ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Επεξεργασία Επερωτήσεων με Συνδυασμό Γεωγραφικής Θέσης και Λέξεις-Κλειδιά στην Hbase

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

ΧΑΡΙΤΟΥΔΗ ΑΠΟΣΤΟΛΟΥ και ΓΚΡΕΚΑ ΒΑΣΙΛΕΙΟΥ

Επιβλέπων: Αν. Καθηγήτρια Βλάχου Ακριβή

Μέλη εξεταστικής επιτροπής: Αν. Καθηγητής Κρητικός Κυριάκος, Επ. Καθηγητής Σκούτας Δημήτριος

Σάμος, Σεπτέμβριος 2020

Πρόλογος και ευχαριστίες

Η παρούσα διπλωματική εργασία, με τίτλο «Επεξεργασία Επερωτήσεων με Συνδυασμό Γεωγραφικής Θέσης και Λέξεις-Κλειδιά στην Hbase», εκπονήθηκε από τους προπτυχιακούς φοιτητές Απόστολο Χαριτούδη και Βασίλειο Γκρέκα του τμήματος Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων του Πανεπιστημίου Αιγαίου με έδρα την Σάμο. Η εκπόνηση της εργασίας, είναι υποχρεωτική και αναπτύχθηκε στα πλαίσια του μαθήματος «Διπλωματική Εργασία» για την απόκτηση διπλώματος.

Επιπρόσθετα, εκφράζουμε τις ευχαριστίες μας στην επόπτρια καθηγήτρια κ. Βλάχου Ακριβή, από την οποία προτάθηκε και το θέμα της εργασίας και για την δυνατότητα που μας έδωσε να ασχοληθούμε με ένα αρκετά ενδιαφέρον θέμα.

Επιπλέον, ευχαριστούμε θερμά τις οικογένειες μας, που βοήθησαν τόσο οικονομικά όσο και ενθαρρυντικά όλων των ετών φοίτησης και που μας έδωσαν την ευκαιρία να διευρύνουμε τους πνευματικούς μας ορίζοντες.

Πίνακας περιεχομένων

Πίνακας εικόνων

Λίστα Πινάκων

Πίνακας ορολογιών

Ορολογία	Σημασία
HDFS (Hadoop Distributed File System)	Αρχιτεκτονική κατανεμημένης διαχείρισης αρχείων
Χωρικά δεδομένα	Στοιχεία που απαρτίζονται από συντεταγμένες
PAM (Point Access Method)	Μέθοδοι που κάνουν εφικτή την πρόσβαση σε σημεία αποτελούμενα από συντεταγμένες
SFC (Space Filling Curves)	Μέθοδοι που επιτρέπουν την αναπαράσταση πολλών διαστάσεων σε μία διάσταση
Μέθοδοι αποθήκευσης δεδομένων	Μέθοδοι που προτείνουν αποδοτικούς και αποτελεσματικούς τρόπους αποθήκευσης των δεδομένων
Γεωγραφικές συντεταγμένες	Χρησιμοποιούνται για τον προσδιορισμό θέσης των αντικειμένων
Μοναδικό αναγνωριστικό	Χρησιμοποιούνται κυρίως στις βάσεις δεδομένων για τον προσδιορισμό ενός και μόνο στοιχείου
Ερώτημα	Βασικό εργαλείο των συστημάτων βάσεων δεδομένων
Big Data	Μεγάλα δεδομένα
Latitude	Γεωγραφικό πλάτος
Longitude	Γεωγραφικό μήκος

Περίληψη

Η σημερινή σύγχρονη εποχή χαρακτηρίζεται από την συνεχώς αυξανόμενη τάση των πληροφοριών και των δεδομένων που αποθηκεύονται σε δομές αποθήκευσης και προσφέρονται προς τους χρήστες. Ο κύριος λόγος που παρατηρείται αυτή η αύξηση είναι η ψηφιοποίηση της πληροφορίας που υπήρχε πριν και μετά την δημιουργία του διαδικτύου. Ένας αρκετά μεγάλος όγκος πληροφορίας είναι τα δεδομένα που αφορούν τις γεωγραφικές συντεταγμένες της γης, καλούμενα ως χωρικά δεδομένα.

Η παρούσα διπλωματική εργασία αφορά την διαχείριση, την αποθήκευση και την ανάκτηση των χωρικών δεδομένων, με την χρήση χωρικών ερωτημάτων και κειμένου. Τα κείμενα που θα χρησιμοποιηθούν αφορούν λέξεις - κλειδιά (keywords) και έχουν αρκετά σημαντικό ρόλο στην αναζήτηση των δεδομένων, διότι προσδίδουν ένα επιπλέον χαρακτηριστικό αυτών. Με τον όρο διαχείριση δεδομένων, εννοείται η αποθήκευση, η διαγραφή, η εισαγωγή, η τροποποίηση και η αναζήτηση των δεδομένων. Επομένως, στην εργασία θα παρουσιαστούν μέθοδοι που επιτυγχάνουν τα παραπάνω με αποτελεσματικό και αποδοτικό τρόπο.

Λέξεις κλειδιά: γεωγραφικές συντεταγμένες, χωρικά δεδομένα, μεγάλα δεδομένα, τύπος ερωτημάτων, δομή αποθήκευσης, αποτελεσματικότητα, αποδοτικότητα, μέθοδοι αποθήκευσης και αναζήτησης, μοναδικό αναγνωριστικό, ερωτήματα

Abstract

Today's modern era is characterized by the ever-increasing trend of information and data stored in storage structures and offered to users. The main reason for this increase is the digitization of information that existed before and after the creation of the internet. A fairly large volume of information is the data concerning the geographical coordinates of the earth, by spatial data. This thesis concerns the management and storage of the aforementioned data in combination with their extraction using spatial queries and text. The texts that will be used are related to keywords and have a very important role in the search of data, because they provide an additional feature of them. The term data management means the storage, deletion, import, modification and retrieval of data. Therefore, the paper will present methods that achieve the above in an effective and efficient manner.

Keywords: geographic coordinates, spatial data, big data, query type, storage structure, efficiency, effectiveness, storage and search methods, unique identifier, queries

1

Εισαγωγή

Σε αυτό το κεφάλαιο θα αναφερθεί το αντικείμενο της εργασίας, οι προβληματισμοί που υπάρχουν και προσπαθεί να επιλύσει. Επιπλέον, θα αναφερθεί ο σκοπός και ο στόχος της εργασίας καθώς επίσης και μία περιγραφή του περιεχομένου που θα αφορά την δομή της διπλωματικής.

1.1 Αντικείμενο εργασίας

Το αντικείμενο της εργασίας αφορά τη διαχείριση μεγάλου όγκου δεδομένων που αφορούν γεωγραφικές συντεταγμένες. Η διαχείριση επικεντρώνεται, κυρίως, στην μέθοδο αποθήκευσης και ανάκτησης χωρικών δεδομένων, με την χρήση συγκεκριμένων μεθόδων και συστημάτων αποθήκευσης καθώς επίσης και ειδικού τύπου ερωτημάτων. Επιπλέον, η εργασία εμπλέκει και την χρήση κειμένου για την αναζήτηση και ανάκτηση των δεδομένων.

Συγκεκριμένα, έχουμε στην διάθεσή μας μία πληθώρα δεδομένων που αφορούν καταστήματα εστίασης, έχοντας ως χαρακτηριστικά το όνομα (Brand Name), την βαθμολογία (Rating), τις ακριβείς γεωγραφικές συντεταγμένες (Latitude και Longitude) καθώς επίσης και τα προϊόντα προς πώληση (keywords) κάθε καταστήματος.

Στην εργασία θα αναπτυχθεί ένα λογισμικό που θα βοηθά κάποιο χρήστη στην αναζήτηση και στην εύρεση των πλησιέστερων καταστημάτων, με κριτήρια αναζήτησης: ένα σημείο αναφοράς που καθορίζεται από τον χρήστη, τις προτιμήσεις των προϊόντων που επιθυμεί ο ίδιος και την απόσταση - ακτίνα από το σημείο αναφοράς του.

Δίνεται ιδιαίτερη έμφαση στον τρόπο ανάκτησης των δεδομένων, έτσι ώστε η αναζήτηση αυτών να είναι αποτελεσματική και αποδοτική χρησιμοποιώντας συγκεκριμένους τύπους ερωτημάτων που θα τίθενται στην βάση δεδομένων. Επιπρόσθετα, εστιάζει στην μοναδικότητα της κάθε εγγραφής, έτσι ώστε κάθε εγγραφή να αντιπροσωπεύεται από ένα μοναδικό αναγνωριστικό, κάτι το οποίο είναι ιδιαίτερης σημασίας και παίζει σημαντικό ρόλο στην αποθήκευση των δεδομένων.

1.2 Καθορισμός προβλήματος

Το πρόβλημα που προσπαθεί να επιλυθεί στην συγκεκριμένη εργασία αφορά την διαχείριση χωρικών δεδομένων (Spatial Data) [1] που αφορούν φυσικά καταστήματα ανά την υφήλιο (όπως αναφέρεται στην ενότητα 1.1). Ο όρος διαχείριση χωρικών δεδομένων χρησιμοποιείται για να γίνει αναφορά στον τρόπο με τον οποίο μπορούν να επεξεργασθούν αποδοτικά τα δεδομένα (όπως είναι οι γεωγραφικές συντεταγμένες) που αφορούν το χώρο. Τα δεδομένα αυτά δημιουργούν δύο βασικά προβλήματα τα οποία δυσκολεύουν τον τρόπο με τον οποίο πρέπει αυτά να διαχειριστούν και για αυτό θέλουν ιδιαίτερη μεταχείριση σε σύγκριση με πιο απλά δεδομένα. Το πρώτο πρόβλημα που εμφανίζεται αφορά την φύση των χωρικών δεδομένων, δηλαδή το γεγονός ότι αφορούν δεδομένα με πολύπλοκη δομή εφόσον αφορούν σημεία στον χώρο (π.χ. οι συντεταγμένες ενός καταστήματος) με αποτέλεσμα ο υπολογισμός των συσχετίσεων μεταξύ τους να απαιτεί αρκετό χρόνο ώστε να μην μπορούν να χρησιμοποιηθούν αποδοτικά. Το δεύτερο πρόβλημα που παρατηρείται αφορά τον όγκο των δεδομένων που προκύπτουν από την ύπαρξη όλων των καταστημάτων ανά την υφήλιο, με αποτέλεσμα να δημιουργείται ένας υπερμεγέθης όγκος δεδομένων. Από τα παραπάνω γίνεται αντιληπτό ότι ο όγκος των δεδομένων που καλούμαστε να διαχειριστούμε είναι μεγάλος και έτσι το πρόβλημα ανάγεται στη διαχείριση δεδομένων μεγάλου όγκου (Big Data), με αποτέλεσμα να κατευθυνθούμε προς την αναζήτηση τεχνολογιών και μεθόδων πέρα από τις συμβατικές, προκειμένου να επιλύσουμε το πρόβλημα της διαχείρισής τους.

1.3 Σκοπός της εργασίας

Σκοπός της εργασίας αυτής είναι να κατανοήσουμε την έννοια των μεγάλων δεδομένων (Big Data), των χωρικών δεδομένων, την χρησιμότητα τους και την αναγκαιότητα για εύρεση μεθόδων και συστημάτων που είναι σχεδιασμένα για την διαχείρισή τους.

Πιο συγκεκριμένα θα πρέπει να αναλυθούν και να σχεδιαστούν μέθοδοι, που θα αποθηκεύουν αποδοτικά και αποτελεσματικά τα δεδομένα στην βάση δεδομένων, με απώτερο σκοπό την ευρετηριοποίηση των εγγραφών και την αποτελεσματική αναζήτηση αυτών.

1.4 Στόχος της εργασίας

Η εργασία έχει ως στόχο την ανάκτηση των δεδομένων που αντιστοιχούν στο ερώτημα που θέτει ο χρήστης. Πιο συγκεκριμένα, η παρούσα εργασία έχει ως στόχο την εύρεση του κατάλληλου ερωτήματος που θα τίθεται στην βάση με αποτέλεσμα να επιστρέφονται αποτελέσματα που σχετίζονται άμεσα με την γεωγραφική θέση από την οποία τίθεται το ερώτημα, παίρνοντας ως παράμετρο και τις προτιμήσεις του χρήστη.

Συνοπτικά, στόχος της εργασίας είναι η εύρεση ενός αξιόπιστου, αποδοτικού και αποτελεσματικού ερωτήματος που θα χρησιμοποιηθεί για την αποτελεσματική και αποδοτική ανάκτηση των δεδομένων βάσει της πληροφορίας που αναζητά ο χρήστης.

1.5 Δομή της εργασίας

Η εργασία είναι δομημένη σε έντεκα (11) κεφάλαια, συμπεριλαμβανομένου και της εισαγωγής, όπου στο κάθε ένα περιγράφονται οι μέθοδοι και τα εργαλεία που χρησιμοποιήθηκαν για την διεκπεραίωση της διπλωματικής καθώς επίσης περιέχονται η πειραματική μελέτη των μεθόδων που επιλέχθηκαν, οι συγκρίσεις αυτών, τα συμπεράσματα της εργασίας καθώς επίσης και η βιβλιογραφία που χρησιμοποιήθηκε. Πιο αναλυτικά:

- **Κεφάλαιο 1:** Το Κεφάλαιο 1 πρόκειται για μια εισαγωγή της εργασίας, έτσι ώστε ο αναγνώστης να αντιληφθεί εξ΄ αρχής το αντικείμενο της εργασίας, τις δυσκολίες που προσπαθεί να επιλύσει, τον στόχο και τον σκοπό της εκπόνησής της.
- **Κεφάλαιο 2:** Στο Κεφάλαιο 2 δίνονται οι ορισμοί και η χρήση των μεγάλων δεδομένων (Big Data) και των χωρικών δεδομένων, έτσι ώστε ο αναγνώστης να κατανοήσει τις έννοιες αυτών και να γίνει περισσότερο αντιληπτό το περιεχόμενο της εργασίας.
- **Κεφάλαιο 3:** Σε αυτό το κεφάλαιο παρέχονται πληροφορίες σχετικά με το σύστημα διαχείρισης μεγάλου όγκου δεδομένων που χρησιμοποιείται για την αποθήκευση των δεδομένων μας καθώς επίσης και η σημαντικότητα ενός τέτοιου συστήματος.
- **Κεφάλαιο 4:** Στο Κεφάλαιο 4 δίνεται ο ορισμός της μεθόδου Z-Order που χρησιμοποιείται για να προσδίδει στα δεδομένα μας ένα μοναδικό αναγνωριστικό. Επιπλέον, περιγράφεται η διαδικασία εξαγωγής μοναδικού χαρακτηριστικού, κάνοντας χρήση αυτής της μεθόδου, καθώς επίσης δίνεται και ένα παράδειγμα δημιουργίας μιας τιμής Z-Order και αναφέρονται οι λόγοι χρήσης μιας τέτοιας μεθόδου.
- **Κεφάλαιο 5:** Στο Κεφάλαιο 5 περιγράφονται οι μέθοδοι αποθήκευσης που επιλέχθηκαν να χρησιμοποιηθούν στην παρούσα εργασία.
- **Κεφάλαιο 6:** Στο Κεφάλαιο 6 θα δοθεί η έννοια του ερωτήματος, που τίθεται σε μία βάση δεδομένων. Επιπλέον, θα αναλυθούν μερικοί από τους τύπους ερωτημάτων και θα αναφερθούν οι λόγοι και οι τύποι ερωτημάτων που επιλέχθηκαν για την παρούσα εργασία.
- **Κεφάλαιο 7:** Στο Κεφάλαιο 7 θα περιγραφεί η υλοποίηση των μεθόδων και θα αναφερθούν τα δεδομένα που έχουμε στην διάθεσή μας.
- **Κεφάλαιο 8:** Στο Κεφάλαιο 8 θα διεξαχθεί πειραματική μελέτη των μεθόδων δίνοντας φωτογραφικό υλικό που σχετίζεται με την αποτελεσματικότητα και αποδοτικότητα των μεθόδων.

- **Κεφάλαιο 9:** Στο Κεφάλαιο 9 θα πραγματοποιηθεί η συγκριτική μελέτη των μεθόδων δίνοντας παράλληλα συγκριτικά διαγράμματα της αποδοτικότητας κάθε μίας.
- **Κεφάλαιο 10:** Στο κεφάλαιο 10 θα αναφερθούν τα συμπεράσματα και η γνώση η οποία αποκομίστηκε από την διεκπεραίωση της εργασίας.
- **Βιβλιογραφία:** Θα αναφερθούν οι πηγές που χρησιμοποιήθηκαν.

2

Μεγάλα δεδομένα και χωρικά δεδομένα

Σε αυτό το κεφάλαιο θα αναλυθούν και θα δοθούν οι ορισμοί των εννοιών των μεγάλων δεδομένων και των χωρικών δεδομένων.

2.1 Μεγάλα δεδομένα (Big Data)

Κάτι που χαρακτηρίζει την σημερινή εποχή είναι οι συνεχής αύξηση της πληροφορίας και των δεδομένων και η ανάγκη της αποθήκευσης αυτών. Η αύξηση των δεδομένων οφείλεται στην ψηφιοποίηση της πληροφορίας, που υπήρχε πριν και μετά την δημιουργία του διαδικτύου. Επιπλέον, δημοφιλής και μη εταιρίες συλλέγουν καθημερινά ένα τεράστιο όγκο δεδομένων από τους ίδιους τους χρήστες. Μερικά από τα δεδομένα, αυτά, είναι το όνομα, το επώνυμο, η ηλικία, το φύλο, η τοποθεσία και οι προτιμήσεις των χρηστών. Τα δεδομένα αυτά συλλέγονται και δημιουργούν κάποια πληροφορία για κάθε χρήστη, που είναι αρκετά χρήσιμη για την εκάστοτε εταιρία και την βοηθάει τόσο στην βελτίωση και στην ανάπτυξη των προϊόντων και των υπηρεσιών της όσο και στην προώθηση αυτών.

Τα δεδομένα μεγάλου όγκου (Big Data) αφορούν δεδομένα για τα οποία οι συμβατικοί μέθοδοι και τεχνικές κρίνονται μη αποδοτικές και συνεπώς μη ικανές να τα διαχειριστούν [2]. Επομένως, από το παραπάνω συμπεραίνεται ότι τα μεγάλα δεδομένα απαιτούν εξελιγμένη τεχνολογία και σύγχρονες μεθόδους για την διαχείρισή τους.

Καθημερινά, λόγω της αύξησης των χρηστών του διαδικτύου, τα δεδομένα συνεχώς αυξάνονται με συνέπεια το πρόβλημα διαχείρισης των Big Data να μεγεθύνεται ακόμα περισσότερο. Επιπλέον, οι τεχνολογίες και υπηρεσίες Smart Home και Smart Cities αυξάνουν ολοένα και περισσότερο τον όγκο πληροφοριών. Επιπρόσθετα, μεγάλο ρόλο στην αύξηση των δεδομένων παίζουν τα μέσα μαζικής ενημέρωσης, στον χώρο του διαδικτύου, που καθημερινά αναρτούν ειδήσεις και οπτικοακουστικό υλικό. Από το παραπάνω γίνεται αντιληπτό ότι τα μεγάλα δεδομένα δεν αποτελούν ένα στατικό πρόβλημα αλλά ένα δυναμικό που συνεχώς αυξάνεται.

Ωστόσο, τα μεγάλα δεδομένα δεν περιορίζονται μονάχα στις δημοφιλείς και μη εταιρίες ή οργανισμούς αλλά απασχολούν και τις κυβερνήσεις των κρατών, για την αποθήκευση και γενικότερα την διαχείριση των δεδομένων που αφορούν τους πολίτες. Αυτά τα δεδομένα τα χρησιμοποιούν αρκετές δημόσιες υπηρεσίες και χρήζουν αντιμετώπισης από σύγχρονες και εξελιγμένες τεχνολογίες και μεθόδους.

Ένα από τα πλεονεκτήματα των μεγάλων δεδομένων είναι ότι η επεξεργασία των δεδομένων που αφορούν τους χρήστες έχει σαν αποτέλεσμα την εξαγωγή πληροφοριών. Αυτές τις πληροφορίες, όπως αναφέρθηκε και προηγουμένως, είναι εφικτό να τις εκμεταλλευτούν οργανισμοί και εταιρίες για την βελτίωση και την ανάπτυξη νέων υπηρεσιών και προϊόντων καθώς επίσης και για την προώθηση αυτών. Επομένως, τα μεγάλα δεδομένα ανοίγουν τον δρόμο της Διαφήμισης (Marketing) για την δημιουργία νέων τεχνολογιών και υπηρεσιών.

Τα μεγάλα δεδομένα χαρακτηρίζονται από συνολικά έξι (6) χαρακτηριστικά, εκ των οποίων τα τρία (3) είναι τα πιο σημαντικά. Τα πρώτα τρία και κύρια είναι τα παρακάτω [2]:

- **Όγκος:** Αυτό που κάνει τα δεδομένα να χαρακτηρίζονται ως Big Data είναι ξεκάθαρα το μέγεθος του. Για παράδειγμα, η εταιρία Google αποθηκεύει και επεξεργάζεται καθημερινά δεδομένα που αφορούν τις προτιμήσεις εκατομμυρίων χρηστών.

- **Ταχύτητα:** Η ταχύτητα είναι ένα αναπόσπαστο χαρακτηριστικό των Big Data. Με τον όρο ταχύτητα νοείται η ραγδαία αύξηση δημιουργίας και κατ΄ επέκταση αποθήκευσης και επεξεργασίας των δεδομένων. Για παράδειγμα, όπως έχει αναφερθεί και παραπάνω, οι χρήστες του διαδικτύου αυξάνονται καταιγιστικά και συνεχώς με αποτέλεσμα, θέλοντας και μη, να δημιουργούν κάποιο διαδικτυακό προφίλ με συνέπεια να αυξάνεται και ο όγκος της πληροφορίας.

- **Ποικιλία:** Η ποικιλία αφορά το περιεχόμενο των δεδομένων. Έχει να κάνει με την διαφορετικότητα των τύπων των δεδομένων. Για παράδειγμα, καθημερινά υπάρχει αλληλεπίδραση μεταξύ των χρηστών στα κοινωνικά δίκτυα. Η ποικιλία αυτής της αλληλεπίδραση, αφορά μηνύματα κειμένου, ηχητικά μηνύματα, βίντεο και φωτογραφίες.

Τα τρία επιπλέον χαρακτηριστικά των μεγάλων δεδομένων είναι αυτά που παρουσιάζονται παρακάτω:

- **Μεταβλητότητα:** Η μεταβλητότητα των δεδομένων εκφράζει τις παραπάνω από μία έννοιες μιας λέξης. Για παράδειγμα η λέξη, «διάσπαση» έχει παραπάνω από μία σημασία. Μία περίπτωση είναι η λέξη διάσπαση να αναφέρεται στην διάσπαση ενός ατόμου στην επιστήμη της φυσικής, ενώ μία δεύτερη περίπτωση είναι η λέξη διάσπαση να αναφέρεται στην διάσπαση προσοχής, όσον αφορά την συγκέντρωση ενός ανθρώπου.

- **Ακρίβεια:** Η ακρίβεια των δεδομένων παίζει ένα αρκετά σημαντικό ρόλο σε όλων των ειδών δεδομένων (είτε μεγάλων δεδομένων είτε όχι). Για παράδειγμα αν έχουμε στην διάθεσή μας δεδομένα που δεν έχουν ακρίβεια και πραγματοποιηθεί η ανάλυση αυτών, τότε θα εξαχθούν λανθασμένα

αποτελέσματα. Επομένως, η ανακρίβεια κάνει τα δεδομένα αυτομάτως αναξιόπιστα.

- **Αξία:** Τα δεδομένα πρέπει να μετατρέπονται με τέτοιο τρόπο έτσι ώστε να προσδίδεται αξία σε αυτά. Αν για παράδειγμα ένας βηματομετρητής έδειχνε μόνο τον αριθμό βημάτων και καμία άλλη πληροφορία τότε αυτό το δεδομένο θα ήταν εντελώς αδιάφορο. Αντίθετα ο βηματομετρητής μετατρέπει τον αριθμό των βημάτων σε πόσες θερμίδες καταναλώθηκαν, δίνοντας αξία στα δεδομένα.

2.2 Χωρικά δεδομένα

Τα χωρικά δεδομένα, λόγω του μεγάλου τους όγκου στις περισσότερες περιπτώσεις που χρησιμοποιούνται, μπορεί να ειπωθεί ότι ανήκουν στην κατηγορία των μεγάλων δεδομένων (Big Data). Αποτελούν ένα τεράστιο όγκο δεδομένων που δυσκολεύει αρκετά την αποθήκευσή τους και γενικότερα την διαχείρισή τους.

Τα δεδομένα αυτά έχουν κάποια συγκεκριμένα χαρακτηριστικά όπως είναι η τοποθεσία και η απόσταση, με αποτέλεσμα να απαιτούν ένα μεγάλο χώρο αποθήκευσης. Χρησιμοποιούνται αρκετά συχνά σε συστήματα γεωγραφικών πληροφοριών και βρίσκουν εφαρμογή τόσο σε συστήματα πλοήγησης (όπως το GPS) όσο και σε στην επεξεργασία εικόνας, δύο τομείς οι οποίοι εκ πρώτης όψεως δεν συγκλίνουν. Επιπλέον, η ανάλυση αυτών είναι ιδιαίτερα σημαντική, έτσι ώστε η αποθήκευση και γενικότερα η επεξεργασία τους να γίνεται κατά το δυνατό με αποτελεσματικότερο και αποδοτικότερο τρόπο [3].

Ένα παράδειγμα χωρικών δεδομένων είναι η διεύθυνση ενός καταστήματος, η οποία προσδιορίζεται από την ονομασία του δρόμου, ένα αριθμό καθώς και ένα ταχυδρομικό κώδικα. Αυτά τα δεδομένα, όμως, δεν είναι αρκετά για να προσδιορίσουν την ακριβή τοποθεσία ενός αντικειμένου στον χώρο και για αυτόν τον λόγο χρησιμοποιούνται οι γεωγραφικές συντεταγμένες. Οι γεωγραφικές συντεταγμένες που χρησιμοποιούνται για να εκφράσουν ένα σημείο στην γη αποτελούνται από δύο τιμές. Αυτές οι τιμές είναι το Latitude και το Longitude που αντιπροσωπεύουν το γεωγραφικό πλάτος και γεωγραφικό μήκος, αντίστοιχα.

Όπως αναφέρθηκε παραπάνω, τα χωρικά δεδομένα πρόκειται για μεγάλα δεδομένα. Αυτό συμβαίνει διότι σε καθημερινή βάση οι εταιρίες και οι οργανισμοί με την χρήση των έξυπνων συσκευών (έξυπνα κινητά, ρολόγια κ.α.) καταγράφουν την τοποθεσία των χρηστών, δημιουργώντας ένα μεγάλο όγκο πληροφοριών τον οποίο αποθηκεύουν και επεξεργάζονται προκειμένου να παράγουν κάποια αποτελέσματα που αφορούν τον χρήστη. Αυτή η πληροφορία είναι ιδιαίτερα σημαντική καθώς μπορεί να χρησιμοποιηθεί για την προώθηση και την προβολή διαφημίσεων για προϊόντα και υπηρεσίες διαδικτυακών καταστημάτων καθώς επίσης και για καταστήματα με φυσική παρουσία.

Επιπρόσθετα, τα χωρικά δεδομένα είναι χρήσιμα και σε διάφορες υπηρεσίες. Για παράδειγμα, μία υπηρεσία που καθιστά αναγκαία την χρήση των χωρικών δεδομένων είναι το εθνικό κτηματολόγιο, στο οποίο οι χρήστες μπορούν να δηλώσουν με ακρίβεια τις εκτάσεις των ακίνητων που έχουν στην κατοχή τους. Έπειτα αυτές οι πληροφορίες αποθηκεύονται και επεξεργάζονται από την υπηρεσία.

Άλλη μία χρήση των χωρικών δεδομένων, η οποία χρησιμοποιείται από τους ανθρώπους είναι εύρεση κοντινών τοποθεσιών ή καταστημάτων από ένα σημείο αναφοράς που δίνεται από τον χρήστη.

3

Περιγραφή των εργαλείων

Σε αυτό το κεφάλαιο θα περιγραφούν τα εργαλεία με τα οποία ασχοληθήκαμε και χρησιμοποιήθηκαν για την αποθήκευση των δεδομένων. Επιπλέον, θα γίνει αναφορά στην αρχιτεκτονική η οποία υιοθετείται καθώς επίσης θα ακολουθήσουν τα πλεονεκτήματα, τα μειονεκτήματα και οι λόγοι χρήσης ενός τέτοιου εργαλείου.

3.1 Σύστημα διαχείρισης βάσεων δεδομένων

Βάση δεδομένων ονομάζεται η δομή στην οποία αποθηκεύονται συλλογές δεδομένων με τέτοιο τρόπο έτσι ώστε τα δεδομένα να σχετίζονται μεταξύ τους. Επιπλέον, οι βάσεις δεδομένων, εκτός της αποθήκευσης, παρέχουν την δυνατότητα της προσπέλασης και της ανάκτησης των δεδομένων, με την χρήση σχετικών ερωτημάτων προς αυτές [4]. Ουσιαστικά, η βάση δεδομένων περιέχει δεδομένα σχετιζόμενα μεταξύ τους, προκειμένου να παράγεται κάποια πληροφορία και η οποία καθιστά εφικτή την ανάκτηση και προσπέλαση των δεομένων, έτσι ώστε να εξάγονται αποτελέσματα.

Ένα σύστημα διαχείρισης βάσεων δεδομένων, πρόκειται για λογισμικό με το οποίο πραγματοποιείται η δημιουργία και ο χειρισμός μιας βάσης δεδομένων [4]. Με τον όρο «δημιουργία» νοείται η κατασκευή μιας βάσης δεδομένων, ενώ ο όρος «χειρισμός» αποσκοπεί στην διαχείριση της βάσης και περιέχει τις λειτουργίες της εισαγωγής, διαγραφής και τροποποίησης των δεδομένων. Επιπλέον, τα συστήματα διαχείρισης βάσεων δεδομένων παρέχουν την δυνατότητα της συντήρησης των δεδομένων και την ανάκτηση αυτών.

Ένας τρόπος για να ομαδοποιηθούν τα συστήματα διαχείρισης βάσεων δεδομένων είναι ανάλογα με τον τρόπο οργάνωσης των δεδομένων τους. Οι βασικές κατηγορίες είναι τρεις. Η πρώτη αφορά δεδομένα που είναι δομημένα και μπορούν να αναπαρασταθούν σε σχέσεις μεταξύ τους, για αυτό και ονομάζονται σχεσιακές βάσεις δεδομένων. Η δεύτερη αφορά δεδομένα που δεν είναι οργανωμένα και συνήθως έχουν μορφή κλειδιού-τιμής (key-value pair). Επίσης, δεν είναι απαραίτητο τα δεδομένα να έχουν σχέσεις μεταξύ τους για αυτό οι βάσεις που φιλοξενούν τέτοια δεδομένα καλούνται ως no-sql βάσεις δεδομένων. Τέλος, υπάρχουν οι κατανεμημένες βάσεις δεδομένων στις οποίες τα δεδομένα αποθηκεύονται ανά ομάδες σύμφωνα με κριτήρια που εξυπηρετούν την καλύτερη αποδοτικότητα του συστήματος. Στην

παρούσα εργασία χρησιμοποιείται το τελευταίο μοντέλο βάσεων δεδομένων που αναφέρθηκε παραπάνω.

Συμπερασματικά, η βάση δεδομένων είναι μία δομή η οποία περιέχει μια συλλογή δεδομένων σχετιζόμενες μεταξύ τους και με την βοήθεια συστημάτων διαχείρισης βάσεων δεδομένων πραγματοποιείται η διαχείρισή τους.

3.2 Γιατί έχουν αναπτυχθεί εργαλεία διαχείρισης μεγάλου όγκου δεδομένων;

Όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο η αύξηση της ψηφιοποίησης της πληροφορίας έχει σαν αποτέλεσμα την δημιουργία ενός μεγάλου όγκου δεδομένων. Αυτό έχει σαν αποτέλεσμα να κάνει την αποθήκευσή τους αρκετά δύσκολη και ταυτόχρονα να μειώνει την αποδοτικότητα της ανάλυσης, της αναζήτησης και γενικότερα της επεξεργασίας αυτών με την χρήση συμβατικών εργαλείων διαχείρισης βάσεων δεδομένων. Επιπλέον, οι εταιρίες, οι επιχειρήσεις και οι οργανισμοί επιθυμούν να διαχειρίζονται όσο το δυνατό πιο αποδοτικά τα δεδομένα που έχουν στην διάθεσή τους. Τα δεδομένα αυτά, λόγω του όγκου που καταλαμβάνουν, είναι ανέφικτο να διαχειριστούν από τα παραδοσιακά συστήματα διαχείρισης και για τον λόγο αυτό αναπτύχθηκαν συστήματα διαχείρισης μεγάλου όγκου δεδομένων [5].

Τα συστήματα διαχείρισης μεγάλου όγκου δεδομένων, αναπτύχθηκαν με τέτοιο τρόπο έτσι ώστε να υποστηρίζεται η παράλληλη και κατανεμημένη επεξεργασία των δεδομένων με απώτερο σκοπό την ταχύτερη εξαγωγή αποτελεσμάτων. Ουσιαστικά τα δεδομένα ομαδοποιούνται και κατανέμονται σε έναν αριθμό τερματικών, που έχουν τον ρόλο του εξυπηρετητή (server) και υποστηρίζουν την παράλληλη επεξεργασία των δεδομένων, εξοικονομώντας αρκετό χρόνο.

3.3 Περιγραφή HBase

Η HBase αναπτύχθηκε από την Apache και διατέθηκε τον Μάρτιο του 2008. Ο λόγος της δημιουργίας της ήταν εξαιτίας της αύξησης του όγκου των δεδομένων.

Η HBase είναι ένα σύστημα διαχείρισης μεγάλου όγκου δεδομένων το οποίο ανταποκρίνεται επάξια στην ταχύτητα επεξεργασίας, αποθήκευσης και γενικότερα της διαχείρισης δεδομένων μεγάλου όγκου.

Η HBase είναι μία βάση δεδομένων τυχαίας πρόσβασης και ανάγνωσης, δηλαδή επιτρέπει την πρόσβαση σε δεδομένα που είναι αποθηκευμένα οπουδήποτε και αν βρίσκονται στην βάση και πρόκειται για μία κατανεμημένη βάση δεδομένων, η οποία αποθηκεύει τα δεδομένα της σε διαφορετικούς εξυπηρετητές. Ο λόγος που πραγματοποιείται αυτό είναι για να επιτυγχάνεται η κατανεμημένη επεξεργασία των δεδομένων, κάνοντας ταχύτερη την διαχείριση αυτών. Γενικότερα η HBase βασίζεται στο σύστημα του Hadoop το οποίο με την σειρά υιοθετεί την αρχιτεκτονική των

κατανεμημένων αρχείων (HDFS-Hadoop Distributed File System). Το Hadoop καθώς επίσης και η αρχιτεκτονική του HDFS θα αναλυθούν περαιτέρω σε παρακάτω ενότητα αυτού του κεφαλαίου.

Ένα κύριο χαρακτηριστικό της HBase είναι ότι πρόκειται για μία noSQL βάση δεδομένων ή αλλιώς non-relational βάση δεδομένων. Αυτό σημαίνει ότι δεν χρησιμοποιείται το παραδοσιακό μοντέλο των σχεσιακών βάσεων δεδομένων (RDBMS-Relational Database Management System) [6]. Βάσει του προαναφερθέντος, μία noSQL βάση δεδομένων όπως είναι η HBase χρησιμοποιείται για ημι-δομημένα δεδομένα, δηλαδή για δεδομένα που δεν είναι καλά ορισμένα και αυτό διότι τα συστήματα διαχείρισης μεγάλου όγκου δεδομένων έχουν ως σκοπό την ταχύτερη προσπέλαση και την επεξεργασία των δεδομένων δίχως να δίνεται σημασία για την σχετικότητα αυτών.

Επιπλέον, λόγω του ότι η HBase χρησιμοποιείται και για δεδομένα που δεν είναι καλά ορισμένα, διαθέτει την ικανότητα να ομαδοποιεί τα δεδομένα στη βάση χρησιμοποιώντας τις οικογένειες στηλών (Column Family). Με αυτόν τον τρόπο δίνοντας τους ένα αναγνωριστικό όνομα στη στήλη ομαδοποιεί τα δεδομένα που βρίσκονται στην βάση [6]. Επιπλέον, κάθε εγγραφή της HBase διαθέτει ένα μοναδικό αναγνωριστικό (key) το οποίο αντιστοιχεί και ταυτοποιεί μία και μόνο μία εγγραφή.

Περιληπτικά, η HBase είναι ένα σύστημα τυχαίας πρόσβασης και ανάγνωση με δυνατότητες διαχείρισης μεγάλου όγκου δεδομένων. Επιπρόσθετα, βασίζεται στην κατανεμημένη αποθήκευση και επεξεργασία με αποτέλεσμα να επιτυγχάνει τις αποδοτικότερες διαδικασίες διαχείρισης.

Επισημαίνεται, ότι στην παρούσα εργασία δεν θα ασχοληθούμε με την παράλληλη επεξεργασία των δεδομένων.

3.4 Περιγραφή Hadoop

Το Hadoop είναι ένα ανοιχτού κώδικα λογισμικό το οποίο κυκλοφόρησε τον Απρίλιο του 2006 από την Apache [7]. Η κύρια χρησιμότητα του είναι ότι έχει σημαντικό ρόλο στην επίλυση προβλημάτων που συναντώνται στα μεγάλου όγκου δεδομένα και είναι υπεύθυνο για την εύρυθμη λειτουργία των εξυπηρετητών στους οποίους φιλοξενούνται τα δεδομένα.

Επιπλέον, το Hadoop αποθηκεύει τα δεδομένα σε αρχεία και υιοθετεί την αρχιτεκτονική HDFS. Βάσει αυτής της αρχιτεκτονικής δημιουργεί μπλοκ και διανέμει τα αρχεία στους κόμβους που έχει στην διάθεσή του. Με την έννοια του κόμβου, νοείται το τερματικό (εξυπηρετητής), που είναι υπεύθυνο για την αποθήκευση των δεδομένων. Επιπρόσθετα, όλοι οι κόμβοι, συνυπάρχουν στο ίδιο δίκτυο και αυτό δίνει την δυνατότητα της παράλληλης επεξεργασίας των δεδομένων και κατ΄ επέκταση την ταχύτερη και αποδοτική επεξεργασία.

Επιπλέον, το Hadoop, διαθέτει εργαλεία παρακολούθησης (monitoring) τα οποία ελέγχουν την εύρυθμη λειτουργία των κόμβων για την αποφυγή δυσλειτουργιών. Επιπρόσθετα, έχει την δυνατότητα να επιλύει αυτόματα και ανεξάρτητα τις αστοχίες του συστήματος και είναι ανεκτό σε σφάλματα.

3.5 Αρχιτεκτονική HDFS

Το HDFS (Hadoop Distributed File System), είναι μία αρχιτεκτονική η οποία αποσκοπεί στην κατανεμημένη επεξεργασία δεδομένων μεγάλου όγκου. Με την χρήση του HDFS γίνεται ο διαμερισμός των δεδομένων σε μπλοκ και κατανέμονται σε εξυπηρετητές με απώτερο σκοπό την παράλληλη επεξεργασία αυτών. Επιπλέον, διατηρεί αντίγραφο των δεδομένων, κάτι το οποίο είναι αρκετά χρήσιμο για την αντιμετώπιση κάποιας αστοχίας.

Η αρχιτεκτονική αυτή διαθέτει μηχανισμούς που ελέγχουν την ακεραιότητα των δεδομένων στους κόμβους καθώς επίσης και μηχανισμούς που ελέγχουν την εύρυθμη λειτουργία του συστήματος. Επιπρόσθετα, έχει την δυνατότητα εντοπισμού και αυτόματης επίλυσης σφαλμάτων.

Επιπλέον, το HDFS είναι βασισμένο στην αρχιτεκτονική Master-Slave. Στο HDFS υπάρχουν δύο κόμβοι οι οποίοι είναι υπεύθυνοι για την διαχείριση των αρχείων. Αυτοί οι κόμβοι είναι ο NameNode και ο DataNode. Ο NameNode αποτελείται από τουλάχιστον έναν ενεργό εξυπηρετητή (server) όπου αποθηκεύονται μεταδεδομένα που μπορούν να ταυτοποιήσουν ένα αρχείο. Τέτοια δεδομένα είναι το όνομα του αρχείου, το μπλοκ στο οποίο εμπεριέχεται το αρχείο και η τοποθεσία του αρχείου στην βάση. Από την άλλη πλευρά ο DataNode είναι υπεύθυνος για τις διεργασίες της ανάγνωσης, εγγραφής και επεξεργασίας των δεδομένων κατ΄εντολή του NameNode.

Ουσιαστικά στην αρχιτεκτονική Master-Slave που υιοθετεί η αρχιτεκτονική HDFS, ο κόμβος NameNode είναι ο Master και ο DataNode είναι ο Slave.

3.6 Λεπτομέρειες της HBase

Σε αυτή την ενότητα θα αναφερθούν ο τρόπος λειτουργίας του συστήματος διαχείρισης HBase, οι λόγοι χρήσης, τα πλεονεκτήματα και τα μειονεκτήματα ενός τέτοιου συστήματος.

3.6.1 Τρόπος λειτουργίας της HBase [8]

Η HBase βασίζεται στην αρχιτεκτονική του Hadoop και χρησιμοποιεί πίνακες για την αποθήκευση των δεδομένων, σε αντίθεση με το Hadoop που χρησιμοποιεί αρχεία. Οι πίνακες που αποθηκεύονται στην HBase είναι πολυδιάστατοι όπου κάθε γραμμή

περιέχει ένα κλειδί προκειμένου να εξασφαλίζεται η μοναδικότητά της και έναν αριθμό στηλών. Επιπλέον, η HBase θέτει ένα χρονικό στιγμιότυπο (timestamp) σε κάθε εγγραφή-γραμμή των πινάκων.

Επίσης, η HBase ομαδοποιεί τα χαρακτηριστικά των εγγραφών σε μία ή περισσότερες οικογένειες στηλών (Column Family). Προκειμένου να γίνει αντιληπτή αυτή η τεχνική, δίνεται το ακόλουθο παράδειγμα. Έστω ότι ένας πίνακας περιέχει τα χαρακτηριστικά Ονομασία οδού, Αριθμός, Πόλη και Ταχυδρομικός κώδικας, τότε το Column Family που θα περιείχε αυτά τα στοιχεία θα ονομαζόταν «Διεύθυνση». Επιπλέον, είναι προτιμότερο η δήλωση των Column Family, να γίνεται κατά την δημιουργία ενός πίνακα και όχι αφετέρου της δημιουργίας. Επιπρόσθετα, η χρήση των Column Family είναι ιδιαίτερα σημαντική, όσον αφορά την ανάλυση των δεδομένων.

Η αναζήτηση, η επεξεργασία και γενικότερα η πρόσβαση στις εγγραφές των πινάκων γίνεται κυρίως με το κλειδί που διακρίνει κάθε εγγραφή. Επιπλέον, κάθε εγγραφή σε ένα πίνακα της HBase αποτελείται από ένα κλειδί (rowkey), από μία Column Family και από μία χρονική σήμανση (timestamp) [9]. Επίσης, κάθε πίνακας της HBase χωρίζεται σε περιοχές δίνοντας ένα αναγνωριστικό στην κάθε μία για να γίνεται πιο εύκολη η αναζήτηση των αντικειμένων. Άλλο ένα χαρακτηριστικό της HBase είναι ότι σε περίπτωση τροποποίησης των δεδομένων που περιέχει, διατηρεί τις προηγούμενες τιμές, ως αντίγραφο ασφαλείας (backup), κάνοντάς το ανεκτό σε σφάλματα.

Επιπλέον, όπως το Hadoop, έτσι και η HBase υιοθετεί και αυτή την αρχιτεκτονική Master-Slave, όπου ο HMaster, ο οποίος είναι ένας εξυπηρετητής που ευθύνεται για αιτήματα εκχώρησης, διαγραφής ή τροποποίησης των δεδομένων στην βάση που πραγματοποιούνται με την βοήθεια του HRegionServer. Ο HRegionServer έχει την ευθύνη για την επεξεργασία των αιτημάτων του HMaster καθώς επίσης είναι ο ίδιος που μπορεί να γράψει και να διαβάσει στην βάση, κατόπιν αιτήματος του HMaster. Ουσιαστικά ο HMaster είναι ο Master και ο HRegionServer είναι ο Slave, όσον αφορά την αρχιτεκτονική Master-Slave.

Επιπλέον, η HBase διαθέτει άλλο ένα εργαλείο του Hadoop, το οποίο ονομάζεται Zookeeper και είναι υπεύθυνο για να ενημερώνει την κατάσταση των RegionServers καθώς και για τις διάφορες ενέργειες που γίνονται σε αυτούς [8]. Επισημαίνεται ότι η αναζήτηση στην HBase είναι ταχύτερη και αποτελεσματική.

3.6.2 Πλεονεκτήματα της HBase

Τα πλεονεκτήματα της HBase είναι ότι πρόκειται για ένα σύστημα διαχείρισης μεγάλου όγκου δεδομένων, που διευκολύνει την διαχείρισή τους. Επιπλέον, είναι κατανεμημένο σύστημα, το οποίο διαμοιράζει στους κόμβους τις διεργασίες εξοικονομώντας χρόνο. Διατηρεί αντίγραφα των δεδομένων κάνοντάς το ανεκτικό σε

σφάλματα του χρήστη και τις αστοχίες του συστήματος. Επιπλέον, είναι ένα σύστημα με υψηλή αποδοτικότητα.

3.6.3 Μειονεκτήματα της HBase

Τα μειονεκτήματα της HBase είναι ότι πρόκειται για ένα σύστημα αρκετά δύσκολο ως προς την κατανόηση του τρόπου λειτουργίας του, με αποτέλεσμα να κρίνεται αναγκαία η χρήση του από εξειδικευμένο προσωπικό απαιτώντας αρκετή εμπειρία. Επιπλέον, είναι ένα σύστημα το οποίο, για να είναι πλήρως λειτουργικό θα πρέπει να εγκατασταθούν επιπλέον λογισμικά, δυσκολεύοντας κατά πολύ την εγκατάσταση του, επειδή αυτά τα προγράμματα θα πρέπει να είναι συνεργάσιμα μεταξύ τους.

3.6.4 Λόγοι χρήσης της HBase

Βάσει των παραπάνω, γίνεται αντιληπτό, ότι η χρήση του συστήματος διαχείρισης της HBase είναι αναγκαία για την σημερινή εποχή. Οι περισσότερες εταιρίες, οργανισμοί ή ακόμα και επιχειρήσεις έχουν την ανάγκη για την αποθήκευση, την διαχείριση και γενικότερα την επεξεργασία μεγάλου όγκου δεδομένων. Ο χρόνος απόκρισης της επεξεργασίας των δεδομένων είναι αρκετά σημαντικός θέτοντας σκόπιμο την υιοθέτηση ενός διαφορετικού συστήματος διαχείρισης βάσεων δεδομένων που ανταποκρίνεται ανελλιπώς στις απαιτήσεις. Η HBase λόγω του μοντέλου στο οποίο είναι βασισμένη, ανταποκρίνεται θετικά στα προαναφερθέντα ζητήματα συμβάλλοντας έτσι αρκετά στην μείωση του χρόνου επεξεργασίας των δεδομένων.

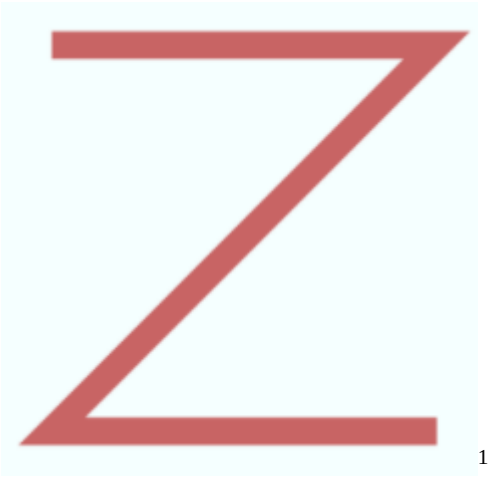
4

Μέθοδος Z-Order

Σε αυτό το κεφάλαιο θα περιγραφεί και θα αναλυθεί η μέθοδος του Z-Order. Αυτή η μέθοδος χρησιμοποιείται στην παρούσα εργασία προκειμένου να ανατεθεί σε κάθε εγγραφή, των δεδομένων, ένα μοναδικό αναγνωριστικό.

4.1 Τι είναι η μέθοδος Z-order [10]

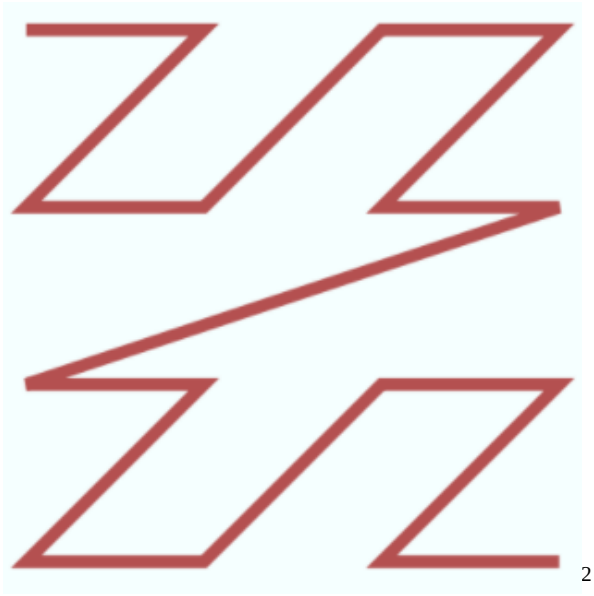
Η μέθοδος Z-Order είναι μία ευρέως γνωστή μέθοδος ως προς την ταξινόμηση γεωγραφικών συντεταγμένων και ανήκει στην κατηγορία των Space Filling Curves μεθόδων. Χρησιμοποιείται για την μετατροπή δεδομένων πολλών διαστάσεων σε μία. Με την μέθοδο Z-Order δίνεται η δυνατότητα ταξινόμησης γεωγραφικών συντεταγμένων, με κριτήριο την τιμή Z, που παράγεται για κάθε εγγραφή. Πιο συγκεκριμένα, η ταξινόμηση των δεδομένων γίνεται με την τιμή Z της κάθε μίας εγγραφής, δημιουργώντας το σύμβολο Z. Επισημαίνεται ότι κάθε γωνία του Z αναπαριστά ένα σημείο στο επίπεδο. Στις παρακάτω εικόνες, φαίνεται ο γενικός τρόπος λειτουργίας του Z-Order.



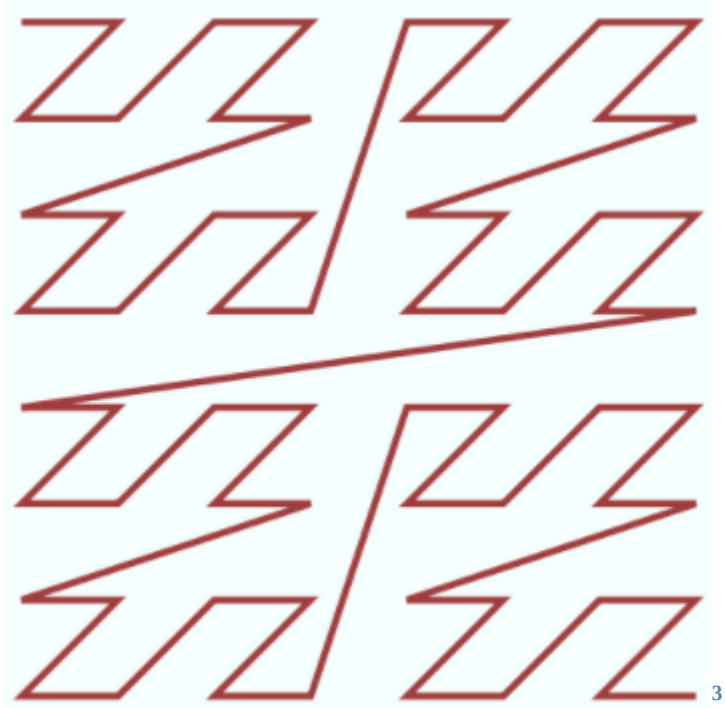
[1]

Εικόνα 1. Ταξινόμηση τεσσάρων (4 ή 4^1) γεωγραφικών σημείων (Z-Order)

[1] https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Four-level_Z.svg/1200px-Four-level_Z.svg.png


[2]

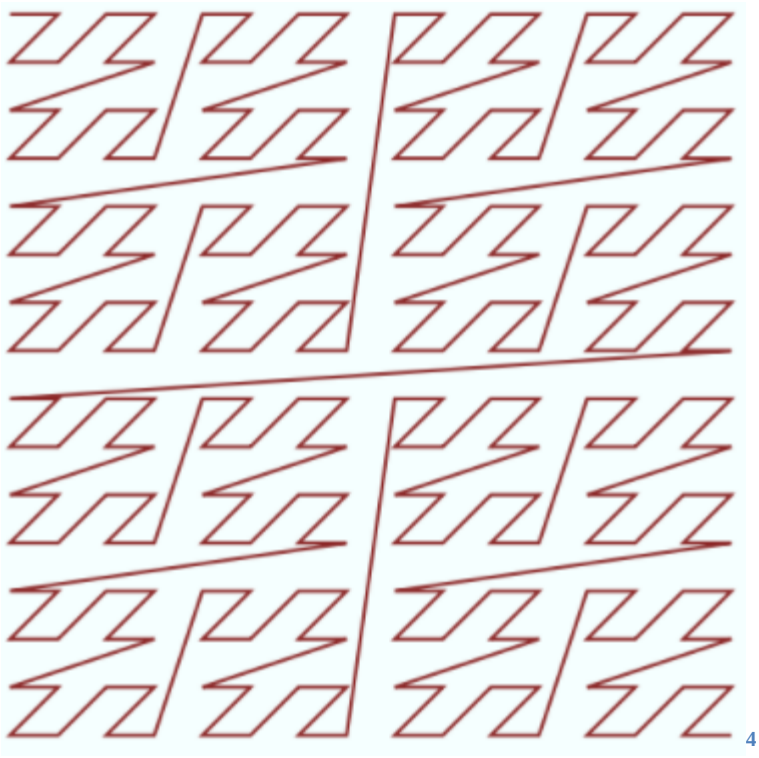
Εικόνα 2. Ταξινόμηση δεκαέξι (16 ή 4^2) γεωγραφικών σημείων (Z-Order)


[3]

Εικόνα 3. Ταξινόμηση εξήντα-τεσσάρων (64 ή 4^3) γεωγραφικών σημείων (Z-Order)

[2] https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Four-level_Z.svg/1200px-Four-level_Z.svg.png

[3] https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Four-level_Z.svg/1200px-Four-level_Z.svg.png



Εικόνα 4. Ταξινόμηση διακοσίων πενήντα έξι (256 ή 4^4) γεωγραφικών σημείων (Z-Order)

Επιπλέον, η μέθοδος, αυτή, όπως αναφέρθηκε παραπάνω, χρησιμοποιείται για την μετατροπή πολλών διαστάσεων σε μία. Πιο συγκεκριμένα ο σκοπός της χρήσης της μεθόδου Z-Order, στην παρούσα εργασία, είναι η αναπαράσταση των συντεταγμένων (Latitude και Longitude) σε μία τιμή με στόχο κάθε εγγραφή να αντιπροσωπεύεται και να αντιστοιχίζεται με αυτήν την μοναδική τιμή. Αυτό το χαρακτηριστικό ήταν και ο κύριος λόγος που επιλέχθηκε, αυτή η μέθοδος, για την παρούσα εργασία.

4.2 Περιορισμοί εφαρμογής της μεθόδου Z-Order

Η μέθοδος Z-Order έχει τρεις (3) βασικούς περιορισμούς που αφορούν την εφαρμογή της:

- Ο πρώτος είναι ότι η μέθοδος εφαρμόζεται σε τιμές ακέραιες και όχι σε δεκαδικές.
- Ο δεύτερος είναι ότι η μέθοδος είναι εφικτά υλοποιήσιμη σε θετικές τιμές και όχι σε αρνητικές.
- Ο τρίτος είναι ότι εφαρμόζεται μόνο σε δυαδικής (binary) μορφής δεδομένα.

[4] https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/Four-level_Z.svg/1200px-Four-level_Z.svg.png

4.3 Παραμετροποίηση των δεδομένων για την δημιουργία του Z-Order

Τα δεδομένα που έχουμε στην διάθεσή μας αφορούν καταστήματα εστίασης. Ένα χαρακτηριστικό αυτών των εγγραφών είναι ότι κάθε μία εγγραφή διαθέτει συντεταγμένες γεωγραφικού μήκους και πλάτους (Latitude και Longitude). Οι συντεταγμένες αυτές αφορούν δεκαδικές, θετικές ή αρνητικές τιμές.

Βάσει των παραπάνω περιορισμών που αναφέρθηκαν, αναγκαστικά θα πρέπει να πραγματοποιηθεί κάποια παραμετροποίηση στα δεδομένα έτσι ώστε να καθίσταται εφικτή η εφαρμογή της Z-Order.

Οι μετατροπές που θα πραγματοποιηθούν στα δεδομένα είναι η μετατροπή των γεωγραφικών συντεταγμένων από αρνητικές σε θετικές τιμές και από δεκαδικές σε ακέραιες.

4.3.1 Απαλοιφή αρνητικού προσήμου

Παρατηρώντας τα δεδομένα του αρχείου, όπου περιέχονται τα χαρακτηριστικά των δεδομένων που έχουμε στην διάθεσή μας και αναζητώντας στο διαδίκτυο, συμπεραίνεται ότι οι τιμές Latitude έχουν εύρος τιμών από -90 έως 90, για το Νότιο και Βόρειο ημισφαίριο αντίστοιχα, ενώ οι τιμές Longitude έχουν εύρος τιμών από -180 έως 180, καθορίζοντας τις Ανατολικές και Δυτικές τιμές.

Προκειμένου να πραγματοποιηθεί η μετατροπή των μεταβλητών Latitude και Longitude σε τιμές με θετικό πρόσημο, εξαλείφοντας το αρνητικό πρόσημο, προστίθεται σε κάθε μεταβλητή Latitude η τιμή 90 και σε κάθε μεταβλητή Longitude η τιμή 180. Με αυτό τον τρόπο οι τιμές της μεταβλητής Latitude θα έχουν εύρος τιμών από 0 (μηδέν) έως 180 και οι τιμές της μεταβλητής Longitude από 0 (μηδέν) έως 360, χωρίς να αλλάξει η σχέση (απόσταση) μεταξύ των σημείων.

Αυτό έχει ως αποτέλεσμα όλες οι τιμές να έχουν θετικό πρόσημο έτσι ώστε να ικανοποιείται η συνθήκη του Z-Order προκειμένου να διεξάγεται ομαλά και ορθά η δημιουργία του.

4.3.2 Μετατροπή των δεκαδικών τιμών σε ακέραιες τιμές

Για την μετατροπή του δεκαδικού αριθμού σε ακέραιο, πραγματοποιείται πολλαπλασιασμός των τιμών Latitude και Longitude με δυνάμεις του 10 (δηλαδή 10^n).

Το n αντιπροσωπεύει τον εκθέτη της δύναμης του 10. Με την βοήθεια του n θα δημιουργείται μία τιμή (κάποιο πολλαπλάσιο του 10), που όταν αυτή

πολλαπλασιάζεται με μία τιμή Latitude ή Longitude να αποδίδει μία τιμή με ικανοποιητική ακρίβεια.

Μία τιμή που μπορεί να αποδοθεί στο n είναι το έξι (6) και αυτό, διότι αν πολλαπλασιαστεί μία τιμή Latitude ή Longitude, με 10^6 αποδίδει μία ικανοποιητική ακρίβεια διατηρώντας τα πρώτα 6 σημαντικά ψηφία, πριν της υποδιαστολής, για κάθε τιμή. Με αυτό τον τρόπο, επιτυγχάνεται το ζητούμενο, όλες οι τιμές των Latitude και Longitude να είναι ακέραιες, έτσι ώστε να εφαρμοστεί, ορθά, η μέθοδος της Z-Order.

4.4 Τρόπος δημιουργίας του Z-Order

Η μέθοδος του Z-Order εφαρμόζεται σε τιμές με δυαδική μορφή, όπως αναφέρθηκε και παραπάνω. Επομένως, έπειτα της μετατροπής σε ακέραιες και θετικές τιμές των Latitude και Longitude των εγγραφών, μετατρέπουμε τις τιμές αυτές σε δυαδική μορφή (Binary) των 32 bits για κάθε εγγραφή. Σε περίπτωση που μία μεταβλητή έχει λιγότερα από 32 bits τότε σε αυτή, επισυνάπτονται μηδενικά bits, μέχρι η μεταβλητή να διαθέτει 32 bits.

Έπειτα, της διεκπεραίωσης της προαναφερθείσας ενέργειας δημιουργούμε το Z-Order. Για την δημιουργία του Z-Order εκχωρούμε εναλλάξ τα bits του Latitude και Longitude σε μία νέα μεταβλητή. Η μεταβλητή που παράγεται είναι η τιμή του Z-Order και αντιστοιχεί σε μία και μόνο μία τοποθεσία και αντιπροσωπεύει μία μοναδική εγγραφή.

4.5 Παράδειγμα δημιουργίας τιμής Z-Order με εφαρμογή στα δεδομένα μας

Έστω ότι έχουμε τις παρακάτω μεταβλητές:

Latitude = 37.755557

Longitude = -122.250360846519

Αρχικά, ο σκοπός μας είναι η εξάλειψη του αρνητικού προσήμου (-). Οπότε προσθέτουμε στις τιμές Latitude και Longitude τις τιμές 90 και 180 αντίστοιχα. Επομένως, οι μεταβλητές που παράγονται είναι οι παρακάτω:

Latitude = 37.755557 **+ 90** = 127.755557

Longitude = -122.250360846519 **+ 180** = 57.749639153481

Στην συνέχεια πολλαπλασιάζουμε τις παραπάνω τιμές με την τιμή 10^6. Επομένως οι τιμές που προκύπτουν είναι οι παρακάτω:

Latitude = 127.755557 *** 10^6** = 127755557

Longitude = 57.749639153481 *** 10^6** = 57749639.153481

Έπειτα, μετατρέπουμε τους αριθμούς σε ακέραιους, κρατώντας μόνο τα ψηφία πριν την υποδιαστολή, επομένως έχουμε:

Latitude = 127755557

Longitude = 57749639

Εν συνεχεία, μετατρέπουμε τις τιμές σε δυαδική μορφή των 32 bits:

Latitude = 00000111 10011101 01100101 00100101

Longitude = 00000011 01110001 00110000 10000111

Τέλος, η τιμή του Z-Order παράγεται από την εναλλάξ τοποθέτηση των bits των Latitude και Longitude στην μεταβλητή Z-Order και είναι η παρακάτω:

Latitude = 00000111 10011101 01100101 00100101
Longitude = 00000011 01110001 00110000 10000111

Z_Order=0000000000101111100101111010001100101101001000100100100001101111

5

Μέθοδοι αποθήκευσης των δεδομένων

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι μέθοδοι αποθήκευσης που εφαρμόστηκαν στην παρούσα εργασία.

5.1 Τι είναι οι μέθοδοι αποθήκευσης δεδομένων;

Κυρίως όταν αναφερόμαστε σε βάσεις δεδομένων, αναφερόμαστε σε δεδομένα που είναι αποθηκευμένα σε κάποιο τερματικό. Ωστόσο, στα τερματικά που διατηρούν μεγάλου όγκου δεδομένα μας ενδιαφέρει ως επί το πλείστων ο τρόπος με τον οποίο αποθηκεύονται τα δεδομένα, έτσι ώστε να επιτυγχάνεται η επεξεργασία τους και γενικότερα η διαχείρισή τους. Αυτό, αναλαμβάνεται από τις μεθόδους αποθήκευσης των δεδομένων.

Οι μέθοδοι αποθήκευσης των δεδομένων, προτείνουν εναλλακτικούς τρόπους αποθήκευσης των δεδομένων. Ουσιαστικά είναι υπεύθυνες για την οργάνωση των δεδομένων στις βάσεις δεδομένων. Αυτή η οργάνωση αποσκοπεί στην ταξινόμηση και στην ομαδοποίηση των δεδομένων, προκειμένου η επεξεργασία και η διαχείριση αυτών να γίνεται περισσότερο αποδοτική και αποτελεσματική.

Όπως γίνεται αντιληπτό, από τα παραπάνω, η επιλογή και η εφαρμογή μιας μεθόδου αποθήκευσης δεδομένων είναι αρκετά σημαντική διότι βάσει αυτών εξαρτάται η οργάνωση των δεδομένων. Η οργάνωση των δεδομένων επιδρά σε μεγάλο βαθμό στην επεξεργασία και γενικότερα στην διαχείριση αυτών. Επομένως, κατά την διαδικασία επιλογής μια τέτοιας μεθόδου πρέπει να λαμβάνονται υπόψη ο χρόνος που απαιτεί κάθε μέθοδος και ο τρόπος οργάνωσης αυτών των δεδομένων. Στις παρακάτω ενότητες θα παρουσιαστούν οι μέθοδοι αποθήκευσης που προτιμήθηκαν και υιοθετήθηκαν στην παρούσα εργασία.

5.2 Μέθοδος αποθήκευσης Ordered Indexing [11]

Η συγκεκριμένη μέθοδος είναι υπεύθυνη για την οργάνωση των δεδομένων στην βάση δεδομένων της HBase. Αρχικά, τα δεδομένα θα ταξινομούνται σύμφωνα με έναν αλγόριθμο ταξινόμησης και στην συνέχεια ομαδοποιούνται και εισάγονται στους πίνακες της HBase. Η ταξινόμηση και η ομαδοποίηση των δεδομένων θα γίνεται με βάση το μοναδικό αναγνωριστικό που έχει δημιουργηθεί για κάθε μία από αυτές (τιμή Z-Order). Σκοπός της Ordered Indexing είναι η ομαδοποίηση των

δεδομένων σε περισσότερους από έναν πίνακες, δηλαδή ο διαμερισμός και η κατανομή των δεδομένων. Η χρήση αυτής της μεθόδου έχει ως στόχο να μειώσει τον όγκο των δεδομένων που χρειάζεται να προσπελασθούν κατά την εκτέλεση ενός ερωτήματος.

Η πρώτη διαδικασία αφορά την ταξινόμηση [12] των δεδομένων σύμφωνα με ένα μοναδικό κλειδί κάθε εγγραφής. Στην συγκεκριμένη διαδικασία είναι απαραίτητο να καθοριστεί η κατάλληλη μέθοδος με την οποία θα ταξινομηθούν τα δεδομένα ώστε να παρέχεται η μέγιστη δυνατή αποδοτικότητα.

Οι τρεις μέθοδοι ταξινόμησης που μπορούν να χρησιμοποιηθούν είναι η Merge Sort, η Quick Sort και η Heap Sort. Για να επιλεγεί η κατάλληλη μέθοδος ταξινόμησης για την υλοποίηση που προαναφέρθηκε θα πρέπει να γίνει σύγκριση των τριών μεθόδων ως προς την απόδοση της χρονικής και χωρικής πολυπλοκότητας καθώς επίσης και στην σταθερότητα που προσδίδουν αυτές. Με τον όρο σταθερότητα εννοούμε την ικανότητα του αλγορίθμου να μπορεί να διατηρεί την σειρά κατάταξης των αντικειμένων που διαθέτουν το ίδιο κλειδί, σύμφωνα με κάποιο δευτερεύον κριτήριο. Βέβαια, στην περίπτωση της παρούσας εργασίας, δεν θα μας απασχολήσει η σταθερότητα που αποδίδουν οι αλγόριθμοι, διότι υπάρχει ένα μοναδικό κριτήριο που θα καθορίσει την κατάταξη των αντικειμένων.

5.2.1 Τρόπος λειτουργίας **Ordered Indexing**

Αρχικά γίνεται ταξινόμηση των δεδομένων που διατίθενται βάσει ενός μοναδικού κλειδιού που αντιπροσωπεύει μία και μόνο μία εγγραφή. Έπειτα της ταξινόμησης, των εγγραφών, τα δεδομένα διανέμονται και αποθηκεύονται σε πίνακες. Εν συνεχεία, δημιουργείται ένας επιπλέον πίνακας, ο οποίος περιέχει τις αρχικές και τις τελικές τιμές (πρώτη και τελευταία θέση) κάθε πίνακα. Αυτό αποσκοπεί στην δημιουργία ενός ευρετηρίου των πινάκων για ταχύτερη αναζήτηση των δεδομένων κατά την εκτέλεση ενός ερωτήματος που τίθεται από τον χρήστη. Για παράδειγμα όταν ο χρήστης θέτει ένα ερώτημα στην βάση, πρώτα αναζητείται στο ευρετήριο με σκοπό να βρεθεί σε ποιον πίνακα ανήκουν τα δεδομένα που επιθυμεί να ανακτήσει και έπειτα πραγματοποιείται η αναζήτηση σε συγκεκριμένους πίνακες.

5.3 Επιλογή αλγορίθμου ταξινόμησης

Στον προγραμματισμό, με τον όρο «ταξινόμηση» εννοούμε την αύξουσα ή φθίνουσα σειρά των αντικειμένων. Εφαρμόζεται σε αντικείμενα τα οποία έχουν μία ακανόνιστη και τυχαία θέση, μέσα σε μία δομή αποθήκευσης, με απώτερο σκοπό την ταχύτερη αναζήτηση αυτών και την βέλτιστη διαχείριση τους. Υπάρχουν αρκετοί μέθοδοι που χρησιμοποιούνται για αυτό τον σκοπό, όπου η κάθε μία αποδίδει διαφορετικά ως προς την αποδοτικότητα και την πολυπλοκότητα της.

Σε αυτή την ενότητα θα παρουσιαστούν τα χαρακτηριστικά των αλγορίθμων καθώς επίσης θα γίνει μία συγκριτική αναφορά αυτών, η οποία θα δικαιολογεί την επιλογής μας. Οι τρεις αλγόριθμοι ταξινόμησης που αποδίδουν καλύτερα και θα παρουσιαστούν παρακάτω είναι η Merge Sort, η Quick Sort και η Heap Sort.

5.3.1 Αλγόριθμος ταξινόμησης Merge Sort

Η Merge Sort είναι ένας αλγόριθμος ταξινόμησης που βασίζεται στην έκφραση «Διαίρει και Βασίλευε». Εκτελεί διαδοχικές διαιρέσεις ενός πίνακα με σκοπό κάθε στοιχείο, που περιέχεται σε αυτόν, να απομονώνεται. Στην συνέχεια εκτελεί συγκρίσεις μεταξύ αυτών των στοιχείων και τοποθετεί τα στοιχεία ταξινομημένα σε πίνακες μικρού μεγέθους. Με αυτό τον τρόπο κάθε πίνακας που δημιουργείται διατηρεί τα στοιχεία του ταξινομημένα. Έπειτα, συγκρίνει ξανά τα στοιχεία των νέων πινάκων που δημιουργούνται, ξεκινώντας τις συγκρίσεις από αριστερά προς τα δεξιά και τοποθετεί τα στοιχεία με αύξουσα ή φθίνουσα σειρά. Ουσιαστικά συγχωνεύει τα στοιχεία των πινάκων. Η προαναφερθείσα διαδικασία έχει ως αποτέλεσμα την ταξινόμηση του αρχικού πίνακα.

Ο χρόνος πολυπλοκότητας αυτού του αλγορίθμου στην χειρότερη, στην μέση και στην καλύτερη περίπτωση είναι O(n$\log n$) ενώ η χωρική πολυπλοκότητα είναι O(n).

5.3.2 Αλγόριθμος ταξινόμησης Quick Sort

Η Quick Sort είναι μία αρκετά διαδεδομένη τεχνική ταξινόμησης όπου βασίζεται στις συγκρίσεις και χρησιμοποιεί δείκτες θέσης (pointers position). Αρχικά θέτει στην τελευταία θέση του πίνακα έναν δείκτη με το όνομα pivot. Έπειτα θέτει στην πρώτη και στην προτελευταία θέση του πίνακα δύο δείκτες (pointer1 και pointer2 αντίστοιχα) και συγκρίνει το περιεχόμενο αυτών των δύο μεταξύ τους. Στην περίπτωση της αύξουσας ταξινόμησης, αν το στοιχείο στην πρώτη θέση είναι μεγαλύτερο από το στοιχείο της προτελευταίας θέσης, τότε μεταφέρει το στοιχείο της πρώτης θέσης στην προτελευταία θέση και το στοιχείο της προτελευταίας θέσης στην πρώτη θέση (swap). Εν συνεχεία, αλλάζει τις θέσεις των δεικτών (pointer1 και pointer2) κατά συν ένα (+1) του δείκτη pointer1 και κατά μείον ένα (-1) του pointer2, συνεχίζοντας την ίδια διαδικασία συγκρίσεων μέχρις ότου οι δύο δείκτες να αναφέρονται στην ίδια θέση του πίνακα. Έπειτα, συγκρίνεται το περιεχόμενο της θέσης που αναφέρονται και οι δύο δείκτες με το περιεχόμενο της τελευταίας θέσης (pivot) εκτελώντας την διαδικασία swap αν χρειαστεί. Με παρόμοιο τρόπο εκτελείται και φθίνουσα ταξινόμηση.

Αποτέλεσμα της παραπάνω διαδικασίας είναι η ταξινόμηση των στοιχείων και έχει χρόνο πολυπλοκότητας στην μέση και στην καλύτερη περίπτωση O(n$\log n$) ενώ στην χειρότερη O(n^2). Επιπλέον, η χωρική πολυπλοκότητα του αλγορίθμου είναι O($\log n$).

5.3.3 Αλγόριθμος ταξινόμησης Heap Sort

Η Heap Sort πρόκειται για μία μέθοδο ταξινόμησης των δεδομένων, όπου βασίζεται στις συγκρίσεις και στην δομή δεδομένων της σωρού. Η σωρός πρόκειται για μία δυαδική δενδρική δομή αποθήκευσης των δεδομένων και διαχωρίζεται σε δύο κατηγορίες, την σωρού μεγίστου (Max Heap) και την σωρού ελαχίστου (Min Heap) [12].

Η σωρός μεγίστου αφορά την ταξινόμηση των αντικειμένων με φθίνουσα σειρά (δηλαδή από την μεγαλύτερη προς την μικρότερη τιμή), η οποία απαιτεί ο κόμβος ρίζα του δέντρου να είναι η μέγιστη τιμή της σωρού και κάθε κόμβος του δέντρου να έχει τιμή μεγαλύτερη ή ίση από τον επόμενο, με αποτέλεσμα στα φύλλα του δέντρου να αποθηκεύονται οι μικρότερες τιμές.

Αντίθετα, η σωρός ελαχίστου αφορά την ταξινόμηση των αντικειμένων με αύξουσα σειρά (δηλαδή από την μικρότερη προς την μεγαλύτερη τιμή), η οποία απαιτεί ο κόμβος ρίζα του δέντρου να είναι η μικρότερη τιμή της σωρού και κάθε κόμβος του δέντρου να έχει τιμή μικρότερη ή ίση από τον επόμενο, με αποτέλεσμα στα φύλλα του δέντρου να αποθηκεύονται οι μέγιστες τιμές της σωρού. Ο χρόνος πολυπλοκότητας της Heap Sort τόσο στην χειρότερη όσο στην μέση και την καλύτερη περίπτωση είναι O($\log n$). Επιπλέον, όσον αφορά την χωρική πολυπλοκότητα του αλγορίθμου, είναι O(n).

5.3.4 Επιλογή αλγορίθμου

Από τα παραπάνω προκύπτει ο παρακάτω συγκεντρωτικός πίνακας, με τα χαρακτηριστικά των αλγορίθμων ταξινόμησης:

Αλγόριθμος ταξινόμησης	**Χειρότερη περίπτωση**	**Μέση περίπτωση**	**Καλύτερη περίπτωση**	**Χωρική πολυπλοκότητα**
Merge Sort	O($n\log n$)	O($n\log n$)	O($n\log n$)	O(n)
Heap Sort	O($n\log n$)	O($n\log n$)	O($n\log n$)	O(n)
Quick Sort	O(n^2)	O($n\log n$)	O($n\log n$)	O($\log n$)

Πίνακας 1. Σύγκριση μεθόδων ταξινόμησης

Από τα παραπάνω στοιχεία του εκάστοτε αλγορίθμου παρατηρείται ότι η επιλογή του αλγορίθμου ταξινόμησης θα γίνει μεταξύ της Heap Sort και της Merge Sort, διότι παρουσιάζουν την καλύτερη χρονική και χωρική πολυπλοκότητα. Και οι δύο αλγόριθμοι έχουν την ίδια χρονική πολυπλοκότητα όσον αφορά και τις τρεις περιπτώσεις (χειρότερη, μέση και καλύτερη) καθώς επίσης, έχουν και την ίδια χωρική πολυπλοκότητα. Επομένως, εφόσον η χρονική και χωρική πολυπλοκότητα των δύο αλγορίθμων είναι ίδια αποφασίζεται να χρησιμοποιηθεί ο αλγόριθμος της Heap Sort, διότι η υλοποίησή της είναι αρκετά πιο απλή και κατανοητή. Επιπλέον, η Heap Sort

θα χρησιμοποιηθεί σε συνδυασμό με τον σωρό ελαχίστου (Min Heap), στην παρούσα εργασία.

5.3.5 Λεπτομερής περιγραφή της Heap Sort

Επειδή, ο αλγόριθμος ταξινόμησης που επιλέχθηκε είναι η Heap Sort είναι σκόπιμο να παρουσιαστεί μια λεπτομερής περιγραφή αυτής.

Η ταξινόμηση της Heap Sort σε συνδυασμό με τη σωρού ελαχίστου, αναζητά την ελάχιστη τιμή των δεδομένων και την εκχωρεί στο πρώτο στοιχείο της σωρού [12]. Έπειτα, συνεχίζει την αναζήτηση των στοιχείων, εκχωρώντας τις μικρότερες τιμές προς την κορυφή του δέντρου, δίνοντας ως ρίζα την ελάχιστη τιμή και τις μεγαλύτερες τιμές προς τα κάτωθεν επίπεδα του δέντρου, προωθώντας τις μέγιστες τιμές στα φύλλα του δέντρου. Η σωρός μπορεί να είναι μία λίστα ή ένας πίνακας όπου εσωτερικά εκχωρούνται τα στοιχεία, από αριστερά προς τα δεξιά και από το μικρότερο προς το μεγαλύτερο (επειδή χρησιμοποιείται σωρό ελαχίστου-Min Heap).

Η ρίζα του δέντρου είναι το πρώτο στοιχείο του πίνακα ή της λίστας καθώς επίσης το δεύτερο και τρίτο στοιχείο είναι τα παιδιά της ρίζας. Υποθετικά, αν ένα στοιχείο που βρίσκεται στην θέση i του πίνακα και είναι πατέρας τότε το στοιχείο που βρίσκεται στην θέση 2*i του πίνακα είναι το αριστερό παιδί του πατέρα-κόμβου και το στοιχείο που βρίσκεται στην θέση 2*i+1 του πίνακα είναι το δεξί παιδί του πατέρα-κόμβου [12].

Αντίστροφα, αν επιθυμούμε να βρούμε τον πατέρα κόμβο ενός στοιχείο τότε χρειάζεται να βρεθεί η θέση i του στοιχείου. Έπειτα της εύρεσης της θέσης του στοιχείου, το $\lfloor i/2 \rfloor$ υποδηλώνει τον πατέρα-κόμβο.

Κατά την ταξινόμηση και ελέγχου της σωρού είναι δυνατόν να πραγματοποιηθεί εναλλαγή της θέσης των στοιχείων με τους πατέρες κόμβους και τα παιδιά. Αν διαπιστωθεί ότι υπάρχει κάποιο στοιχείο μικρότερο από κάποιο άλλο του δέντρου, τότε οι θέσεις των στοιχείων εναλλάσσονται μεταξύ τους (swap).

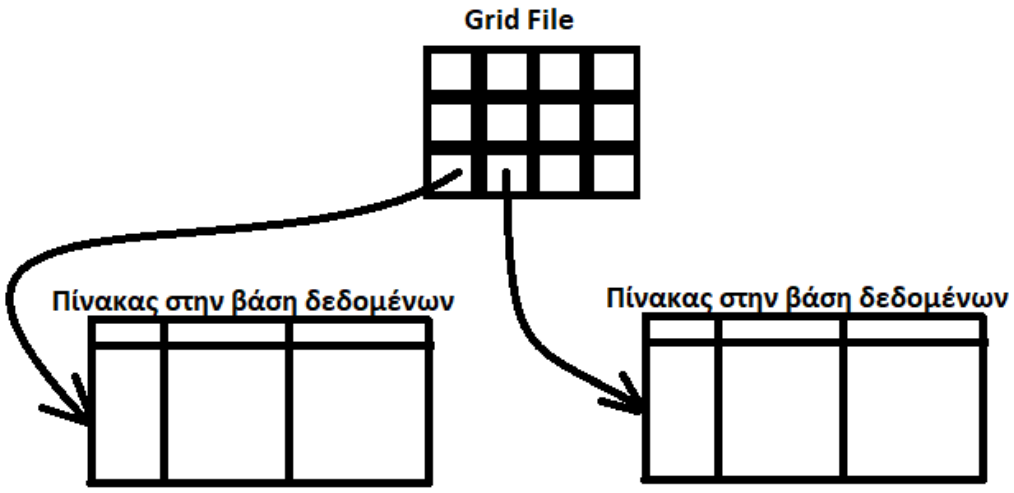
5.4 Μέθοδος Grid File

Η μέθοδος Grid File ανήκει στην κατηγορία μεθόδων πρόσβασης σημείου (Point Access Method) και χρησιμοποιείται αρκετά συχνά για τον διαχωρισμό και την πρόσβαση σε χωρικά δεδομένα [13]. Η μέθοδος Grid File, διαχωρίζει τα δεδομένα σε ένα πλέγμα, το οποίο αποτελείται από κελιά. Εσωτερικά αυτών των κελιών αποθηκεύονται εγγραφές βάσει των γεωγραφικών συντεταγμένων. Αυτό έχει ως σκοπό κάθε κελί να αναφέρεται σε ένα μικρό σύνολο σημείων που αντιστοιχεί σε έναν πίνακα στην βάση δεδομένων.

Επιπρόσθετα, το μέγεθος του πλέγματος καθορίζεται από το πλήθος των γεωγραφικών συντεταγμένων των δεδομένων και δεν πρόκειται για μια σταθερή ή

αυθαίρετη τιμή. Επιπλέον, η μέθοδος του Grid File δίνει την δυνατότητα η αναζήτηση να γίνεται αρκετά αποτελεσματικά και αποδοτικά.

Γενικότερα, η μέθοδος Grid File αποτελείται, συνήθως, από ένα δισδιάστατο πίνακα. Εσωτερικά του πίνακα, κάθε κελί-θέση αντιπροσωπεύει μία πλειάδα εγγραφών που βρίσκονται αποθηκευμένες σε πίνακες της βάσης δεδομένων.



Εικόνα 5. Εικονική αναπαράσταση Grid File

5.4.1 Τρόπος λειτουργίας του Grid File

Όσον αφορά τα δεδομένα που έχουμε στην διάθεση μας, αρχικά θα καθοριστεί το μέγεθος του πλέγματος βάσει του εύρους των γεωγραφικών συντεταγμένων (0-180 latitude, 0-360 longitude). Στην συνέχεια θα δημιουργηθεί ένας διαφορετικός πίνακας για κάθε κελί του πλέγματος. Ουσιαστικά κάθε κελί ομαδοποιεί τα δεδομένα βάσει των συντεταγμένων κάθε εγγραφής, με σκοπό την γρήγορη και αποτελεσματική αναζήτηση.

Κατά την διαδικασία της αναζήτησης, ο χρήστης θα θέτει ένα ερώτημα προς την βάση. Εν συνεχεία γίνεται ανεύρεση των κελιών του πλέγματος με τα πιθανά αποτελέσματα και στην συνέχεια πραγματοποιείται έλεγχος και σύγκριση των εγγραφών στους πίνακες, της βάσης δεδομένων, που υποδεικνύει κάθε κελί, του Grid File, σε σχέση με το ερώτημα του χρήστη.

Όπως αναφέρθηκε και στην ενότητα 5.4, που αφορά την μέθοδο του Grid File, κάθε κελί του πλέγματος αντιπροσωπεύει μια πλειάδα εγγραφών, που βρίσκονται αποθηκευμένες σε πίνακες στην βάση δεδομένων. Στην παρούσα εργασία, κάθε κελί χαρακτηρίζεται από ένα γεωγραφικό εύρος τιμών, όπου, αντιπροσωπεύει μία πλειάδα εγγραφών, που ανήκουν σε αυτό το εύρος και είναι αποθηκευμένες σε έναν πίνακα στην βάση. Όσον αφορά την αναζήτηση, αρχικά ελέγχουμε τις τιμές εισόδου του

χρήστη, σε ποιο γεωγραφικό εύρος ανήκουν στο πλέγμα και έπειτα, η αναζήτηση, συνεχίζει στους πίνακες που μας υποδεικνύει το κελί του πλέγματος.

6

Ανάκτηση δεδομένων με χρήση ερωτημάτων

Σε αυτό το κεφάλαιο θα περιγραφεί η ανάκτηση των δεδομένων, μέσω των οποίων διευκολύνεται η εξαγωγή αποτελεσμάτων. Πιο συγκεκριμένα θα προσδιοριστεί ο ορισμός των ερωτημάτων. Επιπλέον, θα δοθεί η σημασία και η χρησιμότητα των χωρικών ερωτημάτων καθώς επίσης και οι τύποι αυτών, που προτείνονται και χρησιμοποιήθηκαν. Επιπρόσθετα, θα παρουσιαστεί η μέθοδος που χρησιμοποιήθηκε στην παρούσα εργασία.

6.1 Ανάκτηση χωρικών δεδομένων

Με τον όρο ανάκτηση δεδομένων χαρακτηρίζεται η διαδικασία κατά την οποία μπορούν να αποκτηθούν δεδομένα από κάποιο σύστημα διαχείρισης βάσεων δεδομένων, σε μία βάση δεδομένων.

Η ανάκτηση των δεδομένων είναι πολύ σημαντική, σε ένα σύστημα διαχείρισης βάσεων δεδομένων, διότι βάσει αυτής εξάγονται αποτελέσματα και συμπεράσματα που βασίζονται στα δεδομένα. Χρησιμοποιούνται κυρίως για την αναπαράσταση των δεδομένων έτσι ώστε το αποτέλεσμα αυτών να επιστρέφει κάποια πληροφορία. Επιπλέον, χρησιμοποιούνται για την εξόρυξη συγκεκριμένων δεδομένων, που επιθυμούμε, από μία βάση δεδομένων.

Επιπρόσθετα, σε αυτή την διαδικασία μπορούν να εφαρμοστούν τεχνικές φιλτραρίσματος ώστε να επιστραφεί το κατάλληλο αποτέλεσμα για τον εκάστοτε σκοπό. Σημαντικό είναι ότι μέσω αυτής διαδικασίας τα δεδομένα μπορούν να ομαδοποιηθούν ώστε να παρέχεται η μέγιστη δυνατή οργανωμένη πληροφορία στο τελικό αποτέλεσμα.

Όπως γίνεται αντιληπτό, η ανάκτηση των δεδομένων είναι μία ιδιαίτερα σημαντική διαδικασία για την εξαγωγή συμπερασμάτων. Αυτό συμβαίνει διότι δίνεται η δυνατότητα εξαγωγής συμπερασμάτων, στατιστικών στοιχείων και δεδομένων για τα δεδομένα που περιέχονται σε μία βάση.

Οι παραπάνω διαδικασίες και γενικότερα η ανάκτηση των δεδομένων επιτυγχάνονται με την βοήθεια των ερωτημάτων. Τα ερωτήματα διαχωρίζονται σε διάφορες κατηγορίες ανάλογα με τον σκοπό που εξυπηρετούν και το είδος των δεδομένων.

6.2 Ερωτήματα βάσεων δεδομένων

Με τον όρο ερώτημα (query) αναφερόμαστε στην επιλογή κατάλληλων κριτηρίων ώστε να επιστραφούν τα αποτελέσματα με την μέγιστη δυνατή ακρίβεια στις απαιτήσεις του χρήστη. Τα ερωτήματα ικανοποιούν τις παρακάτω ιδιότητες ενός συστήματος διαχείρισης βάσεων δεδομένων:

- πρόσβαση στα δεδομένα
- προσπέλαση των δεδομένων
- ανάκτησης των δεδομένων
- προβολής των δεδομένων
- κριτήρια επιλογής δεδομένων και
- συγκεκριμενοποίησης των δεδομένων προς προβολή

Για τους προαναφερθέντες λόγους, γίνεται κατανοητό ότι η χρήση των ερωτημάτων είναι αρκετά σημαντικοί και υψίστης σημασίας για τις βάσεις δεδομένων και κατ΄ επέκταση των χρηστών και διαχειριστών αυτών.

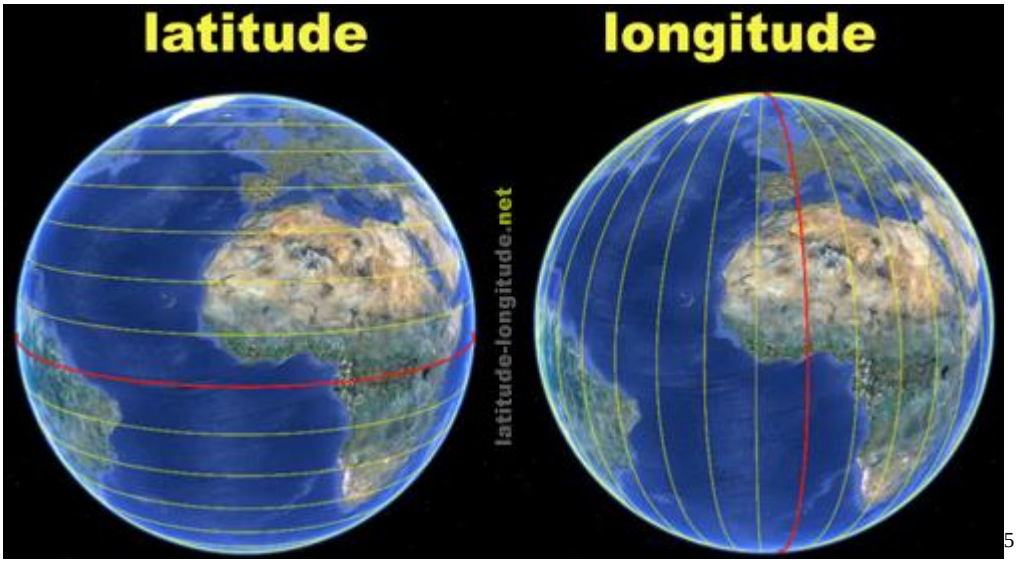
6.3 Χωρικά ερωτήματα

Τα χωρικά ερωτήματα αφορούν την κατηγορία των ερωτημάτων που ασχολείται με χωρικά δεδομένα. Στην παρούσα εργασία δίνεται ιδιαίτερη έμφαση σε αυτά μιας και το αντικείμενο της εργασίας κρίνει ουσιαστική την χρήση τους.

Τα χωρικά ερωτήματα χρησιμοποιούνται τόσο για την εύρεση γεωγραφικών περιοχών όσο και στην υπολογιστική όραση, για την διόρθωση και βελτίωση των εικονοστοιχείων (pixel) μιας εικόνας. Η πιο διαδεδομένη χρήση τους είναι σε συστήματα γεωγραφικών πληροφοριών. Επιπλέον είναι υπεύθυνα για την μέθοδο συγγραφής των ερωτημάτων που θέτονται σε μία βάση χωρικών δεδομένων [14].

Σκοπός τους είναι να προσανατολίσουν ορθά την συγγραφή των ερωτημάτων με στόχο την ανάκτηση δεδομένων από μία βάση, σύμφωνα με το ερωτήμα που τίθεται. Πιο συγκεκριμένα θέτει ένα ερώτημα σε μία βάση δεδομένων με απώτερο σκοπό να επιστρέψει τις εγγραφές που βρίσκονται πλησιέστερα ενός σημείου αναφοράς ή να επιστρέψει αποτελέσματα σχετικά με το ερώτημα, ανεξάρτητα της απόστασης.

Επιπρόσθετα, τα χωρικά ερωτήματα στην παρούσα εργασία βασίζονται σε δύο παραμέτρους προσδιορισμού και μιας γεωγραφικής περιοχής. Αυτές οι παράμετροι είναι το γεωγραφικό μήκος (Latitude) και γεωγραφικό πλάτος (Longitude) μιας τοποθεσίας. Επίσης, αν επιθυμούμε τα αποτελέσματα του ερωτήματος να είναι ακριβής, τότε πρέπει να δοθεί ιδιαίτερη σημασία και στην ακρίβεια των τιμών Latitude και Longitude.



[5]

Εικόνα 6. Γεωγραφικό πλάτος και μήκος

Επιπλέον, υπάρχουν τύποι ερωτημάτων που αναζητούν πλησιέστερα αντικείμενα, βάσει εύρους, από ένα σημείο αναφοράς που επιθυμεί ο χρήστης. Στις παρακάτω υποενότητες θα αναφερθούν κάποιοι τύποι ερωτημάτων χωρικών δεδομένων.

6.3.1 Ερώτημα σημείου (Point Query)

Τα ερωτήματα σημείου δίνουν την δυνατότητα να ανακτηθούν χαρακτηριστικά των δεδομένων μιας βάσης, για μια τοποθεσία [15]. Πιο συγκεκριμένα εφαρμόζεται για δεδομένα που έχουμε στην διάθεση μας και επιθυμούμε να εμφανίσουμε κάποια χαρακτηριστικά που αφορούν αυτό.

Για παράδειγμα, έστω ότι έχουμε ένα σύνολο εγγραφών, που περιέχουν τα χαρακτηριστικά κάποιων καταστημάτων εστίασης. Τέτοια χαρακτηριστικά θα μπορούσε να είναι οι γεωγραφικές συντεταγμένες και η ονομασία του καταστήματος. Αν θέλαμε να εφαρμόσουμε ένα ερώτημα σημείου (Point Query) στα δεδομένα αυτά τότε θα έπρεπε να αναζητήσουμε βάσει των συντεταγμένων αυτών των δεδομένων. Δηλαδή, μία αντιπροσωπευτική μορφή ενός ερωτήματος σημείου θα ήταν: «Εμφάνισε το όνομα του καταστήματος που έχει γεωγραφικές συντεταγμένες Latitude=5 και Longitude=8».

Βάσει των παραπάνω γίνεται αντιληπτό, ότι τα ερωτήματα σημείου (Point Query) εφαρμόζονται και τίθενται όταν υπάρχουν ακριβείς τιμές των συντεταγμένων με απώτερο σκοπό την επιστροφή κάποιων πληροφοριών για το συγκεκριμένο σημείο.

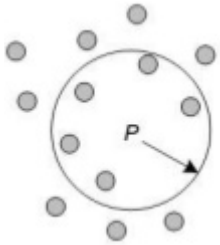
[5] https://www.latitude-longitude.net/latitude-longitude-lines.jpg

6.3.2 Ερώτημα εύρους (Range Query)

Τα ερωτήματα εύρους ανήκουν και αυτά στον τύπο των χωρικών ερωτημάτων. Εφαρμόζονται και χρησιμοποιούνται κυρίως σε χωρικά δεδομένα με σκοπό την εύρεση γεωγραφικών τοποθεσιών που ανήκουν εντός μιας ακτίνας δοσμένης από ένα σημείο αναφοράς του χρήστη. Το μήκος της ακτίνας καθορίζεται από τον χρήστη και ξεκινά από το σημείο αναφοράς του χρήστη και εκτείνεται μέχρι την απόσταση που επιθυμεί ο ίδιος.

Η εύρυθμη λειτουργία, η αποτελεσματικότητα και η αποδοτικότητα αυτών των ερωτημάτων εξαρτάται από την αποθήκευση και την ανάλυση που έχει προηγηθεί στα δεδομένα [16]. Βάσει του παραπάνω χαρακτηριστικού, είναι σημαντικό τα δεδομένα να αναλύονται και να αποθηκεύονται ταξινομημένα έτσι ώστε η αναζήτηση να απαιτεί λιγότερο χρόνο και κατ΄ επέκταση να γίνεται πιο αποδοτική.

Για παράδειγμα, έστω ότι έχουμε ένα σύνολο εγγραφών, που περιέχουν τα χαρακτηριστικά κάποιων καταστημάτων εστίασης. Τέτοια χαρακτηριστικά θα μπορούσε να είναι οι γεωγραφικές συντεταγμένες και η ονομασία του καταστήματος. Αν θέλαμε να εφαρμόσουμε ένα ερώτημα εύρους (Range Query) στα δεδομένα αυτά, τότε θα έπρεπε να αναζητήσουμε βάσει μιας ευρύτερης περιοχής από ένα σημείο αναφοράς. Δηλαδή, μία αντιπροσωπευτική μορφή ενός ερωτήματος εύρους θα ήταν: «Εμφάνισε τα καταστήματα που βρίσκονται σε ακτίνα 5 χιλιομέτρων από το σημείο με γεωγραφικές συντεταγμένες Latitude=5 και Longitude=8». Επισημαίνεται ότι οι συντεταγμένες Latitude και Longitude αφορούν το σημείο αναφοράς του χρήστη. Επιπρόσθετα, παρακάτω δίνεται μία γραφική απεικόνιση του Range Query, όπου P είναι το σημείο αναφοράς του χρήστη, το βέλος αντιπροσωπεύει την ακτίνα και οι κουκίδες τα καταστήματα εστίασης.



[6]

Εικόνα 7. Γραφική αναπαράσταση Range Query

Όπως αντιλαμβανόμαστε και από την γραφική αναπαράσταση τα ερωτήματα εύρους δημιουργούν ένα εικονικό κυκλικό σχήμα, με την βοήθεια της ακτίνας και εμφανίζει τα στοιχεία που υπάρχουν εσωτερικά αυτού του κύκλου.

[6] https://image.slidesharecdn.com/spatialdatabases-180515183153/95/spatial-databases-29-638.jpg?cb=1526409592

6.3.3 Ερώτημα παραθύρου (Window Query)

Τα ερωτήματα παραθύρου υιοθετούν την λογική και την μέθοδο των ερωτημάτων εύρους (Range Query). Η κύρια διαφορά τους είναι ότι δεν λειτουργούν με ένα εικονικό κυκλικό σχήμα βάσει μιας ακτίνας δοσμένη από τον χρήστη αλλά δημιουργούν ένα σχήμα τετραγώνου.

Για παράδειγμα, έστω ότι έχουμε ένα σύνολο εγγραφών, που περιέχουν τα χαρακτηριστικά κάποιων καταστημάτων εστίασης. Τέτοια χαρακτηριστικά θα μπορούσε να είναι οι γεωγραφικές συντεταγμένες και η ονομασία του καταστήματος. Αν θέλαμε να εφαρμόσουμε ένα ερώτημα παραθύρου (Window Query) στα δεδομένα αυτά τότε θα έπρεπε να αναζητήσουμε βάσει μιας ευρύτερης περιοχής από ένα σημείο αναφοράς. Δηλαδή, μία αντιπροσωπευτική μορφή ενός ερωτήματος εύρους θα ήταν: «Εμφάνισε τα καταστήματα που βρίσκονται σε απόσταση 5 χιλιομέτρων από το σημείο με γεωγραφικές συντεταγμένες Latitude=5 και Longitude=8». Επισημαίνεται ότι οι συντεταγμένες Latitude και Longitude αφορούν το σημείο αναφοράς του χρήστη.

Επιπρόσθετα, παρακάτω δίνεται μία γραφική απεικόνιση του Window Query, όπου η κόκκινη κουκίδα αναπαριστά το σημείο αναφοράς του χρήστη, τα βέλη αντιπροσωπεύει την απόσταση και οι κουκίδες τα καταστήματα εστίασης.

Εικόνα 8. ΑναπαράστασηWindow Query

6.3.4 Ερώτημα εύρεσης πλησιέστερου γείτονα (Nearest Neighbor Query)

Τα ερωτήματα εύρεσης πλησιέστερων γειτόνων είναι ένας τρόπος αναζήτησης πλησιέστερων κόμβων και θα μπορούσαν να θεωρηθούν και ως ερωτήματα γεωγραφικής προσέγγισης. Η εύρεση πλησιέστερων κόμβων είναι ένα συνηθισμένο και αρκετά χρήσιμο ερώτημα που τίθεται σε χωρικά δεδομένα. Επιπλέον, τα ερωτήματα αυτά βρίσκουν εφαρμογή σε δεδομένα που είναι ήδη ταξινομημένα με την χρήση αντίστοιχων μεθόδων (όπως τα δέντρα αναζήτησης).

Επομένως, η εφαρμογή, τέτοιων ερωτημάτων, απαιτεί τα δεδομένα να είναι ταξινομημένα και να δίνεται ιδιαίτερη έμφαση στην επιλογή της μεθόδου ταξινόμησης, έτσι ώστε τα δεδομένα να είναι συνδεδεμένα και ομαδοποιημένα μεταξύ τους, με προϋπολογισμένες αποστάσεις [17], προκειμένου τα αποτελέσματα του ερωτήματος να είναι αποτελεσματικά.

Γενικότερα, τα ερωτήματα εύρεσης πλησιέστερων γειτόνων, εφαρμόζονται κυρίως σε χωρικά δεδομένα και τίθενται στην βάση, με σκοπό την εύρεση πλησιέστερων γειτονικών κόμβων από ένα σημείο αναφοράς, χρησιμοποιώντας δενδρικές δομές δεδομένων (όπως B-trees, B+ trees, κ.α.) [18].

Για παράδειγμα, με δεδομένα που έχουμε στην διάθεσή μας, αν θέταμε ένα ερώτημα εύρεσης πλησιέστερων γειτόνων στην βάση από ένα συγκεκριμένο σημείο αναφοράς, θα πρέπει να μας επιστραφούν k καταστήματα που βρίσκονται πλησιέστερα του δοθέντος σημείο, δίχως το ¨πλησιέστερα¨ να καθορίζεται από τον χρήστη.

6.3.5 Ερώτημα χωρικής συνδεσιμότητας (Spatial Join Query)

Τα ερωτήματα χωρικής συνδεσιμότητα, εκτελούνται για την εύρεση περιοχών που συνδέονται άμεσα με μία συγκεκριμένη τοποθεσία.

Για παράδειγμα, αν εφαρμοστεί ερώτημα χωρικής συνδεσιμότητας στον χάρτη της Αυστραλίας, δίνοντας ως είσοδο την Βόρεια περιοχή (Northern Territory), θα επιστραφούν οι περιοχές που είναι άμεσα συνδεδεμένες με αυτήν. Αυτό σημαίνει ότι θα μας επέστρεφε τις εξής περιοχές:

- Δυτική Αυστραλία (Western Australia)
- Νότια Αυστραλία (North Australia)
- Κουίνσλαντ (Queensland)

Παρακάτω δίνεται η γραφική απεικόνιση του ερωτήματος, τονίζοντας με κόκκινο τις περιοχές που θα μας επέστρεφε το ερώτημα, δηλαδή τις περιοχές που είναι άμεσα συνδεδεμένες με την περιοχή του δοθέντος ερωτήματος.

[7]

Εικόνα 9. Απεικόνιση ερωτήματος χωρικής συνδεσιμότητας

6.4 Ερωτήματα που επιλέχθηκαν προς εφαρμογή

Ο τύπος χωρικού ερωτήματος που επιλέχθηκε για την παρούσα εργασία είναι το ερώτημα παραθύρου (Window Query). Ο κύριος λόγος επιλογής αυτού του τύπου ερωτήματος είναι επειδή δίνει την δυνατότητα στον χρήστη να επιλέγει ο ίδιος το γεωγραφικό εύρος που επιθυμεί. Επιπλέον, λόγω του εικονικού (τετραγώνου) σχήματος που χρησιμοποιεί, εξασφαλίζει την εμφάνιση περισσότερων επιλογών προς τον χρήστη, σε αντίθεση με τον τύπο ερωτημάτων εύρους (Range Query).

Επιπρόσθετα, το αντικείμενο της εργασίας επικεντρώνεται και στην αναζήτηση των δεδομένων με την χρήση όρων (keyword). Με την έννοια «όρος» αναφερόμαστε στις λέξεις-κλειδιά που χαρακτηρίζουν το είδος του κάθε καταστήματος. Για αυτό τον λόγο εφαρμόζεται άλλος ένα τύπος ερωτήματος που αφορά αποκλειστικά και μόνο τις επιλογές του χρήστη.

Ένας τύπος ερωτήματος που θα μπορούσε να χρησιμοποιηθεί, για την εύρεση των λέξεων κλειδιών είναι τα Top-k ερωτήματα. Τα Top-k ερωτήματα εφαρμόζονται, συνήθως, ως δευτερεύον αναζήτηση σε δεδομένα που έχουν ανακτηθεί ήδη από μία βάση δεδομένων. Η βασική ιδέα των Top-k ερωτημάτων είναι η δημιουργία μιας τιμής (Rank) για κάθε πιθανό αποτέλεσμα και του ερωτήματος που τίθεται στην βάση, με την χρήση μιας συνάρτησης F. Στην συνέχεια, ταξινομεί τα πιθανά αποτελέσματα βάσει αυτής της τιμής και επιστρέφει στον χρήστη τα πρώτα k αποτελέσματα των οποίων οι τιμές σχετίζονται άμεσα με το αρχικό ερώτημα του χρήστη [20]. Οι τύποι των Top-k ερωτημάτων είναι τα:

[7] https://www.australia.com/content/dam/australia/navigation/placestogo/megamenu_default.png

- **Top-k Selection:** Τα Top-k ερωτήματα επιλογής χρησιμοποιούνται για την επιστροφή αποτελεσμάτων που σχετίζονται περισσότερο με την βαθμολογία που έχει δημιουργηθεί για το ερώτημα. Πιο συγκεκριμένα επιστρέφει κάποια αποτελέσματα που σχετίζονται με το ερώτημα και στην συνέχεια τα ταξινομεί. Έπειτα εμφανίζει τα πρώτα k-στοιχεία της ταξινόμησης. Ένα παράδειγμα ενός Top-k Selection ερωτήματος είναι [19]:

  ΕΜΦΑΝΙΣΕ «Το όνομα των καταστημάτων»
  ΤΟΥ ΠΙΝΑΚΑ «Καταστήματα»
  ΌΠΟΥ «Βρίσκονται εντός ακτίνας 5 χιλιομέτρων»
  ΤΑΞΙΝΟΜΩΝΤΑΣ «Βάσει της βαθμολογίας της συνάρτησης F»
  ΜΕ ΟΡΙΟ «Δέκα καταστήματα»

  Όπου το όριο, αντιπροσωπεύει την τιμή k των επιστρεφόμενων αποτελεσμάτων.

- **Top-k Join Query:** Τα Top-k ερωτήματα συνδεσιμότητας, υιοθετούν τον ίδιο τρόπο εκτέλεσης με τα Top-k ερωτήματα επιλογής, με την διαφορά ότι έχουν την δυνατότητα εκτέλεσης σε περισσότερους από έναν πίνακες. Χάρις του τελευταίου χαρακτηριστικού ονομάζεται και ερώτημα συνδεσιμότητας, διότι συνδέει και εμφανίζει αποτελέσματα από πολλούς πίνακες. Ένα παράδειγμα ενός Top-k Join ερωτήματος είναι [19]:

  ΕΜΦΑΝΙΣΕ «Το όνομα των καταστημάτων»
  ΤΩΝ ΠΙΝΑΚΩΝ «Καταστήματα_1 και Καταστήματα_2»
  ΌΠΟΥ «Βρίσκονται εντός ακτίνας 5 χιλιομέτρων»
  ΤΑΞΙΝΟΜΩΝΤΑΣ «Βάσει της βαθμολογίας της συνάρτησης F»
  ΜΕ ΟΡΙΟ «Δέκα καταστήματα»

  Όπου το όριο, αντιπροσωπεύει την τιμή k των επιστρεφόμενων αποτελεσμάτων.

- **Top-k Aggregate Query:** Τα Top-k συγκεντρωτικά ερωτήματα υιοθετούν τον τρόπο εκτέλεσης των Top-k ερωτημάτων επιλογής και των Top-k ερωτημάτων συνδεσιμότητας, έχοντας μία επιπλέον δυνατότητα. Αυτή η δυνατότητα είναι να ομαδοποιεί και στην συνέχεια να ταξινομεί και να εμφανίζει τα πρώτα k αποτελέσματα. Ένα παράδειγμα ενός Top-k Aggregate ερωτήματος είναι [19]:

  ΕΜΦΑΝΙΣΕ «Το όνομα των καταστημάτων»
  ΤΩΝ ΠΙΝΑΚΩΝ «Καταστήματα_1 και Καταστήματα_2»
  ΌΠΟΥ «Βρίσκονται εντός ακτίνας 5 χιλιομέτρων»
  ΟΜΑΔΟΠΟΙΩΝΤΑΣ «Βάσει των προτιμήσεων του χρήστη»
  ΤΑΞΙΝΟΜΩΝΤΑΣ «Βάσει της βαθμολογίας της συνάρτησης F»
  ΜΕ ΟΡΙΟ «Δέκα καταστήματα»

Όπου το όριο, αντιπροσωπεύει την τιμή k των επιστρεφόμενων αποτελεσμάτων. Ο σκοπός των Top-k ερωτημάτων είναι να παράγει πιο ακριβή αποτελέσματα, προκειμένου να αποδίδεται στον χρήστη βεβαιότητα και αποτελεσματικότητα.

Ένας ακόμη τύπος ερωτημάτων κειμένου (keywords) είναι τα Boolean Queries. Τα ερωτήματα Boolean χρησιμοποιούν τους τελεστές OR, AND και NOT για την πραγματοποίηση της αναζήτησης και έχουν αρκετά απλή και κατανοητή δομή [21]. Επιπλέον, τα Boolean ερωτήματα έχουν σχετικά απλή υλοποίηση και θεωρούνται αρκετά αποτελεσματικά. Επιπρόσθετα, εξάγουν ακριβή και αξιόπιστα αποτελέσματα. Συνήθως εφαρμόζονται σε ήδη ανακτήσιμα πιθανά αποτελέσματα, με σκοπό να «φιλτράρουν» αυτά, αποδίδοντας αποτελέσματα που σχετίζονται άμεσα με το ερώτημα που τίθεται σε μία βάση δεδομένων.

Στην παρούσα εργασία εφαρμόζεται ο συνδυασμός των ερωτημάτων Window Query και Boolean Query. Το Window Query είναι υπεύθυνο για την επιστροφή αποτελεσμάτων που σχετίζονται με τις γεωγραφικές συντεταγμένες που θέτει ο χρήστης και το Boolean Query με τις προτιμήσεις του χρήστη.

Με τον παραπάνω συνδυασμό των δύο ερωτημάτων, επιτυγχάνεται η αποτελεσματικότητα και η ακρίβεια των εξαγόμενων δεδομένων από την βάση.

Προκειμένου να γίνει κατανοητή η χρήση των προαναφερθέντων ερωτημάτων, ακολουθεί ένα παράδειγμα:

Έστω ότι διαθέτουμε μία βάση δεδομένων που εμπεριέχει τα χαρακτηριστικά κάποιων καταστημάτων με φυσική παρουσία. Τέτοια χαρακτηριστικά είναι η ονομασία του καταστήματος, η γεωγραφική θέση του καταστήματος και τα προϊόντα που διαθέτει προς πώληση κάθε κατάστημα. Αν ο χρήστης θέσει το εξής ερώτημα στην βάση δεδομένων: «*Εμφάνισε τα ονόματα των καταστημάτων που βρίσκονται εντός ακτίνας 5 χιλιομέτρων (από το σημείο που βρίσκεται ο χρήστης) που διαθέτουν τα προϊόντα προς πώληση 'Pizza', 'Coffee'*» τότε το Window Query θα επιστρέψει όλα τα καταστήματα που βρίσκονται εντός ακτίνας 5 χιλιομέτρων από σημείο αναφοράς του χρήστη, ως πιθανά αποτελέσματα. Στην συνέχεια θα πραγματοποιηθεί η εφαρμογή του Boolean Query στα πιθανά αποτελέσματα, με σκοπό να ελέγξει ποια από τα καταστήματα, που επέστεψε το Window Query, προσφέρουν τα προϊόντα 'Pizza' ή 'Coffee' ή 'Pizza' και 'Coffee'. Με την ολοκλήρωση και του Boolean Query, επιστρέφονται στον χρήστη όλα τα καταστήματα που βρίσκονται εντός ακτίνας 5 χιλιομέτρων, από τον ίδιο, τα οποία προσφέρουν τα προϊόντα που επιθυμεί ο ίδιος.

7

Περιγραφή υλοποίησης

Σε αυτό το κεφάλαιο θα περιγραφεί λεπτομερώς η υλοποίηση και των δύο μεθόδων που υιοθετήθηκαν, για την αποτελεσματική και αποδοτική αποθήκευση των δεδομένων, στην παρούσα εργασία καθώς επίσης και οι παραμετροποιήσεις που πραγματοποιήθηκαν για την δημιουργία του ερωτήματος με σκοπό την αποτελεσματική ανάκτηση των δεδομένων. Επιπλέον, θα γίνει αναφορά στα δεδομένα που έχουμε στην διάθεσή μας.

7.1 Περιγραφή πειραματικών δεδομένων

Τα δεδομένα που έχουμε στην διάθεσή μας αφορούν τα χαρακτηριστικά καταστημάτων εστίασης με φυσική παρουσία και περιέχονται σε ένα συγκεντρωτικό αρχείο κειμένου. Τα χαρακτηριστικά αυτών των καταστημάτων είναι:

- η ονομασία (Brand name)
- η βαθμολογία (Rating)
- το ακριβές γεωγραφικό πλάτος (Latitude)
- το ακριβές γεωγραφικό μήκος (Longitude) και
- τα προϊόντα προς πώληση (Keywords)

Τα δεδομένα προέρχονται από την ιστοσελίδα: https://www.factual.com/

Τα παραπάνω χαρακτηριστικά, αναφέρθηκαν με την σειρά που παρουσιάζονται και στο αρχείο. Η μορφή απεικόνισης αυτών των καταστημάτων είναι η παρακάτω:

```
Hinano Café|4|33.979293|-118.466451|Burgers, Pub Food, Sandwiches
Palms|4|34.397338|-119.520776|Californian, Seafood, American, Chicken, Steak
Fuzhou Super Buffet|4|38.09562|-122.562614|Chinese, Buffet
Chiaramonte's Deli and Sausages|4|37.353099|-121.886424|Deli, Catering, Sandwiches
Philippe The Original|4.5|34.059767|-118.23675|Sandwiches, American, Salad, Deli, French, Soup
Barstow Station Liquor|3.5|34.891606|-117.000213|Fast Food
Luigi's|4.5|35.374537|-118.993195|Italian, Deli, American
McDonald's|1.5|37.781337|-122.420093|American, Burgers, Fast Food
Swan Oyster Depot|4.5|37.79072|-122.42071|Seafood, American, Traditional
Clearman's Galley|4|34.128218|-118.073139|American, Seafood
Balboa Cafe|4|37.798894|-122.43589|American, Burgers, Californian, Mediterranean
Sam's for Play Cafe and Catering On Cleveland Ave|4.5|38.462831|-122.728381|Catering, Diner
Farmer Boys Restaurant|3.5|33.832059|-117.985020875931|American, Burgers, Fast Food
The Farm Of Beverly Hills|4|34.072022|-118.35754|American, Californian, Coffee, Sandwiches, Tea
```

Εικόνα 10. Απεικόνιση των δεδομένων που έχουμε στην διάθεσή μας

7.2 Υλοποίηση Grid File

Σε αυτή την ενότητα θα περιγραφεί η υλοποίηση του Grid File καθώς επίσης και οι διαδικασίες που θα πρέπει αν επιτευχθούν για την άρτια και ομαλή λειτουργία του.

7.2.1 Ανάκτηση δεδομένων από το αρχικό αρχείο

Η μέθοδος του Grid File υλοποιήθηκε με την χρήση της αντικειμενοστραφούς γλώσσα προγραμματισμού Java. Ο λόγος που επιλέχθηκε αυτή η γλώσσα προγραμματισμού είναι κυρίως η συμβατότητα που παρουσιάζει με την HBase, κάτι το οποίο κάνει την δημιουργία των πινάκων και την εισαγωγή των δεδομένων στην βάση, μία εύκολη διαδικασία. Επιπλέον, υπενθυμίζεται ότι το Grid File είναι ένας πίνακας, του οποίου κάθε κελί αντιπροσωπεύεται από γεωγραφικά (Latitude και Longitude) όρια και υποδεικνύει τον πίνακα στην βάση δεδομένων που περιέχει τα στοιχεία που ανήκουν σε αυτό το όριο.

Αρχικά, με την χρήση ενός *Buffer Reader* πραγματοποιείται η ανάγνωση του αρχείου. Η ανάγνωση γίνεται σειριακά και ξεκινά από την πρώτη εγγραφή του αρχείου, που περιέχει τα καταστήματα, μέχρι την τελευταία.

Όπως έχει αναφερθεί, ήδη, κάθε στοιχείο απαρτίζεται από πέντε (5) χαρακτηριστικά (όνομα, βαθμολογία, γεωγραφικό πλάτος, γεωγραφικό μήκος και τα προϊόντα προς πώληση) τα οποία θα πρέπει να εκχωρηθούν σε μία μεταβλητή που θα αντιπροσωπεύει κάθε εγγραφή του αρχείου. Για να πραγματοποιηθεί αυτό δημιουργήθηκε μία κλάση, της οποίας τα αντικείμενα περιέχουν όλα τα χαρακτηριστικά των στοιχείων. Ουσιαστικά, κάθε αντικείμενο αυτής της κλάσης περιέχει τα χαρακτηριστικά κάθε καταστήματος και αντιπροσωπεύει μία και μόνο μία εγγραφή του αρχικού αρχείου που έχουμε στην διάθεσή μας.

Κατά την δημιουργία των αντικειμένων, πραγματοποιείται και η δημιουργία της τιμής Z-order των συντεταγμένων κάθε καταστήματος, της οποίας η διαδικασία έχει περιγραφεί σε προηγούμενη ενότητα (Βλέπε ενότητα 4.5) και έχει ως σκοπό την δημιουργία ενός μοναδικού αναγνωριστικού για κάθε εγγραφή. Επιπλέον, για κάθε εγγραφή δημιουργείται άλλη μία τιμή, πάλι με την βοήθεια των συντεταγμένων, η οποία στην συνέχεια συμβάλλει στην διαδικασία της ομαδοποίησης των εγγραφών. Αυτή την τιμή την ονομάζεται Hash, γιατί όπως θα περιγραφεί και παρακάτω βοηθά στην δημιουργία ενός κατακερματισμού των εγγραφών.

Επομένως, συνοπτικά, κάθε αντικείμενο της κλάσης περιέχει την μοναδική τιμή Z-Order, την τιμή Hash, το όνομα, την βαθμολογία, τις συντεταγμένες και τα προϊόντα προς πώληση κάθε καταστήματος.

7.2.2 Δημιουργία τιμής Hash

Η τιμής Hash είναι μία αρκετά χρήσιμη τιμή για την υλοποίηση του Grid File. Σκοπός της Hash είναι να υπολογίσει για κάθε εγγραφή μία τιμή, με την οποία θα προσδιορίζεται ο πίνακας στον οποίο θα αποθηκευτεί στην HBase, η εγγραφή. Η δημιουργία αυτή της τιμής είναι αρκετά εύκολη και εφικτή.

Για την δημιουργία, της Hash, ανακτούμε τις τιμές των συντεταγμένων κάθε εγγραφής. Στην συνέχεια κάθε τιμή Latitude την αυξάνουμε κατά 90 και κάθε τιμή Longitude κατά 180, προκειμένου να γίνει η απαλοιφή του αρνητικού προσήμου έτσι ώστε οι τιμές Latitude να εκτείνονται από 0 (μηδέν) έως το 180 και το Longitude από 0 (μηδέν) έως 360. Έπειτα της πρόσθεσης, διαιρούνται ξεχωριστά οι τιμές που προκύπτουν με το δέκα (10) και διατηρείται μόνο το ακέραιο μέρος. Στην συνέχεια πραγματοποιείται η σύνδεση των δύο τιμών, ως αλφαριθμητικά, και προκύπτει η τιμή Hash. Επισημαίνεται ότι στην παρούσα εργασία, κρίθηκε η τιμή του δέκα (10) ως η κατάλληλη τιμή που θα χρησιμοποιηθεί για την δημιουργία της τιμής Hash. Η τιμή αυτή καθορίζεται από τον προγραμματιστή και μπορεί να είναι οποιαδήποτε αυτός επιθυμεί, με απώτερο σκοπό την αύξηση της αποδοτικότητας και της αποτελεσματικότητας της μεθόδου.

Για την καλύτερη κατανόηση αυτής δίνεται ένα παράδειγμα δημιουργίας Hash. Έστω ότι έχουμε τις συντεταγμένες:

Latitude = 37.755557

Longitude = -122.250360846519

Αρχικά προστίθεται σε αυτές οι τιμές 90 και 180 στο γεωγραφικό πλάτος (Latitude) και γεωγραφικό μήκος (Longitude) αντίστοιχα.

Latitude = 37.755557 **+ 90** = 127,755557

Longitude = -122.250360846519 **+ 180** = 58,250360846519

Στην συνέχεια, οι παραπάνω τιμές, διαιρούνται με το 10:

Latitude = 127,755557 **/ 10** = 12,7755557

Longitude = 58,250360846519 **/ 10** = 5,8250360846519

Από αυτές τις τιμές συγκρατείται μόνο το ακέραιο μέρος αυτών, δηλαδή:

Latitude = |12,755557| = 12

Longitude = |5,8250360846519| = 5

Στην συνέχεια πραγματοποιείται η ένωση των τιμών αυτών σε μία, η οποία αντιπροσωπεύει την τιμή Hash:

Hash = 125

Επομένως η τιμή Hash των συντεταγμένων Latitude = 37.755557 και Longitude = -122.250360846519 είναι 125.

7.2.3 Διαγραφή προηγούμενων πινάκων

Στην παρούσα εργασία η δημιουργία και οι ονομασίες των πινάκων βασίζονται στην τιμή Hash κάθε αντικειμένου. Επομένως, όλοι οι πίνακες που θα δημιουργηθούν θα έχουν ως ονομασία τις τιμές Hash.

Επιπλέον, πριν δημιουργηθούν οι πίνακες και εκχωρηθούν τα δεδομένα σε αυτούς πραγματοποιείται έλεγχος, για το αν υπάρχουν ήδη πίνακες με ονομασία κάποιας τιμής Hash.

Με αυτό τον τρόπο αποφεύγεται η δημιουργία πινάκων με την ίδια ονομασία και απομακρύνεται ο κίνδυνος κωλυμάτων τα οποία θα προέκυπταν κατά την εκτέλεση του προγράμματος.

Για να πραγματοποιηθεί η παραπάνω διαδικασία, πρώτα δημιουργείται μία σύνδεση στην βάση της HBase και στην συνέχεια εξετάζονται όλοι οι πίνακες που βρίσκονται στην HBase με σκοπό τον έλεγχο ύπαρξης πινάκων με ονομασία κάποιου αριθμού. Στην περίπτωση που ισχύει ο προαναφερθέν ισχυρισμός, τότε πραγματοποιείται η διαγραφή των συγκεκριμένων πινάκων.

7.2.4 Δημιουργία πινάκων

Όσον αφορά την υλοποίηση του Grid File, η δημιουργία των πινάκων καθώς επίσης και η εισαγωγή των δεδομένων σε αυτούς είναι οι διαδικασίες, οι οποίες εάν δεν διαμορφωθούν και δεν υλοποιηθούν σωστά θα προκύψουν σημαντικά προβλήματα που θα επηρεάσουν τα αποτελέσματα των μεταγενέστερων ερωτημάτων. Για αυτό τον λόγο, αφού έχει πραγματοποιηθεί ο έλεγχος ύπαρξης πινάκων με ονομασία αριθμών και η διαγραφή αυτών τότε προχωράμε στην διαδικασία της δημιουργία των πινάκων.

Επισημαίνεται, ότι μία τιμή Hash μπορεί να ανατίθεται σε περισσότερα από ένα ζεύγη συντεταγμένων. Επομένως, πριν την δημιουργία των πινάκων διεξάγεται νέος έλεγχος για την ύπαρξη ή μη πίνακα με ονομασία μιας συγκεκριμένης τιμής Hash.

Στην περίπτωση που δεν υπάρχει πίνακας με ονομασία ενός συγκεκριμένου Hash, που έχει προκύψει για μια εγγραφή, τότε αυτεπάγγελτα δημιουργείται ένας νέος πίνακας με ονομασία αυτή την τιμή, στην HBase. Επιπλέον, για την δημιουργία ενός πίνακα στην HBase θα πρέπει πρώτα να πραγματοποιηθεί σύνδεση με την βάση.

Αντίθετα, στην περίπτωση που υπάρχει ήδη ένας πίνακας με ονομασία μιας τιμής Hash τότε προχωράμε στην διαδικασία εισαγωγής των στοιχείων στον πίνακα.

Και στις δύο περιπτώσεις οι πίνακες θα πρέπει να διαθέτουν οικογένειες στηλών (Column Family), έτσι ώστε τα δεδομένα να σχετίζονται μεταξύ τους, καθώς επίσης και να επιτυγχάνεται η διαδικασία της αναζήτησης.

7.2.5 Δημιουργία Column Family των πινάκων

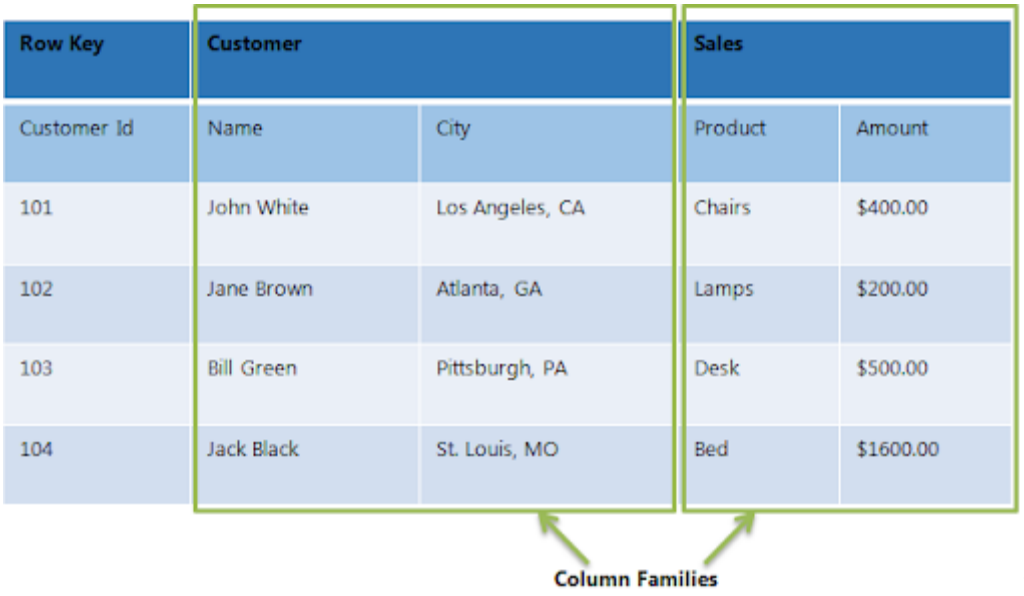
Εφόσον, αναφερόμαστε σε πίνακες της HBase, η δημιουργία των Column Family είναι μία αρκετά σημαντική διαδικασία. Χρησιμοποιείται κυρίως για την ομαδοποίηση των δεδομένων και συμβάλει κατά πολύ στην αναζήτηση και την ανάκτηση των δεδομένων.

Στην παρούσα εργασία έχουν δημιουργηθεί τρεις (3) Column Family για την ομαδοποίηση των χαρακτηριστικών των καταστημάτων. Πιο αναλυτικά τα χαρακτηριστικά αυτά, έχουν κατανεμηθεί στις Column Family ως εξής:

Column Family	**Χαρακτηριστικά**
shopDescription	Όνομα καταστήματος
	Βαθμολογία
shopLocation	Γεωγραφικό πλάτος
	Γεωγραφικό μήκος
keywords	Προϊόντα προς πώληση

Πίνακας 2. Κατανομή χαρακτηριστικών στις Column Family

Παρακάτω δίνεται ένα παράδειγμα απεικόνιση μιας Column Family, για την καλύτερη κατανόηση της χρήσης της.



[8]

Εικόνα 11. Αναπαράσταση Column Family της HBase

7.2.6 Εισαγωγή στοιχείων στους πίνακες

Αφού πραγματοποιηθούν οι απαραίτητοι έλεγχοι των πινάκων, που έχουν αναφερθεί και δημιουργηθούν οι Column Family των πινάκων στην συνέχεια γίνεται η εισαγωγή των στοιχείων στους πίνακες της βάσης, δηλαδή στην HBase.

Για την διεξαγωγή της διαδικασίας της εισαγωγής ενός στοιχείου σε ένα πίνακα, χρειάζεται αρχικά να γίνει σύγκριση της Hash τιμής του στοιχείων με τις ονομασίες όλων των πινάκων στην βάση. Αυτό πραγματοποιείται προκειμένου να εντοπιστεί ο πίνακας, προς αποθήκευση, που αντιστοιχεί σε κάθε στοιχείο. Στην συνέχεια πραγματοποιείται η σύνδεση με τον αντίστοιχο πίνακα και αφού πραγματοποιηθεί και σύνδεση με τον πίνακα, τότε ξεκινά η διαδικασία εισαγωγής των στοιχείων.

Κατά την διαδικασία εισαγωγής των στοιχείων, εκχωρούνται τα χαρακτηριστικά αυτών με μέγιστη προσοχή. Αρχικά εισάγεται η τιμή Z-Order κάθε εγγραφής στον πίνακα που αντιστοιχεί. Υπενθυμίζεται ότι κάθε τιμή Z-Order αντιστοιχεί σε μία και μόνο μία εγγραφή. Στην συνέχεια εκχωρούνται τα υπόλοιπα χαρακτηριστικά κάθε εγγραφής. Αυτό το σημείο απαιτεί μεγάλη προσοχή διότι κατά την εισαγωγή των στοιχείων θα πρέπει να δίνεται ιδιαίτερη σημασία στην δήλωση της Column Family που ανήκουν τα χαρακτηριστικά τους.

Επιπρόσθετα, μετά το τέλος της εισαγωγής των στοιχείων στους αντίστοιχους πίνακες που ανήκουν, εφαρμόζεται μία μέθοδος της οποίας σκοπός είναι η εύρεση διπλότυπων εγγραφών. Με αυτό τον τρόπο εξασφαλίζεται ότι κάθε εγγραφή

[8] https://lh3.googleusercontent.com/proxy/V9xS6PqyhzblYYHnNjKC2uF_oh6nwaE1HcucmUiPoNw0DwSHQImXpIodQL2MtL_2J2NlTUJFEoTNkHnrGxJPwpbUp91x

αντιπροσωπεύεται από μία και μόνο μία τιμή (Z-Order). Επιπλέον, διαβεβαιώνεται ότι δεν έχει γίνει κάποιο λάθος κατά την εισαγωγή των χαρακτηριστικών των στοιχείων στην βάση δεδομένων.

7.2.7 Δημιουργία ερωτήματος προς τα δεδομένα

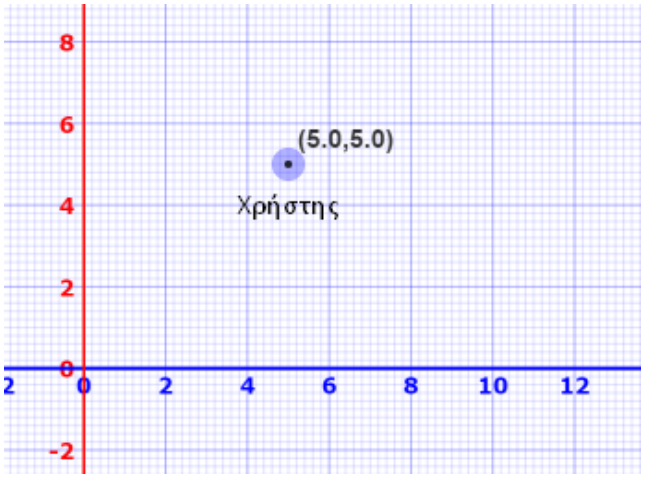
Η δημιουργία του ερωτήματος είναι το κυριότερο κομμάτι της εργασίας και δόθηκε ιδιαίτερη σημασία σε αυτό. Ο τύπος του ερωτήματος που τίθεται στην βάση είναι τύπου παραθύρου (Window Query). Όπως έχει αναφερθεί και σε προηγούμενη ενότητα (Βλέπε υποενότητα 6.3.3) πρόκειται για έναν τύπο ερωτήματος που δημιουργεί ένα εικονικό παράθυρο βάσει των συντεταγμένων και της απόστασης (εύρος) που επιθυμεί ο χρήστης.

Στην παρούσα εργασία, αρχικά το σύστημα δέχεται τρία (3) δεδομένα από τον χρήστη. Το πρώτο είναι η συντεταγμένες της τοποθεσίας του, που περιλαμβάνουν τις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude). Το δεύτερο δεδομένο είναι το εύρος της αναζήτησης που επιθυμεί ο χρήστης και το τρίτο είναι οι προτιμήσεις του χρήστη, οι οποίες αντιστοιχίζονται με τα προϊόντα προς πώληση των καταστημάτων.

Μετά την εισαγωγή των παραπάνω δεδομένων, πραγματοποιείται η διαδικασία της δημιουργίας του ερωτήματος. Για την δημιουργία αυτού, επεξεργάζονται οι συντεταγμένες του χρήστη βάσει του εύρους που έχει εισάγει ο ίδιος. Με τον όρο εύρος, νοείται μία τιμή που αντιπροσωπεύει το εύρος της αναζήτησης σε χιλιόμετρα. Πιο αναλυτικά, στις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude) του χρήστη, αφαιρείται από την κάθε μία η τιμή του εύρους και την αποθηκεύουμε σε μία λίστα. Στην συνέχεια προστίθεται στις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude) του χρήστη το εύρος που επιθυμεί και αποθηκεύουμε και αυτή την τιμή στην λίστα. Με αυτό τον τρόπο δημιουργούμε την κάτω αριστερή και την πάνω δεξιά γωνία του παραθύρου. Εφόσον δημιουργήθηκαν αυτές οι δύο τιμές, το παράθυρο του ερωτήματος έχει σχηματιστεί (εικονικά) και η αναζήτηση περιορίζεται εντός αυτού του τετραγώνου.

Προκειμένου να γίνει περισσότερο κατανοητή η παραπάνω διαδικασία, δίνεται ένα παράδειγμα της χρήσης του.

Έστω ότι ο χρήστης δίνει τις τιμές Latitude=5 και Longitude=5 με εύρος αναζήτησης 2. Επομένως, η τοποθεσία του χρήστη είναι:



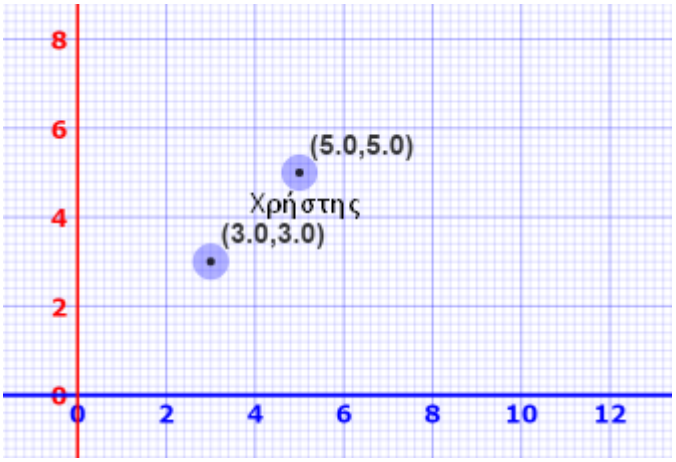
Εικόνα 12. Παράδειγμα δημιουργίας Window Query

Στην συνέχεια, βάσει του εύρους αναζήτησης του χρήστη, που είναι 2, διαμορφώνεται η κάτω αριστερή γωνία και η πάνω δεξιά γωνία του παραθύρου. Για την κάτω αριστερή γωνία, αφαιρείται από τις Latitude και Longitude η τιμή του εύρους. Άρα:

Latitude_Bottom_Left_Corner = Start_Latitude – 2 = 5 – 2 = 3

Longitude_Bottom_Left_Corner = Start_Longitude -2 = 5 – 2 = 3

Και η γραφική απεικόνιση είναι η παρακάτω:



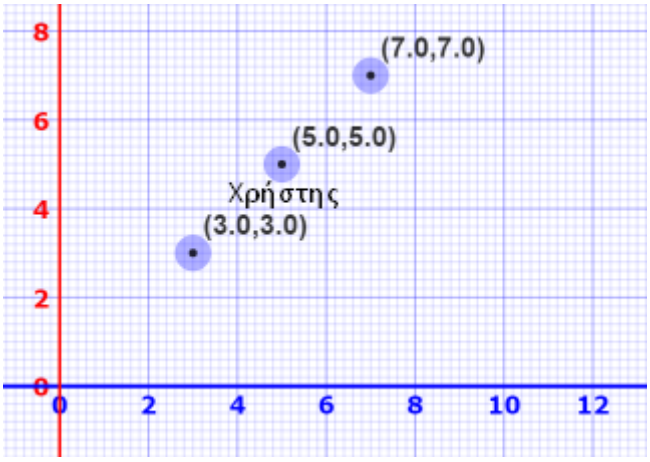
Εικόνα 13. Δημιουργία κάτω αριστερής γωνίας του Window Query

Εν συνεχεία δημιουργείται η πάνω δεξιά γωνία του παραθύρου, προσθέτοντας την τιμή του εύρους στις αρχικές τιμές των συντεταγμένων. Άρα:

Latitude_Top_Right_Corner = Start_Latitude + 2 = 5 + 2 = 7

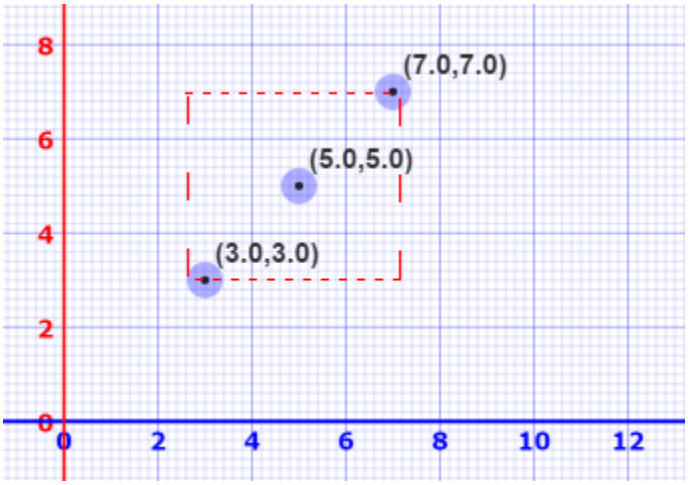
Longitude_Top_Right_Corner = Start_Longitude + 2 = 5 + 2 = 7

και η γραφική απεικόνιση είναι η παρακάτω:



Εικόνα 14. Δημιουργία πάνω δεξιάς γωνίας του Window Query

Επομένως, το τετράγωνο που δημιουργείται και στο οποίο θα εφαρμοστεί η αναζήτηση είναι το παρακάτω:



Εικόνα 15. Γραφική απεικόνιση Window Query

Σημείωση: Το εργαλείο που χρησιμοποιήθηκε για την γραφική απεικόνιση του παραθύρου βρίσκεται στην ιστοσελίδα https://www.mathsisfun.com/data/cartesian-coordinates-interactive.html.

7.2.8 Αναζήτηση βάσει της μεθόδου Grid File και Window Query

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, η αναζήτηση βασίζεται στις τιμές που εκχωρεί ο χρήστης ως είσοδο στο σύστημα, δηλαδή τις τιμές του γεωγραφικού πλάτους και γεωγραφικού μήκους. Επομένως, έπειτα της εισαγωγής αυτών των τιμών στο σύστημα και αφού έχει προηγηθεί η δημιουργία του ερωτήματος παραθύρου ξεκινά η διαδικασία της αναζήτησης.

Η αναζήτηση, αρχικά δημιουργεί την τιμή Hash των συντεταγμένων της κάτω αριστερής και της πάνω δεξιάς γωνίας του παραθύρου. Η δημιουργία των τιμών Hash έχει περιγραφεί σε προηγούμενη ενότητα (Βλέπε υποενότητα 7.2.2). Η διαδικασία αυτή πραγματοποιείται προκειμένου να αναζητήσουμε σε ποιους πίνακες της HBase ανήκουν οι τιμές, του παραθύρου. Ουσιαστικά βάσει της τιμής Hash γίνεται ανάκτηση των ονομάτων των πινάκων της HBase που περιέχουν τα πιθανά αποτελέσματα. Έπειτα της ανάκτησης της ονομασίας των πινάκων, πραγματοποιείται αναζήτηση βάσει των συντεταγμένων των εγγραφών, των πινάκων που ανακτήθηκαν. Πιο συγκεκριμένα, γίνεται σύγκριση μεταξύ των συντεταγμένων των καταστημάτων που έχουμε στην διάθεσή μας (εγγραφές στην βάση) και των συντεταγμένων του χρήστη. Με αυτό τον τρόπο επιστρέφουμε τα καταστήματα που ανήκουν εντός του εύρους του Window Query.

Στην συνέχεια αφού επιστραφούν όλα τα πιθανά αποτελέσματα, εφαρμόζεται μία επιπλέον αναζήτηση. Αυτή η αναζήτηση χρησιμοποιεί ερωτήματα τύπου Boolean Query (Βλέπε ενότητα 6.4) και εφαρμόζεται για την ανάκτηση συγκεκριμένων αποτελεσμάτων, τα οποία σχετίζονται με τις επιθυμίες του χρήστη προς τα προϊόντα προς πώληση των καταστημάτων.

Εν κατακλείδι, η αναζήτηση έχει τρία σκέλη. Το πρώτο σκέλος είναι να ανακτήσει των ονομάτων των πινάκων, που περιέχουν τα δεδομένα που βρίσκονται εντός των ορίων του Window Query. Το δεύτερο είναι να αναζητήσει και να επιστρέψει τα καταστήματα που βρίσκονται εντός των ορίων που θέτονται από το ερώτημα, συγκρίνοντας τις συντεταγμένες (Latitude και Longitude). Ενώ το τρίτο σκέλος εφαρμόζει σε όλα τα πιθανά αποτελέσματα, που επιστρέφει το Window Query, ένα επιπλέον ερώτημα, τύπου Boolean, για να επιστρέψει στον χρήστη μόνο τα δεδομένα που σχετίζονται με το κείμενο που έδωσε ο ίδιος. Με τον όρο κείμενο, εννοούνται ο τύπος των προϊόντων που επιθυμεί ο χρήστης να αναζητήσει.

Για παράδειγμα, ας υποθέσουμε ότι διαθέτουμε μία βάση δεδομένων με πολλούς πίνακες, οι οποίο περιέχουν τα χαρακτηριστικά κάποιων επίγειων καταστημάτων και έστω ότι ο χρήστης θέτει ένα ερώτημα, με τις εξής τιμές Latitude=5, Longitude=5, Range=2 και Κείμενο: Pizza, Coffee, Tea. Σε αυτή την περίπτωση, η αναζήτηση θα πρέπει να επιστρέψει πρώτα τα ονόματα των πινάκων που περιέχουν τα πιθανά αποτελέσματα. Στην συνέχεια να επιστρέψει τα καταστήματα που βρίσκονται εντός δύο (2) χιλιομέτρων, από το σημείο αναφοράς αυτού του χρήστη (Latitude=5, Longitude=5) και τέλος να εφαρμόσει μία επιπλέον αναζήτηση, στα καταστήματα αυτά, προκειμένου να εμφανίσει τα καταστήματα που διαθέτουν προς πώληση τα προϊόντα Pizza, Coffee και Tea.

7.2.9 Παραλλαγή του Grid File

Όπως έχει αναφερθεί και σε προηγούμενη ενότητα (Βλέπε ενότητα 5.4) το Grid File αποτελείται συνήθως από έναν πίνακα. Εσωτερικά αυτού του πίνακα φιλοξενούνται

κελιά, όπου το κάθε ένα από αυτά διαθέτει μία τιμή και προσδιορίζει έναν πίνακα στην βάση δεδομένων. Επομένως, το Grid File αποτελείται από δύο σκέλη. Το πρώτο είναι τα κελιά του πίνακα που διαθέτει και το δεύτερο οι δείκτες που υποδεικνύουν τον πίνακα της βάσης δεδομένων. Με αυτό τον τρόπο η αναζήτηση αρχίζει αναζητώντας το κελί που προσδιορίζει μία πλειάδα δεδομένων και έπειτα αναζητά στον αντίστοιχο πίνακα. Αυτό επιτυγχάνει την αναζήτηση κάνοντάς την αρκετά αποδοτική, διότι περιορίζει τον αριθμό των πινάκων και κατ΄ επέκταση των δεδομένων που πρέπει να προσπελαστούν.

Στην παρούσα εργασία εφαρμόζεται μια παραλλαγή του Grid File. Όπως έχει αναφερθεί, κάθε κελί του Grid File έχει μία ονομασία αποτελούμενη από αριθμούς. Προκειμένου να αποφύγουμε την δημιουργία του πλέγματος θέτονται οι τιμές των κελιών ως ονομασίες των πινάκων. Οι τιμές αυτές προκύπτουν από τον υπολογισμό της τιμής Hash των εγγραφών. Αυτό έχει ως αποτέλεσμα να αναζητούνται κατευθείαν οι πίνακες που επιθυμούμε στην βάση δεδομένων. Επομένως, το παραπάνω, δεν επηρεάζει την αποδοτικότητα της αναζήτησης και ταυτόχρονα εξοικονομείται αποθηκευτικός χώρος μιας και η δημιουργία του πλέγματος δεν πραγματοποιείται.

7.3 Υλοποίηση Ordered Indexing

Σε αυτή την ενότητα θα περιγραφεί η υλοποίηση της μεθόδου Ordered Indexing καθώς επίσης και ο τρόπος που πραγματοποιείται η αναζήτηση σε αυτήν την μέθοδο.

7.3.1 Ανάκτηση δεδομένων από το αρχικό αρχείο

Όπως η μέθοδος του Grid File έτσι και η Ordered Indexing υλοποιήθηκε με την χρήση της αντικειμενοστραφούς γλώσσα προγραμματισμού Java. Ο λόγος που επιλέχθηκε αυτή η γλώσσα προγραμματισμού είναι κυρίως η συμβατότητα την οποία παρουσιάζει με το σύστημα διαχείρισης της HBase.

Αρχικά, με την χρήση ενός *Buffer Reader* πραγματοποιείται η ανάγνωση του αρχείου. Η ανάγνωση γίνεται σειριακά και ξεκινά από την πρώτη εγγραφή του αρχείου, που περιέχει τα καταστήματα, μέχρι την τελευταία.

Όπως έχει αναφερθεί, ήδη, κάθε στοιχείο απαρτίζεται από πέντε (5) χαρακτηριστικά (όνομα, βαθμολογία, γεωγραφικό πλάτος, γεωγραφικό μήκος και τα προϊόντα προς πώληση) τα οποία θα πρέπει να εκχωρηθούν σε μία μεταβλητή που θα αντιπροσωπεύει κάθε εγγραφή του αρχείου. Για να πραγματοποιηθεί αυτό δημιουργήθηκε μία κλάση, της οποίας τα αντικείμενα περιέχουν όλα τα παραπάνω χαρακτηριστικά των στοιχείων. Ουσιαστικά, κάθε αντικείμενο αυτής της κλάσης περιέχει τα χαρακτηριστικά κάθε καταστήματος και αντιπροσωπεύει μία και μόνο μία εγγραφή του αρχικού αρχείου που έχουμε στην διάθεσή μας.

7.3.2 Δημιουργία τιμής Z-order

Προκειμένου, να επιτυγχάνεται η μοναδικότητα της κάθε εγγραφής δημιουργείται μία τιμή Z-Order για κάθε εγγραφή.

Κατά την δημιουργία των αντικειμένων, πραγματοποιείται και η δημιουργία της τιμής Z-Order κάθε καταστήματος. Αυτό υλοποιείται με την βοήθεια των συντεταγμένων (Latitude και Longitude), της οποίας η διαδικασία έχει περιγραφεί σε προηγούμενη ενότητα (Βλέπε ενότητα 4.5). Ο λόγος χρήσης της είναι να αναθέσει ένα μοναδικό αναγνωριστικό για κάθε εγγραφή. Αυτή, είναι μία υψίστης σημασίας τιμή και συμβάλλει στην ταξινόμηση και αναζήτηση των αντικειμένων. Επιπρόσθετα, η τιμή αυτή αποτελεί ένα επιπλέον χαρακτηριστικό των εγγραφών και επισυνάπτεται σε κάθε αντικείμενο αυτών.

Πιο συγκεκριμένα, κατά την ανάκτηση των στοιχείων, από το αρχείο που περιέχει τα στοιχεία που αφορούν τα καταστήματα εστίασης, ανακτώνται και οι τιμές των συντεταγμένων (Latitude και Longitude). Έπειτα της ανάκτησης των τιμών αυτών, δημιουργείται η τιμή Z-Order. Με την δημιουργία αυτής της τιμής επιτυγχάνεται η μοναδικότητα κάθε εγγραφής και χρησιμοποιείται ως κριτήριο της ταξινόμησης των εγγραφών.

Στην συνέχεια, αφού έχει πραγματοποιηθεί η ανάκτηση και η υλοποίηση των τιμών Z-Order των εγγραφών, δημιουργείται το αντίστοιχο αντικείμενο. Ουσιαστικά, σε κάθε αντικείμενο προσδίδεται ένα επιπλέον χαρακτηριστικό και αυτό είναι η τιμή του Z-order.

7.3.3 Ταξινόμηση των αντικειμένων

Έπειτα της ανάκτησης, της δημιουργίας των τιμών Z-Order και των αντικειμένων των καταστημάτων, καταχωρούνται όλα τα αντικείμενα σε μία προσωρινή λίστα. Στην συνέχεια με την βοήθεια της ταξινόμησης της Heap Sort (Βλέπε υποενότητες 5.3.3 και 5.3.5) ταξινομείται αυτή η λίστα, βάσει τη τιμής Z-Order κάθε αντικειμένου που περιέχεται σε αυτήν.

Η παραπάνω διαδικασία έχει σαν αποτέλεσμα μία αύξουσα ταξινομημένη λίστα.

7.3.4 Δημιουργία πινάκων στην HBase

Όπως έχει αναφερθεί και στην περιγραφή της μεθόδου της Ordered Indexing, πρόκειται για μία μέθοδο που ο σκοπός της είναι η διάσπαση των εγγραφών σε περισσότερους από έναν πίνακες που θα αποθηκευτούν στην βάση της HBase. Το μέγεθος, αυτών των πινάκων εξαρτάται από τους σχεδιαστές και ο μόνος περιορισμός που υπάρχει είναι να μην υπερφορτωθεί κάποιος πίνακας με αποτέλεσμα να επιδρά αρνητικά στην αναζήτηση των εγγραφών.

Στην παρούσα εργασία αποφασίστηκε ότι ένα κατάλληλο μέγεθος και όριο αποθήκευσης εγγραφών κάθε πίνακα, το οποίο δεν θα επηρεάζει την αποδοτικότητα της αναζήτησης, είναι οι δέκα χιλιάδες (10.000) εγγραφές. Αυτό σημαίνει ότι κάθε πίνακας θα έχει μέγιστο όριο αποθήκευσης τις 10.000 εγγραφές.

Επιπλέον, το αρχείο το οποίο έχουμε στην διάθεσή μας για την διεξαγωγή των πειραμάτων, περιέχει εβδομήντα οχτώ χιλιάδες εννιακόσια ογδόντα ένα (78.981) εγγραφές, με περιεχόμενο τα χαρακτηριστικά κάποιων καταστημάτων εστίασης. Επομένως, βάσει το όριο αποθήκευσης που έχει τεθεί σε κάθε πίνακα, θα χρειαστούμε οχτώ (8) πίνακες για την αποθήκευση των δεδομένων, διότι:

78.981 / 10.000 =7,8 = ⌈7,8⌉ = 8

(συνολικές εγγραφές) (όριο εγγραφών κάθε πίνακα)

Άρα κάθε πίνακας περιέχει 10.000 εγγραφές, με εξαίρεση τον τελευταίο (όγδοο) πίνακα που θα περιέχει 8.981 εγγραφές. Επιπλέον, οι εγγραφές ανακτώνται και αποθηκεύονται από την ταξινομημένη λίστα. Πιο αναλυτικά:

Πίνακας	**Αριθμός εγγραφών**
Πίνακας 1	Από την *1η* μέχρι την *9.999η* εγγραφή της λίστας
Πίνακας 2	Από την *10.000η* μέχρι την *19.999η* εγγραφή της λίστας
Πίνακας 3	Από την *20.000η* μέχρι την *29.999η* εγγραφή της λίστας
Πίνακας 4	Από την *30.000η* μέχρι την *39.999η* εγγραφή της λίστας
Πίνακας 5	Από την *40.000η* μέχρι την *49.999η* εγγραφή της λίστας
Πίνακας 6	Από την *50.000η* μέχρι την *59.999η* εγγραφή της λίστας
Πίνακας 7	Από την *60.000η* μέχρι την *69.999η* εγγραφή της λίστας
Πίνακας 8	Από την *70.000η* μέχρι την *78.981η* εγγραφή της λίστας

Πίνακας 3. Κατανομή εγγραφών στους πίνακες

7.3.5 Αποθήκευση των μέτα-δεδομένων των πινάκων

Στην περίπτωση που τεθεί ένα ερώτημα στους πίνακες της βάσης δεδομένων, προκειμένου να επιστραφούν τα αποτελέσματα, θα πρέπει να προσπελαστούν όλοι οι πίνακες επηρεάζοντας αρκετά την αποδοτικότητά της. Βάσει αυτού θα έπρεπε να προσπελαστούν και όλες οι εγγραφές που περιέχονται στην βάση και των οχτώ πινάκων.

Προκειμένου, να μην επηρεάζεται η αποδοτικότητα της αναζήτησης, οι πρώτες και οι τελευταίες τιμές Z-Order των εγγραφών κάθε πίνακα αποθηκεύονται σε έναν διαφορετικό πίνακα, δημιουργώντας ένα ευρετήριο. Με αυτό τον τρόπο η αναζήτηση ξεκινά από το ευρετήριο των πινάκων αναζητώντας σε ποιον πίνακα ανήκουν οι τιμές του ερωτήματος και αναζητά μόνο σε αυτούς που είναι απαραίτητο.

7.3.6 Δημιουργία Column Family των πινάκων

Εφόσον, αναφερόμαστε σε πίνακες της HBase, η δημιουργία των Column Family είναι μία αρκετά σημαντική διαδικασία. Χρησιμοποιείται κυρίως για την ομαδοποίηση των δεδομένων και συμβάλει κατά πολύ στην αναζήτηση και την ανάκτηση των δεδομένων.

Στην παρούσα εργασία έχουν δημιουργηθεί τρεις (3) Column Family για την ομαδοποίηση των χαρακτηριστικών των καταστημάτων. Η κατανομή των χαρακτηριστικών σε κάθε Column Family είναι η ίδια που εφαρμόστηκε και στην μέθοδο του Grid File (Βλέπε Εικόνα 2 της ενότητας 7.2.5).

7.3.7 Εισαγωγή των στοιχείων στους πίνακες

Οι εγγραφές, που διατίθενται για την παρούσα εργασία είναι πλέον αποθηκευμένες με τις τιμές Z-Order της κάθε μίας σε μία λίστα. Υπενθυμίζεται ότι αυτή λίστα είναι ταξινομημένη με βάση τις τιμές Z-Order.

Η εισαγωγή των στοιχείων στους πίνακες της βάσης δεδομένων, γίνεται με την ανάγνωση της προαναφερθείσας λίστας. Ουσιαστικά διατρέχουμε την λίστα που περιέχει τις εγγραφές και εισάγουμε τα στοιχεία αυτών στους οχτώ πίνακες τις HBase, ανά 10.000 εγγραφές.

Μετά το τέλος, της διαδικασίας της εισαγωγής των στοιχείων, στους πίνακες ανακτώνται οι πρώτες και οι τελευταίες εγγραφές κάθε πίνακα. Έπειτα της ανάκτησης αυτής, εκχωρούνται οι τιμές των Z-Order των (πρώτων και τελευταίων) εγγραφών. Με αυτό τον τρόπο δημιουργείται ένα HashMap (ή αλλιώς ευρετήριο) το οποίο συμβάλλει αρκετά στην αύξηση της αποδοτικότητας της αναζήτησης.

7.3.8 Δημιουργία του ερωτήματος προς τα δεδομένα

Ο τύπος του ερωτήματος που τίθεται στην βάση είναι τύπου παραθύρου (Window Query). Όπως έχει αναφερθεί και σε προηγούμενη ενότητα (Βλέπε υποενότητα 6.3.3) πρόκειται για έναν τύπο ερωτήματος που δημιουργεί ένα εικονικό παράθυρο βάσει των συντεταγμένων και της απόστασης (εύρος) που επιθυμεί ο χρήστης.

Στην παρούσα εργασία, αρχικά το σύστημα δέχεται τρία (3) δεδομένα από τον χρήστη. Το πρώτο είναι η συντεταγμένες της τοποθεσίας του, που περιλαμβάνουν τις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude). Το δεύτερο δεδομένο είναι το εύρος της αναζήτησης που επιθυμεί ο χρήστης και το τρίτο

είναι οι προτιμήσεις του χρήστη, οι οποίες αντιστοιχίζονται με τα προϊόντα προς πώληση των καταστημάτων.

Έπειτα της εισαγωγής των παραπάνω δεδομένων, πραγματοποιείται η διαδικασία της δημιουργίας του ερωτήματος. Για την δημιουργία αυτού, επεξεργάζονται οι συντεταγμένες του χρήστη βάσει του εύρους που έχει εισάγει ο ίδιος. Με τον όρο εύρος, νοείται μία τιμή που αντιπροσωπεύει το εύρος της αναζήτησης σε χιλιόμετρα.

Πιο αναλυτικά, στις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude) του χρήστη, αφαιρείται από την κάθε μία η τιμή του εύρους και την αποθηκεύουμε σε μία λίστα. Στην συνέχεια προστίθενται στις τιμές του γεωγραφικού πλάτους (Latitude) και γεωγραφικού μήκους (Longitude) του χρήστη το εύρος που επιθυμεί και αυτή η τιμή αποθηκεύεται στην λίστα. Με αυτό τον τρόπο δημιουργείται η κάτω αριστερή και η πάνω δεξιά γωνία του παραθύρου. Εφόσον δημιουργηθούν αυτές οι δύο τιμές, το παράθυρο του ερωτήματος έχει σχηματιστεί και έχει τεθεί το όριο της αναζήτησης που περιορίζεται εντός αυτού του τετραγώνου.

Επισημαίνεται, ότι οι τιμές του παραθύρου, σε αυτήν την μέθοδο (Ordered Indexing), δεν παραμένουν και δεν αποθηκεύονται ως συντεταγμένες. Αντίθετά, δημιουργείται και αποθηκεύεται η τιμή Z-Order αυτών. Ο τρόπος δημιουργίας του Z-Order έχει ήδη περιγραφεί σε προηγούμενη ενότητα (Βλέπε 4.5). Αυτή η διαδικασία πραγματοποιείται διότι η αναζήτηση στην παρούσα μέθοδο βασίζεται στις Z-Order τιμές.

7.3.9 Αναζήτηση βάσει της μεθόδου Ordered Indexing και Window Query

Η αναζήτηση βασίζεται στις τιμές που εκχωρεί ο χρήστης ως είσοδο στο σύστημα, δηλαδή τις τιμές του γεωγραφικού πλάτους και γεωγραφικού μήκους. Επομένως, έπειτα της εισαγωγής αυτών των τιμών στο σύστημα και αφού έχει προηγηθεί η δημιουργία του ερωτήματος παραθύρου ξεκινά η διαδικασία της αναζήτησης. Κατά την διαδικασία της αναζήτησης ανακτώνται οι τιμές Z-Order της κάτω αριστερής και της πάνω δεξιάς γωνίας του τετραγώνου. Βάσει αυτών των τιμών, η αναζήτηση διατρέχει τον πίνακα των μέτα-δεδομένων των πινάκων (ευρετήριο). Αυτό έχει σκοπό την εύρεση των πινάκων που περιέχουν τις τιμές των ορίων του τετραγώνου του Window Query. Το παραπάνω, επιτυγχάνεται με την σύγκριση των τιμών Z-Order των συντεταγμένων και των αρχικών και τελικών Z-Order τιμών των πινάκων. Έπειτα της εύρεσης και της επιστροφής, του ονόματος, των προαναφερθέντων πινάκων, η αναζήτηση συνεχίζει με την προσπέλαση των συγκεκριμένων πινάκων και επιστρέφει τα αποτελέσματα.

Τα αποτελέσματα αυτά ανταποκρίνονται ως προσωρινά και πιθανά αποτελέσματα και αντιστοιχούν στα καταστήματα που ανήκουν εντός του ερωτήματος τετραγώνου. Προκειμένου να επιστραφούν τα τελικά αποτελέσματα

χρησιμοποιείται μία επιπλέον αναζήτηση, η οποία συγκρίνει τις προτιμήσεις του χρήστη ως προς τα προϊόντα που επιθυμεί με τα προϊόντα προς πώληση των καταστημάτων που επιστράφηκαν, κατά την πρώτη αναζήτηση.

Η αναζήτηση αυτή, είναι τύπου Boolean Query (Βλέπε ενότητα 6.4) και εφαρμόζεται για την ανάκτηση συγκεκριμένων αποτελεσμάτων, τα οποία σχετίζονται με τις επιθυμίες του χρήστη προς τα προϊόντα προς πώληση των καταστημάτων.

Για να γίνει περισσότερο κατανοητή η λειτουργία της αναζήτησης, παρακάτω, δίνεται ένα παράδειγμα.

Το δεδομένα τα οποία έχουμε στην διάθεσή μας αφορούν καταστήματα εστίασης, με περιεχόμενο τα χαρακτηριστικά τους και ταξινομημένα σε πίνακες βάσει της τιμής Z-Order της κάθε μίας.

Ας υποθέσουμε ότι ο χρήστης θέτει το εξής ερώτημα στην βάση της HBase:

Latitude=5, Longitude=5, Range=2 και Λέξεις-κλειδιά (keywords): Pizza, Coffee, Tea, όπου Latitude και Longitude είναι το σημείο αναφοράς του χρήστη, το Range είναι το εύρος αναζήτησης του ερωτήματος σε χιλιόμετρα και λέξεις-κλειδιά οι προτιμήσεις των προϊόντων που επιθυμεί.

Κατά την εκτέλεση του ερωτήματος, αρχικά, με την βοήθεια των συντεταγμένων του χρήστη και του εύρους αναζήτησης, δημιουργείται το παράθυρο της αναζήτησης (Window Query) δημιουργώντας τις τιμές της κάτω αριστερής γωνίας και της πάνω δεξιάς γωνίας, μετατρέποντας αυτές τις τιμές σε Z-Order, έχοντας σαν αποτέλεσμα τα όρια του ερωτήματος.

Στην συνέχεια, αναζητείται στον πίνακα των μέτα-δεδομένων, βάσει των τιμών Z-Order σε ποιον πίνακα ανήκουν τα όρια (κάτω αριστερή και πάνω δεξιά γωνία) του παραθύρου με σκοπό να μας επιστραφούν τα ονόματα αυτών.

Έπειτα της εύρεση των ονομάτων των πινάκων, αναζητούνται οι εγγραφές που ανήκουν σε αυτούς τους πίνακες και επιστρέφονται τα πιθανά αποτελέσματα, που ανήκουν εντός των ορίων του παραθύρου.

Εν συνεχεία, εφαρμόζεται σε όλα τα πιθανά αποτελέσματα ερωτήματα Boolean Query για να ανακτηθούν τα αποτελέσματα που σχετίζονται με τις επιθυμίες ως προς τα προϊόντα που επιθυμεί ο ίδιος.

7.4 Παραμετροποίηση ερωτήματος

Όπως έχει ήδη αναφερθεί, η δημιουργία του ερωτήματος βασίζεται σε τρία κριτήρια τα οποία εκχωρεί ο χρήστης ως είσοδο στο σύστημα. Επιπλέον, το ερώτημα χωρίζεται σε δύο, κυρίως, σκέλη. Το πρώτο είναι η δημιουργία του ερωτήματος παραθύρου και η εύρεση των πινάκων που περιέχουν τα στοιχεία που επιθυμεί ο ίδιος

και το δεύτερο είναι η εύρεση των στοιχείων, βάσει των προτιμήσεων του χρήστη ως προς τα προϊόντα.

Όσον αφορά το πρώτο σκέλος, η δημιουργία του ερωτήματος παραθύρου και η εύρεση των πινάκων με περιεχόμενο τις τιμές που ανήκουν εντός του παραθύρου, εξαρτάται από τις τιμές του γεωγραφικού πλάτους και γεωγραφικού μήκους του χρήστη, καθώς επίσης και το εύρος αναζήτησης που δίνει ο χρήστης. Επιπρόσθετα, το εύρος που δίνει ο χρήστης, από μόνο του είναι μία τιμή. Επομένως, αυτή η τιμή πρέπει να μετατραπεί σε μια μονάδα του SI προκειμένου να υπάρχει μία συνάφεια του εύρους του ερωτήματος με την απόσταση στην επιφάνεια της γης.

Η μονάδα SI, μετατροπής του εύρους, που επιλέχθηκε είναι το χιλιόμετρο (km) λόγω του γεγονότος ότι παρέχει την απαραίτητη ακρίβεια που απαιτείται για τη συγκεκριμένη εφαρμογή. Άρα, βάσει την τιμή του εύρους σε χιλιόμετρα θα σχηματιστεί και το παράθυρο του ερωτήματος.

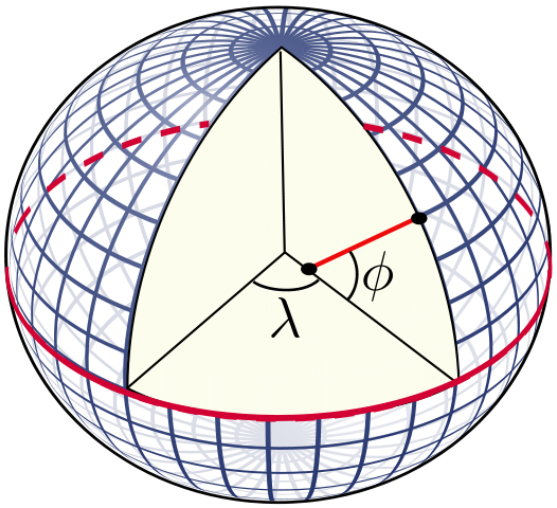
Έπειτα από μελέτη που πραγματοποιήθηκε, αποφάνθηκε ότι η αλλαγή μίας μοίρας της τιμής του γεωγραφικού πλάτους (Latitude) μπορεί να μετατραπεί σε χιλιόμετρα στο εύρος (110.567-111.699). Στην συγκεκριμένη υλοποίηση επιλέγεται ο μέσος όρος των παραπάνω ακραίων τιμών ώστε να αντιστοιχισθεί η αλλαγή μιας μοίρας γεωγραφικούς πλάτους σε χιλιόμετρα. Επομένως, η αλλαγή γεωγραφικού πλάτους θεωρείται σταθερή με τιμή $\frac{110.567-111.699}{2} = 111.133$ Km. Χρησιμοποιώντας το δεδομένο που αναφέρθηκε προηγουμένως μπορεί να βρεθεί η μετατόπιση στον άξονα Y (Latidude) από το σημείο αναφοράς του χρήστης βάσει της απόστασης(συμβολίζεται ως d (distance)), επομένως προκύπτει ότι $transY = \frac{distance}{111.3}_{\circ}$.

Στην συνέχεια, θα πρέπει να βρεθεί η μετατόπιση στον άξονα X (γεωγραφικό μήκος-Longitude) ώστε να βρεθούν οι τέσσερις γωνίες για να δημιουργηθεί το ερώτημα παραθύρου (Window Query). Στην συγκεκριμένη συντεταγμένη δημιουργείται πρόβλημα διότι το εύρος στο οποίο κυμαίνεται είναι (111.320 km-0.000 km), όπως φαίνεται στον παρακάτω πίνακα.

ϕ	Δ^1_{lat}	Δ^1_{long}
0°	110.574 km	111.320 km
15°	110.649 km	107.550 km
30°	110.852 km	96.486 km
45°	111.132 km	78.847 km
60°	111.412 km	55.800 km
75°	111.618 km	28.902 km
90°	111.694 km	0.000 km

Εικόνα 16. Τιμές διακύμανσης του γεωγραφικού μήκους (Longitude)

Είναι απαραίτητο να σημειωθεί ότι η γωνιά φ αφορά την γωνία που σχηματίζεται μεταξύ της επιφάνειας της γης και του επιπέδου του ισημερινού, όπως φαίνεται στην παρακάτω εικόνα.



[9]

Εικόνα 17. Απεικόνιση γωνίας

Αυτό έχει ως αποτέλεσμα να είναι απαραίτητη η χρήση εξισώσεων συναρτήσει του γεωγραφικού πλάτους και της απόστασης (distance) ώστε να βρεθεί η μετατόπιση στον άξονα X (Longitude). Μία μέθοδος για την επίλυση του προβλήματος που αναφέρθηκε είναι η ισοορθογώνια προβολή [20]. Η συγκεκριμένη απλοποιεί το σχεδόν σφαιρικό σχήμα της σε κυλινδρικό ώστε να είναι απλοποιημένες και οι τριγωνομετρικές εξισώσεις που εφαρμόζονται σε αυτή. Η επιλογή της μεθόδου

[9] https://en.wikipedia.org/wiki/File:Latitude_and_longitude_graticule_on_a_sphere.svg

πραγματοποιήθηκε βάσει της ακρίβειας που απαιτείται από το σύστημα καθώς και το υπολογιστικό κόστος για να πραγματοποιηθούν οι απαραίτητες πράξεις των εξισώσεων. Επομένως, χρησιμοποιώντας αυτή την μέθοδο μειώνεται κατά το ελάχιστο η ακρίβεια υπολογισμού της μετατόπισης στον άξονα Χ αλλά κερδίζεται υπολογιστικό κόστος που δεν θα υπήρχε αν επιλεγόταν κάποιες εξισώσεις με μεγαλύτερη πολυπλοκότητα. Έτσι οι τρεις εξισώσεις που χρησιμοποιούνται είναι οι εξής:

(1): $x = (\lambda 2 - \lambda 1) * \cos\left(\frac{\varphi 1+\varphi 2}{2}\right)$

(2): $y = \varphi 2 - \varphi 1$

(3): $d = R * \sqrt{x^2 + y^2}$

Συμβολισμοί των παραπάνω εξισώσεων:

φ1: γεωγραφικό πλάτος(latitude) σημείου του χρήστη

φ2: μέγιστο γεωγραφικό πλάτος(latitude) δεδομένου της απόστασης του ερωτήματος

λ1: γεωγραφικό μήκος (longitude) σημείου του χρήστη

λ2: μέγιστο γεωγραφικό μήκος (longitude) που προκύπτει από τις παραπάνω εξισώσεις

d: απόσταση του ερωτήματος

R: ακτίνα της γης(σταθερά): 6371

Επομένως το ζητούμενο είναι το Δλ=λ2-λ1

Παρακάτω παρουσιάζεται, ένα παράδειγμα, για τον υπολογισμό του ερωτήματος παραθύρου.

Δεδομένα παραδείγματος:

φ1=34.1919, λ1=-118.9375, d=1 Km

Επίλυση:

Αρχικά υπολογίζεται η μετατόπιση του γεωγραφικού πλάτους(Latitude-Άξονας Y) σύμφωνα με την δεδομένη απόσταση του χρήστη:

$$\Delta\varphi=y=\frac{1}{111.13}=8.998*10^{-3}$$

Επομένως χρησιμοποιώντας την εξίσωση (2) υπολογίζεται η μέγιστη τιμή του γεωγραφικού πλάτος, όπως φαίνεται παρακάτω:

$$(2) \rightarrow \varphi 2 = 8.998 * 10^{-3} + \varphi 1 = 34.20089$$

Στην συνέχεια, χρησιμοποιείται η εξίσωση (3) για τον υπολογισμό της μετατόπισης του γεωγραφικού μήκους (Longitude - Άξονας X):

$$(3) \rightarrow (\frac{1}{6731})^2 = \chi^2 + 8.0964 * 10^{-5} \leftrightarrow \chi = 8.9966 * 10^{-3}$$

$$(1) \rightarrow \frac{8.9966 * 10^{-3}}{-0.93550} = -9.6169 * 10^{-3}$$

Άρα $\lambda 2 = \Delta\lambda + \lambda 1 = -118.9471$

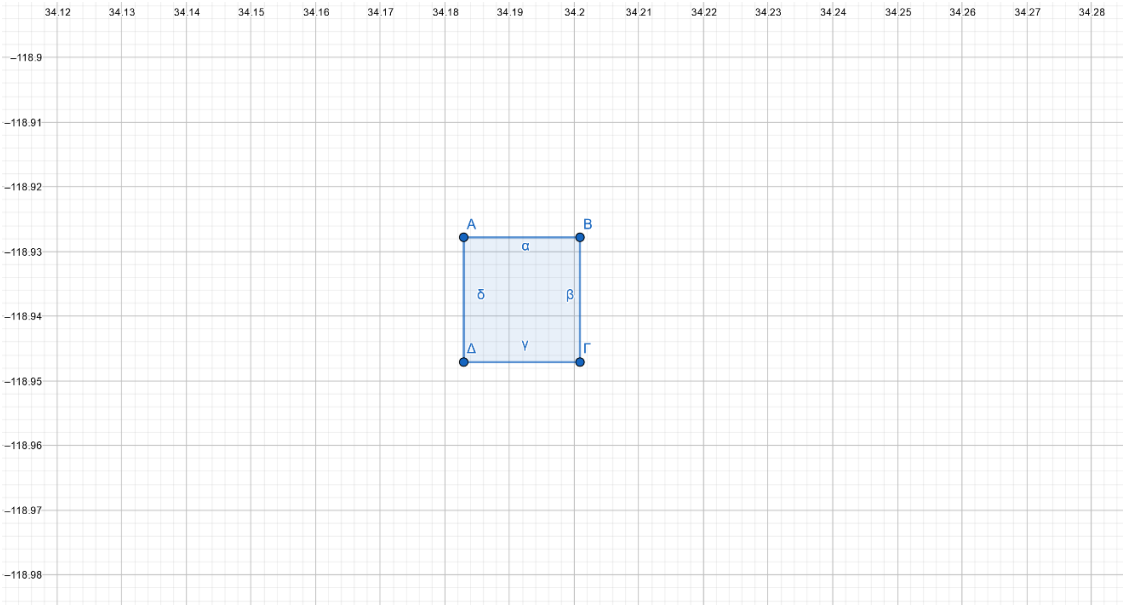
Τέλος, μπορούν να υπολογιστούν οι ελάχιστες τιμές του γεωγραφικού πλάτους (φ3) και γεωγραφικού μήκους (λ3), όπως φαίνεται παρακάτω.

$$\varphi 3 = \varphi 1 - 8.998 * 10^{-3} = 34.1829$$

$$\lambda 3 = \lambda 1 - \Delta\lambda = -118.9375 + 9.6169 * 10^{-3} = -118.9278$$

Γωνίες τετραγώνου:

- Κάτω αριστερή γωνία-Σημείο A $(34.1829, -118.9278)$
- Πάνω αριστερή γωνία-Σημείο B $(34.20089, -118.9278)$
- Πάνω δεξιά γωνία-Σημείο C $(34.20089, -118.9471)$
- Κάτω δεξιά γωνία-Σημείο D $(34.1829, -118.9471)$



Εικόνα 18. Απεικόνιση ερωτήματος παραθύρου με δεδομένα παραδείγματος

8

Αποτελέσματα πειραμάτων

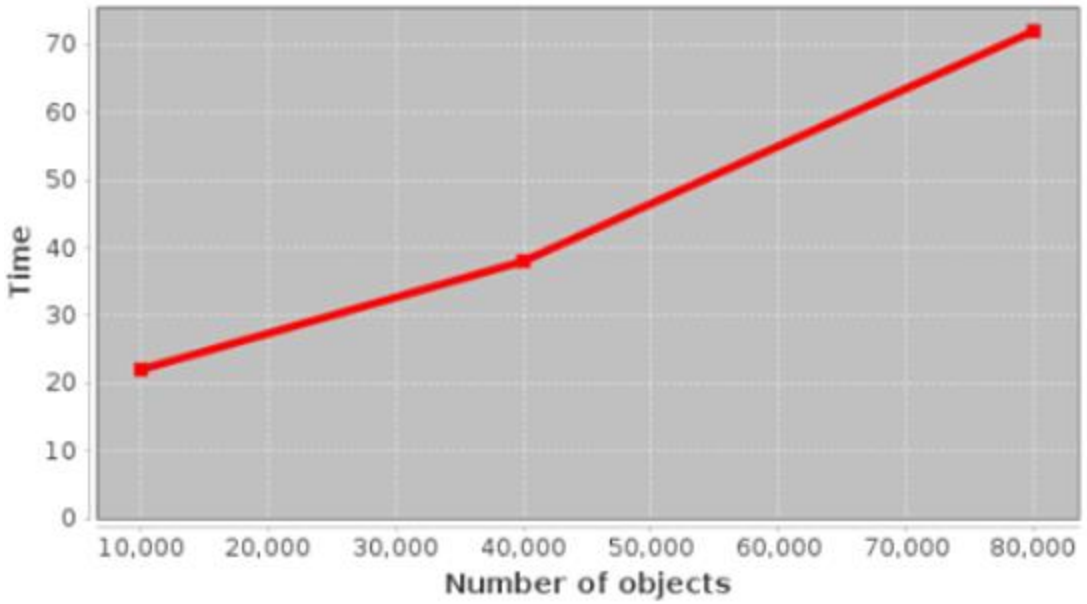
Σε αυτό το κεφάλαιο θα παρουσιαστούν τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν, με την μορφή διαγραμμάτων που θα απεικονίζουν τους χρόνους απόκρισης των δύο μεθόδων, τόσο στην αναζήτηση όσο και στην εισαγωγή των δεδομένων στην HBase. Για την διεξαγωγή των πειραμάτων χρησιμοποιήθηκε ένα αρχείο με σχεδόν 80.000 εγγραφές, οι οποίες έχουν ως περιεχόμενο μερικά χαρακτηριστικά κάποιων επίγειων καταστημάτων εστίασης (Βλέπε ενότητα 7.1).

8.1 Εισαγωγή δεδομένων στην βάση δεδομένων (HBase)

Σε αυτήν την ενότητα θα παρουσιαστούν τα διαγράμματα, τα οποία θα αποτυπώνουν τον χρόνο που χρειάστηκε κάθε μέθοδος, να εισάγει τα στοιχεία στην βάση, σε συνάρτηση με το πλήθος των δεδομένων.

8.1.1 Εισαγωγή δεδομένων στην βάση με χρήση της μεθόδου **Grid File**

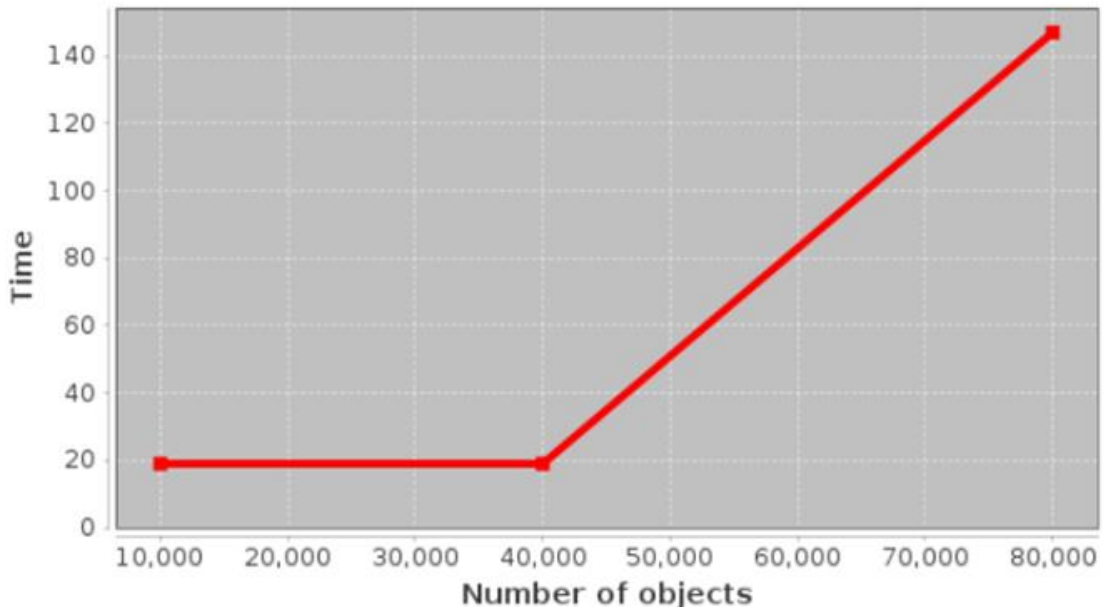
Η πρώτη πειραματική μελέτη πραγματοποιήθηκε για την εισαγωγή των δεδομένων στους πίνακες της HBase. Επιπλέον, λαμβάνεται υπόψιν και ο χρόνος που χρειάστηκε να ομαδοποιηθούν τα δεδομένα. Ο ακριβής αριθμός των εγγραφών είναι 78.981. Στο παρακάτω διάγραμμα, ο άξονας Χ αντιπροσωπεύει των αριθμό των εγγραφών που εισάγονται στην βάση δεδομένων, ενώ ο άξονας Υ αντιπροσωπεύει τον χρόνο σε δευτερόλεπτα (seconds), ο οποίος απαιτείται για την εισαγωγή των δεδομένων. Πιο συγκεκριμένα, παρατηρείται ότι οι 78.981 εγγραφές χρειάζονται λίγα περισσότερα από εβδομήντα (70) δευτερόλεπτα για την ολοκληρωθεί η εισαγωγή τους στην βάση δεδομένων.



Εικόνα 19. Διάγραμμα απεικόνισης χρόνου εισαγωγής των δεδομένων με την χρήση της μεθόδου Grid File

8.1.2 Εισαγωγή δεδομένων στην βάση με χρήση της μεθόδου Ordered Indexing

Όπως φαίνεται και από το παρακάτω διάγραμμα, παρατηρείται ότι απαιτείται περισσότερος χρόνος, προκειμένου να ολοκληρωθεί η εισαγωγή των στοιχείων στην βάση δεδομένων. Αυτό συμβαίνει διότι στην μέθοδο της Ordered Indexing, εκτός της δημιουργίας της τιμής Z-Order χρειάζεται και η ταξινόμηση αυτών των τιμών πριν εκχωρηθούν στους πίνακες της HBase. Επομένως, από το παρακάτω διάγραμμα είναι φανερό ότι μέθοδος Ordered Indexing απαιτεί τον κάτι παραπάνω από εκατόν σαράντα (140) δευτερόλεπτα για την εισαγωγή 78.981 εγγραφών στην βάση.



Εικόνα 20. Διάγραμμα απεικόνισης χρόνου εισαγωγής των δεδομένων με την χρήση της μεθόδου Ordered Indexing

8.2 Χρόνος απόκρισης του ερωτήματος του ερωτήματος

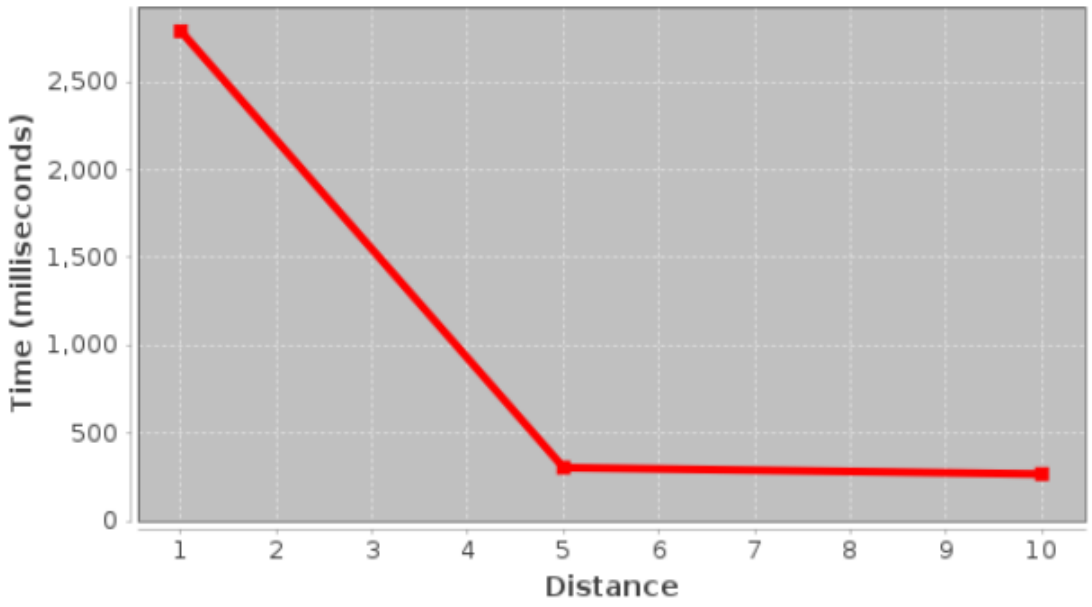
Σε αυτήν την ενότητα θα παρουσιαστούν τα διαγράμματα, τα οποία θα αποτυπώνουν τον χρόνο που χρειάστηκε η εκτέλεση του ερωτήματος σε κάθε μέθοδο, έτσι ώστε να ελέγξει τις εγγραφές των πινάκων που βρίσκονται στην βάση και να επιστρέψει τα αποτελέσματα στον χρήστη.

8.2.1 Χρόνος απόκρισης του ερωτήματος εφαρμοσμένο στην μέθοδο αποθήκευσης Grid File

Στο παρακάτω διάγραμμα απεικονίζεται ο χρόνος που απαιτείται για την ολοκλήρωση ενός ερωτήματος που θέτει ο χρήστης προς το σύστημα, όπου τα δεδομένα καταχωρήθηκαν στην βάση με την μέθοδο του Grid File. Επισημαίνεται ότι ο χρόνος, αυτός περιλαμβάνει το χωρικό ερώτημα που τίθεται στα δεδομένα, δηλαδή την δημιουργία του παραθύρου (Window Query) και το Boolean ερώτημα που σχετίζεται με τις επιθυμίες του χρήστη ως προς τα προϊόντα προς πώληση που θέλει ο ίδιος. Επιπλέον, ο χρόνος περιλαμβάνει και τον χρόνο που απαιτεί η επιστροφή των αποτελεσμάτων στον χρήστη.

Όπως φαίνεται και παρακάτω, ο χρόνος που απαιτεί η εκτέλεση του ερωτήματος για εύρος αναζήτησης ενός χιλιομέτρου (1km) είναι περίπου 2.700 milliseconds, δηλαδή 2.7 δευτερόλεπτα.

Αντίθετα, με εύρος αναζήτησης τα πέντε χιλιόμετρα (5km), ο χρόνος απόκρισης είναι περίπου 300 milliseconds, δηλαδή 0.3 δευτερόλεπτα. Το ίδιο ισχύει και για εύρος δέκα χιλιομέτρων (10km).



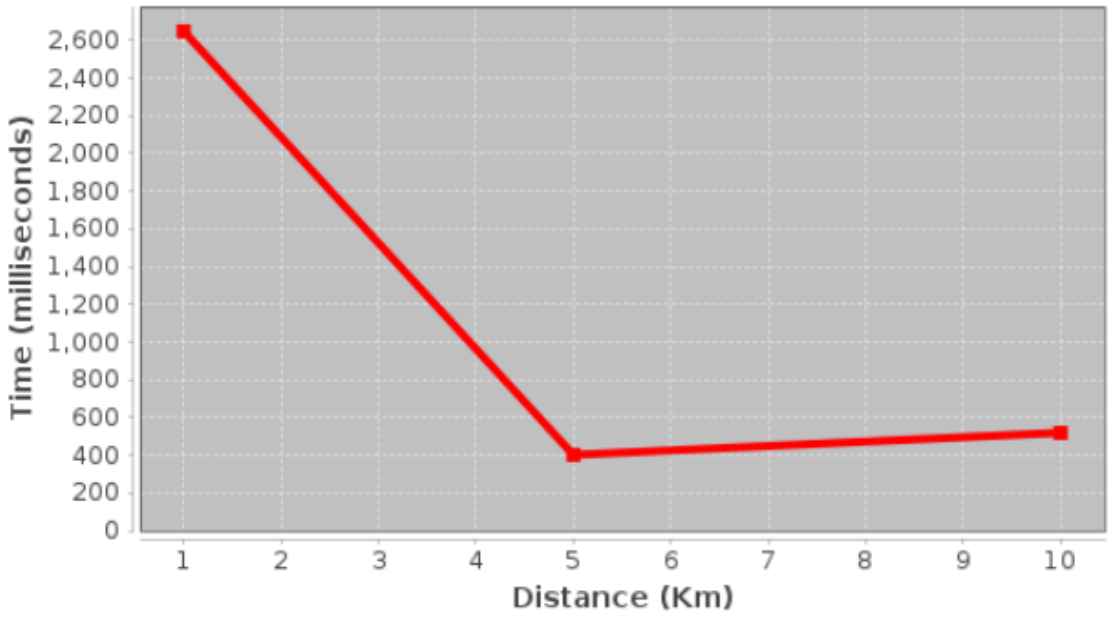
Εικόνα 21. Χρόνος απόκρισης ερωτήματος στην μέθοδο Grid File

8.2.2 Χρόνος απόκρισης του ερωτήματος εφαρμοσμένο στην μέθοδο αποθήκευσης Ordered Indexing

Στο παρακάτω διάγραμμα απεικονίζεται ο χρόνος που απαιτείται για την ολοκλήρωση ενός ερωτήματος που θέτει ο χρήστης προς το σύστημα, όπου τα δεδομένα καταχωρήθηκαν στην βάση με την μέθοδο του Ordered Indexing. Επισημαίνεται ότι ο χρόνος, αυτός περιλαμβάνει το χωρικό ερώτημα που τίθεται στα δεδομένα, δηλαδή την δημιουργία του παραθύρου (Window Query) και το Boolean ερώτημα που σχετίζεται με τις επιθυμίες του χρήστη ως προς τα προϊόντα προς πώληση που θέλει ο ίδιος. Επιπλέον, ο χρόνος περιλαμβάνει και τον χρόνο που απαιτεί η επιστροφή των αποτελεσμάτων στον χρήστη.

Όπως φαίνεται και παρακάτω, χρόνος που απαιτεί η εκτέλεση του ερωτήματος για εύρος αναζήτησης ενός χιλιομέτρου (1km) είναι περίπου 2.700 milliseconds, δηλαδή 2.7 δευτερόλεπτα.

Αντίθετα, με εύρος αναζήτησης τα πέντε χιλιόμετρα (5km), ο χρόνος απόκρισης είναι περίπου 400 milliseconds, δηλαδή 0.4 δευτερόλεπτα. Ενώ για εύρος δέκα χιλιομέτρων (10km) χρειάζεται περίπου 600 milliseconds, δηλαδή 0.6 δευτερόλεπτα.



Εικόνα 22. Χρόνος απόκρισης ερωτήματος στην μέθοδο Ordered Indexing

8.3 Πλήθος αποτελεσμάτων του ερωτήματος

Σε αυτήν την ενότητα θα παρουσιαστούν τα διαγράμματα, τα οποία θα αποτυπώνουν το πλήθος των αποτελεσμάτων, έπειτα της εκτέλεσης του ερωτήματος και των δύο μεθόδων, έτσι ώστε να επιστρέψει τα αποτελέσματα στον χρήστη.

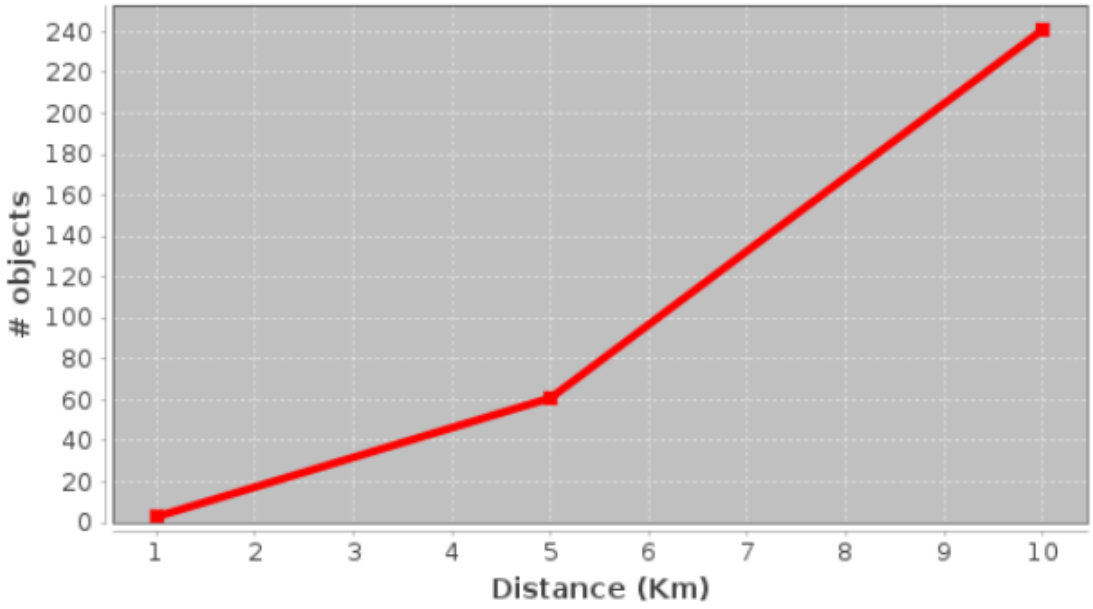
8.3.1 Πλήθος αποτελεσμάτων του ερωτήματος εφαρμοσμένο στην μέθοδο αποθήκευσης Grid File

Σε αυτήν την υποενότητα θα παρουσιαστεί το διάγραμμα που θα σχετίζεται με τον αριθμό των αποτελεσμάτων του ερωτήματος σε σχέση με το εύρος της αναζήτησης.

Όπως φαίνεται και από το παρακάτω διάγραμμα, για εύρος ενός χιλιομέτρου (1km) εμφανίζει λιγότερα από είκοσι (20) αποτελέσματα.

Αντίθετα, για εύρος πέντε χιλιομέτρων (5km) επιστρέφονται εξήντα (60) αποτελέσματα και για εύρος δέκα χιλιομέτρων (10km) επιστρέφονται σχεδόν διακόσια σαράντα (240) αποτελέσματα.

Τα ερωτήματα τέθηκαν στον σύστημα με το ίδιο σημείο αναφοράς του χρήστη και το ίδιο κείμενο. Επομένως, γίνεται αντιληπτό ότι όσο μεγαλύτερο είναι το εύρος τόσα περισσότερα αποτελέσματα επιστρέφονται.



Εικόνα 23. Πλήθος αποτελεσμάτων στην μέθοδο Grid File

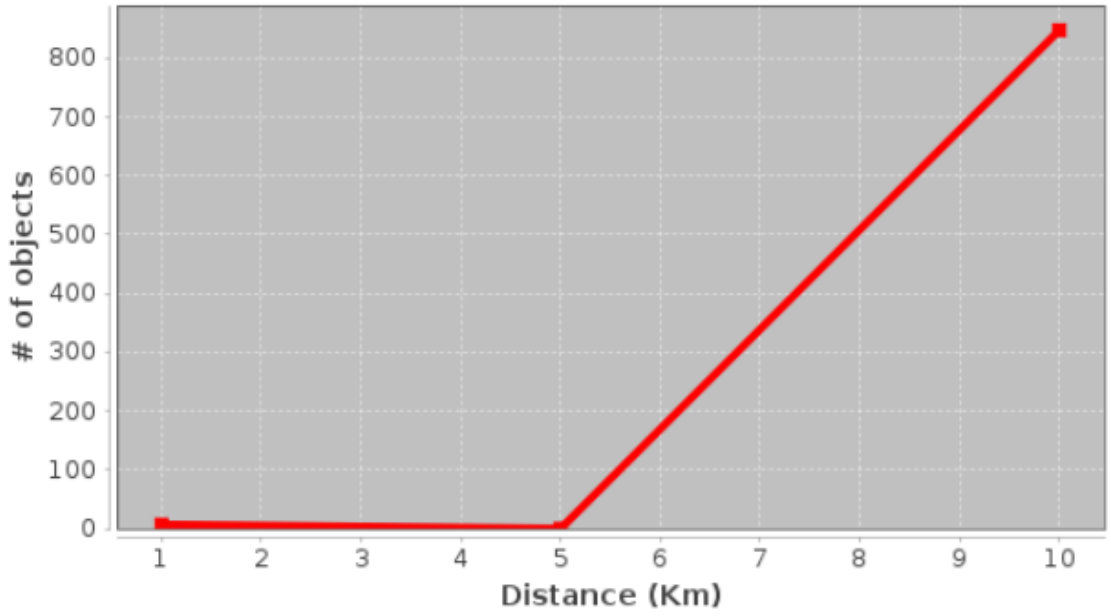
8.3.2 Πλήθος αποτελεσμάτων του ερωτήματος εφαρμοσμένο στην μέθοδο αποθήκευσης Ordered Indexing

Σε αυτήν την υποενότητα θα παρουσιαστεί το διάγραμμα που θα σχετίζεται με τον αριθμό των αποτελεσμάτων του ερωτήματος σε σχέση με το εύρος της αναζήτησης, εφαρμοσμένο στην μέθοδο Ordered Indexing.

Όπως φαίνεται και από το παρακάτω διάγραμμα, για εύρος ενός χιλιομέτρου (1km) και πέντε χιλιομέτρων (5km) εμφανίζει λιγότερα από πενήντα (50) αποτελέσματα.

Αντίθετα, για εύρος δέκα χιλιομέτρων (10km) επιστρέφονται περισσότερα από οχτακόσια (800) αποτελέσματα.

Τα ερωτήματα τέθηκαν στον σύστημα με το ίδιο σημείο αναφοράς του χρήστη και το ίδιο κείμενο. Επομένως, η εκτόξευση του αριθμού των αποτελεσμάτων στην Ordered Indexing προκαλεί μία ανησυχία ως προς την αξιοπιστία του. Αυτό συμβαίνει διότι η αναζήτηση γίνεται βάσει της τιμής Z-Order των εγγραφών, η οποία προκύπτει από τις συντεταγμένες των καταστημάτων. Ωστόσο, η τιμή Z-Order αποδίδει τιμές στις εγγραφές της βάσης μικρότερες ή και μεγαλύτερες από αυτές που ανταποκρίνονται στην πραγματικότητα. Αυτός είναι και ο λόγος δυσλειτουργίας του ερωτήματος στην συγκεκριμένη μέθοδο.



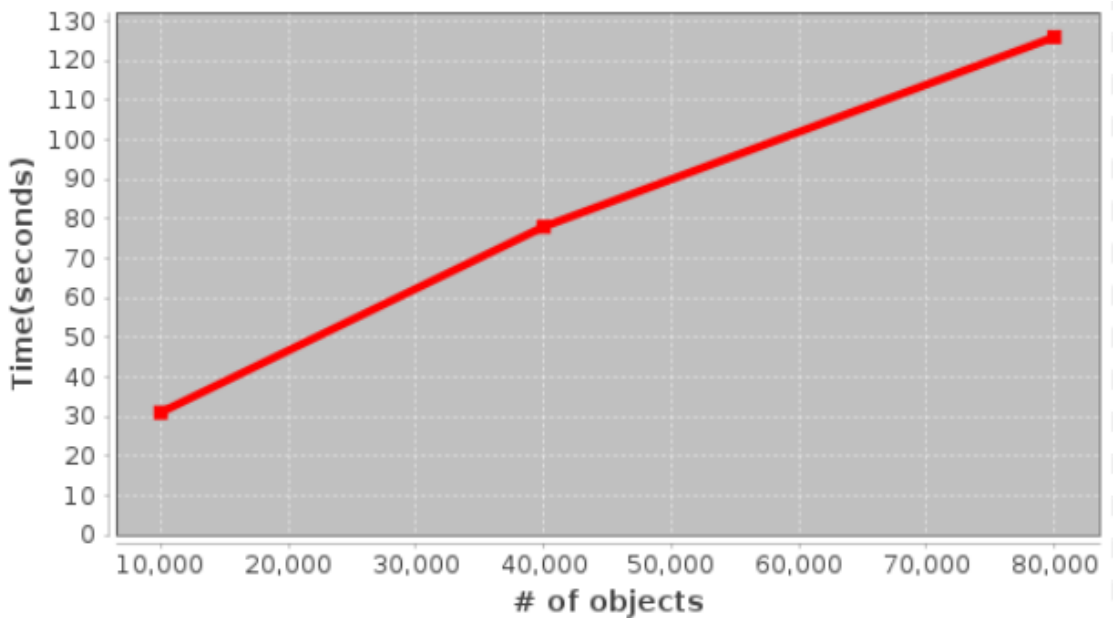
Εικόνα 24. Πλήθος αποτελεσμάτων στην μέθοδο Ordered Indexing

8.4 Αυξομειώσεις του αριθμού των πινάκων που απαιτούνται στην μέθοδο του Grid File

Σε αυτό το κεφάλαιο θα παρουσιαστούν αποτελέσματα με την μορφή διαγραμμάτων που απεικονίζουν τον χρόνο που απαιτείται για την εισαγωγή και την εκτέλεση του ερωτήματος προς την βάση δεδομένων.

8.4.1 Δημιουργία τιμής Hash με συντελεστή πέντε (5)

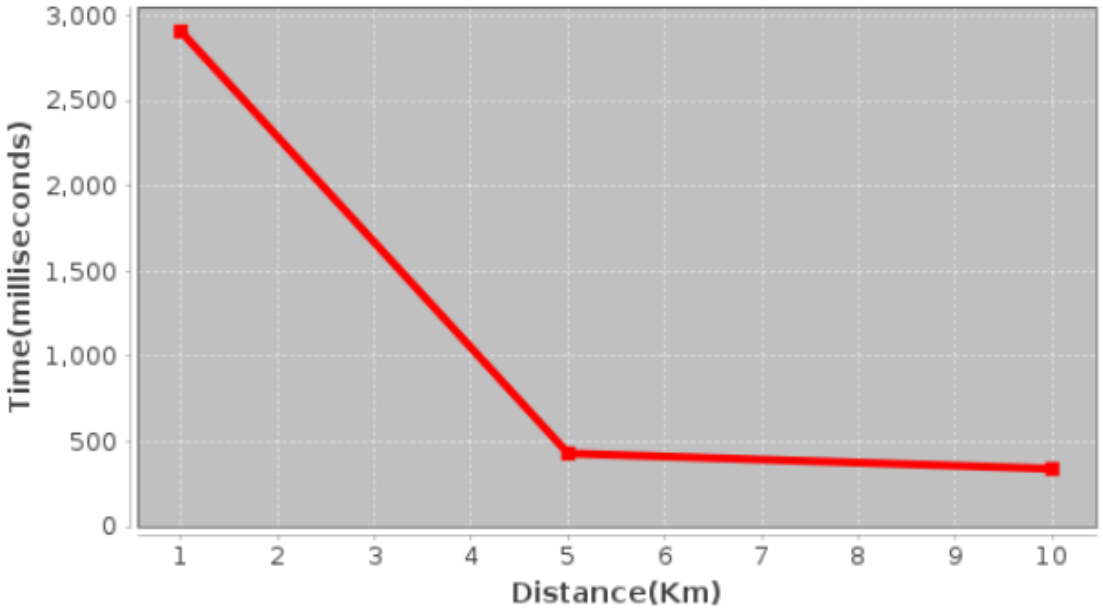
Σε αυτήν την ενότητα δημιουργούνται οι τιμές Hash, των εγγραφών, με την συντελεστή πέντε (5). Αυτό σημαίνει ότι, βάσει αυτού του συντελεστή θα καθοριστεί και το πλήθος των πινάκων που θα αποθηκευτούν στην βάση δεδομένων. Η διαδικασία, της δημιουργίας της Hash, έχει περιγραφεί σε προηγούμενη ενότητα (Βλέπε 7.2.2). Όπως γίνεται αντιληπτό, αν η τιμή Hash δημιουργηθεί με συντελεστή πέντε (5), τότε ο αριθμός των πινάκων στους οποίους θα αποθηκευτούν τα δεδομένα θα είναι αισθητά μεγαλύτερος. Στο παρακάτω διάγραμμα απεικονίζεται ο χρόνος που απαιτείται για την δημιουργία των τιμών Hash κάθε εγγραφής, την δημιουργία των πινάκων και την εισαγωγή των δεδομένων σε αυτούς. Όπως παρατηρείται, ο χρόνος που απαιτείται και για τις 78.981 είναι αρκετά μεγαλύτερος από τον χρόνο που παρουσιάζει ο συντελεστής 10, που χρησιμοποιήθηκε στην ενότητα 8.1.1. Αυτό συμβαίνει λόγω της αύξηση του αριθμού των πινάκων χρειάζεται περισσότερο χρονικό διάστημα για την δημιουργία τους και περαιτέρω αναζήτηση των ονομάτων τους για την εισαγωγή των στοιχείων σε αυτούς.



Εικόνα 25. Χρόνος εισαγωγής δεδομένων στην βάση, με την μέθοδο Grid File και δημιουργία τιμής Hash με συντελεστή πέντε

Επιπλέον, ο χρόνος που απαιτείται για την εκτέλεση ενός ερωτήματος των δεδομένων στους πίνακες της HBase φαίνεται να αυξάνεται ελάχιστα σε σχέση με τον

συντελεστή δέκα (10) της υποενότητας 8.2.1 και αυτό παρατηρείται από το παρακάτω διάγραμμα.

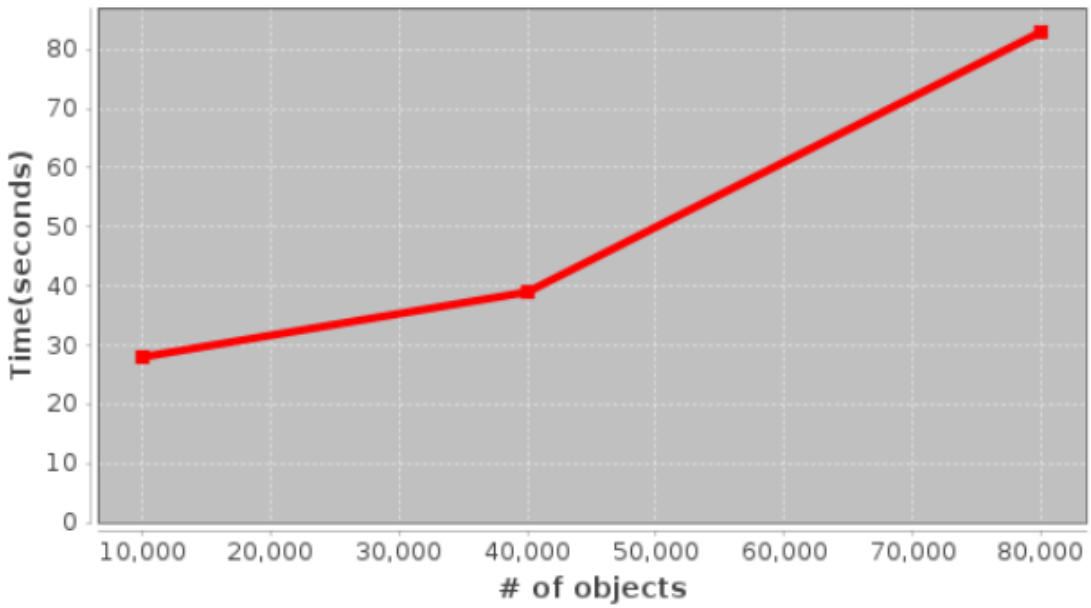


Εικόνα 26. Χρόνος απόκρισης ενός ερωτήματος στην μέθοδο Grid File και δημιουργία τιμής Hash με συντελεστή πέντε

Ωστόσο, αξίζει να σημειωθεί ότι τα αποτελέσματα που επιστρέφονται είναι τα ίδια που επιστρέφονται, με συντελεστή δέκα (10) του Hash.

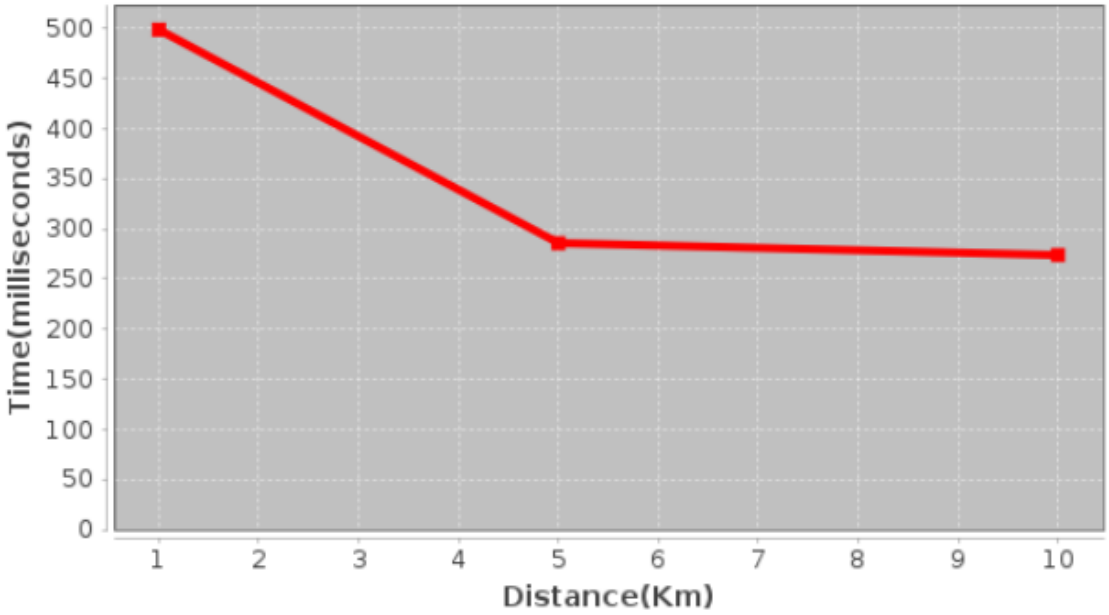
8.4.2 Δημιουργία τιμής Hash με συντελεστή είκοσι (20)

Σε αυτήν την ενότητα δημιουργούνται οι τιμές Hash, των εγγραφών, με την συντελεστή είκοσι (20). Αυτό σημαίνει ότι, βάσει αυτού του συντελεστή θα καθοριστεί και το πλήθος των πινάκων που θα αποθηκευτούν στην βάση δεδομένων. Όπως γίνεται αντιληπτό, αν η τιμή Hash δημιουργηθεί με συντελεστή είκοσι (20), τότε ο αριθμός των πινάκων στους οποίους θα αποθηκευτούν τα δεδομένα θα είναι αισθητά μικρότερος. Όπως φαίνεται και από το παρακάτω διάγραμμα, ο χρόνος δημιουργίας των τιμών Hash, η δημιουργία των πινάκων και η εισαγωγή των δεδομένων σε αυτούς, φαίνεται να μην επηρεάζεται σχεδόν καθόλου σε σχέση με τον συντελεστή δέκα (10) της Hash (Βλέπε 8.1.1).



Εικόνα 27. Χρόνος εισαγωγής δεδομένων στην βάση, με την μέθοδο Grid File και δημιουργία τιμής Hash με συντελεστή είκοσι

Αυτό που είναι άξιο σημασίας είναι ότι ο χρόνο απόκρισης του ερωτήματος μειώνεται αρκετά, όπως φαίνεται και στο παρακάτω διάγραμμα.

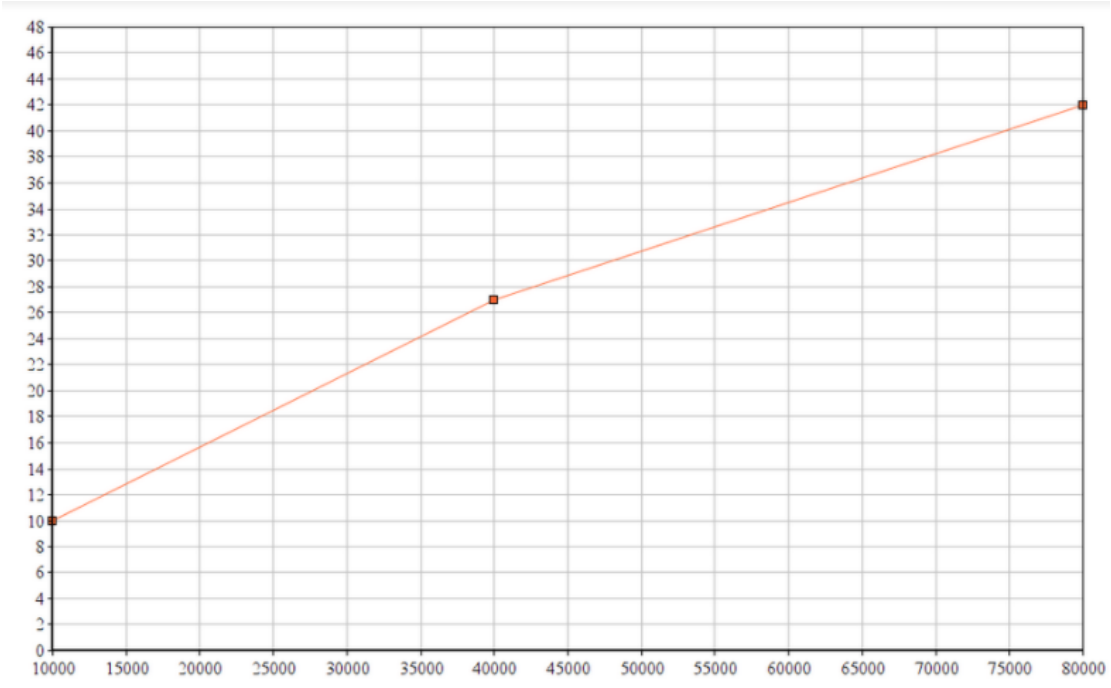


Εικόνα 28. Χρόνος απόκρισης ενός ερωτήματος στην μέθοδο Grid File και δημιουργία τιμής Hash με συντελεστή είκοσι

Ωστόσο, αξίζει να σημειωθεί ότι τα αποτελέσματα που επιστρέφονται είναι τα ίδια στην περίπτωση με συντελεστή δέκα (10) του Hash.

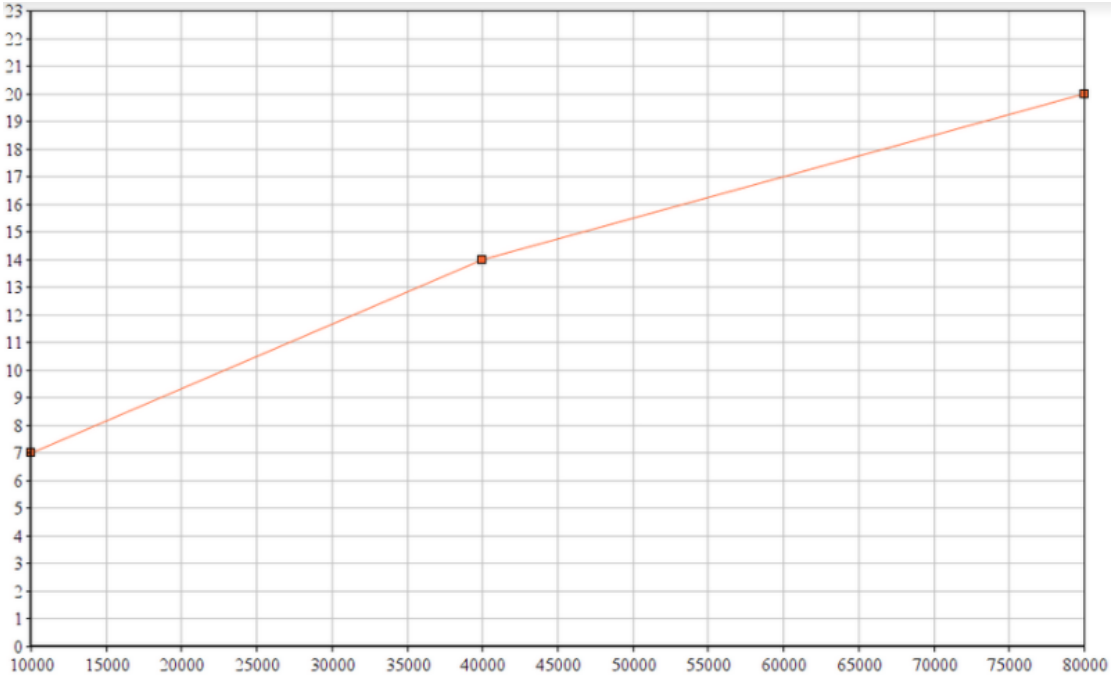
8.4.3 Απαιτούμενος αριθμός πινάκων για συντελεστή 5, 10 και 20 της τιμής Hash

Για συντελεστή 5 της τιμής Hash απαιτείται η δημιουργία 42 πινάκων όπως φαίνεται και στο παρακάτω διάγραμμα.



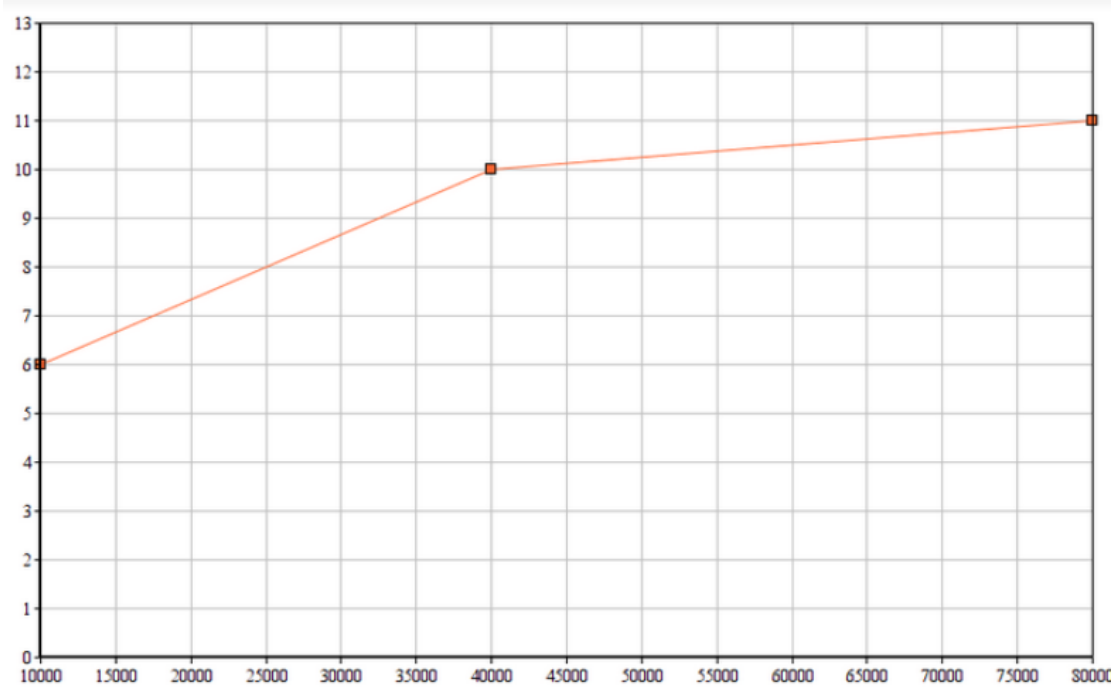
Εικόνα 29. Απαιτούμενος αριθμός πινάκων για συντελεστή 5

Για συντελεστή 10 της τιμής Hash απαιτείται η δημιουργία 20 πινάκων όπως φαίνεται και στο παρακάτω διάγραμμα.



Εικόνα 30. Απαιτούμενος αριθμός πινάκων για συντελεστή 10

Ενώ για συντελεστή 20 της τιμής Hash απαιτείται η δημιουργία 11 πινάκων όπως φαίνεται και στο παρακάτω διάγραμμα.



Εικόνα 31. Απαιτούμενος αριθμός πινάκων για συντελεστή 20

9

Σύγκριση Grid File και Ordered Indexing μεθόδων

Λαμβάνοντας υπόψιν όλα τα παραπάνω, σε αυτό το κεφάλαιο θα παρουσιαστεί η σύγκριση των δύο μεθόδων αποθήκευσης, βάσει της πειραματικής μελέτης που πραγματοποιήθηκε.

9.1 Σύγκριση χρόνου εισαγωγής και επεξεργασίας των δεδομένων των δύο μεθόδων

Η επεξεργασία των δεδομένων στην μέθοδο Grid File, αφορά την δημιουργία της τιμής Hash κάθε εγγραφής και της τιμής Z-Order η οποία αποδίδει την μοναδικότητα της κάθε εγγραφής. Εν συνεχεία με την τιμή Hash κάθε εγγραφής καθορίζεται ο αριθμός των πινάκων που θα δημιουργηθούν και θα εκχωρηθούν οι εγγραφές σε αυτούς. Επομένως πρόκειται για μια αρκετά βέλτιστη επεξεργασία και εισαγωγή των δεομένων.

Αντίθετα, στην μέθοδο του Ordered Indexing η επεξεργασία των δεδομένων, αφορά την δημιουργία του μοναδικού αναγνωριστικού Z-Order κάθε εγγραφής, την ταξινόμηση, την εισαγωγή στους πίνακες της HBase και η αποθήκευση των μέτα-δεδομένων των πινάκων. Όσον αφορά την ταξινόμηση των δεδομένων, γίνεται αντιληπτό ότι πρόκειται για μία διαδικασία που καθυστερεί αρκετά την επεξεργασία και την αποθήκευση των δεδομένων στην βάση.

Παρατηρώντας τα αποτελέσματα της πειραματικής μελέτης που αφορούν την εισαγωγή των δεδομένων (Βλέπε 8.1) στους πίνακες της HBase, διακρίνεται ότι η μέθοδος του Ordered Indexing λόγω της περαιτέρω επεξεργασίας των δεδομένων, που πραγματοποιεί, απαιτεί τον διπλάσιο χρόνο εισαγωγής των στοιχείων στην βάση δεδομένων, σε σύγκριση με την μέθοδο του Grid File.

9.2 Σύγκριση χρόνου απόκρισης του ερωτήματος

Όσον αφορά τον χρόνο απόκρισης των ερωτημάτων, από την πειραματική μελέτη (Βλέπε 8.2), που τίθενται στην βάση δεδομένων, είτε έχει δημιουργηθεί με την μέθοδο Grid File είτε με την μέθοδο Ordered Indexing δεν παρατηρείται κάποια

μεγάλη διαφορά στους χρόνους τους. Ωστόσο, η μέθοδος Grid File εμφανίζει ελάχιστα βελτιωμένο χρόνο απόκρισης σε συγκρίσει της μεθόδου Ordered Indexing.

9.3 Σύγκριση πλήθος αποτελεσμάτων των δύο μεθόδων

Βάσει των αποτελεσμάτων που προκύπτουν από την πειραματική μελέτη του πλήθους των αποτελεσμάτων (Βλέπε 8.3), γίνεται αντιληπτό ότι η μέθοδος του Grid File επιστρέφει καλύτερα αποτελέσματα σε σχέση με την μέθοδο Ordered Indexing. Λαμβάνοντας υπόψιν το παραπάνω, η μέθοδος Grid File είναι περισσότερη αποτελεσματική έναντι της Ordered Indexing. Αυτό συμβαίνει διότι η αναζήτηση σε κάθε μέθοδο εφαρμόζεται διαφορετικά και θα περιγραφεί στην παρακάτω ενότητα.

9.4 Σύγκριση εφαρμογής της αναζήτησης των δύο μεθόδων

Όπως έχει αναφερθεί και σε προηγούμενη ενότητα, η αναζήτηση σε κάθε μέθοδο εφαρμόζεται διαφορετικά. Πιο συγκεκριμένα, η αναζήτηση στη μέθοδο Grid File γίνεται βάσει των συντεταγμένων της κάθε εγγραφής, ενώ στην μέθοδο του Ordered Indexing γίνεται βάσει της τιμής Z-Order.

Επισημαίνεται ότι η τιμή Z-Order χρησιμοποιείται και στις δύο μεθόδους, προκειμένου να αποδοθεί σε κάθε εγγραφή ένα μοναδικό αναγνωριστικό που προκύπτει από τις τιμές των συντεταγμένων τους. Επιπλέον, η τιμή Z-Order μπορεί να αποδώσει μία τιμή σε κάθε εγγραφή χωρίς όμως να εγγυάται ότι αυτή η τιμή ανταποκρίνεται επαρκώς στην πραγματικότητα.

Για παράδειγμα, μπορεί να αποδώσει μία τιμή μικρότερη σε ένα σημείο που είναι μεγαλύτερο από κάποιο άλλο και αντίστροφα, μπορεί να αποδώσει μία τιμή μεγαλύτερη για ένα σημείο που στην πραγματικότητα είναι μικρότερο από κάποιο άλλο. Το αντίστοιχο συνέβη και στην περίπτωση του Ordered Indexing, με αποτέλεσμα η αναζήτηση να αναζητά τιμές που ανήκουν εντός του ορίου του Window Query και να επιστρέφονται περισσότερα και λανθασμένα αποτελέσματα προς στον χρήστη.

9.5 Σύγκριση υλοποίησης των δύο μεθόδων

Όπως προκύπτει από την υλοποίηση και των δύο μεθόδων, η ανάπτυξη της Grid File μεθόδου φαίνεται περισσότερο κατανοητή και απλή. Αντίθετα η ανάπτυξη της Ordered Indexing μεθόδου δίνει την αίσθηση μία περίπλοκης, σύνθετης και αρκετά χρονοβόρα διαδικασία. Αυτό συμβαίνει, διότι στην Ordered Indexing θα πρέπει ληφθούν υπόψιν αρκετές διαδικασίες για την ανάπτυξη της μεθόδου. Σε αντίθεση, η Grid File μέθοδος χρειάζεται λιγότερο χρόνο υλοποίησης και πρόκειται για μία αρκετά διακριτή μέθοδος.

9.6 Σύγκριση συντελεστών της τιμής Hash στο Grid File

Όπως παρατηρείται και από το κεφάλαιο της πειραματικής μελέτης όσο περισσότερο μειώνεται ο συντελεστής της τιμής Hash τόσο περισσότεροι πίνακες δημιουργούνται στην βάση δεδομένων και συνάμα αυξάνεται ο απαιτούμενος χρόνος εισαγωγής των δεδομένων. Επιπλέον, παρατηρήθηκε ότι όσο περισσότεροι είναι οι πίνακες τόσο περισσότερο μειώνεται και ο χρόνος απόκρισης του ερωτήματος στην βάση δεδομένων. Ωστόσο, ο συντελεστής της τιμής Hash αλλά και το πλήθος των πινάκων δεν φάνηκαν να επηρεάζουν το πλήθος των αποτελεσμάτων.

10

Συμπεράσματα

Βάσει όλων των προαναφερθέντων, γίνεται αντιληπτό ότι τα μεγάλα δεδομένα αποτελούν ένα αναπόσπαστο κομμάτι της σημερινής σύγχρονης εποχής λόγω της ανάγκης για ψηφιοποίηση της πληροφορίες. Επιπλέον, τα μεγάλου όγκου δεδομένα παρέχουν και μία πληθώρα πληροφοριών που είναι χρήσιμα τόσο για τους χρήστες του διαδικτύου όσο και για τις εταιρίες και οργανισμούς.

Το αντικείμενο της παρούσας εργασίας επικεντρώνεται κυρίως στην αποθήκευση και διαχείριση των χωρικών δεδομένων. Τα χωρικά δεδομένα, όπως έχει αναφερθεί είναι εντελώς χρήσιμα και πολύτιμα για όλους, διότι υποβοηθούν και βελτιώνουν την ποιότητα ζωής των ανθρώπων. Τα δεδομένα αυτά λόγω του όγκου τους κατατάσσονται στην κατηγορία των μεγάλων δεδομένων, επομένως αντιμετωπίζουν τα ίδια προβλήματα που παρουσιάζουν και τα Big Data.

Τα μεγάλα δεδομένα απαιτούν ολοένα και περισσότερο αποθηκευτικό χώρο, όμως το κύριο πρόβλημα τους είναι η δυσκολία που παρουσιάζουν ως προς την επεξεργασία τους, με τα συμβατικά και παραδοσιακά συστήματα διαχείρισης. Αυτές οι δυσκολίες αποσκοπούν στην χρονική καθυστέρηση που επιφέρει η επεξεργασία των δεδομένων. Επομένως, γίνεται κατανοητή η ανάπτυξη και ύπαρξη αντίστοιχων μεθόδων και συστημάτων διαχείρισης μεγάλου όγκου δεδομένων, σχεδιασμένες κατάλληλα με σκοπό να επιταχύνονται οι διαδικασίες επεξεργασίας.

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε το εργαλείο διαχείρισης της HBase, το οποίο χάρις την αρχιτεκτονική HDFS που υιοθετεί ανταποκρίνεται επάξια των προσδοκιών. Πιο συγκεκριμένα, λόγω της κατανεμημένης αποθήκευσης των δεδομένων, επηρεάζει θετικά την απόδοση της επεξεργασία των δεδομένων. Επιπλέον, η HBase σε συνδυασμό με αντίστοιχες μεθόδους (όπως το Grid File) ενισχύει σε σημαντικό βαθμό την απόδοση επεξεργασίας των δεδομένων και την απόδοση των εκτελούντων ερωτημάτων, προς την βάση, που χρησιμοποιούνται για την εξαγωγή αποτελεσμάτων. Επιπρόσθετα, πρόκειται για ένα πλήρως χρήσιμο εργαλείο με εύκολα κατανοητή λειτουργία.

Επιπλέον, βάσει των πεπραγμένων συμπεραίνεται ότι η μέθοδος Grid File υπερτερεί έναντι της μεθόδου Ordered Indexing, τόσο στον χρόνο που απαιτείται για την επεξεργασία και την εισαγωγή των δεδομένων στους πίνακες της HBase όσο και στους χρόνους απόκρισης των ερωτημάτων. Επομένως η μέθοδος Grid File, λόγω του τρόπου δημιουργίας των πινάκων και η εισαγωγή των δεδομένων σε αυτούς, κρίνεται

ως μία αρκετά αποδοτική μέθοδος, η οποία επεξεργάζεται, αποθηκεύει και ανακτά τα δεδομένα από την βάση σε σύντομο χρονικό διάστημα.

Επιπρόσθετα, όσον αφορά την αναζήτηση των δύο μεθόδων, η Grid File δίνει αποτελέσματα που σχετίζονται απόλυτα με ερώτημα που τίθεται στην βάση σε σύγκριση της μεθόδου Ordered Indexing, κάνοντάς την αποτελεσματική. Αυτό συμβαίνει λόγω της διαφορετικής προσέγγισης των στοιχείων. Στην Grid File μέθοδο, η αναζήτηση βασίζεται στις συγκρίσεις των γεωγραφικών συντεταγμένων των δεδομένων που βρίσκονται αποθηκευμένες στην βάση, με τις γεωγραφικές συντεταγμένες του ερωτήματος ενώ στην μέθοδο Ordered Indexing βασίζεται στις συγκρίσεις των τιμών Z-Order κάθε εγγραφής στην βάση με την τιμή Z-Order του ερωτήματος. Επομένως, η τιμή Z-Order που δημιουργείται, μπορεί να χρησιμοποιηθεί για να αποδώσει ένα μοναδικό αναγνωριστικό στην κάθε εγγραφή αλλά κρίνεται ως αναξιόπιστη ως προς την ταξινόμηση και την αναζήτηση, διότι οι τιμές αυτές δεν ανταποκρίνονται στην πραγματικότητα και δεν μπορούν να αντιστοιχίσουν ρεαλιστικά ένα σημείο στον χάρτη.

Επιπλέον, η αναζήτηση των γεωγραφικών συντεταγμένων σε συνδυασμό με το κείμενο (keywords), που αφορά τις επιθυμίες του χρήστη, κάνει τα αποτελέσματα περισσότερο ακριβή αποδίδοντας αποτελεσματικά.

Συμπερασματικά, η Grid File πρόκειται για μία αποτελεσματική και αποδοτική μέθοδο και μπορεί να χρησιμοποιηθεί για την διαχείριση δεδομένων μεγάλου όγκου. Επίσης, στην μέθοδο Grid File, ο συνδυασμός των χωρικών ερωτημάτων με κείμενο εκτοξεύει την αποτελεσματικότητα του ερωτήματος, δίνοντας έγκυρα αποτελέσματα και κρίνοντάς το ως αξιόπιστο. Τέλος, είναι σημαντικό να αναφερθεί ότι ο συντελεστής που επιλέγεται για την δημιουργία της Hash τιμής των δεδομένων και καθορίζει το πλήθος των πινάκων, στην μέθοδο Grid File, δεν θα πρέπει να είναι αρκετά μεγάλος ή αρκετά μικρός διότι δημιουργούνται πίνακες που θα τελικά υπερφορτώνονται καθυστερώντας στην χειρότερη περίπτωση την αναζήτηση ή αυξάνεται κατά πολύ ο χρόνος εισαγωγής των δεδομένων στην βάση, αντίστοιχα.

Βιβλιογραφία

[1] Belussi, A., Catania, B., Clementini, E., & Ferrari Spatial Data on the Web: Issues and Challenges

[2] Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. Mobile Networks and Applications, 19(2), 171–209

[3] Jin, J., An, N., & Sivasubramaniam, A. (n.d.). Analyzing range queries on spatial data. Proceedings of 16th International Conference on Data Engineering

[4] Azhar Susanto, Meiryani. Database Management System. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 8, ISSUE 06, JUNE2019

[5] Mehul Nalin Vora. (2011). Hadoop-HBase for large-scale data. Proceedings of 2011 International Conference on Computer Science and Network Technology

[6] Jing Han, Haihong E, Guan Le, & Jian Du. (2011). Survey on NoSQL database. 2011 6th International Conference on Pervasive Computing and Applications

[7] Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)

[8] Taylor, R. C. (2010). An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC Bioinformatics, 11(Suppl 12), S1

[9] Jianling Sun, & Qiang Jin. (2010). Scalable RDF store based on HBase and MapReduce. 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)

[10] Morton curve segments produce no more than twodistinct face-connected subdomains

[11] Yongqiang Zou, Jia Liu, Shicai Wang, Li Zha, and Zhiwei Xu. CCIndex: AComplemental Clustering Index onDistributed Ordered Tables for Multi-dimensional Range Queries

[12] Skiena, S. S. (2012). Sorting and Searching. The Algorithm Design Manual, 103–144

[13] Nievergelt, J., Hinterberger, H., & Sevcik, K. C. (1981). The grid file: An adaptable, symmetric multi-key file structure. Lecture Notes in Computer Science, 236–251

[14] D Papadias, Y Tao, J Zhang, Q Shen, N Mamoulis. Novel Forms of Nearest Neighbor Queries in Spatio-Temporal Applications

[15] Porkaew, K., & Chakrabarti, K. (1999). Query refinement for multimedia similarity retrieval in MARS. Proceedings of the Seventh ACM International Conference on Multimedia (Part 1) - MULTIMEDIA '99

[16] Jin, J., An, N., & Sivasubramaniam, A. (n.d.). Analyzing range queries on spatial data. Proceedings of 16th International Conference on Data Engineering

[17] Mohammad R. Kolahdouzan and Cyrus Shahabi. Continuous K Nearest Neighbor Queries in SpatialNetwork Databases

[18] Jagadish, H. V., Ooi, B. C., Tan, K.-L., Yu, C., & Zhang, R. (2005). iDistance. ACM Transactions on Database Systems

[19] Ilyas, I. F., Beskales, G., & Soliman, M. A. (2008). A survey of top-kquery processing techniques in relational database systems. ACM Computing Surveys, 40(4), 1–58

[20] Melita Kennedy. "Equirectangular" p61. Understanding Map ProjectionEquirectangular projection

[21] Kevin Chen-Chuan Chang, Hector Garcia-Molina, Andreas Paepcke Boolean Query Mapping Across Heterogeneous Information Sources