updated dynamically. Each object sends periodically its new location to a central processor, who must give response to several continuous queries. It is highly likely, however, that the position of each object is updated with very high frequency (e.g. every 10 seconds), thus the amount of information accumulated gets big and a data stream becomes too large. Thus, the system must manage data online, and will probably delay the processing of various continuous queries instead of providing instant responses. Also, by taking under consideration that there are finite memory and processing power, one idea is to process only a part of the input stream so as to give online answers to current questions affecting of course the quality of the results.

This thesis aims to design and implement a load shedding algorithm for trajectory streams of moving objects, allowing online answers to spatio-temporal continuous queries, with some discount on accuracy of the responses. This is achieved with finding a heuristic way of selecting to preserve and fully process only the trajectories that lead to the lowest discount in accuracy and at the same time satisfying the time constraints which the system provides. In addition, the proposed algorithm is adaptive, thus the number of trajectories fully processed at each execution cycle might change and specifically adapts such as to satisfy the system's constraints and use the maximum processing power that the system has. The experimental evaluation of our algorithm showed promising results for both time-consumption and accuracy.

Keywords

Data Streams, Moving Objects, Trajectories, Load Shedding, Windows, Synopses.

Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	5
Περιεχόμενα	9
Κατάλογος Σχημάτων	12
Κατάλογος Πινάκων	13
1 Ρεύματα κινούμενων αντικειμένων	19
1.1 Εισαγωγή	19
1.2 Ρεύματα δεδομένων	20
1.2.1 Μοντέλο ρευμάτων δεδομένων	20
1.2.2 Βασικά χαρακτηριστικά ρευμάτων δεδομένων και η συσχέτισή τους με τα 4V των Big Data	21
1.3 Συστήματα διαχείρισης ρευμάτων δεδομένων	22
1.3.1 Προδιαγραφές συστήματος	22
1.3.2 Κυριότερα συστήματα διαχείρισης ρευμάτων δεδομένων	23
1.4 Ερωτήματα σε ρεύματα δεδομένων	24
1.5 Παράθυρα	25
1.5.1 Εισαγωγή στα παράθυρα	25
1.5.2 Ιδιότητες χρονικών παραθύρων	26
1.6 Αλγόριθμοι για ρεύματα δεδομένων	26
1.7 Ανάγκη για απεριόριστη μνήμη	27
1.8 Προσεγγιστικές απαντήσεις σε ερωτήματα	27
1.9 Συνόψεις	28
1.9.1 Μαζική επεξεργασία	28
1.9.2 Περιλήψεις ή συνόψεις δεδομένων	28

2	Διαχείριση κινούμενων αντικειμένων	29
2.1	Εισαγωγή	29
2.2	Χωροχρονικά δεδομένα	29
2.3	Τροχιές κινούμενων αντικειμένων	30
2.4	Ερωτήματα σε κινούμενα αντικείμενα	31
2.4.1	Ερωτήματα θέσης	31
2.4.2	Ερωτήματα τροχιάς	32
2.5	Δεικτοδότηση κινούμενων αντικειμένων	33
3	Αλγόριθμοι απόρριψης φόρτου από ρεύματα κινούμενων αντικειμένων	35
3.1	Απόρριψη φόρτου από ρεύματα δεδομένων	36
3.1.1	Απόρριψη φόρτου με βάση διαγράμματα ποιότητας	36
3.1.2	Απόρριψη φόρτου με δειγματοληψία	36
3.1.3	Απόρριψη φόρτου με χρήση παραθύρων	38
3.1.4	Απόρριψη φόρτου με βάση τεχνικές αυτομάτου ελέγχου	39
3.1.5	Υβριδική απόρριψη φόρτου	39
3.2	Απόρριψη φόρτου σε κινούμενα αντικείμενα	40
3.2.1	Απόρριψη φόρτου κατά περιοχές	40
3.2.2	Απόρριψη φόρτου με τεχνικές εξόρυξης από δεδομένα κινούμενων αντικειμένων	41
3.2.3	Απόρριψη φόρτου με κλιμακώσιμες τεχνικές	43
4	Μηχανισμός απόρριψης φόρτου από τροχιές	45
4.1	Εισαγωγή	45
4.2	Διαχείριση ρευμάτων κινούμενων οχημάτων	46
4.2.1	Το μοντέλο δεδομένων	46
4.2.2	Το μοντέλο ερωτημάτων	46
4.3	Αρχιτεκτονική του συστήματος	47
4.4	Φάση ενημέρωσης	50
4.5	Φάση αποτίμησης ερωτημάτων	52
4.6	Φάση απόρριψης φόρτου	55
5	Πειραματική Αξιολόγηση	61
5.1	Πειραματικό πλαίσιο	61
5.1.1	Παραγωγή συνθετικών δεδομένων	61
5.1.2	Πειραματικά δεδομένα	62
5.2	Αξιολόγηση αποτελεσμάτων	63
5.2.1	Διαστασιολόγηση καννάβου	63
5.2.2	Αναπροσαρμογή φόρτου συστήματος	64
5.2.3	Επίδραση βήματος απόρριψης φόρτου	64
5.2.4	Ποσοστά επιλεγμένων τροχιών βάσει κριτηρίων απόρριψης φόρτου	66

<i>Περιεχόμενα</i>	<i>9</i>
5.2.5 Επίδραση ζώνης ανοχής θ	67
5.2.6 Ποιότητα αποτελεσμάτων	68
6 Επίλογος	69
6.1 Συμπεράσματα	69
6.2 Μελλοντικές επεκτάσεις	70
Βιβλιογραφία	72

Κατάλογος Σχημάτων

1.1 Ένα αφαιρετικό πλαίσιο λειτουργίας για Συστήματα Διαχείρισης Ρευμάτων Δεδομένων [31]	22
2.1 Χωροχρονικό χώρος [20]	30
2.2 Αναπαράσταση τροχιάς κινούμενου αντικειμένου στον τρισδιάστατο χώρο (x, y, t) [20]	31
2.3 Τύποι ερωτημάτων για κινούμενα αντικείμενα [21]	33
3.1 Load Shedding Road Map [25]	37
3.2 Ιδέα μείωσης του αξιοποιήσιμου ρυθμού εισόδου [4]	37
3.3 Κανόνες για Window drop τελεστές [26]	38
3.4 Σύστημα κλειστού ελέγχου [29]	39
3.5 Σύστημα Data Triage [22]	40
3.6 Αρχιτεκτονική του συστήματος LIRA [9]	41
3.7 Αναπαράσταση μίας κινούμενης συστάδας [19]	42
4.1 Τροχιές οχημάτων κατά μήκος οδικών αξόνων	47
4.2 Αρχιτεκτονική απόρριψης φόρτου από τροχιές	48
4.3 Λίστα τροχιάς κινούμενου αντικειμένου	52
4.4 Σύνοψη κινούμενου οχήματος	52
4.5 Δομή EvalStruct	54
4.6 Σύνδεση του EvalStruct και της λίστας τροχιάς ενός κινούμενου αντικειμένου	55
4.7 Δομή για τήρηση αντικειμένων που κινούνται σε κάθε ενεργό ερώτημα	56
4.8 Συνθήκη ενεργοποίησης για απόρριψη φόρτου	57
4.9 Πίνακας αληθείας της απόρριψης φόρτου	59
5.1 Κλιμάκωση χρόνου αποτίμησης για διαφορετικό διάφορες υποδιαιρέσεις καννάβου	63
5.2 Προσαρμογή φόρτου για διάφορες τιμές της χωρητικότητας C	65
5.3 Αριθμός εκτελέσεων απόρριψης φόρτου ανά χρονόσημο για διάφορες τιμές του C	66
5.4 Αριθμός εκτελέσεων απόρριψης φόρτου για διάφορες τιμές του s	66
5.5 Μέσος χρόνος εκτέλεσης κάθε φάσης ανά κύκλο εκτέλεσης για τις διάφορες τιμές του s	67

5.6 Μέσος αριθμός τροχιών που κατακρατούνται από κάθε κριτήριο ανά κύκλο εκτέλεσης	67
5.7 Μέσος αριθμός τροχιών που κατακρατούνται από κάθε κριτήριο ανά κύκλο εκτέλεσης	68

Κατάλογος Πινάκων

5.1	Παράμετροι πειραμάτων	62
5.2	Μνήμη που καταλαμβάνει ο κώδικας για διάφορες κατατιμήσεις (χονδροκομ- μένος υπολογισμός)	64
5.3	Ποιότητα αποτελεσμάτων	68

Εισαγωγή

Τα τελευταία χρόνια, παρατηρείται ραγδαία αύξηση του όγκου των δεδομένων που παράγονται, μεταδίδονται και απαιτούν επεξεργασία από εφαρμογές παρακολούθησης (δίκτυα αισθητήρων, συστήματα χρηματοπιστωτικών συναλλαγών, τηλεπικοινωνίες κ.ά.). Στα πλαίσια των υπηρεσιών γεωγραφικού εντοπισμού (*Location-based Services*), ιδιαίτερο ενδιαφέρον έχει η διαχείριση πληροφοριών από κινούμενα αντικείμενα (*moving objects*), οι οποίες καταφθάνουν στο σύστημα ως ρεύματα δεδομένων (*data streams*).

Για την επεξεργασία τέτοιων ρευμάτων διατυπώνονται ερωτήματα διαρκείας (*continuous queries*) που παραμένουν ενεργά επί μακρόν. Τέτοια χωροχρονικά ερωτήματα (περιοχής, k -εγγύτερων γειτόνων, συναθροιστικά κ.ά.) οφείλουν να εξετάζουν εγκαίρως τον κυμαινόμενο όγκο των εισερχόμενων δεδομένων και να παρέχουν αξιόπιστες και διαρκώς επίκαιρες απαντήσεις.

Τα κινούμενα αντικείμενα θεωρούνται σημειακά και καθώς η θέση τους μεταβάλλεται δυναμικά αποστέλλεται ανά ταχτά χρονικά διαστήματα σ' έναν κεντρικό επεξεργαστή. Υπάρχει ενδεχόμενο το στίγμα των αντικειμένων να καταγράφεται με πολύ μεγάλη συχνότητα (λ.χ. κάθε 10 δευτερόλεπτα). Τότε ο όγκος της πληροφορίας που συσσωρεύεται γίνεται υπερβολικά μεγάλος, και εφόσον το σύστημα οφείλει να διαχειρίζεται online τα δεδομένα, ενδεχομένως να καθυστερεί η επεξεργασία ποικίλων ερωτημάτων. Για τον λόγο αυτόν, δεδομένου ότι η διαθέσιμη μνήμη και η επεξεργαστική ισχύς του συστήματος είναι πεπερασμένες, μία ιδέα είναι το σύστημα να επεξεργάζεται μόνο ένα μέρος της εισόδου. Η τεχνική της απόρριψης φόρτου (*load shedding*) έχει εφαρμοστεί σε συστήματα επεξεργασίας ρευμάτων δεδομένων, μειώνοντας την υστέρηση στην ανανέωση των αποτελεσμάτων, χωρίς σοβαρές απώλειες στην ποιότητά τους.

Αντικείμενο της διπλωματικής

Στόχος της διπλωματικής εργασίας είναι η κατασκευή ενός αυτόρρυθμιζόμενου μηχανισμού ο οποίος θα διαχειρίζεται πολύ μεγάλους όγκους δεδομένων, που αφορούν *τροχιές που εξελίσσονται στο χρόνο*. Επειδή δεν είναι εφικτό να επεξεργαστούν όλες οι τροχιές που υπάρχουν στο σύστημα και να απαντώνται τα ερωτήματα σε πραγματικό χρόνο, υλοποιείται ένας μηχανισμός απόρριψης ολόκληρων τροχιών ή μέρους τροχιών με απώτερο σκοπό να μειωθεί ο φόρτος του συστήματος.

Βεβαίως, μπορούν να χρησιμοποιηθούν διάφορες πολιτικές απόρριψης τροχιών κινούμενων

αντικειμένων. Ωστόσο, απορρίπτοντας λιγότερο σημαντικά ή λιγότερο χρήσιμα κομμάτια τροχιών ώστε να μειωθεί ο συνολικός χρόνος επεξεργασίας των ερωτημάτων και να συμβαδίζει με τα τεράστια ποσά ενημερώσεων στις θέσεις των κινούμενων αντικειμένων είναι απολύτως απαραίτητο σε αυτή την περίπτωση.

Η τήρηση ολόκληρου του ιστορικού κάθε τροχιάς δεν είναι εφικτή τόσο λόγω ελλειπών πόρων συστήματος όσο και απεριόριστων απαιτήσεων σε μνήμη. Επίσης, έχει νόημα να τηρούνται ολόκληρες τροχιές ή μέρος τροχιών μόνο εφόσον είναι χρήσιμες σε πολλά ερωτήματα ταυτόχρονα. Ακόμα και αν εξετάζεται μόνο το πρόσφατο κομμάτι κάθε τροχιάς και πάλι είναι πολύ πιθανό να ανακύπτει πρόβλημα πόρων λόγω της παρακολούθησης πολύ μεγάλου αριθμού κινούμενων αντικειμένων. Η μείωση του όγκου των δεδομένων μπορεί να έχει ήδη εφαρμοστεί από τις πηγές στα κινούμενα αντικείμενα, λ.χ. οδηγώντας σε απλοποίηση τροχιών με χρήση συνόψεων. Ωστόσο, αυτή η διαδικασία κυρίως επηρεάζει την *ποιότητα των απαντήσεων* και το κόστος διακίνησης της πληροφορίας (*communication cost*), παρά την επίδοση των ερωτημάτων που περιλαμβάνει την προσαρμοστικότητα (*adaptivity*) και την κλιμακωσιμότητα (*scalability*), την ικανότητα δηλαδή το σύστημα να μπορεί να διαχειρίζεται και να αναποκρίνεται σε οποιοδήποτε ρυθμό εισόδου στιγμάτων θέσεις των κινούμενων αντικειμένων.

Η επιβάρυνση από τον μηχανισμό απόρριψης φόρτου πρέπει να είναι μικρή, συγκριτικά με την επεξεργασία ερωτημάτων. Η πολιτική απόρριψης φόρτου θα πρέπει γενικά να εξαλείφει πλήρως ή μερικώς αυτές που είναι ελάχισονος σημασίας και δεν συνεισφέρουν πολύ στην αύξηση της ακρίβειας των απαντήσεων.

Είναι λοιπόν επιθυμητό να κρατώνται πολλαπλά τμήματα τροχιών στην κύρια μνήμη, αλλά να αποφασίζεται δυναμικά ποιές από τις τροχιές θα χρησιμεύουν στην απάντηση των ερωτημάτων με μικρή υστέρηση. Αυτού θα μπορούσε να επιτευχθεί με μία απλοποιημένη δεικτοδότηση των τροχιών με σκοπό να εντοπίζεται γρήγορα η χρησιμότητά τους στην επεξεργασία ενεργών ερωτημάτων. Πέραν των επιδόσεων, η επιτυχία του προτεινόμενου μηχανισμού θα εξαρτηθεί επίσης από την ποιότητα των απαντήσεων, επιδιώκοντας την μικρότερη δυνατή απώλεια στην ακρίβεια, σε σύγκριση με την εξαντλητική αποτίμηση των ερωτημάτων.

Τέλος, θα παρουσιάσουμε ένα μηχανισμό που θα είναι όπως προείπαμε προσαρμοστικός και θα δουλεύει σε κάθε ρυθμό εισόδου δεδομένων και θα χρησιμοποιείται για την προσεγγιστική και αυξητική απάντηση τυπικών χωροχρονικών συνεχών ερωτημάτων σε πραγματικό χρόνο και με την μικρότερη δυνατή απώλεια στην ακρίβεια (σε σύγκριση με την εξαντλητική αποτίμηση των ερωτημάτων).

Οργάνωση του τόμου

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε έξι κεφάλαια. Στα κεφάλαια 1, 2, 3 παρέχονται το θεωρητικό υπόβαθρο και πληροφορίες για τα ρεύματα δεδομένων, τα κινούμενα αντικείμενα και την απόρριψη φόρτου τόσο από ρεύματα δεδομένων όσο και από κινούμενα αντικείμενα. Στο κεφάλαιο 4 γίνεται η διατύπωση του προβλήματος και η ανάλυση του αλγορίθμου που αναπτύχθηκε. Τέλος, στο κεφάλαιο 5 γίνεται ανάλυση των πειραματικών αποτελεσμάτων και στο κεφάλαιο 7 εξάγονται τα συνολικά συμπεράσματα της εργασίας και

αναφέρονται οι μελλοντικές επεκτάσεις. Συγκεκριμένα:

Στα κεφάλαια 1 περιγράφεται η έννοια των ρευμάτων δεδομένων, όπου δίνεται βάση στην αδυναμία των συστημάτων διαχείρισης βάσεων δεδομένων στην αποδοτική επεξεργασία δυναμικά μεταβαλλόμενων δεδομένων. Παρουσιάζονται τα χαρακτηριστικά των ρευμάτων δεδομένων καθώς και οι επεκτάσεις που έγιναν στις συμβατικές βάσεις δεδομένων.

Στο κεφάλαιο 2 περιγράφεται η έννοια των κινούμενων αντικειμένων, σκιαγραφείται η ιδιαίτερη φύση τους καθώς και αναλύονται τα χαρακτηριστικά τους.

Στο κεφάλαιο 3 περιγράφονται τα προβλήματα απόρριψης φόρτου σε ρεύματα δεδομένων και σε κινούμενα αντικείμενα. Στην πρώτη κατηγορία έχουμε περίπλοκα δίκτυα τελεστών που ο καθένας επιτελεί κάποιο σκοπό. Στην περίπτωση των κινούμενων αντικειμένων, που είναι και πιο συναφή με την παρούσα εργασία, έχουμε κινητά που ενημερώνουν τη θέση τους και επειδή δεν είναι εφικτή η επεξεργασία και τήρηση όλων των θέσεων που στέλνουν τα αντικείμενα, πραγματοποιείται απόρριψη φόρτου με σκοπό την έγκαιρη απάντηση των τρέχοντων ερωτημάτων που έχουν τίθενται, με την μικρότερη δυνατή έκπτωση στην ακρίβεια των αποτελεσμάτων. Πάνω σε αυτά τα θέματα περιγράφονται διαφορετικές ερευνητικές προσεγγίσεις.

Στο κεφάλαιο 4 διατυπώνεται το πρόβλημα που στοχεύει να λύσει η εργασία, όπου αναλύονται τα μοντέλα των κινούμενων αντικειμένων και των ερωτημάτων και η μορφή των απαντήσεων. Ακόμα, παρουσιάζεται ο πρωτότυπος αλγόριθμος για την απόρριψη φόρτου από ρεύματα τροχιάς κινούμενων αντικειμένων. Ο αλγόριθμος δίνει προσεγγιστικές απαντήσεις, καθώς εξετάζει μέρος της αρχικής εισόδου, για να απαντά online τα ερωτήματα, με κάποια έκπτωση στην ακρίβεια των απαντήσεων.

Στο κεφάλαιο 5 αξιολογείται πειραματικά ο αλγόριθμος πάνω σε συνθετικά δεδομένα και σχολιάζονται διεξοδικά τα αποτελέσματα.

Τέλος, στο κεφάλαιο 6 εκτίθενται τα γενικά συμπεράσματα καθώς και πιθανές μελλοντικές προοπτικές της εργασίας.

Κεφάλαιο 1

Ρεύματα κινούμενων αντικειμένων

1.1 Εισαγωγή

Στις μέρες μας, η εξέλιξη των ασύρματων δικτύων και του Διαδικτύου έχουν καταστήσει δυνατή την ανάπτυξη εφαρμογών, οι οποίες παρέχουν υπηρεσίες σε πραγματικό χρόνο σε χρήστες που μεταβάλλουν την θέση τους συνεχώς. Αυτή η κατηγορία εφαρμογών έχουν υψηλές απαιτήσεις σε ταχύτητα και αποκρισιμότητα, μιας και επεξεργάζονται μεγάλο όγκο δεδομένων, τα οποία δεν είναι στατικά αλλά δυναμικά και ταχύτατα μεταβαλλόμενα. Τα υπάρχοντα Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ) έχουν εγγενή αδυναμία να διαχειριστούν τέτοιον όγκο δεδομένων δυναμικής φύσης και να δίνουν απαντήσεις αξιόπιστα και σε πραγματικό χρόνο. Για αυτό τον λόγο, έχει προταθεί το μοντέλο των ρευμάτων δεδομένων (*data streams*), το οποίο επιχειρεί να διαχειριστεί τέτοιου είδους μεταβαλλόμενες, εφήμερες και ενδεχομένως ελλιπείς πληροφορίες.

Οι εφαρμογές παρακολούθησης στις οποίες εμφανίζεται η ανάγκη για δυναμική επεξεργασία ρευμάτων δεδομένων αναπτύχθηκαν βασιζόμενες στην εξάπλωση και εξέλιξη των δικτύων. Οι πιο χαρακτηριστικές είναι:

- *Δίκτυα αισθητήρων*, που αποστέλουν τις μετρήσεις τους με την μορφή ρευμάτων δεδομένων στους δέκτες για ανάλυση και επεξεργασία.
- *Χρηματιστηριακές ή οικονομικές*, οι οποίες παρακολουθούν πολύ μεγάλο όγκο συναλλαγών και άλλων παραμέτρων όπως τιμές μετοχών και δεικτών.
- *Ασφάλειας του διαδικτύου*, Εξετάζουν τα πακέτα που ανταλλάσσονται με σκοπό την εκτίμηση συμφόρησης και τον εντοπισμό κακόβουλης επίθεσης.
- *Στρατιωτικού περιεχομένου*.
- *Παρακολούθησης κινούμενων αντικειμένων*, λ.χ. οχήματα σε οδικά δίκτυα, πλοία στην θάλασσα, άγρια ζώα σε οικότοπους κλπ.

Στο κεφάλαιο αυτό θα μελετηθεί η φύση των ρευμάτων δεδομένων και θα περιγραφούν οι δυνατότητες που πρέπει να έχει ένα ολοκληρωμένο Σύστημα Διαχείρισης Ρευμάτων Δεδομένων (ΣΔΡΔ). Επιπρόσθετα, θα περιγραφούν τα χρονικά παράθυρα, η προσαρμοστικότητα που πρέπει να διαθέτει κάθε τέτοιο ΣΔΡΔ και η βελτιστοποίηση των ερωτημάτων.

1.2 Ρεύματα δεδομένων

Ρεύμα δεδομένων (*data stream*) λέγεται η συνεχής ροή πληροφορίας που παράγεται από μία ή και περισσότερες πηγές, η οποία διοχετεύεται μέσω ενός καναλιού επικοινωνίας σε κάποιο δέκτη [31].

Ο τύπος της πληροφορίας εξαρτάται από την εκάστοτε εφαρμογή. Είναι δυνατό να αποτελεί την παρατήρηση ενός φυσικού φαινομένου (ταχύτητα κίνησης κλπ) ή δευτερογενής πληροφορία όπως πακέτα δεδομένων σε κάποιο δίκτυο υπολογιστών ή τιμές μετοχών στο χρηματιστήριο. Η πληροφορία μεταφέρεται με μορφή πλειάδων (*tuples*) με σκοπό να μπορεί να αξιοποιηθεί από τα σύγχρονα ψηφιακά συστήματα.

Για παράδειγμα, το σύστημα αισθητήρων θερμοκρασίας ενός σύγχρονου αυτοκινήτου περιλαμβάνει κάποιες υπολογιστικές μονάδες που επιτελούν κάποια επεξεργασία. Η επεξεργαστική μονάδα λαμβάνει τις τιμές των μετρήσεων της θερμοκρασίας και εκτελεί ενέργειες όπως άναμμα λυχνιών για ειδοποίηση του οδηγού σε περίπτωση υπερθέρμανσης κάποιου μέρους του οχήματος. Οι αισθητήρες μπορούν να στέλνουν τιμές είτε σε ταχτά χρονικά διαστήματα (λ.χ. κάθε 10 δευτερόλεπτα) είτε όταν η αλλαγή θερμοκρασίας που λαμβάνουν ξεπερνάει κάποιο κατώφλι (λ.χ. 0.8 C°). Ουσιαστικά η πληροφορία που αποστέλει καθέναν από τους αισθητήρες του αυτοκινήτου είναι της μορφής $\langle \text{χρόνος καταγραφής, κωδικός αισθητήρα, μέτρηση} \rangle$. Ο κεντρικός επεξεργαστής λαμβάνει συνεχώς τέτοιες πλειάδες από τους διάφορους αισθητήρες. Επί του πρακτέου, ο εγκέφαλος του αυτοκινήτου δέχεται ένα ρεύμα δεδομένων.

1.2.1 Μοντέλο ρευμάτων δεδομένων

Γενικά, ένα ρεύμα δεδομένων αποτελείται από πλειάδες τιμών που στις περισσότερες των περιπτώσεων περιέχουν την χρονική στιγμή καταγραφής της πλειάδας, το μοναδικό αναγνωριστικό του αποστολέα (διότι ενδέχεται να είναι περισσότεροι από ένας) και την τιμή ή τις τιμές που μετρώνται. Οι πλειάδες καταφθάνουν στον δέκτη με ακανόνιστη σειρά ενδεχομένως διαφορετική από την χρονική σειρά καταγραφής τους στην πηγή λόγω καθυστερήσεων εντός του δικτύου.

Φορμαλιστικά, η μορφή που έχει κάθε πλειάδα ενός ρεύματος δεδομένων είναι:

$$\langle t_i, s_i, m \rangle$$

όπου:

- t_i : η χρονική στιγμή καταγραφής της πληροφορίας (*timestamp*),
- s_i : το αναγνωριστικό (*identifier*) του αποστολέα,

- m : η μέτρηση που καταγράφεται. Μπορεί να είναι και αυτή μία πλειάδα σε περίπτωση που τα μεγέθη ενδιαφέροντος είναι περισσότερα του ενός. Λ.χ. ένας αισθητήρας μπορεί να μετρά τιμές θερμοκρασίας, υγρασίας, κατεύθυνσης ανέμου κλπ.

1.2.2 Βασικά χαρακτηριστικά ρευμάτων δεδομένων και η συσχέτισή τους με τα 4V των Big Data

Στην εποχή μας όλο και περισσότερα δεδομένα καταγράφονται, αποθηκεύονται και αναλύονται για να λειτουργούν οι τεχνολογικές υπηρεσίες στις οποίες στηρίζεται ολόκληρη η ανθρωπότητα καθημερινά. Τα λεγόμενα “μεγάλα δεδομένα” (*big data*) περιλαμβάνουν πληροφορία από διάφορες πηγές όπως συναλλαγές, κοινωνικά δίκτυα, δίκτυα αισθητήρων και smartphones. Οι εταιρίες μπορούν να αξιοποιήσουν αυτά τα δεδομένα έτσι ώστε να προσαρμόσουν τα προϊόντα και τις υπηρεσίες τους πιο αποτελεσματικά στις ανάγκες των καταναλωτών. Ένας ηγέτης του χώρου, η IBM, αποδίδει στα “μεγάλα δεδομένα” τα εξής χαρακτηριστικά: τον όγκο (*volume*), την ταχύτητα (*velocity*), την ποικιλία (*variety*) και την αβεβαιότητα (*volume*).

1. Μεγάλος όγκος δεδομένων (*volume*)

Οι πλειάδες που περιγράψαμε παραπάνω στέλνονται συνεχώς από πολλαπλές πηγές με αποτέλεσμα η ποσότητα πληροφορίας που καταφτάνει προς επεξεργασία να είναι τεράστια. Και ως εκ τούτου, πολλές φορές η επεξεργασία δεν είναι εφικτό να πραγματοποιηθεί ταχύτητα, με αποτέλεσμα να πρέπει να βρεθούν άλλες τεχνικές. Είναι ευρέως διαδεδομένη αλλά και αναγκαία η χρήση γρήγορων ευρετηρίων ή τεχνικών απόρριψης εισερχόμενης πληροφορίας, μειώνοντας όμως την ακρίβεια των απαντήσεων.

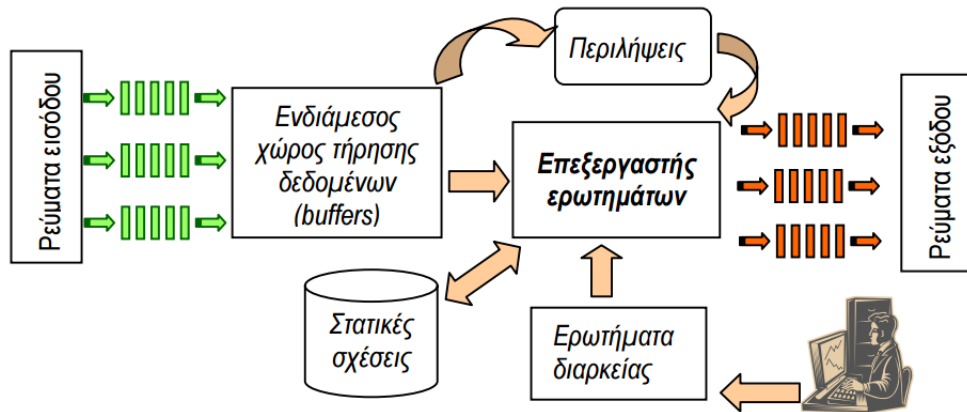
2. Ακαθόριστος ρυθμός άφιξης δεδομένων (*velocity*)

Με γνώμονα το γεγονός πως οι πηγές στέλνουν πλειάδες σε ακαθόριστους και αυθαίρετους χρόνους, η ροή πληροφορίας στον δέκτη δεν είναι εφικτό να ελεγχθεί. Μπορεί κάποιες χρονικές στιγμές ο δέκτης να μην λαμβάνει καθόλου πληροφορία από κάποιες πηγές, ενώ κάποιες άλλες χρονικές στιγμές να λαμβάνει από πάρα πολλές πηγές ταυτόχρονα. Αυτό έχει ως αποτέλεσμα να προξενούνται απότομες αυξομειώσεις στο ρυθμό άφιξης δεδομένων.

Ένα σύστημα που διαχειρίζεται ρεύματα δεδομένων θα πρέπει να είναι σε θέση να παραλάβει και να επεξεργαστεί αμέσως όσες πλειάδες φτάνουν, ασχέτως του ρυθμού άφιξης των στοιχείων (*arrival rate*).

Τα πακέτα που αποστέλνουν οι πηγές μπορεί να φτάσουν στον δέκτη ανά πάσα στιγμή. Κάθε πηγή μπορεί να αποστέλλει δεδομένα με τον δικό της ρυθμό ο οποίος ενδέχεται να είναι κυμαινόμενος. Από την άλλη μπορεί μία πηγή να έχει αποστείλει το πακέτο της αλλά επειδή αυτό μεταφέρεται στον δέκτη μέσω δικτύου (συνήθως ασύρματου), να καθυστερήσει λόγω μεγάλου φόρτου στο δίκτυο. Ο φόρτος μπορεί να επηρεάσει σε μεγάλο βαθμό την δρομολόγηση της πληροφορίας από τις πηγές στον δέκτη.

Ο δέκτης τους ρεύματος πρέπει να γνωρίζει από πριν πως κάποια πακέτα ενδέχεται να καθυστερήσουν ή και να μην φτάσουν ποτέ στον προορισμό τους, με αντίκτυπο να



Σχήμα 1.1: Ένα αφαιρετικό πλαίσιο λειτουργίας για Συστήματα Διαχείρισης Ρευμάτων Δεδομένων [31]

υπάρχει ελλιπής πληροφορία. Επομένως κάθε στιγμή πρέπει να αναμένει για πληροφορία αλλά και να διαχειρίζεται στιγμές στις οποίες δεν φτάνει καθόλου πληροφορία.

3. Αβεβαιότητα πληροφορίας (*variety, veracity*)

Όλες οι πηγές δυνητικά στέλνουν πληροφορίες στον δέκτη. Η πληροφορία που μεταφέρεται όμως δεν είναι ίδια ακριβώς με την πραγματική πληροφορία, όπως ιδανικά θα έπρεπε. Αυτό οφείλεται σε θόρυβο στον διάυλο επικοινωνίας πηγών-δέκτη καθώς και σε σφάλματα οργάνων μέτρησης. Λ.χ. στις εφαρμογές που διαχειρίζονται χωροχρονικά δεδομένα οι μετρήσεις θέσεις που δίνει το GPS του κάθε κινητού δεν είναι πάντα ίδιες με τις πραγματικές αλλά εμπεριέχουν σφάλματα. Ακόμα τα δεδομένα εισόδου είναι αποθηκευμένα σε διάφορα format (XML, CSV...) και ακόμα τα δεδομένα καταφθάνουν από διάφορες πλατφόρμες (κινητά, συσκευές RFID, αισθητήρες...)

Ο δέκτης επομένως οφείλει να λαμβάνει υπόψη του την αβεβαιότητα που χαρακτηρίζει την πληροφορία την οποία παίρνει και όταν είναι εφικτό ή αναγκαίο να τροποποιεί τα συμπεράσματά του βάσει αυτής.

1.3 Συστήματα διαχείρισης ρευμάτων δεδομένων

1.3.1 Προδιαγραφές συστήματος

Τα ρεύματα δεδομένων διαφέρουν από τα παραδοσιακά ΣΔΒΔ σε αρκετά σημεία. Αυτό οφείλεται στις ειδοποιές διαφορές που έχει το μοντέλο ρευμάτων δεδομένων με τα κλασσικά μοντέλα των σχεσιακών ΣΔΒΔ. Οι κυριότερες δυσκολίες που προκύπτουν έχουν να κάνουν με την μετάδοση ολόκληρης της πληροφορίας στο σύστημα, τον υπολογισμό απαιτητικών συναρτήσεων της εισόδου με ρυθμό όμοιο με τον ρυθμό άφιξης της πληροφορίας και την αποθήκευση της εισόδου σε προσωρινές δομές ή περιστασιακά στην εξωτερική μνήμη.

Οι προδιαγραφές ενός Συστήματος Διαχείρισης Ρευμάτων Δεδομένων (ΣΔΡΔ) είναι οι εξής:

- Τα εισερχόμενα στοιχεία ή πλειάδες καταφθάνουν σε **πραγματικό χρόνο (online)**.
- Τα ρεύματα έχουν δυνητικά **απεριόριστο μέγεθος (unbounded size)**.
- Η επεξεργασία οφείλει να γίνεται στην **κύρια μνήμη** για λόγους ταχύτητας αποκρίσεων στα ερωτήματα των χρηστών.
- Το σύστημα θα πρέπει να εξετάζει κάθε στοιχείο που εισέρχεται μόνο μία φορά (**one pass algorithm**).
- Το σύστημα πρέπει να είναι σε θέση να αντιμετωπίσει **τυχόν διακυμάνσεις στο ρυθμό άφιξης της πληροφορίας (arrival rate)**.
- Η χρονική διάταξη των στοιχείων σε ένα ή μεταξύ περισσότερων ρευμάτων δεδομένων, δεν είναι πάντα σωστή, λόγω καθυστερήσεων στη μετάδοση, απωλειών κλπ.
- Το σύστημα οφείλει να ισοσταθμίσει τους πόρους του συστήματος σε χρόνο επεξεργασίας και μνήμη με την ακρίβεια των αποτελεσμάτων που δίνει στα ερωτήματα.
- Η κύρια μνήμη στην οποία γίνεται η επεξεργασία έχει πεπερασμένο μέγεθος.
- Προτιμάται η απάντηση των ερωτημάτων που τίθενται στο σύστημα σε πραγματικό χρόνο, ακόμα και αν αυτό έχει κόστος στην ακρίβεια της.

1.3.2 Κυριότερα συστήματα διαχείρισης ρευμάτων δεδομένων

Η φύση των ρευμάτων δεδομένων απαιτεί ειδική μεταχείριση που δεν μπορούν να την παρέχουν τα κλασικά συστήματα διαχείρισης βάσεων δεδομένων. Έτσι αναπτύχθηκαν πρωτότυπα συστήματα διαχείρισης ρευμάτων δεδομένων τα οποία ταιριάζουν και δουλεύουν αποδοτικά με το μοντέλο ρευμάτων δεδομένων. Τα κυριότερα από αυτά είναι:

- AURORA [2]
- STREAM [3]
- TelegraphCQ [6]
- NiagaraCQ [7]
- Gigascope [8]

Ορισμένα από αυτά αποτελούν συστήματα γενικού σκοπού (AURORA, STREAM), ενώ άλλα απευθύνονται κατά κύριο λόγο σε δικτυακές εφαρμογές και δίκτυα αισθητήρων (TelegraphCQ, NiagaraCQ, Gigascope). Αυτά τα συστήματα χρησιμοποιούν ειδικούς τελεστές που είναι ικανοί να διαχειριστούν καλύτερα και προτείνονται επεκτάσεις στο καθιερωμένο πρότυπο

της SQL οι οποίες υποστηρίζουν λειτουργίες παραθύρων (*windows*). Για παράδειγμα, η CQL (Continuous Query Language), αποτελεί υπερσύνολο της γλώσσας SQL και σχεδιάστηκε στα πλαίσια του STREAM που αναπτύχθηκε στο Stanford, η οποία προσθέτει εξειδικευμένες δομές για υποστήριξη παραθύρων και δειγματοληψίας.

Το TelegraphCQ εστιάζει στην ευελιξία και προσαρμοστικότητα εκτέλεσης ερωτημάτων διαρκείας εισάγοντας τον μηχανισμό Eddy. Απευθύνεται σε έντονα και ασταθή περιβάλλοντα πληροφορίας. Κατά την ανάπτυξη του STREAM ιδιαίτερη βάση δόθηκε στην αποτελεσματική εκμετάλλευση περιορισμένης μνήμης κατά την επεξεργασία ερωτημάτων. Εξετάζει τρόπους παραγωγής προσεγγιστικών αποτελεσμάτων και επιχειρεί να περιορίσει τις απαιτήσεις σε μνήμη. Τέλος, το AURORA επικεντρώνεται στην προσπάθεια παροχής στους χρήστες της δυνατότητας να επιλέγουν δυναμικά την ποιότητα υπηρεσιών που επιθυμούν (Quality of Service), για την εκτέλεση ερωτημάτων και στη συνέχεια να χρησιμοποιούν αυτά τα μέτρα ποιότητας των απαντήσεων για την κατανομή φόρτου εργασίας.

1.4 Ερωτήματα σε ρεύματα δεδομένων

Τα ερωτήματα που υποβάλλονται σε ένα σύστημα διαχείρισης ρευμάτων δεδομένων (ΣΔΡΔ) μπορούν να διακριθούν σε δύο βασικές κατηγορίες:

- *Ερωτήματα στιγμοτύπου* (snapshot or one-time queries), τα οποία εκτελούνται σε τακτά ή μη χρονικά διαστήματα και αφορούν το τρέχον στιγμιότυπο της πληροφορίας. Εξετάζουν δηλαδή την κατάσταση που έχει ένα σύστημα μόνο τη δεδομένη χρονική στιγμή.
- *Ερωτήματα διαρκείας* (continuous queries), τα οποία εκτελούνται συνεχώς από την στιγμή που τίθενται στο σύστημα, και η απάντηση ανανεώνεται για κάθε νέο στοιχείων εισερχόμενων ρευμάτων.

Από τις δύο παραπάνω κατηγορίες τα ερωτήματα διαρκείας είναι αυτά που παρουσιάζουν το μεγαλύτερο ενδιαφέρον όταν υποβάλλονται σε ρεύματα δεδομένων. Τα ερωτήματα στιγμοτύπου δεν διαφέρουν ουσιαστικά από τα ερωτήματα που υποβάλλονται στις σχεσιακές βάσεις δεδομένων. Στα ερωτήματα διαρκείας η απάντηση παράγεται συνεχώς στο χρόνο και αφορά πάντα την πληροφορία που έχει καταφθάσει έως και την τρέχουσα χρονική στιγμή.

Μία δεύτερη κατάταξη των ερωτημάτων γίνεται με βάση το χρόνο υποβολής του ερωτήματος και εξετάζεται αν το ερώτημα διατυπώθηκε και υποβλήθηκε πριν ή μετά από την έναρξη της άφιξης των δεδομένων. Οι κατηγορίες εδώ είναι:

- *Προκαθορισμένα ερωτήματα* (predefined queries), τα οποία είναι γνωστά στο σύστημα πριν ξεκινήσει η άφιξη της πληροφορίας προς επεξεργασία.
- *Μη προβλέψιμα ή περιστασιακά ερωτήματα* (adhoc queries), υποβάλλονται από τον χρήστη ενώ έχουν εισρεύσει ήδη στοιχεία στο σύστημα.

Η διαφοροποίηση των ερωτημάτων με βάση το πότε τέθηκαν στο σύστημα γίνεται γιατί τα προκαθορισμένα ερωτήματα είναι γνωστά εκ των προτέρων και το σύστημα μπορεί αφενός να επιλέξει κατάλληλες δομές για να ανταποκριθεί αποδοτικά και γρήγορα στα ερωτήματα αυτά και αφετέρου να ισοσταθμίσει την κατανάλωση πόρων (επεξεργαστικός χρόνος, μνήμη) με την ακρίβεια στις απαντήσεις ώστε να ανταποκρίνεται στις απαιτήσεις της εκάστοτε εφαρμογής. Από την άλλη πλευρά, τα ερωτήματα διαρκείας δεν είναι εκ των προτέρων γνωστά στο σύστημα και συνεπώς περιπλέκουν αρκετά την επεξεργασία ερωτημάτων, γιατί δεν μπορούν να συμπεριληφθούν στην κατάστρωση και βελτιστοποίηση του πλάνου εκτέλεσης ερωτημάτων (*query optimization*). Ένα ακόμα πρόβλημα που υπάρχει στα ερωτήματα διαρκείας είναι πως ενδέχεται να αναφέρονται σε δεδομένα που έχουν ήδη παρέλθει από το σύστημα και η ανάγκη τους είναι εξαιρετικά δύσκολη. Αυτός ο σκόπελος προσπερνάται είτε με διαβεβαίωση πως το ερώτημα εκτελείται στα δεδομένα που θα καταφθάνουν μετά την υποβολή του, είτε με την χρήση δομών σύνοψης (*summaries, synopses*) που συγκροτούν επιλεκτικά πληροφορία για παρωχημένα δεδομένα.

1.5 Παράθυρα

1.5.1 Εισαγωγή στα παράθυρα

Ένα σύστημα διαχείρισης ρευμάτων δεδομένων δεν μπορεί εκ των προτέρων να προβλέψει πόσες πλειάδες πληροφορίας θα καταφθάνουν σε αυτό και κατ'επέκταση δεν είναι εφικτό να γνωρίζει τον αποθηκευτικό χώρο που θα χρειαστεί η καταγραφή όλου του ρεύματος ή αν διαθέτει τον απαραίτητο χώρο για να υποδεχθεί όλη την εισερχόμενη πληροφορία. Για το λόγο αυτό χρησιμοποιούνται τα παράθυρα (*windows*), τα οποία απομονώνουν ένα πεπερασμένο πλήθος στοιχείων από το απειράριθμο σύνολο που λαμβάνεται από ένα ρεύμα δεδομένων και εκτελείται η επεξεργασία του εκάστοτε ερωτήματος μόνο σε αυτό. Συνήθως, επιλέγονται τα πιο πρόσφατα στοιχεία αντί των παλαιότερων, μιας και οι περισσότερες εφαρμογές ενδιαφέρονται ως επί το πλείστον για την πιο πρόσφατη πληροφορία που είναι διαθέσιμη (λ.χ. δεδομένα θερμοκρασίας κατά την τελευταία ώρα).

Υπάρχουν διάφορες κατηγορίες παραθύρων, ωστόσο οι πιο ενδιαφέρουσες είναι οι εξής:

- Παράθυρα οροσήμου (*landmark windows*), τα οποία ορίζουν σαν αρχή τους κάποια παρελθοντική χρονική στιγμή, αλλά το τέλος τους μεταβάλλεται καθώς εξελίσσεται ο χρόνος. Αυτό έχει σαν αποτέλεσμα το μέγεθος του παραθύρου να αυξάνεται συνεχώς.
- Κυλιόμενα παράθυρα (*sliding windows*), τα οποία έχουν μεταβαλλόμενη αρχή και τέλος που μετακινούνται με το πέρασμα του χρόνου. Το τέλος του παραθύρου βρίσκεται στην πιο πρόσφατη χρονική στιγμή, ενώ η αρχή στην χρονική στιγμή του παρελθόντος, που απέχει από το παρόν έως το τέλος όσο είναι το εύρος του παραθύρου. Κάθε κυλιόμενο παράθυρο διαθέτει ένα εύρος (*range*) ω και ένα βήμα (*slide*) β .

1.5.2 Ιδιότητες χρονικών παραθύρων

Τα παράθυρα σαν μέθοδος προσέγγισης για την επεξεργασία και διαχείριση του απειράριθμου συνόλου δεδομένων που έρχονται από τα ρεύματα, έχουν να επιδείξουν πολλές χρήσιμες ιδιότητες:

- Είναι ντετερμινιστικά, σε αντίθεση με άλλες προσεγγίσεις όπως η τυχαία δειγματοληψία, όπου τα δεδομένα που συνεισφέρουν στην απάντηση διαλέγονται με τυχαίο τρόπο, καταλήγοντας πιθανόν σε μη ρεαλιστικά αποτελέσματα.
- Έχουν σαφή και κατανοητό ορισμό που δεν εγείρει αμφιβολίες ως προς τα δεδομένα που εμπλέκει η απάντηση του ερωτήματος.
- Η σημαντικότερη ιδιότητά τους είναι πως εστιάζουν στα πιο πρόσφατα δεδομένα. Εκτός λοιπόν από μία τεχνική μείωσης του όγκου δεδομένων με σκοπό την καλύτερη απόδοση του συστήματος, τα παράθυρα αποτελούν και ουσιαστικό μέρος της σημασιολογίας του ερωτήματος.

1.6 Αλγόριθμοι για ρεύματα δεδομένων

Όπως προαναφέρθηκε, για την διαχείριση ρευμάτων δεδομένων δεν χρησιμοποιούνται συμβατικά ΣΔΒΔ, καθώς δεν μπορούν να ανταπεξέλθουν στις επεξεργαστικές ανάγκες που εισάγει η δυναμική φύση των ρευμάτων δεδομένων. Έτσι χρησιμοποιούνται ΣΔΡΔ τα οποία απαιτούν νέους αλγορίθμους που θα πρέπει να καλύπτουν τα εξής βασικά χαρακτηριστικά:

- Να γίνεται η ελάχιστη δυνατή κατανάλωση μνήμης, μιας και κάθε φορά ένα μικρό κομμάτι του ρεύματος μπορεί να κρατηθεί στο σύστημα.
- Να “σαρώνουν” τα δεδομένα μόνο μία φορά ,με ένα πέρασμα, και να μην τα προσπελαύνουν πολλές φορές.
- Η ενημέρωση των αποτελεσμάτων να μπορεί να γίνεται προοδευτικά (*incrementally*), με σκοπό να έχουν καλύτερες επιδόσεις και να είναι σε θέση να προσφέρουν στους χρήστες ενδιάμεσα αποτελέσματα.
- Να χαρακτηρίζονται από προσαρμοστικότητα (*adaptivity*). Ανάλογα με το φόρτο, οι αλγόριθμοι πρέπει να καθιστούν το σύστημα ικανό να δώσει απαντήσεις έγκαιρα και ορθά με το μικρότερο δυνατό σφάλμα. Η αποβολή φόρτου (*load shedding*). Πραγματοποιείται σε οποιοδήποτε στάδιο της επεξεργασίας. Αν για παράδειγμα, ένα ρεύμα που προέρχεται από κάποια πηγή τροφοδοτεί έναν δέκτη οποίος δεν προλαβαίνει να το επεξεργαστεί (στο χρόνο που του διατίθεται), ο κόμβος αυτός μπορεί να εφαρμόσει δειγματοληψία (με τυχαίο τρόπο ή σημασιολογικά) αποβάλλοντας έτσι επεξεργαστικό και αποθηκευτικό φορτίο.

1.7 Ανάγκη για απεριόριστη μνήμη

Τα ρεύματα δεδομένων έχουν δυνητικά απεριόριστο μέγεθος. Κατά συνέπεια, το μέγεθος της απαραίτητης μνήμης για την απάντηση ερωτημάτων μπορεί δυνητικά να είναι και αυτό απεριόριστο. Οι αλγόριθμοι που έχουν αναπτυχθεί για επεξεργασία ερωτημάτων με χρήση εξωτερικής μνήμης (λ.χ. σκληρός δίσκος) δεν είναι συμβατοί με το μοντέλο ρευμάτων δεδομένων, αφού δεν είναι ικανοί να παράγουν απαντήσεις στα ερωτήματα διαρκείας σε πραγματικό χρόνο. Η πρόσβαση επομένως στον δίσκο μπορεί να επιδεινώσει δραματικά την απόδοση του συστήματος και για το λόγο αυτό κρίνεται απαγορευτική (*touching the disk kills performance*).

Μία σχετική έρευνα για τα πιο κοινά ερωτήματα, όπως συναθροίσεις (*aggregates*) και συνδέσεις (*joins*) έχει δείξει ότι το μέγεθος της απαιτούμενης μνήμης για την επεξεργασία του ερωτήματος δεν μπορεί να φραχθεί, αν το μέγεθος του ρεύματος δεν είναι εκ των προτέρων γνωστό. Διαισθητικά, αν δεν υπάρχουν περιορισμοί του πεδίου τιμών σε μία σύνδεση, κάθε νέα τιμή μπορεί να συνδέεται δυνητικά με κάθε άλλη του παρελθόντος. Συμπερασματικά, μπορεί να ειπωθεί ότι χωρίς περιορισμούς στο σύνολο των τιμών και χωρίς εκτίμηση του μεγέθους του ρεύματος, δεν είναι δυνατός ο προσδιορισμός της απαιτούμενης μνήμης για την εκτέλεση ενός ερωτήματος.

1.8 Προσεγγιστικές απαντήσεις σε ερωτήματα

Στην ιδανική περίπτωση, ένα σύστημα διαχείρισης ρευμάτων δεδομένων θα έδινε έγκαιρες και ακριβείς απαντήσεις σε κάθε ενεργό ερώτημα. Στην πραγματικότητα όμως, κάτι τέτοιο δεν είναι εφικτό να συμβεί εξαιτίας της ίδιας της φύσης των ρευμάτων δεδομένων:

- Μεγάλες ποσότητες δεδομένων που εισέρχονται στο σύστημα με ασταθή και γρήγορα μεταβαλλόμενο ρυθμό.
- Υπάρχουν περιορισμένοι πόροι σε επεξεργαστική ισχύ και μνήμη για την επεξεργασία και αποτίμηση ερωτημάτων.
- Τα ερωτήματα που υποβάλλονται στο σύστημα μπορεί να είναι πολλά, περίπλοκά και με μεγάλες απαιτήσεις ως προς την επεξεργασία τους.

Επομένως, τα ΣΔΡΔ αντιμετωπίζουν τα εισερχόμενα σε αυτά δεδομένα με εντελώς διαφορετικό τρόπο απ' ό,τι τα ΣΔΒΔ. Δεν επικεντρώνονται στην ακρίβεια των απαντήσεων αλλά απεναντίας επιχειρούν να παράγουν προσεγγιστικές απαντήσεις όσο το δυνατόν ταχύτερα και με την μικρότερη απόκλιση από τις ακριβείς απαντήσεις. Εφόσον διαχειρίζονται βέλτιστα τους διαθέσιμους πόρους τους, βασική επιδίωξη είναι να εντοπίζουν τις πιο ενδιαφέρουσες τάσεις στα εισερχόμενα δεδομένα.

1.9 Συνόψεις

Στο πλαίσιο των προσεγγιστικών απαντήσεων για τα ΣΔΡΔ, έχουν προταθεί διάφορες τεχνικές για τη δημιουργία συνόψεων και την αποδοτική συμπίεση των εισερχόμενων δεδομένων. Πρωταρχικός στόχος είναι να μειωθεί ο όγκος της πληροφορίας προς επεξεργασία. Κατά δεύτερον, οι συνόψεις έχουν σκοπό να διοχετεύσουν την πληροφορία σε κατάλληλους κόμβους για μόνιμη αποθήκευση ως μελλοντική αναφορά σε ιστορικά δεδομένα.

1.9.1 Μαζική επεξεργασία

Η μέθοδος *μαζικής επεξεργασίας* (batch processing), δίνει ακριβείς απαντήσεις με αντίκτυπο το κόστος καθυστέρησης. Το σύστημα δεν επεξεργάζεται κάθε πλειάδα ξεχωριστά μόλις εισέρχεται στο σύστημα, αλλά αναμένει να σχηματιστεί ένα σύνολο από εισερχόμενες πλειάδες και έπειτα τις επεξεργάζεται ομαδικά ώστε να γίνει πιο γρήγορα η εκτέλεση των ερωτημάτων. Χρησιμοποιείται σε περιπτώσεις όπου η άφιξη των δεδομένων εμφανίζει απρόσμενη και καταιγιστική συμπεριφορά. Έτσι η απάντηση είναι προσεγγιστική, μιας και δεν ανανεώνεται τακτικά.

1.9.2 Περιλήψεις ή συνόψεις δεδομένων

Οι συνόψεις επιδιώκουν να αναπαραστήσουν περιληπτικά την εισερχόμενη πληροφορία, με κόστος την μείωση της ακρίβειας του αποτελέσματος αλλά προς όφελος της ταχύτητας επεξεργασίας. Το μέγεθος της σύνοψης θα πρέπει να είναι σημαντικά μικρότερο από το αρχικό μέγεθος της πληροφορίας που αναπαριστά. Θα πρέπει να μπορεί να υπολογίζεται πολύ γρήγορα καθώς διαβάζονται τα εισερχόμενα δεδομένα (μόνο μία φορά) και να ακολουθεί το ρυθμό άφιξής τους.

Παρά την ευελιξία και την βελτίωση της ταχύτητας των υπολογισμών, η χρήση συνόψεων εγείρει αμφιβολίες για το κατά πόσο είναι εφικτό να υπολογιστεί η ακρίβεια των απαντήσεων: ποιες είναι οι πιο κατάλληλες μορφές περίληψης για διάφορες συνθήκες άφιξης ρευμάτων ο συνολικός καταμερισμός μνήμης μεταξύ των διαφόρων τύπων συνόψεων που είναι δυνατό να χρησιμοποιούνται. Αυτός ο καταμερισμός πρέπει να προσαρμόζεται στις αλλαγές που αφορούν τα δεδομένα και τα ερωτήματα που εισέρχονται και τίθενται στο σύστημα αντίστοιχα.

Και οι δύο κατηγορίες που περιγράψαμε παραπάνω, είναι επιθυμητές και είναι πιο γρήγορες συγκριτικά με την εξαντλητική επεξεργασία καθεμιάς πλειάδας που εισέρχεται στο σύστημα από κάθε πηγή. Οι πιο γνωστές κατηγορίες συνόψεων είναι τα σκίτσα (*sketches*), τα κυματίδια (*wavelets*), τα ιστογράμματα (*histograms*) και η δειγματοληψία (*sampling*).

Κεφάλαιο 2

Διαχείριση κινούμενων αντικειμένων

2.1 Εισαγωγή

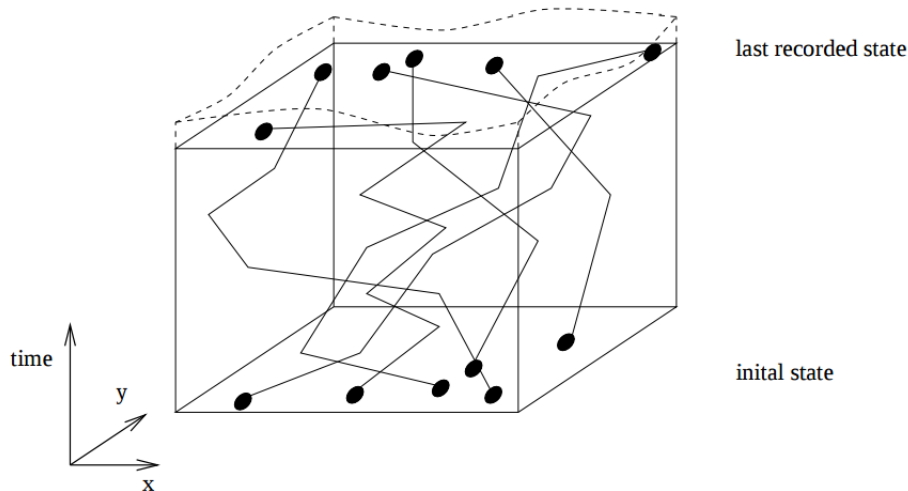
Οι εξελίξεις τόσο στον χώρο των δικτύων όσο και στον χώρο των συστημάτων γεωγραφικού εντοπισμού (GPS) συντέλεσαν στην ανάπτυξη εφαρμογών παρακολούθησης κινούμενων αντικειμένων. Τα κινούμενα αντικείμενα είναι χωρικές οντότητες των οποίων η θέση και η έκταση μπορεί να αλλάζουν συνεχώς με την πάροδο του χρόνου. Την επιστημονική κοινότητα των βάσεων δεδομένων συχνά την έχει απασχολήσει η μοντελοποίηση και επεξεργασία της κίνησης όσον αφορά τη συνεχή μεταβολή της θέσης των αντικειμένων. Η κίνηση είναι χωροχρονικό φαινόμενο, οπότε κατά την ανάλυση και επεξεργασία της θα πρέπει να λαμβάνονται υπόψη τόσο οι χωρικές όσο και οι χρονικές της πτυχές, μιας και είναι άρρηκτα συνδεδεμένες μεταξύ τους.

Αν παρατηρήσει κανείς τον κόσμο τριγύρω θα δει πως τα πάντα κινούνται. Και παρατηρώντας τις κινήσεις διαφόρων κινούμενων αντικειμένων μπορούμε να βρούμε μοτίβα (*patterns*), που αν τα αναλύσουμε και τα κατανοήσουμε, οδηγούμαστε σε σημαντικά συμπεράσματα. Ομάδες κινούμενων αντικειμένων μπορεί να είναι τα οχήματα που κινούνται κατά μήκος οδικών αξόνων, πεζοί σε εμπορικά κέντρα, καρτότσια σε σούπερ μάρκετ, ζώα σε βιότοπο κλπ. Θα πρέπει φυσικά όλα αυτά τα κινούμενα αντικείμενα να είναι εξοπλισμένα με GPS ώστε να μπορούμε ανά πάσα στιγμή να παίρνουμε την θέση τους.

Η κίνηση των αντικειμένων είναι πολύ πιθανό να υπόκειται σε διάφορους περιορισμούς, που είτε έχουν να κάνουν με τον χώρο στον οποίο κινούνται είτε με την φύση τους. Για παράδειγμα τα αυτοκίνητα δεν μπορούν παρά να κινούνται πάνω σε οδικά δίκτυα.

2.2 Χωροχρονικά δεδομένα

Χωροχρονικά δεδομένα είναι μία ειδική κατηγορία δεδομένων η οποία παράγεται από την συστηματική καταγραφή χωροχρονικών φαινομένων. Η κίνηση αποτελεί χωροχρονικό φαινόμενο, αφού στο διάβα του χρόνου, μπορεί να αλλάζει τόσο η θέση όσο και η έκταση ενός



Σχήμα 2.1: Χωροχρονικό χώρο [20]

χωρικού αντικειμένου, το οποίο μπορεί να μοντελοποιείται ως ένα σημείο ή ένα πολύγωνο 2.1.

Ο εντοπισμός της θέσης ενός κινούμενου αντικειμένου είναι δυνατός με την τεχνολογία GPS (*Global Positioning System*). Και η διοχέτευση των παραγόμενων δεδομένων είναι εφικτή χάρη στην ανάπτυξη των ασύρματων τηλεπικοινωνιών. Η μεταβολή της θέσης των αντικειμένων μπορεί να συμβεί είτε με διακριτό τρόπο είτε με συνεχή. Στην παρούσα διπλωματική το ενδιαφέρον επικεντρώνεται στην συνεχή κίνηση, η οποία παρακολουθείται με διακριτό τρόπο. Θεωρούμε λοιπόν ότι με κάποιο ρυθμό, όχι αναγκαστικά σταθερό, ένα κινούμενο αντικείμενο στέλνει στίγμα στον σταθμό κεντρικής παρακολούθησης της κίνησης.

2.3 Τροχιές κινούμενων αντικειμένων

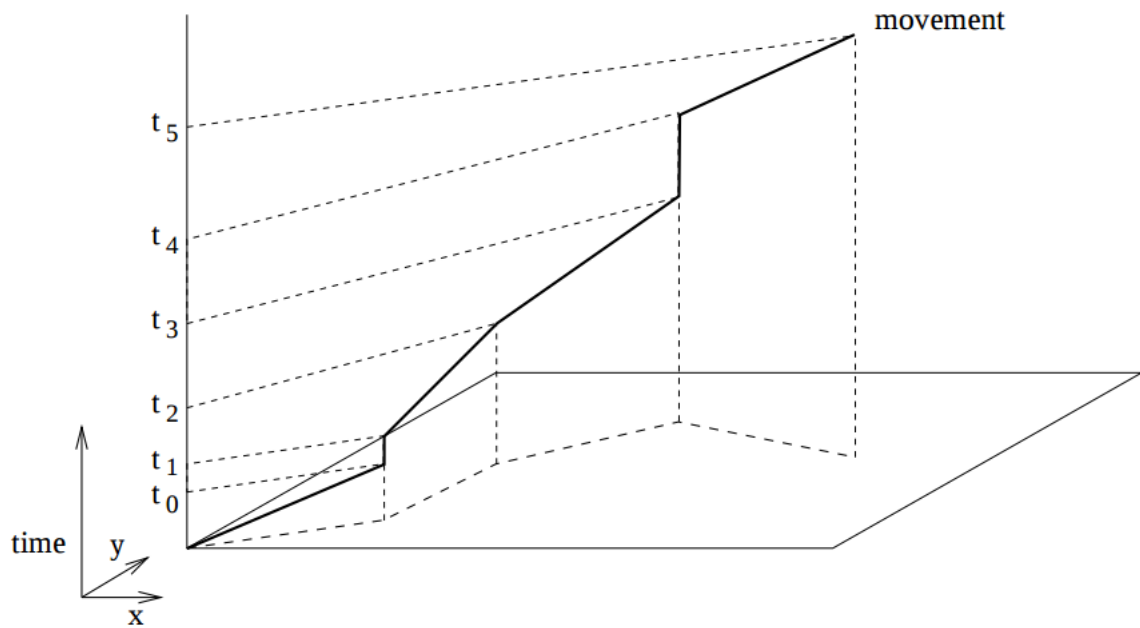
Η κίνηση ενός κινούμενου αντικειμένου στο επίπεδο μπορεί να αναπαρασταθεί με την τροχιά του σε τρισδιάστατο σύστημα αξόνων (Σχήμα 2.2), το οποίο αποτελείται από τις χωρικές συντεταγμένες (x , y) καθώς και από τον άξονα του χρόνου (t).

Μία τροχιά (*trajectory*) μπορεί να μοντελοποιηθεί από μία ακολουθία από πλειάδες της μορφής:

$$\langle \text{Object-id}, \text{Timestamp}, X\text{-coordinate } Y\text{-coordinate} \rangle$$

όπου:

- *Object-id*: το μοναδικό αναγνωριστικό του κινούμενου αντικειμένου
- *Timestamp*: το χρονόσημο που περιγράφει την χρονική στιγμή που αναφέρεται η πλειάδα
- *X-coordinate*: η τετμημένη, η οποία αποτελεί γεωγραφική συντεταγμένη και επί της ουσίας είναι το γεωγραφικό μήκος (*longitude*)



Σχήμα 2.2: Αναπαράσταση τροχιάς κινούμενου αντικειμένου στον τρισδιάστατο χώρο (x , y , t) [20]

- *Y-coordinate*: η τεταγμένη, η οποία αποτελεί γεωγραφική συντεταγμένη και επί της ουσίας είναι το γεωγραφικό πλάτος (*latitude*)

Οι πλειάδες αυτές αποτελούν τα στίγματα που στέλνουν τα αντικείμενα και είναι τα οι ψηφίδες που συνιστούν την τροχιά ενός κινητού.

2.4 Ερωτήματα σε κινούμενα αντικείμενα

2.4.1 Ερωτήματα θέσης

Τα συστήματα διαχείρισης οφείλουν να επεξεργάζονται και να απαντούν διαφόρων ειδών ερωτήματα χρησιμοποιώντας τα δεδομένα που παράγονται από κινούμενα αντικείμενα [21]. Τα ερωτήματα θέσης (*location based queries*) αφορούν την κύρια πληροφορία που καταγράφει το σύστημα. Οι κυριότερες κατηγορίες ερωτημάτων θέσης είναι:

- *Ερωτήματα χωροχρονικού παραθύρου*: Εφαρμόζεται ένα χωρικό και ένα χρονικό παράθυρο συγκεκριμένου εύρους και επιλέγονται τμήματα τροχιών που το τέμνουν.
- *Ερωτήματα εγγύτερου γείτονα*: Εύρεση k -κοντινότερων αντικειμένων με κάποιο αντικείμενο ενδιαφέροντος. Οι γείτονες αυτοί βρίσκονται με χρήση κάποιας μετρικής απόστασης (ευκλείδεια, manhattan, mahalanobis κλπ κλπ) σε αντίστοιχα χρονόσημα.

- *Ερωτήματα κλιμακούμενης απόστασης*: Ίδια με τα ερωτήματα εγγύτερων γειτόνων με την διαφορά ότι δεν αναζητούνται k αντικείμενα, αλλά όλος ο κατάλογος των αντικειμένων με διάταξη βάσει της απόστασης.
- *Ερωτήματα χρονικού τεμαχίου*: Αναζήτηση θέσεων ενός αντικειμένου για συγκεκριμένο χρονικό παράθυρο.

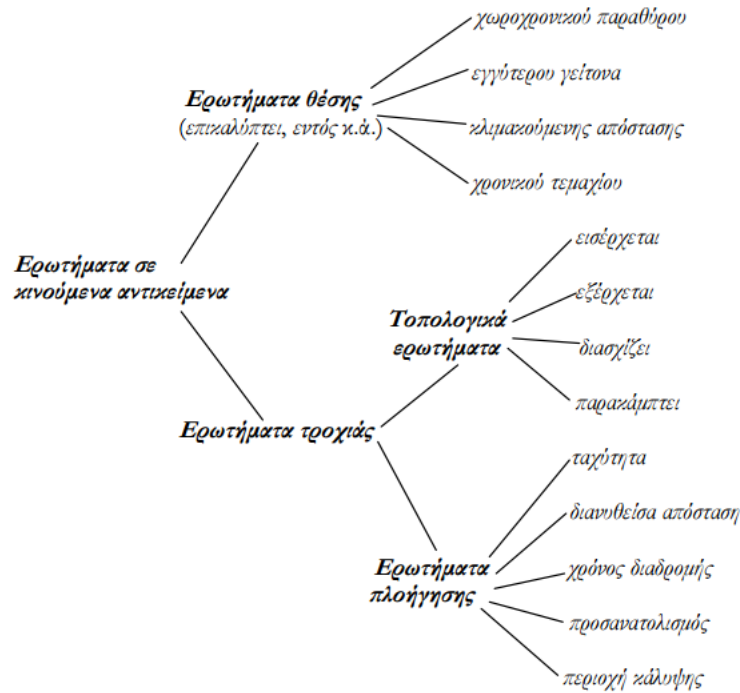
2.4.2 Ερωτήματα τροχιάς

Από την παρακολούθηση της κίνησης των αντικειμένων και την καταγραφή της τροχιάς τους εξάγονται παρατηρήσεις και συμπεράσματα που αφορούν τόσο την αλληλεπίδραση των κινούμενων αντικειμένων μεταξύ τους όσο και με τον χώρο γύρω τους.

Καταγράφουμε και αποθηκεύουμε μέρος της τροχιάς με σκοπό να υποβάλλουμε ερωτήματα προς το σύστημα και να πάρουμε απαντήσεις τις οποίες δεν θα ήταν εφικτό να λάβουμε αν καταγράφαμε μόνο σημειακές θέσεις των κινούμενων αντικειμένων. Τέτοια ερωτήματα έχουν να κάνουν με την κίνηση των αντικειμένων σε σχέση με μία περιοχή ή μεγέθη που χαρακτηρίζουν την ίδια την κίνηση τους όπως η ταχύτητα και η κατεύθυνση κίνησής τους. Τα ερωτήματα που αφορούν τροχιές αντικειμένων μπορούν να χωριστούν στις εξής δύο κατηγορίες:

- *Τοπολογικά ερωτήματα*: Ασχολούνται με την ίδια την τροχιά, με την «τοπολογία» της, δηλαδή με την μορφή της και τα συμπεράσματα που μπορούν να εξαχθούν από αυτήν. Οι βασικοί τύποι τοπολογικών ερωτημάτων είναι:
 - Είσοδος/έξοδος (enter/exit): σε ή από κάποια περιοχή ενδιαφέροντος.
 - Διάβαση (cross): κάποιας περιοχής ενδιαφέροντος.
 - Προσπέρασμα (bypass)
- *Ερωτήματα πλοήγησης*: Δίνουν απαντήσεις σε δυναμικά ερωτήματα, μιάς και αυτές δεν προκύπτουν άμεσα από τα διαθέσιμα δεδομένα αλλά εξάγονται ως δευτερογενής πληροφορία. Οι βασικοί τύποι ερωτημάτων πλοήγησης είναι:
 - Ταχύτητας
 - Κατεύθυνσης
 - Διανυθείσας απόστασης

Τα ερωτήματα πλοήγησης στηρίζονται κυρίως στην χρονική διάσταση της τροχιάς. Στα πλαίσια αυτής της διπλωματικής θα απαντάμε τέτοια ερωτήματα και συγκεκριμένα μέσης ταχύτητας. Για να γίνει κατανοητό πόσο σημαντικός είναι ο χρονικός παράγοντας σε αυτού του τύπου τα ερωτήματα, αρκεί να σκεφτούμε πως η ταχύτητα ενός αντικειμένου για κάποια χρονική περίοδο προκύπτει από την απόσταση που αυτό διάνυσε διά τον χρόνο που χρειάστηκε για να διανύσει την απόσταση αυτή. Τα ερωτήματα πλοήγησης είναι συνήθως χρήσιμα σε εφαρμογές πραγματικού χρόνου. Στην Σχήμα 2.3 συνοψίζονται οι προαναφερθέντες τύποι ερωτημάτων για κινούμενα αντικείμενα.



Σχήμα 2.3: Τύποι ερωτημάτων για κινούμενα αντικείμενα [21]

2.5 Δεικτοδότηση κινούμενων αντικειμένων

Τα συστήματα διαχείρισης ρευμάτων δεδομένων δυσκολεύονται να παρακολουθήσουν και να επεξεργαστούν σε πραγματικό χρόνο (*online*) τον τεράστιο όγκο πληροφορίας που μπαίνει στο σύστημα από τα κινούμενα αντικείμενα, τηρώντας τα πάντα στην κύρια μνήμη. Από την άλλη μεριά, η χρήση δίσκου κρίνεται απαγορευτική για λόγους απόδοσης. Συνεπώς, η επινόηση κατάλληλων δομών δεδομένων δεικτών ή ευρετηρίων (*indexes*) είναι αναγκαία.

Η δεικτοδότηση κινούμενων αντικειμένων βασίζεται κυρίως σε υπάρχουσες τεχνικές για χωρικά αντικείμενα, με κυρίαρχο το R-tree. Παραλλαγές, όπως το STR-tree (Spatio-temporal R-tree), το TB-tree (Trajectory Bundle Tree), το TPR-tree (Time-Parameterized R-tree), το QUAD tree, προσπαθούν να καλύψουν την ανεπάρκεια του R-tree στη δεικτοδότηση. Βέβαια, όταν έχουμε να κάνουμε με τροχιές αντικειμένων, για τις οποίες θα πρέπει να κρατάμε στην μνήμη το πρόσφατο κομμάτι τους, το R-tree και οι παραλλαγές δεν λειτουργούν πολύ καλά. επειδή οι ανανεώσεις των θέσεων της πληθώρας των αντικειμένων προκαλούν πολλά αλληπάλλλα updates στις δενδρικές δομές, επιβαρύνοντας τις επιδόσεις τους.

Μία άλλη τεχνική δεικτοδότησης κινούμενων αντικειμένων, η οποία είναι αμιγώς χωρική είναι η ομοιόμορφη διαμέριση σε κάναβο (*uniform grid partitioning*). Ο χώρος χωρίζεται σε ισομεγέθεις περιοχές. Στα πλαίσια της παρούσας διπλωματικής γίνεται χρήση του καννάβου για ταχεία εύρεση θέσης αντικειμένων και ταυτοποίησης του ερωτήματος στο οποίο εμπίπτει το κάθε αντικείμενο, συνεπώς θα γίνει λεπτομερής περιγραφή αυτού και των επιπρόσθετων λειτουργιών που υλοποιήθηκαν πάνω σε αυτό σε επόμενο κεφάλαιο.

Κεφάλαιο 3

Αλγόριθμοι απόρριψης φόρτου από ρεύματα κινούμενων αντικειμένων

Κάθε σύστημα επεξεργασίας έχει πεπερασμένες δυνατότητες σε μνήμη και επεξεργαστική ισχύ. Επιπλέον, για την απάντηση των διαφόρων ερωτημάτων που τίθενται έχει κάποιο συγκεκριμένο περιθώριο, χρονικό ή επεξεργαστικής ισχύος, εντός του οποίου οφείλει να δίνει απαντήσεις. Απόρριψη φόρτου, είναι η διαδικασία κατά την οποία η επεξεργασία λαμβάνει υπόψιν μόνο ένα μέρος της εισόδου, με σκοπό να σέβεται τους περιορισμένους διαθέσιμους πόρους και να κατορθώνει να δίνει απαντήσεις σε πραγματικό χρόνο.

Γενικά, η απόρριψη φόρτου από ρεύματα δεδομένων περιλαμβάνει την λήψη τριών βασικών αποφάσεων:

- *Πότε γίνεται απόρριψη φόρτου:* Πρέπει τα ΣΔΡΔ να αποφασίζουν γρήγορα τον φόρτο του συστήματος κάθε στιγμή, με βάση κάποιες μετρικές που υιοθετούν. Κάθε φορά που υπάρχει εξέχων φόρτος στο σύστημα, να λαμβάνεται απόφαση για κατάλληλη διαγραφή δεδομένων χωρίς επεξεργασία.
- *Που γίνεται απόρριψη φόρτου:* Οι εισερχόμενες πλειάδες μπορούν να απορριφθούν σε οποιοδήποτε σημείο του δικτύου τελεστών του ΣΔΡΔ. Όταν μία πλειάδα δεν είναι τόσο χρήσιμη στην απάντηση κάποιου ερωτήματος και πρέπει να απορριφθεί, τότε όσο πιο γρήγορα απορριφθεί τόσο το καλύτερο για να μην κάνει το σύστημα περιττούς υπολογισμούς. Βέβαια, όταν έχουμε να κάνουμε με δίκτυο από πολλούς τελεστές ενδέχεται η απόρριψη μίας πλειάδας να είναι καλή για έναν τελεστή ενώ μπορεί να επηρεάζει αρνητικά έναν άλλο.
- *Πόσο φόρτος απορρίπτεται:* Όταν βρεθεί που πρέπει να γίνει απόρριψη φόρτου, το επόμενο βήμα είναι να προσδιοριστεί το ποσοστό των εισερχόμενων πλειάδων που δεν θα ληφθεί καθόλου υπόψιν στην επεξεργασία των ενεργών ερωτημάτων.

Το πόσο φόρτος θα απορριφθεί με τον ένα ή τον άλλο τρόπο προσδιορίζεται και ποσο-

τικοποιείται. Το μείζον ζήτημα είναι να ορίσουμε με σαφή τρόπο πότε και πού θα γίνει η απόρριψη φόρτου. Υπάρχουν διάφορες τεχνικές απόρριψης φόρτου σε ρεύματα δεδομένων και σε κινούμενα αντικείμενα, η καθεμία από τις οποίες εστιάζει σε διαφορετική πτυχή. Ακολουθεί ανασκόπηση των σημαντικότερων και γνωστότερων τεχνικών, οι οποίες αποτελούν την βάση στην οποία στηρίχθηκε η μεθοδολογία της παρούσας διπλωματικής εργασίας, με εξειδίκευση στην απόρριψη φόρτου από ρεύματα κινούμενων αντικειμένων.

3.1 Απόρριψη φόρτου από ρεύματα δεδομένων

3.1.1 Απόρριψη φόρτου με βάση διαγράμματα ποιότητας

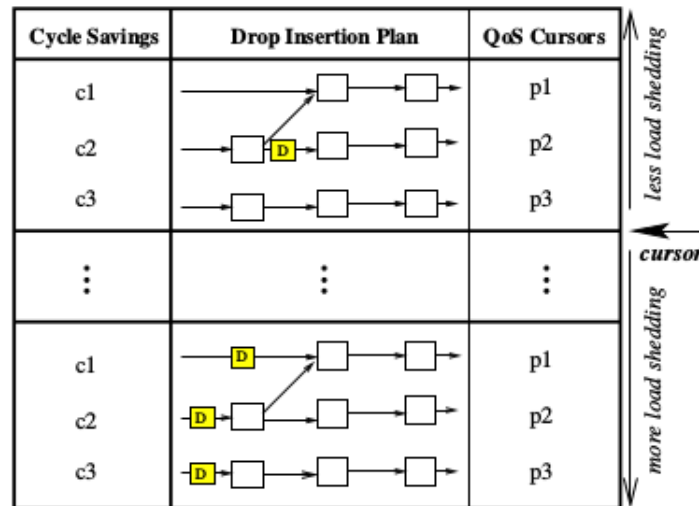
Η συγκεκριμένη τεχνική χρησιμοποιήθηκε στο σύστημα διαχείρισης ρευμάτων δεδομένων του MIT, το AURORA [25]. Αυτό το σύστημα, δέχεται είσοδο από διάφορες πηγές, την επεξεργάζεται για να απαντήσει τα εκκρεμή ερωτήματα και παράγει διαγράμματα ποιότητας (QoS) τα οποία χρησιμοποιεί αργότερα για να δώσει τις νέες απαντήσεις, καθώς και για να κάνει απόρριψη φόρτου όπου χρειάζεται.

Στο AURORA υπάρχει ένα δίκτυο τελεστών που η έξοδος του ενός τροφοδοτεί τον επόμενο μέχρι να εξαχθεί η τελική απάντηση στα εκάστοτε ερωτήματα. Αυτή η απάντηση ουσιαστικά δεν σχηματίζεται από τίποτα άλλο παρά από τις πλειάδες που “επιβιώνουν” διερχόμενες από τους τελεστές του δικτύου. Η απάντηση στο πότε θα γίνει load shedding δίνεται όταν ο φόρτος του συστήματος ξεπεράσει τον μέγιστο επιτρεπτό. Τότε τρέχει μία διαδικασία η οποία επιχειρεί να μειώσει τον φόρτο του συστήματος κατά μία σταθερά, το μέγιστο capacity C . Θα πρέπει επομένως ο αλγόριθμος να είναι προσαρμοζόμενος (*adaptive*) και να εντοπίζει πότε πρέπει να γίνει απόρριψη φόρτου. Εν συνεχεία, ψάχνοντας πού και το πόσο φόρτος πρέπει να απορριφθεί, χρησιμοποιούν μία δομή που την ονομάζουν Load Shedding Road Map (LSRM). Αυτή περιλαμβάνει διάφορα κελιά, με το καθένα να έχει ένα πλάνο για την τοποθέτηση τελεστών απόρριψης φόρτου σε διάφορα σημεία του δικτύου, καθώς και εκτίμηση των κύκλων μηχανής που κερδίζονται κατά την εκάστοτε επιλογή πλάνου, και τα αντίστοιχα QoS. Επιλέγοντας επόμενα κελιά (αφού κατατάσσονται ανάλογα με την επίδραση που έχουν στον φόρτο), η απόρριψη φόρτου εντείνεται. Ο εν λόγω πίνακας απεικονίζεται στο Σχήμα 3.1.

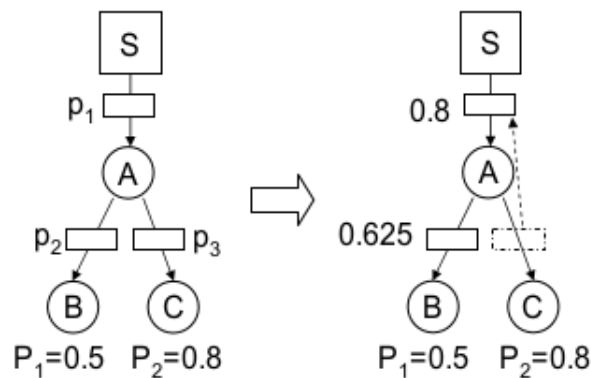
Υπάρχουν δύο πολιτικές απόρριψης φόρτου με βάση το LSRM: μία κατά την οποία η απόρριψη των εισερχόμενων πλειάδων γίνεται με τρόπο τυχαίο (*random*) και μία σημασιολογική (*semantic*) όπου γίνεται απόρριψη των πλειάδων με βάση τα QoS γραφήματα, ώστε να δίνονται απαντήσεις που χρησιμοποιούν την κατάλληλη πληροφορία.

3.1.2 Απόρριψη φόρτου με δειγματοληψία

Η συγκεκριμένη τεχνική χρησιμοποιήθηκε στο σύστημα διαχείρισης ρευμάτων δεδομένων του Stanford, το STREAM [4]. Αυτή η τεχνική στηρίζεται στην τοποθέτηση τελεστών απόρριψης φόρτου (*load shedding operators*) εμπόλιμα στο πλάνο εκτέλεσης ερωτημάτων, οι οποίοι επεξεργάζονται την είσοδο για να δώσουν την τελική απάντηση. Κάθε τέτοιος τελεστής παραμετροποιείται με πιθανότητα δειγματοληψίας p . Κάθε τέτοιος τελεστής ρίχνει



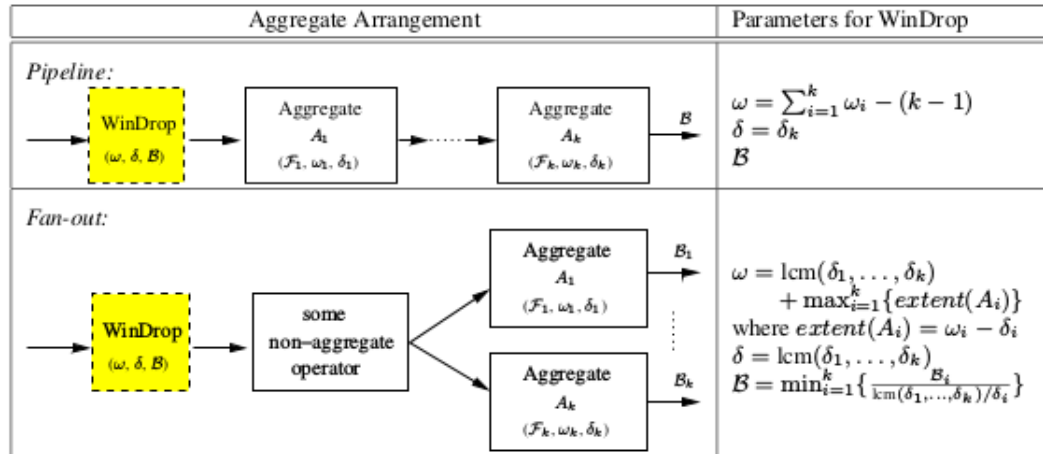
Σχήμα 3.1: Load Shedding Road Map [25]



Σχήμα 3.2: Ιδέα μείωσης του αξιοποιήσιμου ρυθμού εισόδου [4]

ένα κέρμα για κάθε εισερχόμενη πλειάδα και με πιθανότητα p αφήνει την πλειάδα να περάσει στους επόμενους τελεστές του δικτύου, ενώ με πιθανότητα $1 - p$ η πλειάδα απορρίπτεται.

Σκοπός της μεθόδου είναι (α) να προσδιοριστούν για κάθε τελεστή απόρριψης φόρτου η τιμή του p με σκοπό να κατανοηθεί το σφάλμα στα αποτελέσματα, και (β) να καθορίσει το πού θα βάλει τους τελεστές απόρριψης φόρτου στο πλάνο εκτέλεσης με σκοπό να επιτευχθεί η κατάλληλη απόρριψη φόρτου και να δοθούν απαντήσεις σε πραγματικό χρόνο. Ως βέλτιστη λύση προκύπτει πως οι τελεστές φόρτου πρέπει να τοποθετούνται στην αρχή των κοινών τμημάτων των τελεστών. Στο Σχήμα 3.2 φαίνεται η τοποθέτηση τέτοιων τελεστών σε κοινά τμήματα με το ίδιο αποτέλεσμα, αλλά με λιγότερους τελεστές απόρριψης στην μία απ' ότι στην άλλη περίπτωση, με αποτέλεσμα να επιτυγχάνεται η ίδια επιλεκτικότητα (*selectivity*) ελέγχοντας μικρότερο όγκο εισερχόμενων δεδομένων.



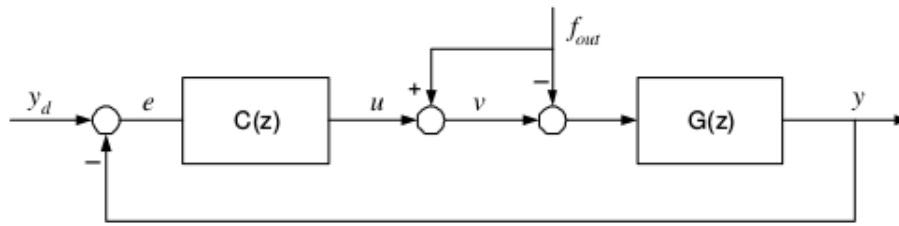
Σχήμα 3.3: Κανόνες για Window drop τελεστές [26]

3.1.3 Απόρριψη φόρτου με χρήση παραθύρων

Η συγκεκριμένη τεχνική δημοσιεύτηκε πρώτη φορά στο άρθρο [26]. Και εδώ έχουν να κάνουν με πλάνο συναθροιστικών ερωτημάτων (*aggregate query plan*). Εισάγουν έναν τελεστή *drop*, ο οποίος είναι ουσιαστικά *window drop* και ουσιαστικά αυτός ο τελεστής σέβεται τις ιδιότητες των παραθύρων των τελεστών που έπονται από αυτόν σε πλάνο ερωτημάτων. Χωρίζει το ρεύμα εισόδου σε παράθυρα και πιθανολογικά αποφασίζει ποιο παράθυρο να αφήσει να περάσει στο υπόλοιπο πλάνο και ποιο να απορρίψει. Σε αντίθεση με τις προηγούμενες μεθόδους, απορρίπτοντας ολόκληρα παράθυρα, έχει την ιδιότητα ότι διατηρεί την ακεραιότητα των παραθύρων και πάντα η απάντηση που δίνει είναι υποσύνολο της ακριβούς απάντησης με ελάχιστο κόστος στην ποιότητα.

Ο ειδικός τελεστής WinDrop σημαδεύει τα παράθυρα που δέχεται από την είσοδο και αποφασίζει αν θα περάσουν ή θα απορριφθούν. Η μέθοδος προνοεί έτσι ώστε παράθυρα τα οποία είναι πίσω στον χρόνο και εκ των προτέρων δεν θα συνισφέρουν στην απάντηση, να εγκαταλείπονται εξ αρχής. Επιπλέον, είναι δυνατόν να αντιμετωπιστούν πολλαπλοί αθροιστικοί τελεστές και εξετάζονται δύο ξεχωριστά σχήματα: το pipeline και το fan-out σχήμα. Στο Σχήμα 3.3 απεικονίζονται τα δύο σχήματα καθώς και οι παράμετροι που θα πρέπει να πληροί ο τελεστής WinDrop για αυτά.

Εν γένει, η απόρριψη φόρτου θα πρέπει να γίνεται στο πιο αρχικό στάδιο του πλάνου ερωτημάτων, με βασικό σκοπό να αποφεύγεται πλεονάζων φόρτος. Στην παρούσα διπλωματική, δεν έχουμε να κάνουμε με πλάνο ερωτημάτων. Έχουμε έναν τύπο ερωτήματος, το οποίο θέλουμε να απαντάμε σεβόμενοι τις παραμέτρους του εκάστοτε παραθύρου. Και από την προαναφερθείσα εργασία, αξιοποιούμε την ιδέα ότι μπορούμε να απορρίπτουμε ολόκληρες τροχιές, να μην τις συμπεριλαμβάνουμε δηλαδή στην απάντηση του ερωτήματός μας, αντί να πετάμε επιλεκτικά κάποια στίγματα της με σημασιολογικό τρόπο.



Σχήμα 3.4: Σύστημα κλειστού ελέγχου [29]

3.1.4 Απόρριψη φόρτου με βάση τεχνικές αυτομάτου ελέγχου

Η συγκεκριμένη τεχνική [29], χρησιμοποιεί τεχνικές αυτομάτου ελέγχου για να διαχειριστεί ρεύματα δεδομένων. Με αρχικό στόχο να απαλλαγθούν από τις αδυναμίες των υπάρχοντων τεχνικών απόρριψης πρότειναν ένα κλειστό βρόγχο με ανάδραση (*feedback*), μέσω της οποίας προσδιορίζουν δυναμικά πόσο απόρριψη πρέπει να κάνουν, για να δίνουν απαντήσεις σε πραγματικό χρόνο και με την μικρότερη δυνατή έκπτωση στην ποιότητά των αποτελεσμάτων. Η μέθοδός τους δεν επηρεάζεται από μεγάλες διακυμάνσεις στο ρυθμό εισόδου δεδομένων στο σύστημα.

Τα βασικά στάδια αυτού του κλειστού βρόγχου που κατασκεύασαν τα εξής:

- Αρχικά, ένας επόπτης μετρά το σήμα στην έξοδο του συστήματος.
- Οι μετρήσεις του επόπτη πηγαίνουν στον ελεγκτή, ο οποίος συγκρίνει την μετρούμενη έξοδο με μία τιμή στόχο (*target value*) και φτιάχνει την διαφορά τους και την στέλνει με ένα σήμα ελέγχου. Αυτό είναι το σφάλμα ελέγχου (*control error*).

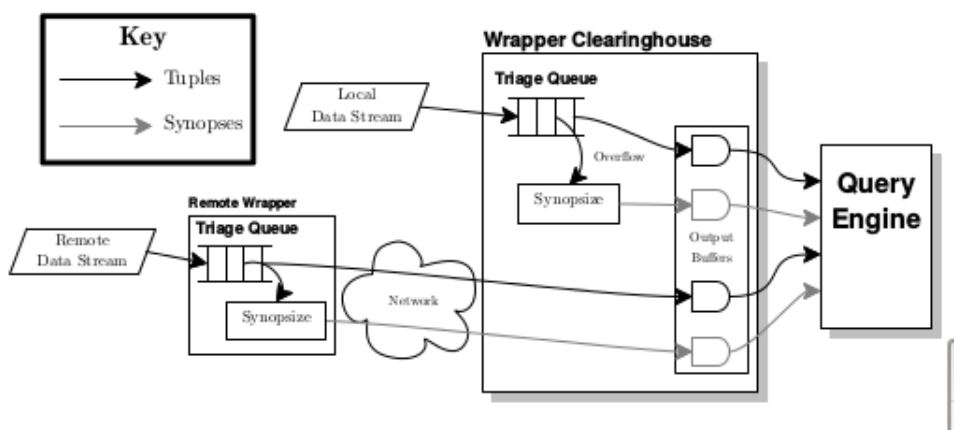
Σκοπός του συστήματος είναι να αντιμετωπίσει τις απρόσμενες διακυμάνσεις και τα απότομα ξεσπάσματα μαζικής εισροής δεδομένων στην είσοδο. Στο Σχήμα 3.4 απεικονίζεται στο παρακάτω σχήμα:

Συγκριτικά με το AURORA, η επίδοση του συστήματος απόρριψης φόρτου με τεχνικές αυτομάτου ελέγχου έδωσε πολύ καλύτερα αποτελέσματα.

3.1.5 Υβριδική απόρριψη φόρτου

Η συγκεκριμένη τεχνική χρησιμοποιήθηκε στο σύστημα TelegraphCQ, το οποίο αναπτύχθηκε στο Berkeley [22]. Είναι δεδομένο πως όταν παρατηρείται μεγάλο ξεσπάσμα στην είσοδο των δεδομένων πρέπει να γίνει απόρριψη φόρτου για να δίνονται εγκαίρως οι απαντήσεις. Η ιδέα που αναπτύχθηκε σε αυτή την εργασία είναι να μην διαγράφονται ολοκληρωτικά οι πλειάδες που πέφτουν θύμα της απόρριψης φόρτου, αλλά να τηρούνται συνόψεις στην μνήμη, οι οποίες δίνουν μία καλή προσέγγιση για την απωλεσθείσα πληροφορία, με σκοπό να χρησιμεύσουν αργότερα στην απάντηση των ερωτημάτων που τίθενται στο σύστημα.

Στόχος είναι το σύστημα να απαντά γρήγορα σε αλλαγές στον φόρτο του συστήματος, να δίνει προσεγγιστικές απαντήσεις, που όμως δεν απέχουν πολύ από τις πραγματικές, να



Σχήμα 3.5: Σύστημα Data Triage [22]

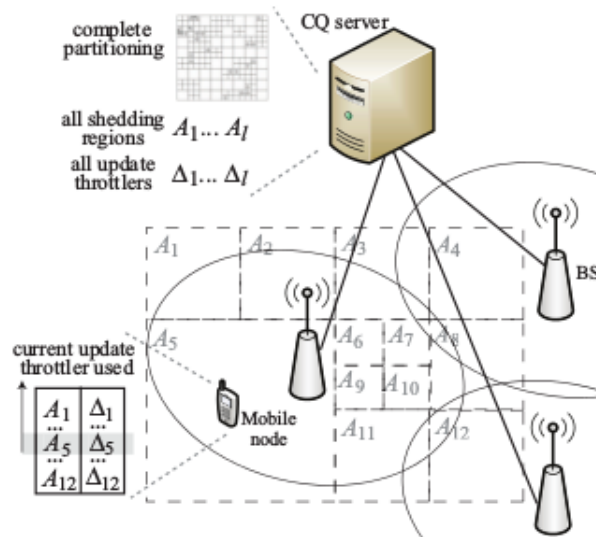
διαχειρίζεται καταστάσεις στις οποίες τα δεδομένα σε ένα ζέσπασμα έχουν διαφορετικά χαρακτηριστικά απ' ότι συνήθως. Η SQL τροποποιείται ώστε να ενσωματώνει δυνατότητα χρησιμοποίησης και των συνόψεων των πλειάδων που έχουν απορριφθεί αλλά και πλειάδων που έχουν κρατηθεί και μπορούν να δώσουν την ακριβή συνεισφορά τους στην απάντηση. Τα βασικά χαρακτηριστικά του συστήματος απεικονίζονται στο Σχήμα 3.5.

3.2 Απόρριψη φόρτου σε κινούμενα αντικείμενα

Τα κινούμενα αντικείμενα θεωρούνται σημειακά και καθώς η θέση τους μεταβάλλεται δυναμικά αποστέλλεται ανά τακτά χρονικά διαστήματα σ' έναν κεντρικό επεξεργαστή. Υπάρχει ενδεχόμενο το στίγμα των αντικειμένων να καταγράφεται με πολύ μεγάλη συχνότητα (λ.χ. κάθε 10 δευτερόλεπτα). Τότε ο όγκος της πληροφορίας που συσσωρεύεται γίνεται υπερβολικά μεγάλος, και εφόσον το σύστημα οφείλει να διαχειρίζεται online τα δεδομένα, ενδεχομένως να καθυστερεί η επεξεργασία ποικίλων ερωτημάτων. Για τον λόγο αυτόν, δεδομένου ότι η διαθέσιμη μνήμη και η επεξεργαστική ισχύς του συστήματος είναι πεπερασμένες, μία ιδέα είναι το σύστημα να επεξεργάζεται μόνο ένα μέρος της εισόδου. Η τεχνική της απόρριψης φόρτου (*load shedding*) έχει εφαρμοστεί σε συστήματα επεξεργασίας ρευμάτων δεδομένων, μειώνοντας την υστέρηση στην ανανέωση των αποτελεσμάτων, χωρίς σοβαρές απώλειες στην ποιότητά τους.

3.2.1 Απόρριψη φόρτου κατά περιοχές

Το LIRA [9] προτείνει μία μέθοδο απόρριψης φόρτου για mobileCQ συστήματα. Χρησιμοποιεί μία region-aware τεχνική, στην οποία διαμερίζεται ο χώρος και προσπαθεί να βρεί από ποιο κομμάτι του χώρου είναι πιο αποδοτικό να διαγράφονται εισερχόμενες πλειάδες (νέα στίγματα) αντικειμένων. Η επιλογή αυτή βασίζεται στην πυκνότητα των στιγμάτων αλλά και των ερωτημάτων που υπάρχουν σε κάθε περιοχή της διαμέρισης αυτής (η οποία βασίζεται στο quad tree).



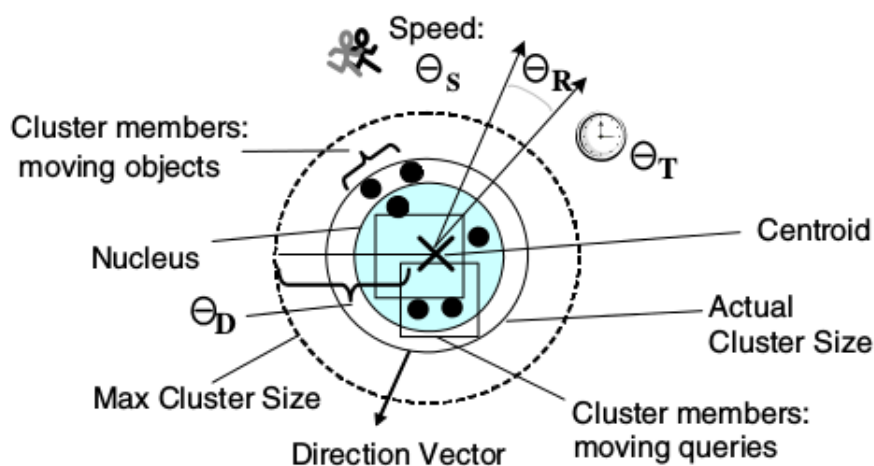
Σχήμα 3.6: Αρχιτεκτονική του συστήματος LIRA [9]

Υπάρχουν εν γένει, δύο κύριες μέθοδοι απόρριψης φόρτου: μία στον εξυπηρετητή και μία στις πηγές. Στην πρώτη μέθοδο, τα νέα στίγματα των κινούμενων αντικειμένων απορρίπτονται από τον ίδιο τον CQ server με σκοπό να συμβαδίζει ο ρυθμός άφιξης των στιγμάτων με τον ρυθμό που μπορεί να παρέχει απαντήσεις ο server. Αυτή η μέθοδος έχει δύο βασικά μειονεκτήματα. Πρώτον, οι πλειάδες που εν τέλει θα απορριφθούν, μεταδίδονται από τα κινούμενα αντικείμενα στον CQ server, σπαταλώντας πόρους δικτύου. Δεύτερον, τα πλεονάζοντα στίγματα, θα φτάσουν έτσι ή αλλιώς στον σερερ πράγμα το οποίο μεγαλώνει τον φόρτο. Η δεύτερη μέθοδος, που είναι source-actuated, απαιτεί μία καλή ενορχήστρωση μεταξύ του σερερ και των κινούμενων αντικειμένων, μιας και οι αποφάσεις για την απόρριψη φόρτου λαμβάνονται από τον server. Το LIRA χρησιμοποιεί την source-actuated τεχνική για load shedding. Η αρχιτεκτονική του συστήματος απεικονίζεται στο Σχήμα 3.6.

Ουσιαστικά η περιοχή μελέτης υποδιαιρείται σε κελιά, χρησιμοποιώντας αρχικά ένα και έπειτα κάθε υποπεριοχή (κελί) τεμαχίζεται με χρήση quadtree, μέχρι όλες οι υποπεριοχές να έχουν την ίδια ή παραπλήσια πυκνότητα. Σε κάθε τέτοιο κομμάτι συλλέγονται κάποια στατιστικά με βάση τα οποία τοποθετούνται load shedder σε κάθε υποπεριοχή και καθορίζεται το ποσοστό απόρριψης των εισερχόμενων πλειάδων που θα έχει ο κάθε load shedder. Ανά τακτά χρονικά διαστήματα εξετάζεται ο φόρτος του συστήματος, η υστέρηση στην απόκριση (*latency*) και άλλες μετρικές, με σκοπό το σύστημά να είναι προσαρμοστικό (*adaptive*) στις αυξομειώσεις του φόρτου.

3.2.2 Απόρριψη φόρτου με τεχνικές εξόρυξης από δεδομένα κινούμενων αντικειμένων

Η μέθοδος ClusterSheddy [19] προτείνει ένα σύστημα που κάνει απόρριψη φόρτου από κινούμενα αντικείμενα χρησιμοποιώντας τεχνικές εξόρυξης δεδομένων (*data mining*). Η κεντρική ιδέα είναι ότι κινούμενα αντικείμενα τα οποία κινούνται με όμοιο τρόπο μπορούν να



Σχήμα 3.7: Αναπαράσταση μίας κινούμενης συστάδας [19]

ομαδοποιούνται σε κινούμενες συστάδες (*clusters*), οι οποίες χρησιμεύουν σαν σύνοψη της κίνησης μελών τους. Λόγω των περιορισμών σε πόρους συστήματος (επεξεργαστική ισχύς και μνήμη), τα μέλη της κάθε συστάδας μπορούν να απορρίπτονται επιλεκτικά, και οι διαδοχικές θέσεις τους θα περιγράφονται προσεγγιστικά από τα αντίστοιχα κεντροειδή.

Η ιδέα βασίζεται στην ομοιότητα της κίνησης μεταξύ αντικειμένων. Η κίνηση όμοιων αντικειμένων προσεγγίζεται με μία κινούμενη συστάδα, και παρακολουθούν τις ιδιότητες σαν σύνολο. Έπειτα, απορρίπτονται στίγματα κινούμενων αντικειμένων, τα οποία είναι κοντά στο κέντρο της συστάδας με αποτέλεσμα να μην επηρεάζεται πολύ η ακρίβεια των απαντήσεων. Άρα η απόφαση αν θα γίνει απόρριψη φόρτου δεν αποφασίζεται με γνώμονα τα μεμονωμένα κινούμενα αντικείμενα, αλλά την συστάδα όπου ανήκουν.

Όταν ένα κινούμενο αντικείμενο ανήκει σε μία συστάδα, αυτό σημαίνει πως έχει παρόμοιες χωροχρονικές ιδιότητες με τα άλλα αντικείμενα στην ίδια συστάδα: ταχύτητα, κατεύθυνση και σχετική ομοιότητα των κινούμενων αντικειμένων. Οι ιδιότητες αυτές συνοψίζουν πώς και πού κινούνται τα αντικείμενα τα οποία βρίσκονται μέσα σε μία συστάδα. Οι συστάδες αναπρο-διορίζονται δυναμικά ανά πάσα χρονική στιγμή, έχοντας υπόψιν τις παραμέτρους και τους διαθέσιμους πόρους του συστήματος. Τα χαρακτηριστικά μίας τέτοιας συστάδας κινούμενων αντικειμένων φαίνεται στο Σχήμα 3.7.

Υλοποιούνται διάφορες πολιτικές απόρριψης φόρτου: 1) τυχαία επιλογή συστάδων προς απόρριψη (*random policy*), 2) πολιτική μετρητή (*count-based policy*) επιλέγοντας με βάση τον αριθμό κινούμενων αντικειμένων που περιέχονται στην συστάδα, 3) πολιτική μεγέθους (*size-based policy*), στην οποία διαλέγονται συστάδες που έχουν μικρότερο μέγεθος με σκοπό να μεγιστοποιηθεί η ακρίβεια και 4) πολιτική βαθμολογίας (*score-based policy*), στην οποία διαλέγονται συστάδες που έχουν το μεγαλύτερο score. Όσον αφορά το ποσοστό απόρριψης από κάθε συστάδα, εισάγεται η έννοια του πυρήνα (*nucleis*) που αυξομειώνεται ανάλογα με τον φόρτο του συστήματος.

3.2.3 Απόρριψη φόρτου με κλιμακώσιμες τεχνικές

Ο αλγόριθμος SOLE [16] επιχειρεί αποδοτική διαχείριση της μνήμης τηρώντας τα αντικείμενα που σημαδεύει με κάποιο κριτήριο ως σημαντικά. Το SOLE χρησιμοποιεί έναν αλγόριθμο απόρριψης φόρτου, με τον οποίο κάποιο από τα αποθηκευμένα κινούμενα αντικείμενα διαγράφονται από την μνήμη.

Η κύρια ιδέα στηρίζεται στην έννοια των σημαντικών αντικειμένων (*significant objects*). Τα αντικείμενα που εισέρχονται στο σύστημα θα εξετάζονται για να εξακριβωθεί πόσο σημαντικά είναι προκειμένου να αποφασιστεί αν θα κρατηθούν στη μνήμη ή θα απορριφθούν κάνοντας χρήση μίας αναπροσαρμοζόμενης βαθμολόγησης (*score*) καθενός τηρούμενου αντικειμένου. Όλα αυτά γίνονται με βάση τις απαιτήσεις που έχει το σύστημα και με τον εκάστοτε φόρτο.

Υπάρχουν δύο τύποι φόρτου εκ των οποίων ο πρώτος εξαρτάται από ερωτήματα (*query load shedding*) και ο δεύτερος από αντικείμενα (*object load shedding*). Στο SOLE εισάγουν την έννοια του cache area, δηλαδή μιας περιοχής που περικλείει την αρχική περιοχή κάθε ερωτήματος, της οποίας τα αντικείμενα τηρούνται στην μνήμη γιατί μελλοντικά ενδέχεται να μπουν στην περιοχή κάποιου ερωτημάτος και κατ' επέκταση να πρέπει να χρησιμοποιηθούν στην απάντηση. Στο query load shedding μειώνεται η cache area γύρω από τα ερωτήματα στα οποία εφαρμόζεται. Αν αρχικά και εφόσον δεν επιτευχθεί η επιθυμητή απόρριψη φόρτου τότε τίθεται σε εφαρμογή το δεύτερο στάδιο, το οποίο είναι να συρρικνώνεται σιγά σιγά η κανονική περιοχή του ερωτήματος έως ότου να φτάσουμε στην επιθυμητή απόρριψη φόρτου. Απεναντίας, στο object load shedding επιλέγονται κατάλληλα αντικείμενα που θα κρατηθούν έτσι ώστε να έχουν την μικρότερη απώλεια σε ακρίβεια απάντησης ερωτημάτων. Η βασική ιδέα είναι να απορρίπτονται τα αντικείμενα που συμμετέχουν σε πολύ λίγα ερωτήματα και αντίθετα να κρατηθούν αυτά που βρίσκονται σε πολλά ερωτήματα ταυτόχρονα. Αν κάποιο αντικείμενο συμμετέχει ταυτόχρονα σε περισσότερα από k ερωτήματα τότε σημαδεύεται ως σημαντικό (*significant*). Ο συνδυασμός των δύο στρατηγικών επιτυγχάνει καλά πειραματικά αποτελέσματα.

Κεφάλαιο 4

Μηχανισμός απόρριψης φόρτου από τροχιές

4.1 Εισαγωγή

Η διπλωματική εργασία προτείνει έναν αυτόρρυθμιζόμενο μηχανισμό ο οποίος θα διαχειρίζεται πολύ μεγάλους όγκους δεδομένων, που αφορούν τροχιές που εξελίσσονται στο χρόνο. Επειδή δεν είναι εφικτό να επεξεργαστούν όλες οι τροχιές που υπάρχουν στο σύστημα και να απαντώνται τα ερωτήματα σε πραγματικό χρόνο, υλοποιείται ένας μηχανισμός απόρριψης ολόκληρων τροχιών ή μέρους τροχιών με απώτερο σκοπό να μειωθεί ο φόρτος του συστήματος.

Βεβαίως, μπορούν να χρησιμοποιηθούν διάφορες πολιτικές απόρριψης τροχιών κινούμενων αντικειμένων. Ωστόσο, απορρίπτοντας λιγότερο σημαντικά ή λιγότερο χρήσιμα κομμάτια τροχιών ώστε να μειωθεί ο συνολικός χρόνος επεξεργασίας των ερωτημάτων και να συμβαδίζει με τα τεράστια ποσά ενημερώσεων στις θέσεις των κινούμενων αντικειμένων είναι απολύτως απαραίτητο σε αυτή την περίπτωση.

Η τήρηση ολόκληρου του ιστορικού κάθε τροχιάς δεν είναι εφικτή τόσο λόγω ελλειψών πόρων συστήματος όσο και απεριόριστων απαιτήσεων σε μνήμη. Επίσης, έχει νόημα να τηρούνται ολόκληρες τροχιές ή μέρος τροχιών μόνο εφόσον είναι χρήσιμες σε πολλά ερωτήματα ταυτόχρονα. Ακόμα και αν εξετάζεται μόνο το πρόσφατο κομμάτι κάθε τροχιάς και πάλι είναι πολύ πιθανό να ανακύπτει πρόβλημα πόρων λόγω της παρακολούθησης πολύ μεγάλου αριθμού κινούμενων αντικειμένων. Η μείωση του όγκου των δεδομένων μπορεί να έχει ήδη εφαρμοστεί από τις πηγές στα κινούμενα αντικείμενα, λ.χ. οδηγώντας σε απλοποίηση τροχιών με χρήση συνόψεων. Ωστόσο, αυτή η διαδικασία κυρίως επηρεάζει την ποιότητα των απαντήσεων και το κόστος διακίνησης της πληροφορίας (*communication cost*), παρά την επίδοση των ερωτημάτων που περιλαμβάνει την προσαρμοστικότητα (*adaptivity*) και την κλιμακωσιμότητα (*scalability*), την ικανότητα δηλαδή το σύστημα να μπορεί να διαχειρίζεται και να αναποκρίνεται σε οποιοδήποτε ρυθμό εισόδου στιγμάτων θέσεις των κινούμενων αντικειμένων.

Η επιβάρυνση από τον μηχανισμό απόρριψης φόρτου πρέπει να είναι μικρή, συγκριτικά με την επεξεργασία ερωτημάτων. Η πολιτική απόρριψης φόρτου θα πρέπει γενικά να εξαλείφει πλήρως ή μερικώς αυτές που είναι ελάσσονος σημασίας και δεν συνεισφέρουν πολύ στην

αύξηση της ακρίβειας των απαντήσεων.

Είναι λοιπόν επιθυμητό να κρατώνται πολλαπλά τμήματα τροχιών στην κύρια μνήμη, αλλά να αποφασίζεται δυναμικά ποιές από τις τροχιές θα χρησιμεύουν στην απάντηση των ερωτημάτων με μικρή υστέρηση. Αυτό θα μπορούσε να επιτευχθεί με μία απλοποιημένη δεικτοδότηση των τροχιών με σκοπό να εντοπίζεται γρήγορα η χρησιμότητά τους στην επεξεργασία ενεργών ερωτημάτων. Πέραν των επιδόσεων, η επιτυχία του προτεινόμενου μηχανισμού θα εξαρτηθεί επίσης από την ποιότητα των απαντήσεων, επιδιώκοντας την μικρότερη δυνατή απώλεια στην ακρίβεια, σε σύγκριση με την εξαντλητική αποτίμηση των ερωτημάτων.

Τέλος, θα παρουσιάσουμε ένα μηχανισμό που θα είναι όπως προείπαμε προσαρμοστικός και θα δουλεύει σε κάθε ρυθμό εισόδου δεδομένων και θα χρησιμοποιείται για την προσεγγιστική και αυξητική απάντηση τυπικών χωροχρονικών συνεχών ερωτημάτων σε πραγματικό χρόνο και με την μικρότερη δυνατή απώλεια στην ακρίβεια (σε σύγκριση με την εξαντλητική αποτίμηση των ερωτημάτων).

4.2 Διαχείριση ρευμάτων κινούμενων οχημάτων

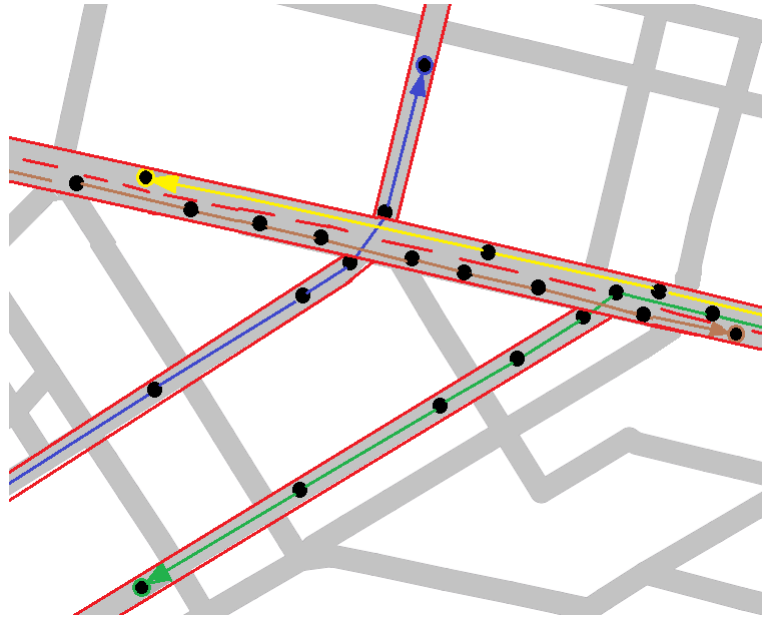
4.2.1 Το μοντέλο δεδομένων

Τα οχήματα που κυκλοφορούν στο οδικό δίκτυο μιας περιοχής θεωρούνται σημειακά αντικείμενα, των οποίων η θέση μεταβάλλεται με την πάροδο του χρόνου. Το χωροχρονικό φαινόμενο της κίνησης κάθε οχήματος στο επίπεδο μπορεί να αναπαρασταθεί με την συνεχή τροχιά του (*trajectory*), η οποία συντίθεται από δύο χωρικές συντεταγμένες (x,y) και μία χρονική ένδειξη (t). Καθώς η συνεχής καταγραφή της τροχιάς είναι αδύνατη, κάθε όχημα αποστέλλει τακτικά το στίγμα του, αν και όχι απαραίτητα με την ίδια συχνότητα με όλα τα υπόλοιπα οχήματα. Πρακτικά, κάθε τροχιά προσεγγίζεται από μία διακριτή ακολουθία πλειάδων $\langle oid, t, x, y \rangle$, όπου *oid* είναι το μοναδικό αναγνωριστικό του οχήματος και t το χρονόσημο (*timestamp*) κατά το οποίο το γεωγραφικό στίγμα (x,y) καταφθάνει στο σύστημα.

Σε κάθε χρονική στιγμή το σύστημα συλλέγει σημειακές ενημερώσεις θέσεων της μορφής $\langle oid, t, x, y \rangle$ από έναν μεγάλο αριθμό N κινούμενων αντικειμένων, τα οποία προφανώς έχουν ενσωματωμένο κάποιον μηχανισμό GPS. Όταν όλες αυτές οι ενημερώσεις (κάθε χρονική στιγμή) φτάνουν σε ένα κεντρικό επεξεργαστή (*server*), συγχροτούν ένα ρεύμα εισόδου (*input stream*) με ταξινομημένες τις πλειάδες με βάση το χρονόσημο t (*timestamp*). Τα στίγματα που εισέρχονται στο σύστημα αποθηκεύονται στην μνήμη και ενημερώνουν τις εκάστοτε τροχιές των κινούμενων αντικειμένων.

4.2.2 Το μοντέλο ερωτημάτων

Ένα πλήθος $M \ll N$ από ερωτήματα διαρκείας (*continuous queries*) τίθενται στο σύστημα και θέτουν χωροχρονικά παράθυρα πάνω στις τροχιές των κινούμενων αντικειμένων. Κάθε παράθυρο W_i καθορίζει μία χωρική περιοχή ενδιαφέροντος A_i , καθώς και ένα χρονικό παράθυρο εύρους ω_i που μετακινείται κάθε β_i χρονικές μονάδες. Οι τροχιές οι οποίες πληρούν τις προϋποθέσεις για να χρησιμοποιηθούν στην απάντηση, δίνουν την πληροφορία που



Σχήμα 4.1: Τροχιές οχημάτων κατά μήκος οδικών αξόνων

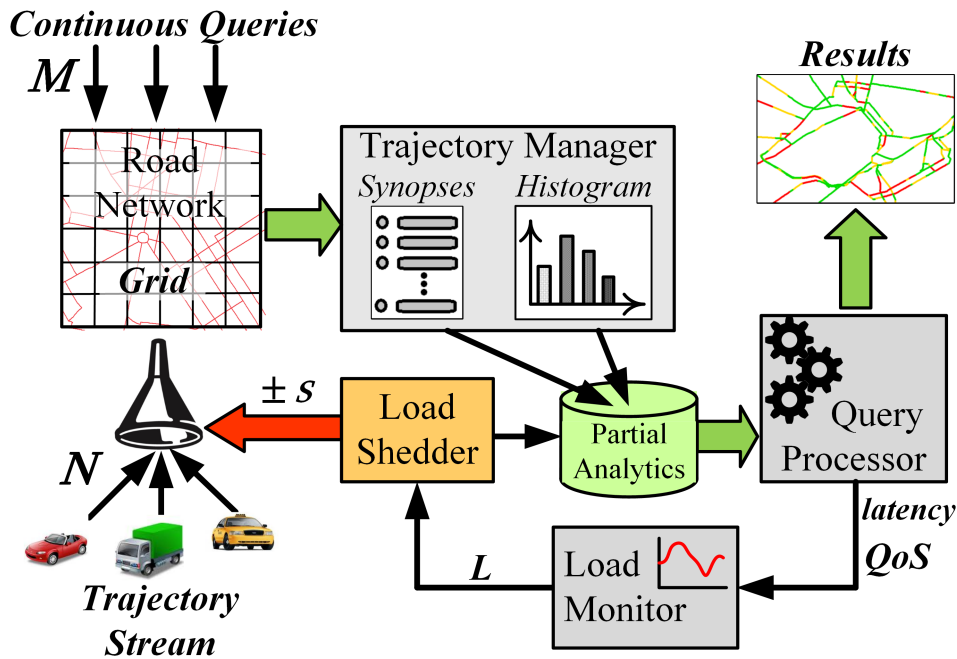
βρίσκεται στα τμήματά τους τα οποία εμπίπτουν στην περιοχή ενδιαφέροντος A_i ανα πάσα χρονική στιγμή εντός εύρους ω_i του παραθύρου.

Ο κεντρικός επεξεργαστής οφείλει να αναγνωρίζει τα τμήματα των τροχιών που διασχίζουν κάθε δρόμο, προκειμένου να υπολογίζει στατιστικά στοιχεία κυκλοφορίας (*traffic analytics*) σε πραγματικό χρόνο. Συγκεκριμένα, απαντώνται συναθροιστικά ερωτήματα μέσης ταχύτητας ανά οδικό άξονα ενδιαφέροντος. Ένα ερώτημα *qid* τίθεται την χρονική στιγμή t , προσδιορίζει το εύρος (*range*) ω του κυλιόμενου παραθύρου (*sliding window*) για το οποίο θα εξετάζει τμήματα τροχιών επί του συγκεκριμένου άξονα, ενώ θα ανανεώνει τις απαντήσεις κάθε β χρονόσημα, όσα προσδιορίζει το βήμα (*slide*) του παραθύρου. Π.χ., ένα ερώτημα μπορεί να ζητεί να υπολογίζεται κάθε λεπτό ($\beta = 1 \text{ min}$) η τρέχουσα μέση ταχύτητα στην οδό Πανεπιστημίου βάσει τροχιών που διήλθαν από εκεί εντός του τελευταίου τετάρτου ($\omega = 15 \text{ min}$). Όπως φαίνεται στο Σχήμα 4.1, τα σχετικά ερωτήματα διαρκείας τίθενται ανά άξονα (με κόκκινο περίγραμμα) και οφείλουν να υπολογίζουν ταχύτητες ανά κατεύθυνση κυκλοφορίας.

4.3 Αρχιτεκτονική του συστήματος

Αφού το σύστημα δέχεται διαρκώς ενημερώσεις θέσης από πλήθος οχημάτων, θα πρέπει να επεξεργάζεται τις δυναμικά εξελισσόμενες τροχιές τους προκειμένου να υπολογίζει τις μέσες ταχύτητες ανά άξονα. Ωστόσο, απότομες διακυμάνσεις (κυρίως ξαφνικές αυξήσεις) στον ρυθμό άφιξης των στιγμάτων μπορεί να καθιστούν απαγορευτική την εξέταση του συνόλου των τροχιών για την έγκαιρη απάντηση των ερωτημάτων. Σε τέτοιες περιπτώσεις, λόγω περιορισμών σε μνήμη και επεξεργαστική ισχύ, γίνεται επιτακτική ανάγκη να απορριφθούν επιλεκτικά κάποιες τροχιές προτού καν εξεταστούν, κατά προτίμηση όσες είναι λιγότερο “χρήσιμες”.

Ο προτεινόμενος μηχανισμός απόρριψης φόρτου λειτουργεί σε κύκλος εκτέλεσης (*execu-*



Σχήμα 4.2: Αρχιτεκτονική απόρριψης φόρτου από τροχιές

tion cycles), καθένας από τους οποίους περιλαμβάνει τρεις διαδοχικές φάσεις: (i) ενημέρωση (update) τροχιών και δομών δεδομένων με τα πρόσφατα στίγματα, (ii) αποτίμηση ερωτημάτων (query evaluation), και (iii) απόρριψη φόρτου (load shedding) όποτε κριθεί απαραίτητη.

Το γενικό διάγραμμα της αρχιτεκτονικής του συστήματος απεικονίζεται στο Σχήμα 4.2. Όπως φαίνεται, κατά την φάση της ενημέρωσης, ο κεντρικός επεξεργαστής δέχεται τα στίγματα έως την χρονική στιγμή του τρέχοντος κύκλου εκτέλεσης. Αναθέτει σε κάθε στίγμα το αναγνωριστικό του δρόμου στον οποίο ανήκει και ενημερώνει τις τροχιές των κινούμενων οχημάτων που υπάρχουν στο σύστημα. Παράλληλα τηρούνται κάποια μερικά συναθροιστικά στατιστικά (partial analytics), τα οποία χρησιμοποιεί κατά της φάση της αποτίμησης με σκοπό να δώσει την μέση ταχύτητα στους άξονες ενδιαφέροντος. Αφού γίνει η αποτίμηση των τρέχοντων ενεργών ερωτημάτων (κατά την φάση της αποτίμησης), εν συνεχεία κατά την φάση της απόρριψης φόρτου, λαμβάνοντας υπόψιν την χρονική υστέρηση (ως μέτρο επίδοσης QoS), αποφασίζεται αν θα τρέξει ο αλγόριθμος της απόρριψης φόρτου, με σκοπό αυξηθεί ή να μειωθεί ο αριθμός τροχιών που τηρεί και επεξεργάζεται το σύστημα.

Ο μηχανισμός του συστήματος του Σχήματος 4.2 είναι προσαρμοστικός (adaptive), δηλαδή δουλεύει σε κάθε ρυθμό εισόδου δεδομένων. Το σύστημα, εκ των προτέρων, έχει ένα συγκεκριμένο χρόνο να εκτελέσει όλες τις λειτουργίες του ανά κύκλο εκτέλεσής. Θα πρέπει το σύστημα να χρησιμοποιεί το μέγιστο των δυνατοτήτων του κάθε φορά, ώστε οι απαντήσεις που δίνει να είναι έγκαιρες και όσο το δυνατόν πιο ακριβείς. Όταν υπερβεί το σύστημα τον χρόνο που του διατίθεται, τότε δυναμικά επιλέγει και απορρίπτει εισερχόμενη πληροφορία από τις λιγότερο “σημαντικές” τροχιές. Αντίστοιχα όταν τα χρονικά πλαίσια του το επιτρέπουν μπορεί να εξετάζει περισσότερες τροχιές.

Algorithm 1 Trajectory-aware Load Shedding

```

1: Procedure SystemMonitor(period  $\delta$ , capacity  $C$ , tolerance  $\theta$ , #buckets  $b$ , step  $s$ )
2: Input: Stream  $P$  of positional updates  $\langle o_i, p_i, \tau \rangle$  from  $N$  moving vehicles
3: Input: Road network represented as a graph  $(V, E)$  with directed edges in  $E$  (i.e., road segments
   carrying traffic flow) connecting pairs of vertices (crossroads) in  $V$ 
4: Input: Query specifications  $Q = \{\langle q_j, \omega_j, \beta_j, \tau_j^0 \rangle, j = 1..M\}$  for average speed along road  $q_j$  every  $\beta_j$ 
   time units over positions received during the past  $\omega$  units
5: Input: Period  $\delta$  (in time units) between successive execution cycles
6: Input: System capacity  $C$ , i.e., CPU cycles dedicated to the monitoring process
7: Input: Tolerance  $\theta$  ( $\%C$ ) for triggering load adjustment
8: Input: Number  $b$  of buckets to be used in trajectory histograms
9: Input: Step  $s$ , expressed as  $\%$  in the  $N$  trajectories to discard or supplement
10: State:  $\mathcal{T} = \{T_i : \text{currently maintained trajectory for each moving object } o_i\}$ 
11: State:  $\mathcal{S} = \{S_i : \text{synopsis of items } \langle q_j, t_{in}, t_{out} \rangle \text{ for each road } q_j \text{ traversed by } o_i\}$ 
12: State:  $\mathcal{K} = \{\forall \text{ query } q, \{o_i : \text{object } o_i \text{ has traversed road } q \text{ during window } q.\omega\}\}$ 
13: State:  $\mathcal{U} = \{u : \forall \text{ object } o_i, \text{ set } u=1 \text{ if } o_i \text{ is kept; } u=0 \text{ if evicted by load shedding}\}$ 
14: Preprocessing: Index all road edges in  $E$  against a grid  $G$  at granularity  $c \times c$ 
15:  $\mathcal{T} \leftarrow \emptyset; \mathcal{S} \leftarrow \emptyset; \mathcal{K} \leftarrow \emptyset;$  //Initialize data structures
16:  $\mathcal{U} \leftarrow \{1, \forall \text{ object } o_i\};$  //By default, try utilizing all trajectories in evaluation
17: repeat
18:    $\tau_c \leftarrow \text{now}();$  //Current timestamp marks this execution cycle
19:    $A \leftarrow \emptyset;$  //Initialize result set for all queries at execution cycle  $\tau_c$ 
20:    $\Delta P \leftarrow \{\langle o_i, p_i, \tau \rangle \in P : \tau \in (\tau_c - \delta, \tau_c]\};$  //Positions received since last cycle
   //Phase #1: Update
21:    $L_u \leftarrow \text{getCPUcycles}();$  //Initialize system load for updates
22:   for each position  $\langle o_i, p_i, \tau \rangle$  do
23:     Update( $\langle o_i, p_i, \tau \rangle$ ); //Update data structures with the given update
24:   end for
25:    $L_u \leftarrow \text{getCPUcycles}() - L_u;$  //Total CPU cost for all updates
   //Phase #2: Evaluation
26:    $L_v \leftarrow \text{getCPUcycles}();$  //Initialize system load for query evaluation
27:   for each query  $q \in Q$  do
28:      $v_q \leftarrow \text{Evaluation}(q, \tau_c);$  //Compute average speed for road  $q$ 
29:      $A \leftarrow A \cup \{q, v_q, \tau_c\};$  //Append result to the answer set
30:   end for
31:    $L_v \leftarrow \text{getCPUcycles}() - L_v;$  //Total CPU cost for query evaluation
   //Phase #3: Load Shedding
32:    $\text{sign} \leftarrow ((L_u + L_v > C) ? +1 : -1);$  //Determine overload (+1) or underload (-1)
33:   if  $|L_u + L_v - C| > \theta \cdot C$  then //Trigger load adjustment
34:     if  $L > C$  then //Overload
35:        $\text{underload} \leftarrow \text{true};$  //Check for potential underload in the next cycle
36:       Compute slopes between pairs of the three most recent load measurements;
37:     end if
38:     if  $(L > C) \vee ((L < C) \wedge \text{underload})$  then //Adjust load
39:        $n \leftarrow |\{u \in \mathcal{U} : u = 1\}|;$  //Number of trajectories just utilized in evaluation
40:       Shedding( $N, n, b, \text{sign} \cdot s$ ); //Shedding or taking in trajectories
41:     end if
42:     if  $\text{underload}$  then //Verify if load is steadily below capacity
43:       Reset  $\text{underload}$  if both slopes for load are decreasing;
44:     end if
45:   end if
   //Phase #4: Reporting
46:   Report  $A$ ; //Disseminate answer at execution cycle  $\tau_c$  to each query
47:   Wait  $\tau_c + \delta - \text{now}()$  time units, if  $< \delta$  units have passed since cycle  $\tau_c$  started;
48: until positional stream  $P$  is exhausted;
49: End Procedure

```

Στον Αλγόριθμο 1 φαίνεται ο ψευδοκώδικας του System Monitor, δηλαδή του ενορχηστρωτή του συστήματος. Είναι ουσιαστικά ο “εγκέφαλος” του συστήματος που προτείνεται. Το σύστημα λειτουργεί σε κύκλους εκτέλεσης, όπου ο καθένας λαμβάνει χώρα κάθε δ χρονικές μονάδες (Αλγόριθμος 1, γραμμή 5). Ιδανικά, η ενημέρωση τροχιών και η αποτίμηση των ερωτημάτων πρέπει να ολοκληρώνονται μέσα στην χρονική περίοδο που θέτει ένας κύκλος εκτέλεσης. Κάτι τέτοιο είναι όμως μάλλον ανέφικτο, όταν το πλήθος των κινούμενων αντικειμένων, ο ρυθμός άφιξης στιγμάτων και ο αριθμός των ερωτημάτων αυξάνει, οπότε η επεξεργασία θα υπερβεί την προκαθορισμένη χωρητικότητα C (*Capacity*) του συστήματος (δηλ. άνω όριο φόρτου). Πριν τεθεί σε λειτουργία το σύστημα, γίνεται μία προεπεξεργασία η οποία περιλαμβάνει την αρχικοποίηση των δομών που χρησιμοποιούνται και τον διαμοιρασμό του οδικού δικτύου σε καννάβο έτσι ώστε να είναι εύκολος και γρήγορος ο εντοπισμός του αναγνωριστικού του δρόμου *qid* στον οποίο ανήκει κάθε εισερχόμενο στίγμα (Αλγόριθμος 1, γραμμές 14-16).

Σε κάθε κύκλο εκτέλεσης η σειρά εκτέλεσης των φάσεων είναι δεδομένη. Πρώτα, για τα εισερχόμενα στίγματα πραγματοποιείται ενημέρωση των αντίστοιχων τροχιών (φάση ενημέρωσης, Αλγόριθμος 1, γραμμές 22-24). Έπειτα, αφού έχουν ενημερωθεί όλες οι τροχιές για τις οποίες κατέφθασαν καινούργια στίγματα στον τρέχοντα κύκλο εκτέλεσης, εκκινεί η διαδικασία αποτίμησης ερωτημάτων (φάση ενημέρωσης, Αλγόριθμος 1, γραμμές 26-30). Σε αυτή τη φάση δίνεται απάντηση στα ενεργά ερωτήματα διαρκείας. Μόλις ολοκληρώνεται η φάση ενημέρωσης και αποτίμησης γίνεται καταμέτρηση του χρόνου που χρειάστηκαν για να ολοκληρωθούν (Αλγόριθμος 1, γραμμές 21,25,26,31). Τέλος, εκτελείται υπό συνθήκες η φάση απόρριψης φόρτου (Αλγόριθμος 1, γραμμές 32-45).

4.4 Φάση ενημέρωσης

Ο ψευδοκώδικας για την φάση ενημέρωσης φαίνεται στον Αλγόριθμο 2. Για κάθε νεοεισερχόμενο στίγμα, το σύστημα κανονικά πρέπει να εντοπίσει τον οδικό άξονα (δηλ. το *qid* κάποιου ερωτήματος) στον οποίο κινήθηκε το όχημα (Αλγόριθμος 2, γραμμή 20). Ακόμη και στην περίπτωση που το στίγμα βρίσκεται πάνω σε διασταύρωση, ο δρόμος προκύπτει από το διάλυσμα κίνησης σε σχέση με το αμέσως προηγούμενο γνωστό στίγμα της τροχιάς. Προκειμένου να είναι εφικτός ο γρήγορος εντοπισμός των υποψηφίων αξόνων, όλη η περιοχή μελέτης και το οδικό δίκτυο διαμερίζονται στα κελιά ενός ομοιόμορφου καννάβου (*grid partitioning*). Κάθε κελί έχει μέγεθος $c \times c$ και του αντιστοιχεί ένα υποσύνολο από τους υπάρχοντες ενεργούς οδικούς άξονες. Αντί να γίνεται σύγκριση κάθε νεοεισερχόμενου στίγματος με όλα τα ενεργά ερωτήματα έως ότου βρεθεί σε ποιο *qid* ανήκει, εντοπίζεται πρώτα με τη βοήθεια του καννάβου το κελί στο οποίο ανήκει και έπειτα συγκρίνεται με τα ερωτήματα του κελιού αυτού.

Με αντίστοιχο τρόπο θα πρέπει να υπολογίζεται η στιγμιαία ταχύτητα \vec{v} του οχήματος. Η δευτερογενής πληροφορία που αφορά τον άξονα *qid* και την στιγμιαία ταχύτητα \vec{v} αποθηκεύονται μαζί με το πρωτογενές στίγμα και αναθεωρούν την κατάσταση της τροχιάς του συγκεκριμένου οχήματος όπως τηρείται απ’ τον επεξεργαστή (*Trajectory Manager*, Σχήμα 4.2) (Αλγόριθμος 2, γραμμές 18-21). Στο Σχήμα 4.3 φαίνεται η τροχιά ενός οχήματος, η

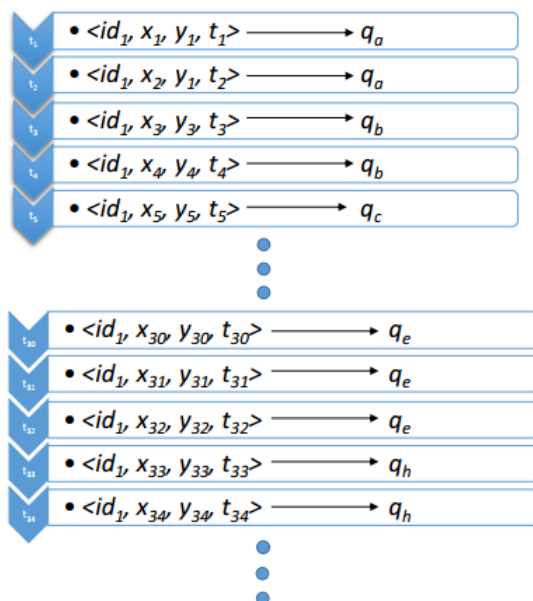
Algorithm 2 Trajectory-aware Load Shedding

```

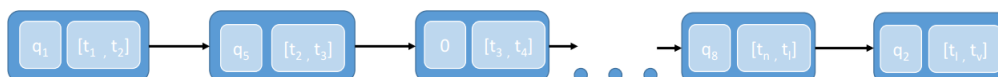
1: Procedure Update (object  $o_i$ , position  $p_i$ , timestamp  $\tau$ )
2: Input: Positional update  $\langle o_i, p_i, \tau \rangle$  from a given moving object with identifier  $o_i$ 
3: State:  $\mathcal{T} = \{T_i : \text{currently maintained trajectory for each moving object } o_i\}$ 
4: State:  $\mathcal{S} = \{S_i : \text{synopsis of items } \langle q_j, t_{in}, t_{out} \rangle \text{ for each road } q_j \text{ traversed by } o_i\}$ 
5: State:  $\mathcal{K} = \{\forall \text{ query } q, \{o_i : \text{object } o_i \text{ has traversed road } q \text{ during window } q.\omega\}\}$ 
6: State:  $G$  : uniform grid partitioning that indexes all road edges
7: Output: Updated states of data structures for  $\mathcal{T}$ ,  $\mathcal{S}$ , and  $\mathcal{K}$ 
8:  $T_i \leftarrow \mathcal{T}.\text{find}(o_i)$ ;           //Fetch trajectory maintained for the given object
9: if  $T_i \neq \emptyset$  then
10:   if  $\mathcal{U}[o_i] = 0$  then                               //Object is not useful
11:     Force object to send notification once it has entered into another road  $q$ ;
12:     if  $q \neq 0$  then                                   //Object has just entered into another road
13:       RefreshSynopsis( $o_i, q, \tau$ ); //Synopsis for this object must be refreshed
14:     else
15:       break; //Skip further processing due to position-level shedding
16:     end if
17:   else                                               //Proceed to update for non-shedded trajectories only
18:      $p_i^{prev} \leftarrow T_i.\text{top}()$ ; //Previously relayed location of  $o_i$  kept in trajectory  $T_i$ 
19:      $\vec{v} \leftarrow \text{Velocity}(p_i^{prev}, p_i)$ ; //Instantaneous velocity from last to current location
20:      $q \leftarrow \text{findRoad}(G, \vec{v})$ ; //Locate the road that this object has been traversing
21:      $T_i.\text{append}(\langle \tau, p_i, \vec{v} \rangle)$ ; //Update trajectory with most recent data
22:     RefreshSynopsis( $o_i, q, \tau$ ); //Check if synopsis for this object needs refresh
23:     if  $\mathcal{K}[q].\text{find}(o_i)$  then //Object  $o_i$  is being monitored for query  $q$ 
24:       Rearrange existing rear & front pointers to trajectory  $T_i$ ;
25:       Update partial aggregates at items involving object  $o_i$  in  $\mathcal{K}[q]$ ;
26:     else //No entry is found in keeper for object  $o_i$ 
27:       Initialize rear & front pointers to trajectory  $T_i$ ;
28:       Create a new entry  $\langle \vec{v}, 1 \rangle$  with speed aggregates under object  $o_i$  in  $\mathcal{K}[q]$ ;
29:     end if
30:   end if
31: else //This object was not being monitored thus far
32:    $T_i \leftarrow \text{constructTrajectory}(o_i)$ ;
33:    $\mathcal{T} \leftarrow \mathcal{T} \cup T_i$ ; //Append new item to trajectory set  $\mathcal{T}$ 
34:    $S_i \leftarrow \text{constructSynopsis}(o_i)$ ;
35:    $\mathcal{S} \leftarrow \mathcal{S} \cup S_i$ ; //Append new item to synopsis set  $\mathcal{S}$ 
36: end if
37: End Procedure

38: Procedure RefreshSynopsis (object  $o_i$ , traversed road  $q$ , timestamp  $\tau$ )
39: State:  $\mathcal{S} = \{S_i : \text{synopsis of items } \langle q_j, t_{in}, t_{out} \rangle \text{ for each road } q_j \text{ traversed by } o_i\}$ 
40:  $S_i \leftarrow \mathcal{S}[o_i]$ ; //Fetch the synopsis maintained for object  $o_i$ 
41:  $\langle q_j, t_{in}, t_{out} \rangle \leftarrow S_i.\text{top}()$ ; //Identify road  $q_j$  where  $o_i$  was traveling until now
42: if  $q \neq q_j$  then //Object just turned into another road
43:    $S_i.\text{pop}()$ ;
44:    $S_i.\text{push}(\langle q_j, t_{in}, \tau \rangle)$ ; //Update synopsis with exit time from previous road
45:    $S_i.\text{push}(\langle q, \tau, \text{nil} \rangle)$ ; //Entrance to new road  $q$  is appended to synopsis
46: end if
47: End Procedure

```



Σχήμα 4.3: Λίστα τροχιάς κινούμενου αντικειμένου



Σχήμα 4.4: Σύνοψη κινούμενου οχήματος

οποία μοντελοποιείται σαν λίστα διαδοχικών ενημερώσεων θέσης.

Παράλληλα, τηρείται σύνοψη (*synopsis*) κάθε εξελισσόμενης τροχιάς (Σχήμα 4.4), η οποία καταγράφει από ποιούς δρόμους πέρασε το όχημα και για πόσο χρονικό διάστημα κινήθηκε σε καθέναν απ' αυτούς (Αλγόριθμος 2, γραμμή 22,38-46). Ουσιαστικά, κάθε φορά που ένα κινούμενο όχημα αλλάζει δρόμο και μεταβαίνει από έναν με qid_1 σε έναν με qid_2 , προσθέτουμε ένα καινούργιο κομμάτι στην σύνοψη με το αντίστοιχο qid_2 και κλείνουμε το χρονικό διάστημα παραμονής του κινητού στον δρόμο qid_1 . Η σύνοψη μοντελοποιείται σαν μία λίστα από "κομμάτια", όπου το κάθε "κομμάτι" έχει σαν στοιχεία (i) το αναγνωριστικό του δρόμου qid και (ii) το χρονικό διάστημα που κινούταν το όχημα πάνω στον δρόμο qid .

Τέλος, σε κάθε κύκλο εκτέλεσης κατασκευάζεται ιστογράμμα (με b κάδους) για το πλήθος των τροχιών που αφορούν κάθε άξονα. Πρακτικά, αυτό χρησιμεύει στον προσδιορισμό αξόνων με μικρό δείγμα τροχιών, ώστε να αποφευχθεί τυχόν απόρριψή τους. Περισσότερα περί του ιστογράμματος που τηρείται θα αναφερθούν στο χωρίο που αφορά την απόρριψη φόρτου.

4.5 Φάση αποτίμησης ερωτημάτων

Ο ψευδοκώδικας για τη φάση αποτίμησης ερωτημάτων δίνεται στον Αλγόριθμο 3. Εφόσον με την έλευση νέων στιγμάτων υπολογίζεται η εκάστοτε στιγμιαία ταχύτητα \vec{v} του αντικειμένου, ενώ επιπλέον αντιστοιχίζεται σε οδικό άξονα qid , αυτά τα στοιχεία θα μπορούσαν να είναι

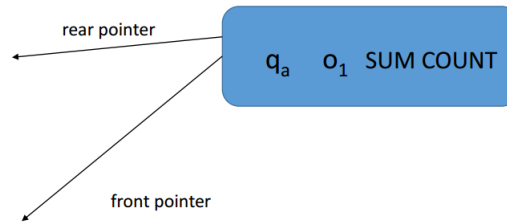
Algorithm 3 Trajectory-aware Load Shedding

```

1: Function Evaluation (query  $q$ , current timestamp  $\tau_c$ )
2: Input: Query specification  $q : \langle \omega, \beta, \tau_0 \rangle$  with sliding window initiated at time  $\tau_0$ 
3: State:  $\mathcal{K} = \{\forall \text{ query } q, \{o_i : \text{object } o_i \text{ has traversed road } q \text{ during window } q.\omega\}\}$ 
4: State:  $\mathcal{U} = \{u : \forall \text{ object } o_i, \text{ set } u=1 \text{ if } o_i \text{ is kept; } u=0 \text{ if evicted by load shedding}\}$ 
5: Output:  $v \leftarrow 0$ ; //Average speed measured along road  $q$  over interval  $q.\omega$ 
6:  $k \leftarrow 0$ ; //Number of objects contributing to speed measurement along road  $q$ 
   //Result is only computed upon each window slide  $q.\beta$  as specified by query  $q$ 
7: if  $(\tau_c \geq \tau_0 + q.\omega) \wedge ((\tau_c - \tau_0) \% q.\beta = 0)$  then
8:   if  $\mathcal{K}[q].\text{size}() > 0$  then //There are objects recently moving along road  $q$ 
9:     for each object  $o_i \in \mathcal{K}[q]$  do
10:       $t^r \leftarrow$  timestamp at trajectory  $T_i$  pointed to by rear pointer of  $o_i$  in  $\mathcal{K}[q]$ ;
11:      if  $t^r < \tau_c - q.\omega$  then
12:         $\mathcal{K}[q].\text{erase}(o_i)$ ; //Trajectory of  $o_i$  has become obsolete for query  $q$ 
13:      else
14:        Rearrange existing rear & front pointers from  $\mathcal{K}[q]$  to trajectory  $T_i$ ;
15:        Calculate average  $V_{avg}$  from partial aggregates retained for  $o_i$  in  $\mathcal{K}[q]$ ;
16:         $v \leftarrow v + V_{avg}$ ; //Sum up speed values from each contributing trajectory
17:         $k \leftarrow k + 1$ ;
18:      end if
19:    end for
20:  end if
21: end if
22: return  $v/k$ ; //Average speed over all  $k$  available trajectories across road  $q$ 
23: End Function

24: Function rankTrajectory (object  $o_i$ )
25: State:  $\mathcal{S} = \{S_i : \text{synopsis of items } \langle q_j, t_{in}, t_{out} \rangle \text{ for each road } q_j \text{ traversed by } o_i\}$ 
26:  $S_i \leftarrow \mathcal{S}[o_i]$ ; //Fetch the synopsis maintained for object  $o_i$ 
27:  $\rho \leftarrow 0$ ; //Initialize score for object  $o_i$ 
28:  $a \leftarrow 1/2$ ; //Ageing-aware weight assigned to each part of the synopsis
29:  $k \leftarrow 1$ ; //Controls the ageing of the weight applied to each part
30:  $\Delta\tau \leftarrow$  overall duration of trajectory synopsis  $S_i$ ;
31: for each  $\langle q_j, t_{in}, t_{out} \rangle \in S_i$  by reverse chronological order do
32:   if  $q_j \neq 0$  then //Trajectory has traversed a road  $q_j$  of interest
33:      $\rho \leftarrow \rho + a^k \cdot (t_{out} - t_{in}) / \Delta\tau$ ; //Use normalized duration for each part
34:      $k \leftarrow k + 1$ ;
35:   end if
36: end for
37: return  $\rho$ ; //Expected utility of trajectory segments from object  $o_i$ 
38: End Function

```



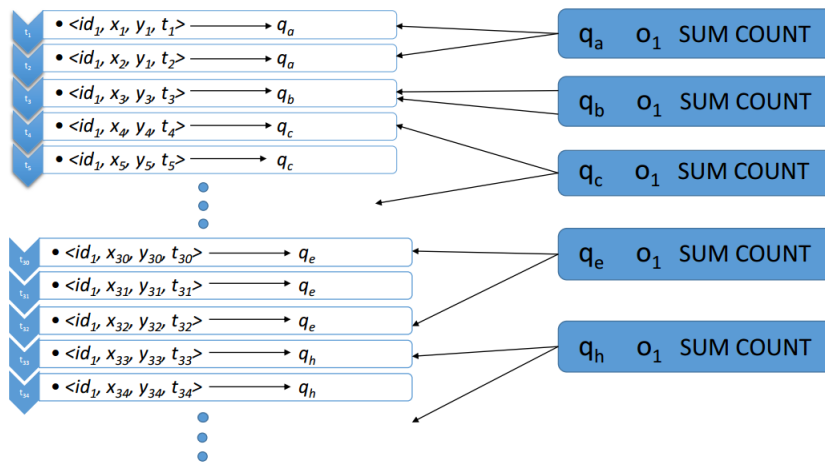
Σχήμα 4.5: Δομή EvalStruct

αμέσως διαθέσιμα για την εκτίμηση μέσων ταχυτήτων. Κάνοντας την παραδοχή πως όταν ένα όχημα εξέλθει από έναν δρόμο qid , τότε δεν θα επανέλθει σύντομα σ' αυτόν, σε κάθε κομμάτι τροχιάς ενός αντικειμένου oid που κινείται στο συγκεκριμένο qid , κρεμάμε δύο δείκτες: (i) έναν στο πρώτο στίγμα της τροχιάς εντός του δρόμου qid , και (ii) έναν άλλον στο τελευταίο στίγμα της τροχιάς αμέσως πριν βγει από τον qid . Άρα αν μία τροχιά πέρασε πρόσφατα από n δρόμους, τότε θα κρέμονται από αυτήν $2n$ δείκτες. Τηρείται λοιπόν μία δομή δεδομένων γι' αυτά τα επιμέρους στατιστικά στοιχεία κίνησης (*partial aggregates*), η οποία πρακτικά κρατάει σε κάθε δρόμο ποια κινούμενα αντικείμενα υπάρχουν κάθε στιγμή, και η οποία ενημερώνεται σε κάθε κύκλο εκτέλεσης με τα πιο πρόσφατα στοιχεία.

Υπάρχει, λοιπόν, μία δομή η οποία ονομάζεται EvalStruct (Σχήμα 4.5) και η οποία έχει δύο δείκτες που δείχνουν σε δύο πλειάδες στη λίστα της τροχιάς ενός κινούμενου οχήματος oid . Από κάθε αντικείμενο oid “κρέμονται” τόσα EvalStructs, όσα είναι και τα ερωτήματα/δρόμοι qid στα οποία έχει βρεθεί (Σχήμα 4.6). Η δομή αυτή ενημερώνεται καθώς εισέρχονται νέες πλειάδες και ενημερώνεται η λίστα του κάθε κινούμενου οχήματος, και επιπρόσθετα, κατά την διαδικασία αυτή κρατείται και ένα *partial sum* και ένα *partial count*, τα οποία αφορούν τις πλειάδες του αντικειμένου oid οι οποίες ανήκουν σε κάποιο συγκεκριμένο ερώτημα qid (Αλγόριθμος 2, γραμμές 23-29 και Αλγόριθμος 3, γραμμές 14). Ένα instance μίας δομής EvalStruct το οποίο αφορά κάποιο αντικείμενο oid , που κινείται σε έναν οδικό άξονα qid , αποδεσμεύεται μόλις ο δείκτης που δείχνει την πιο πρόσφατη πλειάδα εκπέσει από το χρονικό παράθυρο του δρόμου qid (Αλγόριθμος 3, γραμμές 11-12). Φυσικά, με αυτή την προσέγγιση ανά πάσα στιγμή είναι δυνατή η εύρεση της μερικής μέσης ταχύτητας (*partial average speed*) ενός οχήματος, σε έναν δρόμο qid .

Η αποτίμηση ερωτημάτων συνίσταται στον συναθροιστικό υπολογισμό της μέσης ταχύτητας κίνησης ανά δρόμο βάσει των επιμέρους στιγμιαίων ταχυτήτων που έχουν συλλεγεί πρόσφατως (ανάλογα με το εύρος ω του παραθύρου που θέτει κάθε ερώτημα). Υιοθετώντας μία τακτική *lazy evaluation*, οι σχετικοί δείκτες αναπροσαρμόζονται κατά την φάση της ενημέρωσης (Αλγόριθμος 2, γραμμές 23-29), ενώ ταυτόχρονα παρέχουν την συνεισφορά καθενός αντικειμένου στις μέσες ταχύτητες των αξόνων τους οποίους διέσχισε πρόσφατα.

Για να δίνει το σύστημα απαντήσεις στα ενεργά ερωτήματα (*φάση αποτίμησης*), θα πρέπει ανά πάσα στιγμή να γνωρίζει σε κάθε ερώτημα/οδικό άξονα qid ποια αντικείμενα κινούνται. Έχοντας αυτή της πληροφορία σε συνδυασμό και με την δομή EvalStruct, μπορεί πολύ



Σχήμα 4.6: Σύνδεση του EvalStruct και της λίστας τροχιάς ενός κινούμενου αντικειμένου

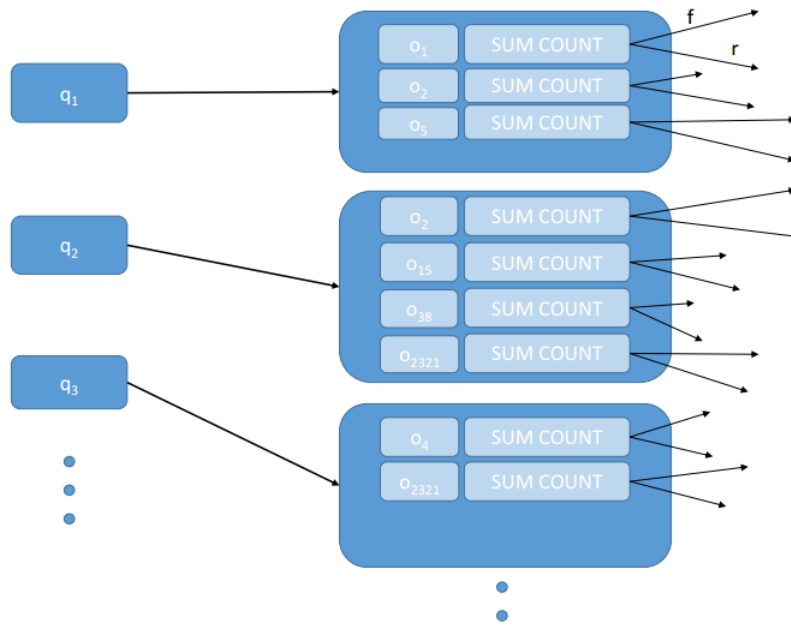
εύκολα να υπολογίζει την μέση ταχύτητα κάθε αντικειμένου σε κάποιον οδικό άξονα qid . Έπειτα αθροίζοντας τις μέσες ταχύτητες των αντικειμένων που κινούνται σε ένα ερώτημα qid υπολογίζει την ζητούμενη συνολική μέση ταχύτητα κίνησης στο qid .

Η πληροφορία για το ποια αντικείμενα κινούνται σε ποια ερωτήματα σχηματίζεται αυξητικά (*incrementally*) σε μία δομή που ονομάζεται K (Σχήμα 4.7) (*Keeper of trajectory movements in each road*). Η K μοντελοποιείται σαν μία map δομή μέσα σε μία άλλη map δομή. Στο εξωτερικό map κάθε εγγραφή έχει ζεύγη κλειδιού–τιμής, με κλειδί το μοναδικό αναγνωριστικό του ερωτήματος qid και τιμή μία άλλη map δομή.

Το δεύτερο εσωτερικό map είναι αυτό που κρατάει τα αντικείμενα που κινούνται στον δρόμο με αναγνωριστικό qid . Αυτό έχει ζεύγη κλειδιού–τιμής, όπου το κλειδί είναι το μοναδικό αναγνωριστικό oid του αντικειμένου, που κινείται στον δρόμο και ως τιμή έχει ένα instance της δομής EvalStruct. Επομένως, διατρέχοντας με ένα πέρασμα την δομή K , το σύστημα δίνει απαντήσεις (Αλγόριθμος 3, γραμμές 8-20) στα ερωτήματα που ικανοποιούν την συνθήκη που ορίζει τότε πρέπει να λάβουν απάντηση (Αλγόριθμος 3, γραμμή 7).

4.6 Φάση απόρριψης φόρτου

Αρχικά, το σύστημα ξεκινά να λειτουργεί με όλες τις διαθέσιμες τροχιές, χωρίς να απορρίπτει καθόλου στοιχεία στην είσοδο. Σε κάθε κύκλο εκτέλεσης, ο μετρούμενος φόρτος L εκφράζει το κόστος σε κύκλους μηχανής (CPU) που απαιτούνται για την ενημέρωση, την αποτίμηση, αλλά και την διαδικασία απόρριψης φόρτου (εφόσον αυτή εκτελείται). Αν το σύστημα εξετάζει κάθε φορά όλες τις τροχιές, πιθανόν το L να προκύπτει συχνά μεγαλύτερο του C . Τότε προκρίνεται η εξέταση ενός μόνο μέρους των δεδομένων εισόδου, επιλέγοντας με σημασιολογικό (και όχι τυχαίο) τρόπο τις κατάλληλες τροχιές (*object-level shedding*). Αυτές μόνο θα είναι όσες θα ενημερωθούν και θα χρησιμεύσουν στην αποτίμηση κατά τον επόμενο κύκλο εκτέλεσης. Για τις υπόλοιπες τροχιές, κάθε εισερχόμενο στίγμα μπορεί να απορρίπτεται χωρίς εξέταση (*position-level shedding*), εκτός εάν το όχημα μόλις έστριψε σε άλλο



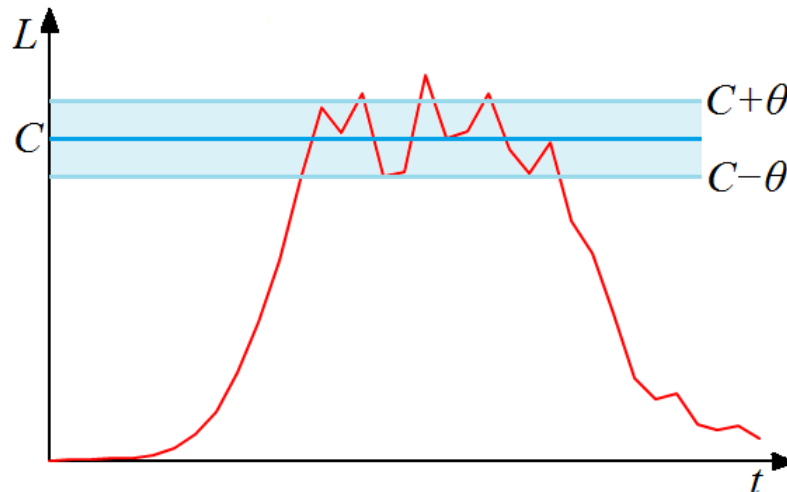
Σχήμα 4.7: Δομή για τήρηση αντικειμένων που κινούνται σε κάθε ενεργό ερώτημα

δρόμο. Κάτι τέτοιο είναι αναγκαίο, διότι εξετάζοντας πλήρως μόνο τα πρώτα στίγματα κάθε δρόμου που εισέρχεται μία ανενεργή τροχιά καθίσταται εφικτό να τηρείται πλήρως η σύνοψή της. Σε περίπτωση που μία τροχιά καταστεί εκ νέου ενεργή, είναι σαφές ότι δεν θα υπάρχει καταγεγραμμένη η πρόσφατη ιστορία της. Για να υπάρχει λοιπόν δυνατότητα υπολογισμού της στιγμιαίας ταχύτητας εκείνη την στιγμή, γίνεται η παραδοχή ότι για όλες τις τροχιές (ενεργές ή μη) πάντοτε τηρείται το αμέσως προηγούμενο γνωστό στίγμα. Με την τήρηση αυτής της πληροφορίας μάλιστα, είναι δυνατή η εύρεση κατά 100% της στιγμιαίας ταχύτητας κάθε στίγματος, που επεξεργαζόμαστε πλήρως.

Προκειμένου να μην ενεργοποιείται άσκοπα η διαδικασία απόρριψης φόρτου, τίθεται μία παράμετρος ανοχής θ (% C) ως προς την ζητούμενη χωρητικότητα C . Όπως φαίνεται στο Σχήμα 4.8, ορίζεται έτσι μία ζώνη ($C - \theta$, $C + \theta$), η οποία περιγράφει τα “φυσιολογικά” επίπεδα στα οποία πρέπει να κινείται ο φόρτος L για να μην τρέχει ο load shedder. Η αναπροσαρμογή φόρτου είναι απαραίτητη όταν παρατηρείται:

- είτε υπερφόρτωση ($L > C + \theta$), οπότε η μέθοδος διατηρεί λιγότερες τροχιές για αποτίμηση ερωτημάτων (Αλγόριθμος 1, γραμμή 33,38),
- είτε υποφόρτωση ($L < C - \theta$), οπότε υπάρχει περιθώριο αύξησης του δείγματος των τροχιών για βελτίωση της ποιότητας των αποτελεσμάτων (Αλγόριθμος 1, γραμμή 33,38).

Για τον προσδιορισμό του πλήθους των τροχιών που προσθαφαιρούνται υιοθετείται μία πολιτική αυξομείωσης κατά βήματα (*step-wise*) με τρεις εναλλακτικές επιλογές. Έτσι, το βήμα s εκφράζει ποσοστό % επί του αριθμού n των τροχιών που είτε (i) έχουν εμφανιστεί έως τώρα στο σύστημα, είτε (ii) του συνολικού προσδοκώμενου αριθμού τροχιών N , είτε (iii) του αριθμού των “χρήσιμων” τροχιών που μόλις συνεισέφεραν στην αποτίμηση κατά τον



Σχήμα 4.8: Συνθήκη ενεργοποίησης για απόρριψη φόρτου

τρέχοντα κύκλο εκτέλεσης. Η διαφοροποίηση στον ορισμό του n επηρεάζει την ταχύτητα σύγκλισης του αλγορίθμου απόρριψης φόρτου, δηλ. πόσο γρήγορα ο παρατηρούμενος φόρτος L θα επανέλθει εντός της ζώνης $(C - \theta, C + \theta)$. Ανάλογα με την κατάσταση του φόρτου (υπερφόρτωση ή υποφόρτωση) τρέχει η διαδικασία Shedding με το κατάλληλο βήμα s (θετικό ή αρνητικό) (Αλγόριθμος 1, γραμμές 32,40).

Σημειώνεται ότι στην περίπτωση που $L < C - \theta$ και ο μετρούμενος φόρτος του συστήματος συνεχώς μειώνεται επειδή δεν υπάρχουν πολλά κινούμενα οχήματα (λ.χ. νυκτερινές ώρες), τότε η λειτουργία απόρριψης πρέπει να απενεργοποιείται (Αλγόριθμος 1, γραμμή 42-44). Βεβαίως, λόγω του αυτορρυθμιζόμενου χαρακτήρα του συστήματος, η απόρριψη τροχιών θα τεθεί πάλι σε ισχύ αν τυχόν σε επόμενο κύκλο ξεπεραστεί το άνω φράγμα της ζώνης ανοχής ($L > C + \theta$) (Αλγόριθμος 1, γραμμές 34-36).

Ακόμα, στον Αλγόριθμο 4 φαίνεται ο ψευδοκώδικας για την φάση της απόρριψης φόρτου. Η σημασιολογική επιλογή των καταλληλότερων τροχιών για την επεξεργασία βασίζεται σε δύο κριτήρια:

1. Το πρώτο προκρίνει την διατήρηση τροχιών που διέρχονται από οδικούς άξονες με “μικρό” δείγμα οχημάτων, ώστε να είναι εφικτό να εκτιμηθεί στοιχειωδώς η μέση ταχύτητα. Η επιλογή του ορίου βάσει του οποίου το δείγμα θεωρείται “μικρό”, βασίζεται σε ιστόγραμμα τιμών του πλήθους τροχιών ανά άξονα. Αυτό μπορεί να είναι απλουστευμένο (ένας κάδος, οπότε λαμβάνεται ως όριο η μέση τιμή ή ο διάμεσος) ή σε percentiles με b κάδους (λ.χ. αν $b = 5$, τότε ως όριο λαμβάνεται η μέγιστη τιμή του πρώτου quantile) (Αλγόριθμος 4, γραμμές 11-25).
2. Το δεύτερο κριτήριο αξιοποιεί την βαθμονόμηση τροχιών με βάση το πλήθος των οδικών αξόνων από τους οποίους έχει διέλθει κάθε όχημα και το διάστημα παραμονής του σ' αυτόν, οπότε ευνοούνται τροχιές που εμπίπτουν σε πολλαπλά ερωτήματα (δηλ. έχουν διασχίσει πολλούς οδικούς άξονες). Αυτή η βαθμονόμηση επιτυγχάνεται εύκολα χάρη

Algorithm 4 Trajectory-aware Load Shedding

```

1: Function Shedding (integer  $N$ , integer  $n$ , #buckets  $b$ , step  $s$ )
2: Input: Maximum number  $N$  of monitored trajectories
3: Input: Number  $n$  of trajectories retained in the current execution cycle
4: Input: Number  $b$  of buckets used in trajectory histogram
5: Input: Percentage  $s$  in  $N$  trajectories to omit (overload) or take in (underload)
6: State:  $\mathcal{T} = \{T_i : \text{currently maintained trajectory for each moving object } o_i\}$ 
7: State:  $\mathcal{K} = \{\forall \text{ query } q, \{o_i : \text{object } o_i \text{ has traversed road } q \text{ during window } q.\omega\}\}$ 
8: Output:  $\mathcal{U} = \{u : \forall \text{ object } o_i, \text{ set } u=1 \text{ if } o_i \text{ is kept; } u=0 \text{ if evicted by shedding}\}$ 
9:  $\mathcal{U}.\text{resetAll}()$ ; //Initially, mark all trajectories as non utilized
10:  $n \leftarrow n + s \cdot N$ ; //Amount of trajectories to retain for the next evaluation cycle
    //Criterion #1: Preserve trajectories useful to queries with few available samples
11:  $\mathcal{H} \leftarrow \emptyset$ ; //Equi-width histogram (with  $b$  buckets) of trajectory counts per query
12: for each query  $q \in \mathcal{K}$  do
13:    $\mathcal{H}.\text{insert}(\mathcal{K}[q].\text{size}())$ ; //Trajectory count for  $q$  inserted into histogram
14: end for
15:  $\xi \leftarrow \max(\{\text{values in bucket } \mathcal{H}(1)\})$ ; //Limit set by the max value in the first bucket
16: for each query  $q \in \mathcal{K}$  do
17:   if  $\mathcal{K}[q].\text{size}() \leq \xi$  then //Queries with  $\leq \xi$  trajectories should take precedence
18:     for each object  $o_i \in \mathcal{K}[q]$  do
19:       if  $n > 0$  then
20:          $\mathcal{U}[o_i] \leftarrow 1$ ; //Trajectory of  $o_i$  survives; will be utilized in the next cycle
21:          $n \leftarrow n - 1$ ;
22:       end if
23:     end for
24:   end if
25: end for
    //Criterion #2: Rank trajectories according to their recent movement across roads
    //Do not consider again trajectories already qualifying to Criterion #1
26:  $R \leftarrow \emptyset$ ; //Temporary priority queue with trajectories ranked by their current utility
27: for each object  $o_i \in \mathcal{T} \setminus \{o_j : \mathcal{U}[o_j] = 1\}$  do
28:    $\rho_i \leftarrow \text{rankTrajectory}(o_i)$ ;
29:    $R.\text{insert}(\langle o_i, \rho_i \rangle)$ ; //R keeps trajectory rankings in descending order
30: end for
31: while  $n > 0$  do //Pick trajectories with top rankings to reach target count  $n$ 
32:    $\mathcal{U}[R.\text{top}()] \leftarrow 1$ ;
33:    $R.\text{pop}()$ ;
34:    $n \leftarrow n - 1$ ;
35: end while
36: return  $\mathcal{U}$ ; //Trajectories to be utilized in query evaluation at the next cycle
37: End Function

```



Σχήμα 4.9: Πίνακας αληθείας της απόρριψης φόρτου

στις τηρούμενες συνόψεις τροχιών (Αλγόριθμος 4, γραμμές 26-37).

Για την βαθμονόμηση της κάθε τροχιάς γίνεται χρήση της ρουτίνας `rankTrajectory` (Αλγόριθμος 3, γραμμές 24-38). Αυτή επιστρέφει ένα *weighted* άθροισμα, που το κάθε στοιχείο του είναι το γινόμενο του συντελεστή a^k , ο οποίος αφορά στην χρονικό διαχωρισμό των κομματιών κάθε τροχιάς προμηδοτώντας τα πιο πρόσφατα, και της *κανονικοποιημένης* διάρκειας του οχήματος σε κάθε δρόμο *gid* που πέρασε.

Μόλις εκτελεστεί η ρουτίνα απόρριψης φόρτου, *Shedding* (Αλγόριθμος 4), αυτό που επιστρέφει είναι ένας πίνακας U (Σχήμα 4.9, Αλγόριθμος 4, γραμμή 8) που έχει τόσες εγγραφές όσες και ο τρέχων αριθμός τροχιών που υπάρχουν στο σύστημα. Όταν η εγγραφή μίας τροχιάς σε αυτόν τον πίνακα ισούται με την μηδέν, αυτό σημαίνει ότι στον επόμενο κύκλο εκτέλεσης (*execution cycle*), και κατά την φάση της ενημέρωσης αυτή η τροχιά δε θα ενημερωθεί πλήρως. Θα ενημερωθεί η σύνοψή της μόνο στην περίπτωση που το νεοεισερχόμενο στίγμα είναι το πρώτο στο καινούργιο δρόμου που αυτή εισήλθε. Απεναντίας, όταν η εγγραφή μίας τροχιάς ισούται με το ένα, τότε κατά τον επόμενο κύκλο εκτέλεσης αυτή η τροχιά θα ενημερωθεί κανονικά. Κατά την φάση αποτίμησης γίνεται χρήση οποιασδήποτε πληροφορίας υπάρχει τη δεδομένη στιγμή στα παράθυρα, χωρίς να είναι απαραίτητο αυτή να προέρχεται από ενεργές τροχιές (δηλαδή τροχιές με μονάδα στον πίνακα της απόρριψης φόρτου). Η απόρριψη φόρτου καταλήγει να είναι η λήψη μίας απόφασης: Ενημέρωσε και εξέτασε κάποια τροχιά ή αγνόησέ την.

Κεφάλαιο 5

Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό παρουσιάζεται η πειραματική αξιολόγηση και ο έλεγχος σωστής λειτουργίας του αλγορίθμου. Γίνεται περιγραφή των χαρακτηριστικών των αρχείων εισόδου, των δεδομένων που χρησιμοποιήθηκαν και γίνεται παράθεση των συγκριτικών πειραμάτων που εκτελέστηκαν, με βάση τα οποία γίνεται αξιολόγηση τόσο των επιδόσεων όσο και των αποδόσεων του αλγορίθμου.

5.1 Πειραματικό πλαίσιο

Στην ενότητα αυτή αξιολογείται πειραματικά ο αλγόριθμος που αναπτύχθηκε και υλοποιήθηκε στο κεφάλαιο 4. Όλες οι δομές υλοποιήθηκαν στη γλώσσα προγραμματισμού C++ και τα πειράματα εκτελέστηκαν σε λειτουργικό σύστημα Linux σε προσωπικό υπολογιστή Intel(R) Core(TM) i5, 2.6 GHz με μνήμη RAM 8 Gb.

5.1.1 Παραγωγή συνθετικών δεδομένων

Τα πειραματικά δεδομένα των κινούμενων αντικειμένων παρήχθησαν βάσει ενός ψηφιακού χάρτη οδικού δικτύου του πολεοδομικού συγκροτήματος Αθηνών. Το συγκεκριμένο ψηφιακό υπόβαθρο αφορά το βασικό οδικό δίκτυο της πρωτεύουσας, προέρχεται από χάρτες κλίμακας 1:5000 και τηρείται σε διανυσματική (vector) μορφή, ενώ καλύπτει έκταση περίπου 625 τ.χλμ. Οι οδικοί άξονες διακρίνονται σε κατηγορίες (λεωφόροι ταχείας κυκλοφορίας, κύριες και δευτερεύουσες αρτηρίες, βοηθητικοί δρόμοι) και χαρακτηρίζονται από την μέση ταχύτητα κίνησης των οχημάτων στη διάρκεια της ημέρας, όπως έχει προκύψει από επιτόπιες μετρήσεις. Αυτό ακριβώς το στοιχείο μπορεί να αξιοποιηθεί για τον υπολογισμό του μέσου χρόνου διαδρομής ενός οχήματος κατά μήκος των συνδέσμων του δικτύου, καθιστώντας αυτήν τη γεωγραφική βάση δεδομένων κατάλληλη για υπολογισμό της βέλτιστης διαδρομής (shortest path) μέσα στην πόλη. Με χρήση του λογισμικού ArcView GIS 3.2 και της επέκτασής του Network Analyst 1.0b, δημιουργήθηκαν συνολικά 100000 τροχιές ισάριθμων αντικειμένων, θέτοντας ως προέλευση και προορισμό κάθε διαδρομής τυχαία επιλεγμένα ζεύγη κόμβων του δικτύου. Οι κινήσεις διεξάγονται ως επί το πλείστον ακτινικά, θεωρώντας ότι τα περισσότερα αντικείμενα ξεκινούν από την περιφέρεια, διέρχονται από το κέντρο της πόλης και κατευθύνονται προς

κάποιο προάστιο. Η διάρκεια κάθε τροχιάς διαφέρει και κυμαίνεται από 1090 έως 4060 δευτερόλεπτα. Αντί να ξεκινάνε την κίνησή τους όλα τα αντικείμενα την χρονική στιγμή $t = 0$, τα στίγματά τους μετατοπίστηκαν κατά μία τυχαία μεταβλητή με σκοπό να λάβουν χώρα μέσα σε μία χρονική περίοδο $T = 4200$ δευτερολέπτων. Κατά την χρονική στιγμή $t = 2100$ sec είναι σε εξέλιξη όλες οι τροχιές των αντικειμένων και μάλιστα βρίσκονται στο μέσο της τροχιάς τους. Αρχικά, κανένα όχημα δεν κινείται, αλλά σταδιακά όλο και περισσότερα τίθενται σε κίνηση. Πρακτικά, στο μέσο του πειράματος όλα τα αντικείμενα κινούνται και σταδιακά τερματίζουν τις διαδρομές τους με αντίστροφη σειρά από την οποία ξεκίνησαν την κίνησή τους. Η κατάσταση αυτή προσομοιώνει τις πραγματικές συνθήκες στους δρόμους της Αθήνας, ειδικά στις κύριες αρτηρίες. Επιπρόσθετα, παρήχθησαν συνολικά 1771 στατικά ερωτήματα/οδικοί άξονες για την ίδια περιοχή. Για κάθε ερώτημα υπάρχει αρχική καταγραφή για $t = 0$, δηλαδή θεωρείται πως όλα τα ερωτήματα είναι ενεργά από την αρχή εκτέλεσης του πειράματος. Όπως περιγράφηκε, σε κάθε ερώτημα “κρέμεται” ένα χωροχρονικό παράθυρο το οποίο έχει συγκεκριμένο εύρος ω και βήμα β .

5.1.2 Πειραματικά δεδομένα

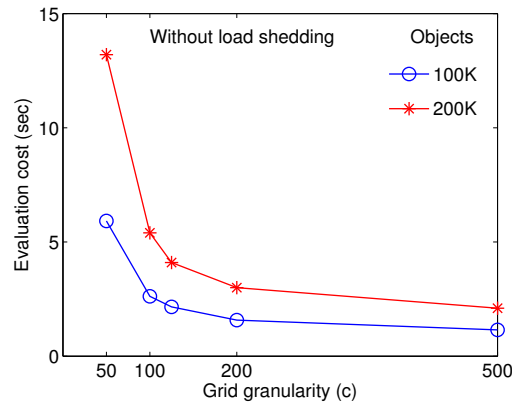
Ο αλγόριθμος απόρριψης φόρτου από τροχιές κινούμενων οχημάτων δοκιμάστηκε για διαφορετικά πλήθη οχημάτων (100000, 200000, και 500000), ενώ η χρονική διάρκεια κάθε τροχιάς κυμαίνεται μεταξύ 1090 και 4060 δευτερολέπτων. Έγιναν πειράματα επιδόσεων και ποιότητας απαντήσεων για διάφορες τιμές παραμέτρων του συστήματος:

- Κατάτμηση καννάβου $c \times c$ καννάβου.
- Επιθυμητή χωρητικότητα συστήματος C .
- Βήμα προσθαφαίρεσης τροχιών s .
- Τρόπος επιλογής ορίου για μικρό δείγμα τροχιών.
- Τιμές ανοχής θ .

Με έντονους χαρακτήρες φαίνονται οι τυπικές τιμές που χρησιμοποιούνται στα πειράματα:

Κατάτμηση καννάβου $c \times c$ καννάβου	50, 100, 125, 200, 500
Επιθυμητή χωρητικότητα συστήματος C ως προς L_{peak}	20%, 33%, 50%, 67% , 80%
Βήμα προσθαφαίρεσης τροχιών s	2%, 5% , 10%, 15%
Τρόπος επιλογής ορίου για μικρό δείγμα τροχιών	<i>mean</i> , <i>median</i> , <i>b-tile</i>
Τιμές ανοχής θ επί του C	2%, 5%, 10% , 15%

Πίνακας 5.1: Παράμετροι πειραμάτων



Σχήμα 5.1: Κλιμάκωση χρόνου αποτίμησης για διαφορετικό διάφορες υποδιαιρέσεις καννάβου

5.2 Αξιολόγηση αποτελεσμάτων

Κάθε πείραμα εκτελέστηκε για όλη τη χρονική διάρκεια των δεδομένων. Στη συνέχεια υπολογίστηκαν μέσες τιμές των μετρήσεων για τα ακόλουθα μεγέθη:

- Ο μέσος χρόνος εκτέλεσης (επίδοση) του αλγορίθμου, για να δοθεί απάντηση στο σύνολο των ερωτημάτων, για την φάση ενημέρωσης, την φάση αποτίμησης ερωτημάτων και την φάση της απόρριψης φόρτου ξεχωριστά, ανά κύκλο εκτέλεσης.
- Η ακρίβεια του αλγορίθμου, η οποία υπολογίζεται μετρώντας την απόκλιση των εκτιμώμενων μέσων ταχυτήτων από αυτές που βασίζονται σε εξαντλητική αποτίμηση.

5.2.1 Διαστασιολόγηση καννάβου

Χρησιμοποιείται η κατάτμηση του χώρου σε ισομεγέθη κελιά (κάνναβος) με σκοπό να γίνεται γρηγορότερα ο εντοπισμός του αναγνωριστικού του δρόμου *qid*, στον οποίο εμπίπτει κάθε εισερχόμενη πλειάδα. Για λεπτότερες υποδιαιρέσεις του καννάβου (περισσότερα κελιά) προκύπτει λιγότερη επιβάρυνση στην φάση αποτίμησης (Σχήμα 5.1), εις βάρος όμως της μνήμης που απαιτείται για την τήρησή του. Με μία χονδροειδή εκτίμηση υποθέτοντας πως έχουμε $1771 \approx 2000$ ερωτήματα και επιπρόσθετα πως σε κάθε κελί του καννάβου κρέμονται δείκτες προς όλα τα ερωτήματα. Ο κάθε δείκτης σε ένα 64-bit σύστημα καταλαμβάνει 8 byte στη μνήμη, άρα κάθε δείκτης σε κελί για να αποθηκευτεί στη μνήμη χρειάζεται 16 KB. Επομένως, κάνοντας τους υπολογισμούς προκύπτει στο “χειρότερο” σενάριο ο πίνακας 5.2 που δείχνει τη μνήμη που καταλαμβάνει ο κάνναβος για διάφορες τιμές κατάτμησης.

Επομένως, επειδή το σύστημα που τρέχουμε τα πειράματα έχει 8 Gb RAM, και πρακτικά η παραπάνω εκτίμηση της μνήμης που πιάνει ο κάνναβος αποτελεί υπερεκτίμηση, επιλέγουμε να κατατμήσουμε τον χώρο σε 500×500 κελιά με σκοπό να γίνεται όσο το δυνατόν γρηγορότερα η ανάθεση *qid* σε κάθε εισερχόμενη πλειάδα κατά την φάση της αποτίμησης.

grid granularity	memory allocated
50 × 50	32 MB
100 × 100	160 MB
125 × 125	250 MB
200 × 200	640 MB
500 × 500	4 GB

Πίνακας 5.2: Μνήμη που καταλαμβάνει ο κάρναβος για διάφορες κατατμήσεις (χονδροκομμένος υπολογισμός)

5.2.2 Αναπροσαρμογή φόρτου συστήματος

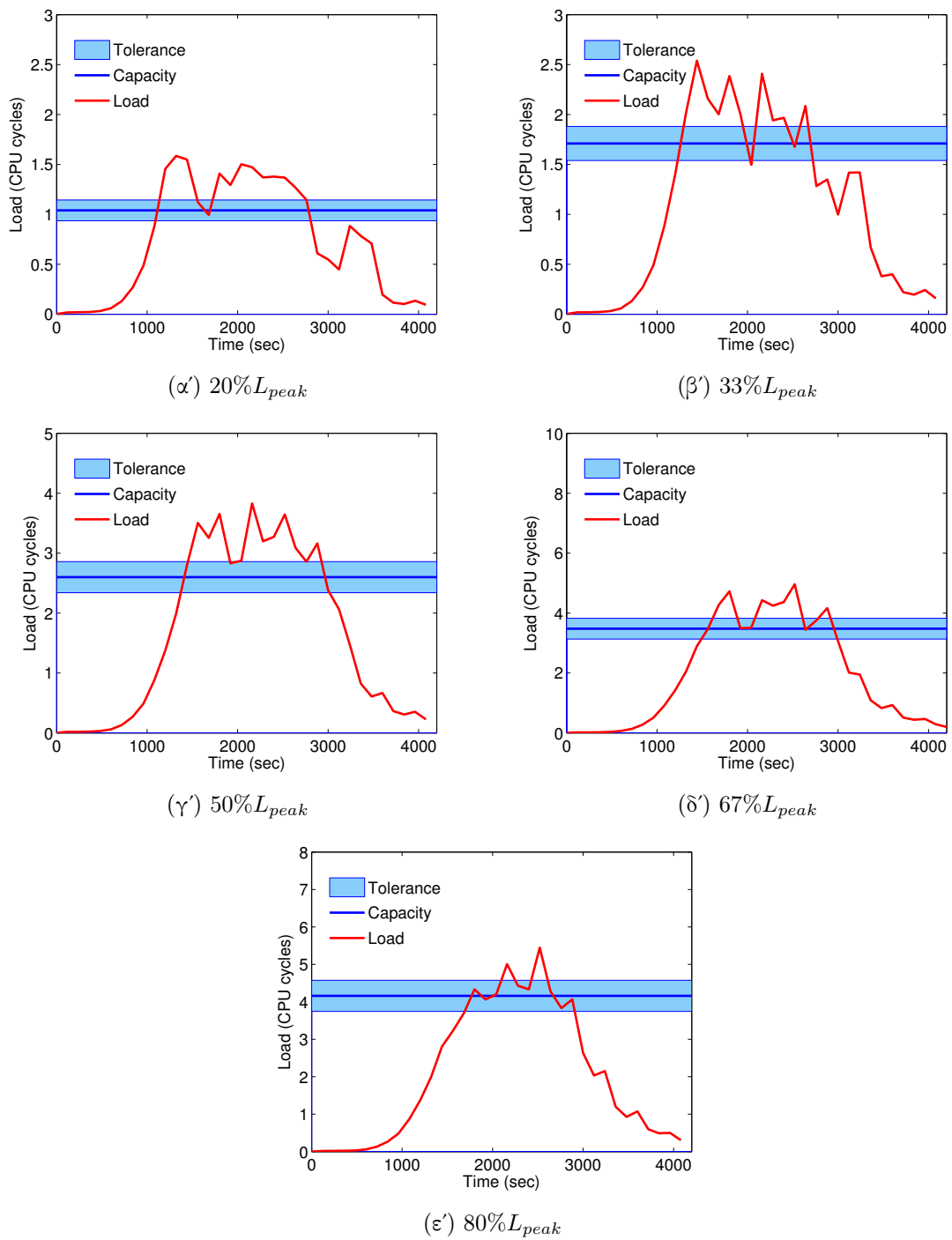
Πραγματοποιήθηκε ένα πείραμα με απενεργοποιημένο τον μηχανισμό της απόρριψης φόρτου, απ' όπου εξάγαμε το μέγιστο φόρτο (L_{peak}) που παρατηρείται στο σύστημα για τα δεδομένα εισόδου. Έπειτα διεξήχθη ένα πείραμα για τυπικές παραμέτρους (grid gran = 500, 100k αντικείμενα, $s = 5\%C$, πολιτική 1^{ou} κριτηρίου = quantile, $\theta = 10\%C$) και για διάφορες τιμές χωρητικότητας συστήματος C (20, 33, 50, 67, $80\%L_{peak}$). Ο προτεινόμενος μηχανισμός γενικά κατορθώνει να αναπροσαρμόζει δυναμικά τον φόρτο ανάλογα με τους διατιθέμενους πόρους (C), όπως φαίνεται στο Σχήμα 5.2. Ωστόσο, επιλέγουμε σαν τυπική παράμετρο χωρητικότητας το $67\%C$.

Επίσης στο Σχήμα 5.3 απεικονίζεται ο αριθμός εκτελέσεων απόρριψης φόρτου ανά χρονόσημο για τις διάφορες τιμές της χωρητικότητας C και τις υπόλοιπες παραμέτρους να λαμβάνουν τις τυπικές τιμές. Όπως φαίνεται και είναι λογικό, για το ίδιο πείραμα όσο μικρότερη είναι η τιμή της χωρητικότητας C τόσο περισσότερες φορές εκτελείται ο μηχανισμός της απόρριψης φόρτου στο χρονικό διάστημα της προσομοίωσης, με απώτερο σκοπό να επαναφέρει το φόρτο στα επιθυμητά επίπεδα.

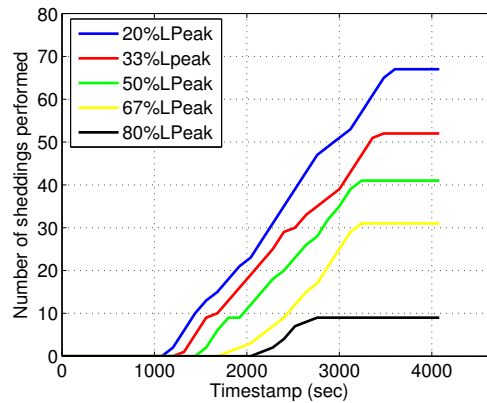
5.2.3 Επίδραση βήματος απόρριψης φόρτου

Έγινε ένα πείραμα με τις τυπικές παραμέτρους και για διάφορες τιμές βήματος απόρριψης φόρτου s . Για μεγαλύτερα βήματα s , η προσαρμογή γίνεται πιο δραστηκά, προσθαφαιρώντας περισσότερες τροχιές προς επεξεργασία. Ωστόσο, όταν το s λαμβάνει ακραίες τιμές (λίγες ή πάρα πολλές τροχιές), τότε πραγματοποιείται load shedding συχνότερα, επειδή βγαίνουμε συχνότερα εκτός ζώνης ανοχής. Όταν το βήμα s εφαρμόζεται επί του συνολικού αριθμού N τροχιών που αναμένονται από το σύστημα, τότε η σύγκλιση στην χωρητικότητα C είναι ταχύτερη (Σχήμα 5.4). Επιλέξαμε σαν τυπική παράμετρο για το βήμα απόρριψης s το 5%.

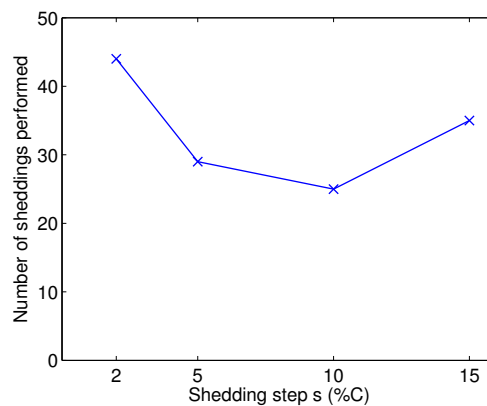
Επιπρόσθετα, για τις διάφορες τιμές του βήματος s , έγινε καταμέτρηση του μέσου χρόνου εκτέλεσης για τις τρεις επιμέρους φάσεις ανά κύκλο εκτέλεσης. Όπως ήταν αναμενόμενο η φάση της ενημέρωσης και της αποτίμησης καταλαμβάνει σεβαστό ποσοστό του χρόνου, μιας και σε αυτές γίνεται η βασική επεξεργασία και ενημέρωση των υπάρχοντων δομών. Ακόμα παρατηρείται ότι σε κάθε περίπτωση ο χρόνος που καταλαμβάνει η φάση απόρριψης φόρτου παραμένει μικρός, όπως θα έπρεπε άλλωστε. Οι επιμέρους μέσοι χρόνοι ανά κύκλο εκτέλεσης



Σχήμα 5.2: Προσαρμογή φόρτου για διάφορες τιμές της χωρητικότητας C



Σχήμα 5.3: Αριθμός εκτελέσεων απόρριψης φόρτου ανά χρονόσημο για διάφορες τιμές του C

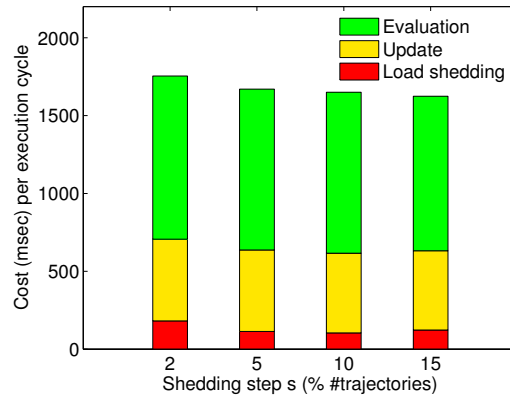


Σχήμα 5.4: Αριθμός εκτελέσεων απόρριψης φόρτου για διάφορες τιμές του s

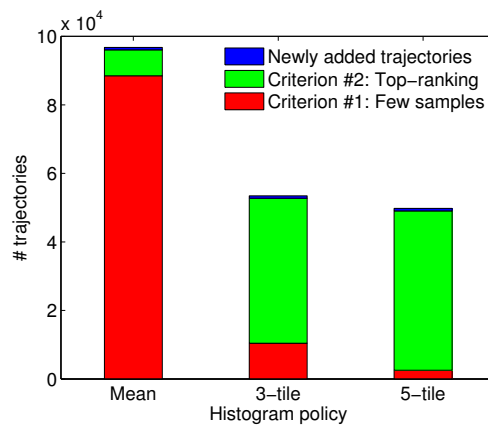
φαίνονται στο Σχήμα 5.5.

5.2.4 Ποσοστά επιλεγμένων τροχιών βάσει κριτηρίων απόρριψης φόρτου

Έγινε ένα πείραμα με τρεις διαφορετικές πολιτικές για το πρώτο κριτήριο της απόρριψης φόρτου: το μέσο όρο (mean), το 3-tile και το 5-tile. Τα αποτελέσματα του σχετικού πειράματος φαίνονται στο Σχήμα 5.6. Όπως βλέπουμε στο σχετικό σχήμα, με μπλέ χρώμα απεικονίζεται ο κατά μέσο όρο αριθμός νεοεισερχόμενων τροχιών ανά κύκλο εκτέλεσης, ο οποίος είναι σταθερός για κάθε πολιτική που υιοθετούμε. Επιπρόσθετα, με κόκκινο και πράσινο χρώμα δείχνεται ο κατά μέσο όρο αριθμός τροχιών που κρατώνται από το 1^ο και το 2^ο κριτήριο αντίστοιχα. Πρώτα γίνεται η επιλογή της κράτησης των αναγκαίων τροχιών και έπειτα απ' όσες μείνουν διαλέγονται οι καλύτερες από το δεύτερο κριτήριο, το οποίο στηρίζεται στη βαθμονόμηση τους. Όπως είναι εμφανές, στην περίπτωση που η πολιτική του 1^{ου} κριτηρίου είναι ο μέσος όρος, αυτό κρατάει πάρα πολλές τροχιές λόγω του ότι ο μέσος όρος επηρεάζεται από τις ακραίες τιμές. Αυτό έχει σαν αποτέλεσμα να μένουν πολύ λίγες τροχιές προς βαθμονόμηση από το δεύτερο κριτήριο, το οποίο είναι και πιο σημαντικό. Επομένως, η επιλογή του



Σχήμα 5.5: Μέσος χρόνος εκτέλεσης κάθε φάσης ανά κύκλο εκτέλεσης για τις διάφορες τιμές του s

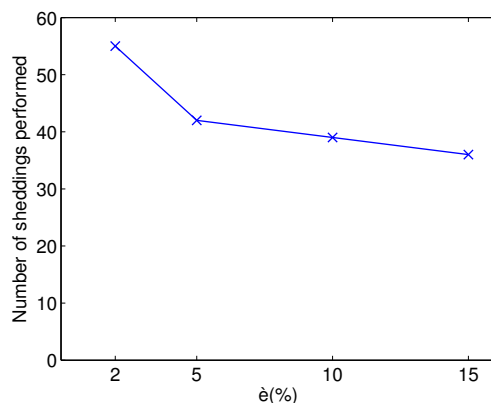


Σχήμα 5.6: Μέσος αριθμός τροχιών που κατακρατούνται από κάθε κριτήριο ανά κύκλο εκτέλεσης

μέσου όρου κρίνεται απαγορευτική. Από την άλλη, στην περίπτωση όπου η πολιτική του πρώτου κριτηρίου είναι το b -tile βλέπουμε πως όσο μεγαλύτερη είναι η τιμή του b τόσο λιγότερες τροχιές κρατώνται από το πρώτο κριτήριο και κατ'επέκταση όλο και περισσότερες μένουν για εκλογή από το δεύτερο. Ως προς την επιλογή ορίου για “μικρό” δείγμα τροχιών ανά άξονα, η πολιτική που βασίζεται σε 5-tile (quantiles) είναι εκείνη που δίνει αντιπροσωπευτικότερες τιμές για τους δρόμους που καλύπτουν αυτό το κριτήριο γιαυτό και την επιλέγουμε ως τυπική παράμετρο.

5.2.5 Επίδραση ζώνης ανοχής θ

Διεξήχθησαν πειράματα για τις τυπικές παραμέτρους αλλά για διάφορες τιμές ανοχής θ . Το συμπέρασμα που εξήχθει είναι πως, όσο μεγαλύτερη είναι η τιμή της ανοχής θ , τόσο λιγότερες φορές εκτελείται load shedding συνολικά στην διάρκεια του πειράματος. Κάτι τέτοιο συμβαίνει διότι στην περίπτωση που το θ είναι μεγάλο, είναι μεγάλη και η ζώνη ανοχής, άρα υπάρχει μεγαλύτερη πιθανότητα ο μετρούμενος φόρτος να βρίσκεται εντός της, επομένως



Σχήμα 5.7: Μέσος αριθμός τροχιών που κατακρατούνται από κάθε κριτήριο ανά κύκλο εκτέλεσης

να μην εκτελείται ο μηχανισμός απόρριψης φόρτου. Αυτό φαίνεται καλύτερα στο Σχήμα 5.7. Προς αυτή την κατεύθυνση, επιλέγεται σαν τυπική παράμετρος για τη ζώνη ανοχής το $\theta = 10\%C$.

5.2.6 Ποιότητα αποτελεσμάτων

Οι εκτιμώμενες μέσες ταχύτητες ανά άξονα εμφανίζουν μικρή απόκλιση από τις τιμές που βασίζονται σε εξαντλητική αποτίμηση. Οι αποκλίσεις είναι ελαφρώς αυξημένες σε μεγάλου μήκους άξονες όπου σημειώνονται μεγαλύτερες διακυμάνσεις ταχυτήτων. Στον Πίνακα 5.3 φαίνονται για το πείραμα που εκτελέστηκε με τις τυπικές τιμές, και για τιμές χωρητικότητας C (50, $67\%L_{peak}$), η μέση διαφορά από τις ακριβείς τιμές και η τυπική απόκλιση. Για παράδειγμα, στην περίπτωση όπου η χωρητικότητα $C = 67\%L_{peak}$, οι διαφορές των προσεγγιστικών από τις πραγματικές ταχύτητες είναι της τάξης των 1-2 km/h.

Comparison	Mean Difference	StdDev Difference
$C = 50\%L_{peak}$	0.921727017262146	1.8448065257291
$C = 67\%L_{peak}$	0.464668440564887	1.17986382514618

Πίνακας 5.3: Ποιότητα αποτελεσμάτων

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Στόχος της διπλωματικής εργασίας ήταν η κατασκευή ενός αυτόρρυθμιζόμενου μηχανισμού ο οποίος θα διαχειρίζεται πολύ μεγάλους όγκους δεδομένων, που αφορούν τροχιές που εξελίσσονται στο χρόνο. Επειδή δεν είναι εφικτό να επεξεργαστεί όλες τις τροχιές που υπάρχουν στο σύστημα και να απαντά στα ερωτήματα σε πραγματικό χρόνο, υλοποιείται ένας μηχανισμός απόρριψης ολόκληρων τροχιών ή μέρους τροχιών με απώτερο σκοπό να μειωθεί ο φόρτος του συστήματος. Από τη μελέτη σχεδίαση του συστήματος, προκύπτουν τα εξής συμπεράσματα:

- Η επιλογή καννάβου ως χωρικού ευρετηρίου και η προσπέλαση των κελιών για την ανάθεση αναγνωριστικού δρόμου σε κάθε εισερχόμενη πλειάδα που επεξεργάζεται το σύστημα, αποδείχτηκε ιδανική. Για λεπτότερες υποδιαιρέσεις του καννάβου (περισσότερα κελιά) προκύπτει λιγότερη επιβάρυνση στην φάση αποτίμησης, εις βάρος όμως της μνήμης που απαιτείται για την τήρησή του.
- Ο προτεινόμενος μηχανισμός γενικά κατορθώνει να αναπροσαρμόζει δυναμικά τον φόρτο ανάλογα με τους διατιθέμενους πόρους (C). Όσο μεγαλύτερο είναι το C , τόσο γρηγορότερα γίνεται η προσαρμογή.
- Απορρίπτει πρωτογενή πληροφορία από “ασήμαντες” τροχιές. Κρατά και τηρεί πλήρως μόνο εκείνες τις τροχιές που συνεισφέρουν τα μέγιστα στην ακρίβεια των αποτελεσμάτων.
- Το μοντέλο απόρριψης φόρτου είναι προσαρμοστικό στις συνθήκες:
 - Πλήθος αντικειμένων
 - Ρυθμός εισροής στιγμάτων
 - Αριθμός και ποικιλία ερωτημάτων
- Οι χρονικές επιδόσεις του αλγόριθμου μετρήθηκαν πειραματικά και αποδείχθηκαν ιδιαιτέρως επαρκείς. Τα αποτελέσματα της ακρίβειας των προσεγγιστικών απαντήσεων ήταν επίσης πολύ ενθαρρυντικά.

6.2 Μελλοντικές επεκτάσεις

Από τη μελέτη του συγκεκριμένου προβλήματος, προκύπτουν ενθαρρυντικές προοπτικές επέκτασής του. Συγκεκριμένα, θα ήταν δυνατό να εξεταστούν τα εξής:

- Αναθεώρηση κριτηρίων βαθμονόμησης συνόψεων με αναπροσαρμογή βαρών, μαζί με την καταγραφή της εκάστοτε κατάστασης των τροχιών (ενεργή ή μη).
- Για περαιτέρω ενίσχυση της ακρίβειας των αποτελεσμάτων θα μπορούσαν να χρησιμοποιηθούν δειγματοληπτικές μέθοδοι, ώστε για τις τροχιές που από ανενεργές μόλις κατέστησαν ενεργές να έχουν τηρηθεί περισσότερα δείγματα θέσεων.
- Υιοθέτηση μεταβλητού βήματος s , με βάση το οποίο πραγματοποιείται η προσθαφαίρεση τροχιών, ώστε να κυμαίνεται ανάλογα με την υπέρβαση φόρτου.
- Τέλος, θα είχε μεγάλο ενδιαφέρον αν ήταν δυνατόν το προτεινόμενο πλαίσιο από συγκεντρωτικό (*centralized*) να εφαρμοστεί και σε κατανεμημένο (*distributed*) περιβάλλον, όπου διάφοροι επεξεργαστικοί κόμβοι θα συνεργάζονταν στην απόρριψη τροχιών και στην κατανομή του φόρτου.

Βιβλιογραφία

- [1] D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, J. Gehrke, L. Haas, A. Halevy, J. Han, H. V. Jagadish, A. Labrinidis, S. Madden, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, K. Ross, C. Shahabi, D. Suciu, S. Vaithyanathan, and J. Widom. Challenges and Opportunities with Big Data - A community white paper developed by leading researchers across the United States, 2012. URL: <http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf> (accessed 14 July 2015).
- [2] D.J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: a New Model and Architecture for Data Stream Management. *VLDB Journal*, 12(2):120-139, August 2003.
- [3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. In *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'02)*, pp.1-16, Madison, Wisconsin, May 2002.
- [4] B. Babcock, M. Datar, and R. Motwani. Load Shedding for Aggregation Queries over Data Streams. In *Proceedings of the 20th IEEE International Conference on Data Engineering (ICDE)*, pp. 350-361, Boston, Massachusetts, March 2004.
- [5] S. Babu and J. Widom. Continuous Queries over Data Streams. *ACM SIGMOD Record*, 30 (3):109-120, September 2001.
- [6] S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S.R. Madden, V. Raman, F. Reiss, and M.A. Shah. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR)*, Asilomar, California, January 2003.
- [7] Jianjun Chen, David J. DeWitt, Feng Tian, Yuan Wang. NiagaraCQ: A Scalable Continuous Query System for Internet Databases. In *Proceeding SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, Pages 379-390, New York, NY, USA, June 2000

- [8] C. Cranor, T. Johnson, O. Spataschek, and V. Shkapenyuk. Gigascope: A Stream Database for Network Applications. In *Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data*, pp. 647-651, San Diego, California, USA June 2003.
- [9] B. Gedik, L. Liu, K.-L. Wu, and P.S. Yu. LIRA: Lightweight, Region-aware Load Shedding in Mobile CQ Systems. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, pp. 286-295, Istanbul, Turkey, April 2007.
- [10] B. Gedik, K.-L. Wu, P.S. Yu, and L. Liu. MobiQual: QoS-aware Load Shedding in Mobile CQ Systems. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, pp. 1121-1130, Cancun, Mexico, April 2008.
- [11] B. Gedik, K.-L. Wu, L. Liu, and P.S. Yu. Load Shedding in Mobile Systems with MobiQual. *IEEE Transactions on Knowledge and Data Engineering*, 23(2): 248-265, February 2011.
- [12] L. Golab and M. Tamer Ozsü. Issues in Data Stream Management. *ACM SIGMOD Record*, 32(2):5-14, June 2003.
- [13] A. Magdy, M.F. Mokbel, S. Elnikety, S. Nath, and Y. He. Mercury: A Memory-Constrained Spatio-temporal Real-time Search on Microblogs. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp. 172-183, Chicago, Illinois, USA, April 2014.
- [14] M.F. Mokbel and W.G. Aref. GPAC: Generic and Progressive Processing of Mobile Queries over Mobile Data. In *Proceedings of the 6th international conference on Mobile Data Management (MDM'05)*, pp. 155-163, Ayia Napa, Cyprus, May 2005.
- [15] M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems (GIS'03)*, pp. 110-117, New Orleans, Louisiana, USA, November 2003.
- [16] M.F. Mokbel and W.G. Aref. SOLE: Scalable On-Line Execution of Continuous Queries on Spatio-temporal Data Streams. *VLDB Journal*, 17(5): 971-995, 2008.
- [17] B. Mozafari and C. Zaniolo. Optimal Load Shedding with Aggregates and Mining Queries. In *Proceedings of the 26th International Conference on Data Engineering (ICDE)*, pp. 76-88, Long Beach, California, USA, March 2010.
- [18] B. Mozafari and C. Zaniolo. Optimal Load Shedding with Aggregates and Mining Queries. In *Proceedings of the 26th International Conference on Data Engineering (ICDE)*, pp. 76-88, Long Beach, California, USA, March 2010.

- [19] R.V. Nehme and E.A. Rundensteiner. ClusterSheddy: Load Shedding Using Moving Clusters over Spatio-temporal Data Streams. In *Proceedings of the 12th International Conference on Database Systems for Advanced Applications (DASFAA)*, pp. 637-651, Bangkok, Thailand, April 2007.
- [20] D. Pfoser and C. Jensen. Capturing the Uncertainty of Moving-Object Representations. In *Proceedings of the 6th International Symposium on Spatial Databases (SSD'99)*, pp. 111-132, Hong Kong, July 1999.
- [21] D. Pfoser, C. S. Jensen and Y. Theodoridis. Novel Approaches to the Indexing of Moving Object Trajectories. In *Proceedings of the 26th International Conference on Very Large Databases*, Cairo, Egypt, 2000.
- [22] F. Reiss and J.M. Hellerstein. Data Triage: An Adaptive Architecture for Load Shedding in TelegraphCQ. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, pp. 155-156, Tokyo, Japan, April 2005.
- [23] H. Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2): 187-260, June 1984.
- [24] M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. *ACM SIGMOD Record*, 34(4):42-47, December 2005.
- [25] N. Tatbul, U. Çetintemel, S. Zdonik, M. Cherniack, and M. Stonebraker. Load Shedding in a Data Stream Manager. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)*, pp. 309-320, Berlin, Germany, September 2003.
- [26] N. Tatbul and S. Zdonik. Window-aware Load Shedding for Aggregation Queries over Data Streams. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pp. 799-810, Seoul, Korea, September 2006.
- [27] Y. Theodoridis. Ten Benchmark Database Queries for Location-based Services. *Computer Journal*, 46(6): 713-725, 2003.
- [28] Y. Theodoridis, T. Sellis, A.N. Papadopoulos and Y.Manolopoulos. Specifications for efficient indexing in spatiotemporal databases. In *Proceedings of 10th International Conference on Scientific and Statistical Database Management (SSDBM 1998)*, pp. 123-132, Capri, Italy, 1998.
- [29] Y.-C. Tu, S. Liu, S. Prabhakar, and B. Yao. Load Shedding in Stream Databases: A Control-Based Approach. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pp. 787-798, Seoul, Korea, September 2006.
- [30] J. Zhang, M. Zhu, D. Papadias, Y. Tao, and D. L. Lee. Location-based Spatial Queries. In *Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data*, pp. 443-454, San Diego, California, June 2003.

-
- [31] Κ. Πατρούμπας. "Συστήματα Ρευμάτων Δεδομένων για Κινούμενα Αντικείμενα", ΔΠΜΣ "Γεωπληροφορική", Αθήνα 2003

Γλωσσάριο

Ελληνικός όρος

ακρίβεια
αναγνωριστικό
απόρριψη φόρτου
αποτίμηση ερωτημάτων
βήμα
διακίνηση της πληροφορίας
δειγματοληψία
δεικτοδότηση
διαχειριστής τροχιών
εξυπηρετητής
εύρος
ερώτημα διαρκείας
ερώτημα στιγμιοτύπου
ιστόγραμμα
κάνναβος
κατανεμημένο
κατώφλι
κινούμενο αντικείμενο
κλιμακωσιμότητα
κύκλος εκτέλεσης
κυλιόμενα παράθυρα
κυματίδια
μαζική επεξεργασία
μοτίβα
παράθυρα
παράθυρα ορόσημου
περιστασιακό ερώτημα
πλειάδα
προκαθορισμένο ερώτημα
προσαρμοστικότητα
ρεύμα δεδομένων

Αγγλικός όρος

accuracy
identifier
load shedding
query evaluation
slide
communication cost
sampling
indexing
trajectory manager
server
range
continuous query
snapshot/onetime query
histogram
grid
distributed
threshold
moving object
scalability
execution cycle
sliding windows
wavelets
batch processing
patterns
windows
landmark windows
adhoc query
tuple
predefined query
adaptivity
data stream

ρυθμός άφιξης	arrival rate
σημασιολογικά	semantically
σκίτσα	sketches
συγκεντρωτικό	centralized
συνάθροιση	aggregation
σύνδεση	join
σύννοψη	synopsis
τροχιά	trajectory
υπηρεσίες εντοπισμού	location-based services
φάση ενημέρωσης	update phase
φάση αποτίμησης	evaluation phase
φάση απόρριψης φόρτου	load shedding phase
χρονόσημο	timestamp
χωρητικότητα	capacity

Απόρριψη φόρτου από ρεύματα τροχιάς κινούμενων αντικειμένων

Σεραφείμ Παπαδιάς
makisntpar@gmail.com

Διπλωματική εργασία στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων
Επιβλέπων: Καθηγητής **I. Βασιλείου**

1 Γενικό πλαίσιο

Τα τελευταία χρόνια, παρατηρείται ραγδαία αύξηση του όγκου των δεδομένων που παράγονται, μεταδίδονται και απαιτούν επεξεργασία από εφαρμογές παρακολούθησης (δίκτυα αισθητήρων, συστήματα χρηματοπιστωτικών συναλλαγών, τηλεπικοινωνίες κ.ά.). Στα πλαίσια των υπηρεσιών γεωγραφικού εντοπισμού (*Location-based Services*), ιδιαίτερο ενδιαφέρον έχει η διαχείριση πληρο-φοριών από κινούμενα αντικείμενα (*moving objects*), οι οποίες καταφθάνουν στο σύστημα ως ρεύματα δεδομένων (*data streams*). Για την επεξεργασία τέτοιων ρευμάτων διατυπώνονται ερωτήματα διάρκειας (*continuous queries*) που παραμένουν ενεργά επί μακρόν. Τέτοια χωροχρονικά ερωτήματα (περιοχής, *k*-εγγύτερων γειτόνων, συναθροιστικά κ.ά.) οφείλουν να εξετάζουν εγκαίρως τον κυμαινόμενο όγκο των εισερχόμενων δεδομένων και να παρέχουν αξιόπιστες και διαρκώς επίκαιρες απαντήσεις.

Τα κινούμενα αντικείμενα θεωρούνται σημειακά και καθώς η θέση τους μεταβάλλεται δυναμικά αποστέλλεται ανά τακτά χρονικά διαστήματα σ' έναν κεντρικό επεξεργαστή. Υπάρχει ενδεχόμενο το στίγμα των αντικειμένων να καταγράφεται με πολύ μεγάλη συχνότητα (λ.χ. κάθε 10 δευτερόλεπτα). Τότε ο όγκος της πληροφορίας που συσσωρεύεται γίνεται υπερβολικά μεγάλος, και εφόσον το σύστημα οφείλει να διαχειρίζεται online τα δεδομένα, ενδεχομένως να καθυστερεί η επεξεργασία ποικίλων ερωτημάτων. Για τον λόγο αυτόν, δεδομένου ότι η διαθέσιμη μνήμη και η επεξεργαστική ισχύς του συστήματος είναι πεπερασμένες, μία ιδέα είναι το σύστημα να επεξεργάζεται μόνο ένα μέρος της εισόδου. Η τεχνική της *απόρριψης φόρτου* (*load shedding*) έχει εφαρμοστεί σε συστήματα επεξεργασίας ρευμάτων δεδομένων (AURORA, STREAM, TelegraphCQ) μειώνοντας την υστέρηση στην ανανέωση των αποτελεσμάτων, χωρίς σοβαρές απώλειες στην ποιότητά τους.

Η εργασία αποσκοπεί στην ανάπτυξη ενός αυτορρυθμιζόμενου συστήματος διαχείρισης μεγάλου όγκου τροχιών κινούμενων αντικειμένων, κάνοντας επιλεκτική απόρριψη ολόκληρων τροχιών ώστε να μειωθεί ο φόρτος του συστήματος (*trajectory-aware load shedding*). Λόγω περιορισμένων πόρων, επιδιώκουμε να μην λαμβάνουμε υπ' όψιν όλες τις τροχιές κατά την επεξεργασία, έτσι ώστε η αποτίμηση ερωτημάτων να κλιμακώνεται για πολύ συχνές ενημερώσεις στιγμάτων από μεγάλο πλήθος κινούμενων αντικειμένων. Απώτερος στόχος είναι η έγκαιρη ανανέωση των απαντήσεων σε συναθροιστικά

ερωτήματα διάρκειας, και ειδικότερα η εύρεση της μέσης ταχύτητας κυκλοφορίας οχημάτων ανά οδικό άξονα. Απόρριψη σημειακών θέσεων από ρεύματα κινούμενων αντικειμένων έχει ήδη χρησιμοποιηθεί σε άλλες τεχνικές (ClusterSheddy, LIRA, SOLE), ωστόσο καμία δεν περι-ελάμβανε απόρριψη φόρτου από εξελισσόμενες τροχιές αντικειμένων, όπως προτείνουμε σ' αυτήν την εργασία.

2 Διαχείριση τροχιών κινούμενων οχημάτων

Τα οχήματα που κυκλοφορούν στο οδικό δίκτυο μιας περιοχής θεωρούνται σημειακά αντικείμενα, των οποίων η θέση μεταβάλλεται με την πάροδο του χρόνου. Το χωροχρονικό φαινόμενο της κίνησης κάθε οχήματος στο επίπεδο μπορεί να αναπαρασταθεί με την συνεχή τροχιά του (*trajectory*), η οποία συντίθεται από δύο χωρικές συντεταγμένες (x,y) και μία χρονική ένδειξη (t). Καθώς η συνεχής καταγραφή της τροχιάς είναι αδύνατη, κάθε όχημα αποστέλλει τακτικά το στίγμα του, αν και όχι απαραίτητα με την ίδια συχνότητα με όλα τα υπόλοιπα οχήματα. Πρακτικά, κάθε τροχιά προσεγγίζεται από μία διακριτή ακολουθία πλειάδων $\langle oid, t, x, y \rangle$, όπου *oid* είναι το μοναδικό αναγνωριστικό του οχήματος και *t* το χρονόσημο (*timestamp*) κατά το οποίο το γεωγραφικό στίγμα (x,y) καταφθάνει στο σύστημα.



Σχήμα 1: Τροχιές οχημάτων κατά μήκος οδικών αξόνων

Ο κεντρικός επεξεργαστής οφείλει να αναγνωρίζει τα τμήματα των τροχιών που διασχίζουν κάθε δρόμο, προκειμένου να υπολογίζει στατιστικά στοιχεία κυκλοφορίας

(traffic analytics) σε πραγματικό χρόνο. Συγκεκριμένα, απαντώνται συναθροιστικά ερωτήματα μέσης ταχύτητας ανά οδικό άξονα ενδιαφέροντος. Ένα ερώτημα *qid* τίθεται την χρονική στιγμή t , προσδιορίζει το εύρος (*range*) ω του κυλιόμενου παραθύρου (*sliding window*) για το οποίο θα εξετάζει τμήματα τροχιών επί του συγκεκριμένου άξονα, ενώ θα ανανεώνει τις απαντήσεις κάθε β χρονό-σημα, όσα προσδιορίζει το βήμα (*slide*) του παραθύρου. Π.χ., ένα ερώτημα μπορεί να ζητεί να υπολογίζεται κάθε λεπτό ($\beta=1$ min) η τρέχουσα μέση ταχύτητα στην οδό Πανεπιστημίου βάσει τροχιών που διήλθαν από εκεί εντός του τελευταίου τετάρτου ($\omega=15$ min). Όπως φαίνεται στο Σχήμα 1, τα σχετικά ερωτήματα διαρκείας τίθενται ανά άξονα (με κόκκινο περίγραμμα) και οφείλουν να υπολογίζουν ταχύτητες ανά κατεύθυνση κυκλοφορίας.

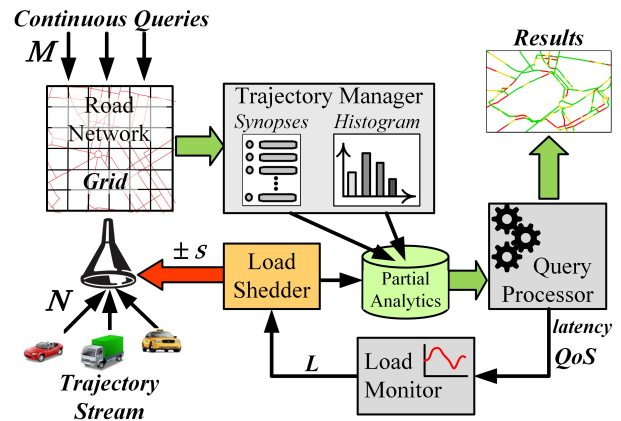
3 Μηχανισμός απόρριψης φόρτου από τροχιές

Αφού το σύστημα δέχεται διαρκώς ενημερώσεις θέσης από πλήθος οχημάτων, θα πρέπει να επεξεργάζεται τις δυναμικά εξελισσόμενες τροχιές τους προκειμένου να υπολογίζει τις μέσες ταχύτητες ανά άξονα. Ωστόσο, απότομες διακυμάνσεις (κυρίως ξαφνικές αυξήσεις) στον ρυθμό άφιξης των στιγμάτων μπορεί να καθιστούν απαγορευτική την εξέταση του συνόλου των τροχιών για την έγκαιρη απάντηση των ερωτημάτων. Σε τέτοιες περιπτώσεις, λόγω περιορισμών σε μνήμη και επεξεργαστική ισχύ, γίνεται επιτακτική ανάγκη να απορριφθούν επιλεκτικά κάποιες τροχιές προτού καν εξεταστούν, κατά προτίμηση όσες είναι λιγότερο «χρήσιμες».

Ο προτεινόμενος μηχανισμός απόρριψης φόρτου λειτουργεί σε κύκλους εκτέλεσης (*execution cycles*), καθένας από τους οποίους περιλαμβάνει τρεις διαδοχικές φάσεις: (i) ενημέρωση (*update*) τροχιών και δομών δεδομένων με τα πρόσφατα στιγμάτα, (ii) αποτίμηση ερωτημάτων (*query evaluation*), και (iii) απόρριψη φόρτου (*load shedding*) όποτε κριθεί απαραίτητη. Στο Σχήμα 2 φαίνεται το γενικό διάγραμμα της αρχιτεκτονικής του συστήματος.

3.1 Φάση ενημέρωσης

Για κάθε νεοεισερχόμενο στίγμα, το σύστημα κανονικά πρέπει να εντοπίσει τον οδικό άξονα (δηλ. το *qid* κάποιου ερωτήματος) στον οποίο κινήθηκε το όχημα. Ακόμη και στην περίπτωση που το στίγμα βρίσκεται πάνω σε διασταύρωση, ο δρόμος προκύπτει από το διάγραμμα κίνησης σε σχέση με το αμέσως προηγούμενο γνωστό στίγμα της τροχιάς. Προκειμένου να είναι εφικτός ο γρήγορος εντοπισμός των υποψηφίων αξόνων, όλη η περιοχή μελέτης και το οδικό δίκτυο διαμερίζονται στα κελιά ενός ομοιόμορφου καννάβου (*grid partitioning*).



Σχήμα 2: Αρχιτεκτονική απόρριψης φόρτου από τροχιές

Με αντίστοιχο τρόπο θα πρέπει να υπολογίζεται η στιγμιαία ταχύτητα \vec{v} του οχήματος. Η δευτερογενής πληροφορία που αφορά τον άξονα *qid* και την στιγμιαία ταχύτητα \vec{v} αποθηκεύονται μαζί με το πρωτογενές στίγμα και αναθεωρούν την κατάσταση της τροχιάς του συγκεκριμένου οχήματος όπως τηρείται απ' τον επεξεργαστή (*Trajectory Manager*). Όπως όμως θα αναλυθεί στην φάση της απόρριψης φόρτου, τέτοιες ενημερώσεις γίνονται μόνο για όσες τροχιές χαρακτηρίζονται ως «χρήσιμες» και δεν απορριφθούν.

Παράλληλα, τηρείται *σύνοψη* κάθε εξελισσόμενης τροχιάς, η οποία καταγράφει από ποιους δρόμους πέρασε το όχημα και για πόσο χρονικό διάστημα κινήθηκε σε καθέναν απ' αυτούς. Τέλος, σε κάθε κύκλο εκτέλεσης κατασκευάζεται *ιστόγραμμα* (με b κάδους) για το πλήθος των τροχιών που αφορούν κάθε άξονα. Πρακτικά, αυτό χρησιμεύει στον προσδιορισμό αξόνων με μικρό δείγμα τροχιών, ώστε να αποφευχθεί τυχόν απόρριψή τους.

3.2 Φάση αποτίμησης ερωτημάτων

Εφόσον με την έλευση νέων στιγμάτων υπολογίζεται η εκάστοτε στιγμιαία ταχύτητα \vec{v} του αντικείμενου, ενώ επιπλέον αντιστοιχίζεται σε οδικό άξονα *qid*, αυτά τα στοιχεία θα μπορούσαν να είναι αμέσως διαθέσιμα για την εκτίμηση μέσων ταχυτήτων. Κάνοντας την παραδοχή πως όταν ένα όχημα εξέλθει από έναν δρόμο *qid*, τότε δεν θα επανέλθει σύντομα σ' αυτόν, σε κάθε κομμάτι τροχιάς ενός αντικείμενου *oid* που κινείται στο συγκεκριμένο *qid*, κρεμάμε δύο δείκτες: (i) έναν στο πρώτο στίγμα της τροχιάς εντός του δρόμου *qid*, και (ii) έναν άλλον στο τελευταίο στίγμα της τροχιάς αμέσως πριν βγει από τον *qid*. Άρα αν μια τροχιά πέρασε πρόσφατα από n δρόμους, τότε θα κρέμονται από αυτήν $2n$ δείκτες. Τηρείται λοιπόν μία δομή δεδομένων γι' αυτά τα επιμέρους στατιστικά στοιχεία κίνησης (*partial aggregates*), η οποία πρακτικά κρατάει σε κάθε δρόμο ποια κινούμενα αντικείμενα υπάρχουν κάθε στιγμή, και η οποία ενημερώνεται σε κάθε κύκλο εκτέλεσης με τα πιο πρόσφατα στοιχεία.

Η αποτίμηση ερωτημάτων συνίσταται στον συναθροιστικό υπολογισμό της μέσης ταχύτητας κίνησης ανά δρόμο βάσει των επιμέρους στιγμιαίων ταχυτήτων που έχουν συλλεγεί προσφάτως (ανάλογα με το εύρος ω του παραθύρου που θέτει κάθε ερώτημα). Υιοθετώντας μία τακτική *lazy evaluation*, οι σχετικοί δείκτες αναπροσαρμόζονται κατά την φάση της ενημέρωσης, ενώ ταυτόχρονα παρέχουν την συνεισφορά καθενός αντικειμένου στις μέσες ταχύτητες των αξόνων τους οποίους διέσχισε πρόσφατα.

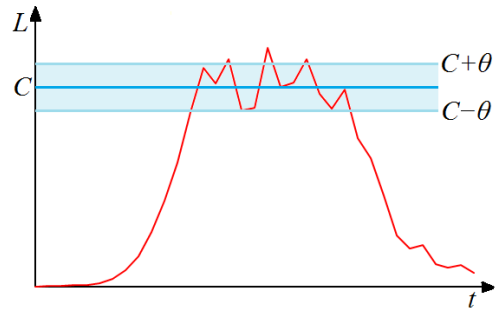
3.3 Φάση απόρριψης φόρτου

Ιδανικά, η ενημέρωση τροχιών και η αποτίμηση των ερωτημάτων πρέπει να ολοκληρώνονται μέσα στην χρονική περίοδο που θέτει ένας κύκλος εκτέλεσης. Κάτι τέτοιο είναι όμως μάλλον ανέφικτο, όταν το πλήθος των κινούμενων αντικειμένων, ο ρυθμός άφιξης στιγμάτων και ο αριθμός των ερωτημάτων αυξάνει, οπότε η επεξεργασία θα υπερβεί την προκαθορισμένη *χωρητικότητα C (Capacity)* του συστήματος (δηλ. άνω όριο φόρτου).

Αρχικά, το σύστημα ξεκινά να λειτουργεί με όλες τις διαθέσιμες τροχιές, χωρίς να απορρίπτεται καθόλου στοιχείο στην είσοδο. Σε κάθε κύκλο εκτέλεσης, ο *μετρούμενος φόρτος L* εκφράζει το κόστος σε κύκλους μηχανής (CPU) που απαιτούνται για την ενημέρωση, την αποτίμηση, αλλά και την διαδικασία απόρριψης φόρτου (εφόσον αυτή εκτελείται). Αν το σύστημα εξετάζει κάθε φορά όλες τις τροχιές, πιθανόν το L να προκύπτει συχνά μεγαλύτερο του C . Τότε προκρίνεται η εξέταση ενός μόνο μέρους των δεδομένων εισόδου, επιλέγοντας με *σημασιολογικό* (και όχι τυχαίο) τρόπο τις κατάλληλες τροχιές (*object-level shedding*). Αυτές μόνο θα είναι όσες θα ενημερωθούν και θα χρησιμεύσουν στην αποτίμηση *κατά τον επόμενο κύκλο εκτέλεσης*. Για τις υπόλοιπες τροχιές, κάθε εισερχόμενο στίγμα μπορεί να απορρίπτεται χωρίς εξέταση (*position-level shedding*), εκτός εάν το όχημα μόλις έστριψε σε άλλο δρόμο. Σε περίπτωση που μία τροχιά καταστεί εκ νέου ενεργή, είναι σαφές ότι δεν θα υπάρχει καταγεγραμμένη η πρόσφατη ιστορία της. Για να υπάρχει λοιπόν δυνατότητα υπολογισμού της στιγμιαίας ταχύτητας εκείνη την στιγμή, γίνεται η παραδοχή ότι για όλες τις τροχιές (ενεργές ή μη) πάντοτε τηρείται το αμέσως προηγούμενο γνωστό στίγμα.

Προκειμένου να μην ενεργοποιείται άσκοπα η διαδικασία απόρριψης φόρτου, τίθεται μία παράμετρος *ανοχής θ (%C)* ως προς την ζητούμενη χωρητικότητα C . Όπως φαίνεται στο Σχήμα 3, ορίζεται έτσι μία ζώνη ($C-\theta$, $C+\theta$), η οποία περιγράφει τα «φυσιολογικά» επίπεδα στα οποία πρέπει να κινείται ο φόρτος L για να μην τρέχει ο *load shedder*. Η αναπροσαρμογή φόρτου είναι απαραίτητη όταν παρατηρείται:

- είτε *υπερφόρτωση* ($L > C+\theta$), οπότε η μέθοδος διατηρεί λιγότερες τροχιές για αποτίμηση ερωτημάτων,



Σχήμα 3: Συνθήκη ενεργοποίησης για απόρριψη φόρτου

- είτε *υποφόρτωση* ($L < C-\theta$), οπότε υπάρχει περιθώριο αύξησης του δείγματος των τροχιών για βελτίωση της ποιότητας των αποτελεσμάτων.

Για τον προσδιορισμό του *πλήθους των τροχιών που προσθαφαιρούνται* υιοθετείται μία πολιτική *αυξομειώσης κατά βήματα (step-wise)* με τρεις εναλλακτικές επιλογές. Έτσι, το *βήμα s* εκφράζει ποσοστό % επί του αριθμού n των τροχιών που είτε (i) έχουν εμφανιστεί έως τώρα στο σύστημα, είτε (ii) του συνολικού προσδοκώμενου αριθμού τροχιών N , είτε (iii) του αριθμού των «χρήσιμων» τροχιών που μόλις συνεισέφεραν στην αποτίμηση κατά τον τρέχοντα κύκλο εκτέλεσης. Η διαφοροποίηση στον ορισμό του n επηρεάζει την ταχύτητα σύγκλισης του αλγορίθμου απόρριψης φόρτου, δηλ. πόσο γρήγορα ο παρατηρούμενος φόρτος L θα επανέλθει εντός της ζώνης ($C-\theta$, $C+\theta$).

Η *σημασιολογική* επιλογή των καταλληλότερων τροχιών για την επεξεργασία βασίζεται σε δύο κριτήρια:

- Το πρώτο προκρίνει την διατήρηση τροχιών που διέρχονται από οδικούς άξονες με «μικρό» δείγμα οχημάτων, ώστε να είναι εφικτό να εκτιμηθεί στοιχειωδώς η μέση ταχύτητα. Η επιλογή του ορίου βάσει του οποίου το δείγμα θεωρείται «μικρό», βασίζεται σε *ιστόγραμμα* τιμών του πλήθους τροχιών ανά άξονα. Αυτό μπορεί να είναι απλουστευμένο (ένας κάδος, οπότε λαμβάνεται ως όριο η μέση τιμή ή ο διάμεσος) ή σε *percentiles* με b κάδους (λ.χ. αν $b=5$, τότε ως όριο λαμβάνεται η μέγιστη τιμή του πρώτου *quantile*).
- Το δεύτερο κριτήριο αξιοποιεί την *βαθμονόμηση τροχιών* με βάση το πλήθος των οδικών αξόνων από τους οποίους έχει διέλθει κάθε όχημα και το διάστημα παραμονής του s' αυτόν, οπότε ευνοούνται τροχιές που εμπίπτουν σε πολλαπλά ερωτήματα (δηλ. έχουν διασχίσει πολλούς οδικούς άξονες). Αυτή η βαθμονόμηση επιτυγχάνεται εύκολα χάρη στις τηρούμενες *συνόψεις* τροχιών.

Σημειώνεται ότι στην περίπτωση που $L < C - \theta$ και ο μετρούμενος φόρτος του συστήματος συνεχώς μειώνεται επειδή δεν υπάρχουν πολλά κινούμενα οχήματα (λ.χ. νυκτερινές ώρες), τότε η λειτουργία απόρριψης πρέπει να απενεργοποιείται. Βεβαίως, λόγω του

αυτορρυθμιζόμενου χαρακτήρα του συστήματος, η απόρριψη τροχιών θα τεθεί πάλι σε ισχύ αν τυχόν σε επόμενο κύκλο ξεπεραστεί το άνω φράγμα της ζώνης ανοχής ($L > C + \theta$).

4 Πειραματική μελέτη

Η υλοποίηση του μηχανισμού πραγματοποιήθηκε σε γλώσσα C++ και δοκιμάστηκε πειραματικά σε *συνθετικά δεδομένα* τροχιών από οχήματα που κινούνται με διάφορες ταχύτητες στο οδικό δίκτυο της Αθήνας. Ο αλγόριθμος δοκιμάστηκε για διαφορετικά πλήθη οχημάτων (100000, 200000, και 500000), ενώ η χρονική διάρκεια κάθε τροχιάς κυμαίνεται μεταξύ 1090 και 4060 δευτερολέπτων. Έγιναν πειράματα επιδόσεων και ποιότητας απαντήσεων για διάφορες τιμές παραμέτρων του συστήματος:

- Κατάτμηση καννάβου (*grid granularity*) cxc : 50x50, 100x100, 125x125, 200x200 και 500x500.
- Επιθυμητή *χωρητικότητα* συστήματος C : 20%, 33%, 50%, 67% και 80% ως προς τον μέγιστο μετρούμενο φόρτο συστήματος χωρίς απόρριψη (L_{peak}).
- Βήμα *προσθαφαίρεσης* τροχιών s : 2%, 5%, 10%, 15%, εφαρμοζόμενο επί διαφόρων συνόλων τροχιών.
- Τρόπος επιλογής ορίου για μικρό δείγμα τροχιών: *mean*, *median*, *b-tile* (για $b = 5$ προκύπτει *quantile*).
- Τιμές ανοχής θ : 2%, 5%, 10%, 15% επί του C .

Από την μελέτη των πειραματικών μετρήσεων προκύπτουν τα εξής:

- Για λεπτότερες υποδιαίρεσεις του καννάβου (περισσότερα κελιά) προκύπτει λιγότερη επιβάρυνση στην φάση αποτίμησης, εις βάρος όμως της μνήμης που απαιτείται για την τήρησή του.
- Ο προτεινόμενος μηχανισμός γενικά κατορθώνει να αναπροσαρμόζει δυναμικά τον φόρτο ανάλογα με τους διατιθέμενους πόρους (C). Όσο μεγαλύτερο είναι το C , τόσο γρηγορότερα γίνεται η προσαρμογή.
- Για μεγαλύτερα βήματα s , η προσαρμογή γίνεται πιο δραστικά, προσθαφαίρωντας περισσότερες τροχιές προς επεξεργασία. Ωστόσο, όταν το s λαμβάνει ακραίες τιμές (λίγες ή πάρα πολλές τροχιές), τότε πραγματοποιείται *load shedding* συχνότερα, επειδή βγαίνουμε συχνότερα εκτός ζώνης ανοχής. Όταν το βήμα s εφαρμόζεται επί του συνολικού αριθμού N τροχιών που αναμένονται από το σύστημα, τότε η σύγκλιση στην χωρητικότητα C είναι ταχύτερη.
- Ως προς την επιλογή ορίου για «μικρό» δείγμα τροχιών ανά άξονα, η πολιτική που βασίζεται σε *quantiles* είναι εκείνη που δίνει αντιπροσωπευτικότερες τιμές για τους δρόμους που καλύπτουν αυτό το κριτήριο.

- Όσο μεγαλύτερη είναι η τιμή της ανοχής θ , τόσο λιγότερες φορές εκτελείται *load shedding* συνολικά στην διάρκεια του πειράματος.
- Οι εκτιμώμενες μέσες ταχύτητες ανά άξονα εμφανίζουν μικρή απόκλιση από τις τιμές που βασίζονται σε εξαντλητική αποτίμηση. Οι αποκλίσεις είναι ελαφρώς αυξημένες σε μεγάλου μήκους άξονες όπου ση-μειώνονται μεγαλύτερες διακυμάνσεις ταχυτήτων.
- Το σύστημα έχει ελεγχθεί και για κλιμακούμενα πλήθη (έως 500000) κινούμενων αντικειμένων, επιβεβαιώνοντας την ανθεκτικότητα της μεθόδου και την δυνατότητα αυτορρύθμισης του φόρτου.

5 Συμπεράσματα – Μελλοντικές κατευθύνσεις

Το προτεινόμενο σύστημα μεριμνά για την διατήρηση του φόρτου στα επιθυμητά επίπεδα, απορρίπτοντας πρωτογενή πληροφορία από «μη σημαντικές» τροχιές και κρατώντας εκείνες που συνεισφέρουν τα μέγιστα στην ακρίβεια των υπολογιζόμενων μέσων ταχυτήτων ανά άξονα. Το μοντέλο απόρριψης φόρτου είναι προσαρμοστικό (*adaptive*) στις εκάστοτε συνθήκες (πλήθος αντικειμένων, ρυθμός εισροής στιγμάτων, αριθμός και ποικιλία ερωτημάτων), έτσι ώστε να παρέχονται έγκαιρες και αξιόπιστες απαντήσεις.

Οι δομές και οι τεχνικές που προτάθηκαν αφήνουν σημαντικά περιθώρια για περαιτέρω διερεύνηση, όπως:

- Αναθεώρηση κριτηρίων βαθμονόμησης συνόψεων με αναπροσαρμογή βαρών, μαζί με την καταγραφή της εκάστοτε *κατάστασης* των τροχιών (ενεργή ή μη).
- Για περαιτέρω ενίσχυση της ακρίβειας των αποτελεσμάτων θα μπορούσαν να χρησιμοποιηθούν *δειγματοληπτικές* μέθοδοι, ώστε για τις τροχιές που από ανενεργές μόλις κατέστησαν ενεργές να έχουν τηρηθεί περισσότερα δείγματα θέσεων.
- Υιοθέτηση *μεταβλητού βήματος* s , ώστε να κυμαίνεται ανάλογα με την υπέρβαση φόρτου ($L-C$).
- Τέλος, θα είχε μεγάλο ενδιαφέρον αν ήταν δυνατόν το προτεινόμενο πλαίσιο από *συγκεντρωτικό* (*centralized*) να εφαρμοστεί και σε *καταμεμημένο* (*distributed*) περιβάλλον, όπου διάφοροι επεξεργαστικοί κόμβοι θα συνεργάζονταν στην απόρριψη τροχιών και στην κατανομή του φόρτου.