

# Επεξεργασία Μεγάλων Δεδομένων: Τεχνικές και Εργαλεία

Ιουν. 2024-25

Αικατερίνη Μενύχτα  
ME2421Χρήστος Δουλκερίδης  
Επιβλέπων Καθηγητής

## ΠΕΡΙΛΗΨΗ - ΕΙΣΑΓΩΓΗ

Η παρούσα εργασία εξετάζει τη χρήση παράλληλης επεξεργασίας για τη βελτιστοποίηση της απόδοσης γεωχωρικών συναρτήσεων σύζευξης με τη χρήση του πλαισίου Apache Spark. Συγκεκριμένα, χρησιμοποιήθηκαν δύο μεγάλα σύνολα γεωγραφικών δεδομένων (RAILS και AREALM), αποτελούμενα από 3,184,948 και 3,848,971 εγγραφές αντίστοιχα, για την υλοποίηση δύο βασικών χωρικών ερωτημάτων (Query A και Query B).

Στο πρώτο ερώτημα (Query A), εντοπίζονται όλα τα ζεύγη αντικειμένων από τα δύο σύνολα, των οποίων η ευκλείδεια απόσταση είναι μικρότερη ή ίση με ένα δεδομένο κατώφλι απόστασης ( $\epsilon$ ). Στο δεύτερο ερώτημα (Query B), εντοπίζονται οι εγγραφές του συνόλου R που έχουν τουλάχιστον k εγγραφές από το σύνολο S εντός της απόστασης  $\epsilon$ . Η λύση που υλοποιήθηκε αξιοποιεί τη grid-based (MBB-based) χωρική διαμέριση για αποδοτική κατανομή των δεδομένων στα partitions του Spark και περιλαμβάνει κατάλληλες στρατηγικές όπως επεκτάσεις γειτονικών κελιών (neighbor-cell expansions) και φιλτράρισμα αποτελεσμάτων με βάση τις ευκλείδειες αποστάσεις.

Πραγματοποιήθηκαν εκτενή πειράματα αξιολόγησης με διάφορες παραμέτρους ( $\epsilon$ , k, αριθμός partitions, executors και cores), τα οποία ανέδειξαν τη σημαντική επίδραση αυτών των παραμέτρων στην απόδοση των ερωτημάτων. Ειδικότερα, αποδείχθηκε ότι η επιλογή κατάλληλων παραμέτρων μπορεί να βελτιώσει σημαντικά το χρόνο εκτέλεσης. Η βέλτιστη απόδοση επιτεύχθηκε με τη χρήση 16 partitions, 4 executors και 1 πυρήνα ανά executor. Επιπλέον, παρατηρήθηκε ασύμμετρη συμπεριφορά των αποτελεσμάτων όταν αλλάζουν οι ρόλοι των συνόλων R και S, λόγω της διαφορετικής κατανομής των δεδομένων.

Συνολικά, τα αποτελέσματα καταδεικνύουν ότι η συγκεκριμένη υλοποίηση παρέχει αποτελεσματική και επεκτάσιμη λύση για την εκτέλεση χωρικών συναρτήσεων σύζευξης σε μεγάλα σύνολα δεδομένων.

Επιτρέπεται η δημιουργία ψηφιακών ή έντυπων αντιγράφων όλου ή μέρους αυτού του έργου για προσωπική ή εκπαιδευτική χρήση, χωρίς χρέωση, υπό την προϋπόθεση ότι τα αντίγραφα δεν διανέμονται για κέρδος ή εμπορικό όφελος και φέρουν αυτήν την ειδοποίηση καθώς και την πλήρη αναφορά στην πρώτη σελίδα.

Τα πνευματικά δικαιώματα για στοιχεία αυτού του έργου που ανήκουν σε τρίτους εκτός από τον/τους συγγραφέα(-είς) πρέπει να τιμώνται. Η αφαίρεση ή συνοπτική αναπαραγωγή με αναφορά επιτρέπεται.

Για οποιαδήποτε άλλη χρήση, όπως αντιγραφή, αναδημοσίευση, ανάρτηση σε διακομιστές ή αναδιανομή σε λίστες, απαιτείται προηγούμενη συγκεκριμένη άδεια και/ή χρέωση. Αιτήσεις για άδειες πρέπει να απευθύνονται στο [permissions@acm.org](mailto:permissions@acm.org).

Απαλλακτική Εργασία, Ιουν. 2024-25, Πανεπιστήμιο Πειραιώς

© 2025 Πνευματικά δικαιώματα διατηρούνται από τον/τους ιδιοκτήτη(-ες)/συγγραφέα(-είς).

Τα δικαιώματα δημοσίευσης έχουν εκχωρηθεί στην ACM.

## ΣΗΜΑΝΤΙΚΕΣ ΕΝΝΟΙΕΣ

- Θεωρία υπολογιστών → Κατανεμημένοι υπολογισμοί και παραλληλία
- Πληροφοριακά συστήματα → Γεωχωρικά συστήματα
- Συστήματα διαχείρισης βάσεων δεδομένων → Κατανεμημένα DBMS
- Υπολογιστικά μοντέλα → MapReduce / Spark

## ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ

Apache Spark, Χωρική Σύζευξη, Κατανεμημένα Συστήματα, Γεωχωρικά Δεδομένα, Παράλληλη Επεξεργασία, Grid Partitioning, Adaptive Replication, Big Data, Κλιμακωσιμότητα

Μορφή Αναφοράς ACM:

Καθηγητής: Κ. Δουλκερίδης, Αικατερίνη Μενύχτα. 2025. Επεξεργασία Μεγάλων Δεδομένων: Τεχνικές και Εργαλεία: Ιουν. 2024-25.

In Proceedings of Επεξεργασία Μεγάλων Δεδομένων: Τεχνικές και Εργαλεία (ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ). ACM, Νέα Υόρκη, NY, ΗΠΑ, 11 σελίδες.

## 1 Υπολογιστικοί Πόροι και Περιορισμοί

Η υλοποίηση της εργασίας έγινε σε ένα σύμπλεγμα (cluster) τεσσάρων εικονικών μηχανών (VMs), το οποίο δημιουργήθηκε στην πλατφόρμα Okeanos [6]. Οι υπολογιστικοί πόροι διαμορφώθηκαν σύμφωνα με τις προδιαγραφές της πλατφόρμας και τις ανάγκες της εφαρμογής ως εξής:

- Αριθμός μηχανών: 4 (node1, node2, node3, node4)
- Λειτουργικό σύστημα: Ubuntu Server 16.04.3 LTS
- Επεξεργαστική ισχύς (CPU): 4 πυρήνες ανά μηχανήμα
- Μνήμη RAM: 6 GB ανά μηχανήμα
- Χώρος αποθήκευσης (HDD): 10 GB ανά μηχανήμα

Η διαδικασία δημιουργίας του cluster περιλάμβανε λεπτομερείς ρυθμίσεις για τη δικτύωση των μηχανών, τη διαμόρφωση πρόσβασης μέσω SSH χωρίς χρήση κωδικού πρόσβασης, καθώς και την εγκατάσταση όλων των απαραίτητων εργαλείων λογισμικού.

Οι περιορισμοί που προέκυψαν κατά τη διαδικασία της υλοποίησης και των πειραμάτων ήταν οι ακόλουθοι:

- Περιορισμένη διαθέσιμη μνήμη: Παρότι η διαμόρφωση του cluster ήταν επαρκής για την εκτέλεση των πειραμάτων της εργασίας, η σχετικά μικρή ποσότητα διαθέσιμης μνήμης ανά μηχανήμα επέβαλε αυστηρούς περιορισμούς στο μέγεθος των ενδιάμεσων αποτελεσμάτων που μπορούσαν να διατηρηθούν στη μνήμη κατά την εκτέλεση. Σύνθετα χωρικά και χωροχρονικά ερωτήματα
- Υπολογιστική ισχύς και χρόνος εκτέλεσης: Λόγω της περιορισμένης επεξεργαστικής ισχύος, απαιτήθηκε λεπτομερής βελτιστοποίηση των αλγορίθμων και της κατανομής των δεδομένων, ώστε να αποφεύγεται η υπερβολική υπολογιστική επιβάρυνση των κόμβων. Ευελιξία στη μοντελοποίηση και αποθήκευση ημιδομημένων δεδομένων
- Διαχειριστικοί περιορισμοί της πλατφόρμας: Η πλατφόρμα Okeanos έχει περιορισμούς όπως η πιθανή διακοπή σύνδεσης ή διαθεσιμότητας πόρων, γεγονός που απαιτούσε συνεχή παρακολούθηση και συχνές

επανεκκινήσεις των μηχανών κατά τη διάρκεια των πειραμάτων.

Παρά τους παραπάνω περιορισμούς, η αρχιτεκτονική και οι επιλεγμένες τεχνικές επέτρεψαν την αποτελεσματική υλοποίηση και αξιολόγηση των αλγορίθμων, αναδεικνύοντας την επάρκεια της προσέγγισης για τη διαχείριση και ανάλυση μεγάλων γεωχωρικών δεδομένων στο πλαίσιο του Apache Spark.

## 2 ΠΕΡΙΓΡΑΦΗ ΥΛΟΠΟΙΗΣΗΣ ΣΥΜΦΩΝΑ ΜΕ ΤΕΚΜΗΡΙΩΜΕΝΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ

Η υλοποίηση της παράλληλης επεξεργασίας των γεωχωρικών δεδομένων βασίζεται στο πλαίσιο Apache Spark και εφαρμόζει στρατηγικές χωρικής κατάτμησης και γειτονικής επέκτασης δεδομένων για αποτελεσματική επίλυση των ζητούμενων ερωτημάτων (Query A και Query B).

### 2.1 Φόρτωση και Προετοιμασία Δεδομένων

Τα δεδομένα από τα σύνολα R (RAILS) και S (AREALM) φορτώνονται ως DataFrames στο Spark, με συγκεκριμένα σχήματα που περιλαμβάνουν αναγνωριστικά αντικείμενων (id) και γεωγραφικές συντεταγμένες (x, y). Τα δεδομένα κατανέμονται σε partitions για αποτελεσματική παράλληλη επεξεργασία.

### 2.2 Χωρική Κατάτμηση μέσω Grid Partitioning

Για να περιοριστεί ο αριθμός των συγκρίσεων και να βελτιστοποιηθεί η απόδοση της επεξεργασίας, χρησιμοποιείται ένα σύστημα ομοιόμορφης χωρικής κατάτμησης του γεωγραφικού χώρου. Ο χώρος διαχωρίζεται σε κελιά (grid cells), με το μέγεθος κάθε κελιού να καθορίζεται με βάση την τιμή της παραμέτρου απόστασης  $\epsilon$  (epsilon). Τα δεδομένα του κάθε συνόλου κατανέμονται σε αυτά τα κελιά με στρογγυλοποίηση των συντεταγμένων σε κατάλληλη ακρίβεια για αποφυγή σφαλμάτων λόγω δεκαδικών ψηφίων.

Η επιλογή αυτής της μεθόδου κατάτμησης εμπνέεται από καθιερωμένες τεχνικές όπως το Partition-based Spatial Merge Join (PBSM), οι οποίες επιτρέπουν μείωση του πλήθους των άσχετων συγκρίσεων και βελτιστοποίηση της απόδοσης.[1]

### 2.3 Γειτονική Επέκταση 3×3

Για την πλήρη κάλυψη των σημείων που βρίσκονται κοντά στα όρια των κελιών, υιοθετείται η προσέγγιση της γειτονικής επέκτασης. Συγκεκριμένα, τα σημεία από το σύνολο S αναπαράγονται και τοποθετούνται στο δικό τους κελί, καθώς και στα οκτώ γειτονικά κελιά ( $\pm 1$  σε κάθε άξονα), ώστε να διασφαλιστεί ότι δεν θα παραλειφθεί κανένα ζεύγος σημείων λόγω οριακών καταστάσεων. Αυτή η μέθοδος είναι εμπνευσμένη από τις τεχνικές Adaptive Replication, όπως παρουσιάζονται από τους Doulkeridis et al. (2025)[2].

### 2.4 Ακρίβεια Στρογγυλοποίησης με Χρήση Λογαρίθμου

Για να αποφευχθούν σφάλματα κατά την ανάθεση σημείων σε κελιά που βρίσκονται ακριβώς στα όρια του grid, εφαρμόζεται ένας υπολογισμός της ακρίβειας της στρογγυλοποίησης βασισμένος στον αρνητικό δεκαδικό λογάριθμο της απόστασης  $\epsilon$ . Αυτή η τεχνική διασφαλίζει ότι κανένα σημείο δεν θα χάσει τους πραγματικούς γείτονές του λόγω προβλημάτων ακρίβειας που μπορεί να προκύψουν από τη στρογγυλοποίηση.

### 2.5 Φιλτράρισμα με βάση την Ευκλείδεια Απόσταση

Μετά την ένωση των κελιών με βάση τις χωρικές τους συντεταγμένες, εφαρμόζεται ένα επιπλέον φίλτρο για να

διατηρηθούν μόνο τα ζεύγη σημείων που η πραγματική ευκλείδεια απόσταση μεταξύ τους είναι μικρότερη ή ίση με το προκαθορισμένο κατώφλι απόστασης  $\epsilon$  (epsilon). Αυτή η φάση υλοποιείται μέσω Spark DataFrame transformations, παρέχοντας αποτελεσματικότητα και ευκολία διαχείρισης.[4]

### 2.6 Εκτέλεση των Ερωτημάτων (Query A και Query B)

Query A: Απαιτείται καταμέτρηση των ζευγαριών (r, s) που πληρούν το κριτήριο απόστασης. Το τελικό αποτέλεσμα είναι η συνολική καταμέτρηση, η οποία γράφεται σε ένα αρχείο εξόδου.

Query B: Βασίζεται στα αποτελέσματα του Query A, ομαδοποιώντας τα σημεία του συνόλου R με βάση τον αριθμό των γειτονικών τους σημείων από το σύνολο S που βρίσκονται εντός της απόστασης  $\epsilon$ . Διατηρούνται μόνο τα σημεία που έχουν τουλάχιστον k γειτονικά σημεία ( $\geq k$ ).

Γιατί επιλέχθηκαν αυτές οι τεχνικές;

Η επιλογή της μεθοδολογίας βασίζεται στις εξής σκέψεις και στη σχετική βιβλιογραφία:

Αποτελεσματική διαχείριση μεγάλων δεδομένων: Το grid partitioning και η γειτονική επέκταση είναι καθιερωμένες τεχνικές για την παράλληλη επεξεργασία χωρικών δεδομένων.[3]

Μείωση περιττών συγκρίσεων: Η γειτονική επέκταση διασφαλίζει ότι όλα τα σημεία κοντά σε όρια κελιών συμπεριλαμβάνονται στις απαραίτητες συγκρίσεις.

Υποστήριξη κλιμακωσιμότητας: Οι συγκεκριμένες τεχνικές είναι σχεδιασμένες να εκμεταλλεύονται πλήρως τις δυνατότητες παραλληλισμού του Spark.

Η υλοποίηση αυτή συμβαδίζει με τις στρατηγικές που παρουσιάζονται σε δημοσιευμένες εργασίες όπως "Parallel Spatial Join Processing with Adaptive Replication" (Doulkeridis et al., 2025).

### 2.7 Σημασία της Παραμετροποίησης

Η παραμετροποίηση της υλοποίησης είναι κρίσιμη για την ευέλικτη προσαρμογή της σε διαφορετικές συνθήκες. Στην παρούσα υλοποίηση, παράμετροι όπως η απόσταση  $\epsilon$  (epsilon), ο αριθμός partitions, η τιμή του k για το Query B, καθώς και κρίσιμες παράμετροι του Apache Spark (μνήμη driver/executors, αριθμός πυρήνων, παράμετροι δικτύου), ορίζονται εύκολα μέσω του κώδικα και της γραμμής εντολών [5]. Αυτή η δυνατότητα επέτρεψε την εκτέλεση πειραμάτων και τη βελτιστοποίηση της απόδοσης για μεγάλα χωρικά δεδομένα.

Επιπλέον, εφαρμόστηκε διαδικασία swapping των συνόλων R και S για τον έλεγχο της ασυμμετρίας στο Query B, αντιστρέφοντας τους ρόλους τους. Η ευέλικτη δομή του κώδικα διευκόλυνε αυτή τη διαδικασία, επιτρέποντας την ποσοτικοποίηση της ασυμμετρίας γρήγορα και αποτελεσματικά. Η παραπάνω υλοποίηση παρέχει ένα ισχυρό και αποτελεσματικό πλαίσιο για την παράλληλη επεξεργασία χωρικών δεδομένων, με άμεση εφαρμογή σε πραγματικά μεγάλα σύνολα δεδομένων.

## 3 Πειραματική Αξιολόγηση

Για την αξιολόγηση της υλοποίησης πραγματοποιήθηκε σειρά πειραμάτων με διαφορετικές τιμές για βασικές παραμέτρους της εφαρμογής, συγκεκριμένα:

- Την τιμή του κατωφλίου απόστασης  $\epsilon$  (epsilon)
- Τον αριθμό των διαμερίσεων (partitions)
- Τον αριθμό των executors και cores
- Την κατεύθυνση επεξεργασίας (μεταξύ R και S)
- Την τιμή κατωφλίου k στο Query B

Τα πειράματα εκτελέστηκαν σε Spark cluster 4 κόμβων, στο υπολογιστικό περιβάλλον Okeanos, με συνδυασμούς ρυθμίσεων που στόχευαν στη βελτιστοποίηση της απόδοσης.

3.1 Υπολογισμός χρόνου εκτέλεσης

Για την αξιολόγηση της απόδοσης κάθε παραμετροποίησης, κρίθηκε απαραίτητο να καταγραφεί με ακρίβεια ο συνολικός χρόνος εκτέλεσης των ερωτημάτων. Η μέτρηση του χρόνου ξεκινά εντός της κύριας συνάρτησης (main) του αρχείου spatial\_join.py, αμέσως πριν την έναρξη του αντίστοιχου Query (A ή B) και μετατρέπεται σε δευτερόλεπτα με ακρίβεια δεκαδικών.

Ο καταγεγραμμένος χρόνος περιλαμβάνει:

- Τη δημιουργία των partitions και την αρχική προετοιμασία των δεδομένων (join και filtering).
- Την εκτέλεση των υπολογισμών για τον εντοπισμό των ζευγών (Query A) ή των εγγραφών με περισσότερους από k γείτονες (Query B).

- Την εγγραφή των τελικών αποτελεσμάτων στο HDFS.

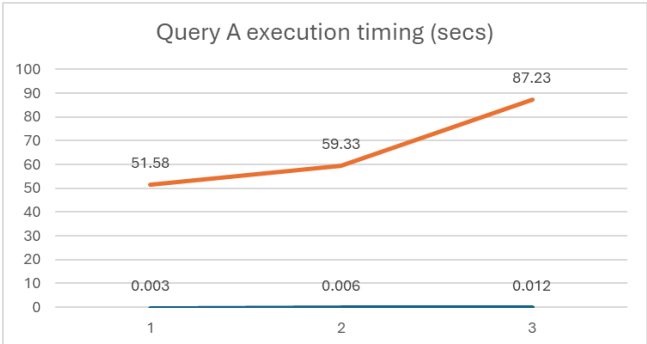
Δεν περιλαμβάνει:

- Την ανάγνωση των αρχείων R και S, καθώς αυτή πραγματοποιείται νωρίτερα στον κώδικα.
- Το setup του Spark session ή την είσοδο των παραμέτρων μέσω argparse.

Με αυτή τη δομή μέτρησης εξασφαλίζεται ότι ο χρόνος που καταγράφεται αντικατοπτρίζει αποκλειστικά τη διάρκεια εκτέλεσης του πυρήνα της αλγοριθμικής επίλυσης των Queries A και B. Έτσι, καθίσταται δυνατή η δίκαιη και αξιόπιστη σύγκριση της απόδοσης μεταξύ διαφορετικών παραμετροποιήσεων.

3.2 Επίδραση του ε

ε	Total Pairs	Matching	Χρόνος Εκτέλεσης (sec)
0.003	2,739,782		51.58
0.006	10,602,167		59.33
0.012	41,927,160		87.23

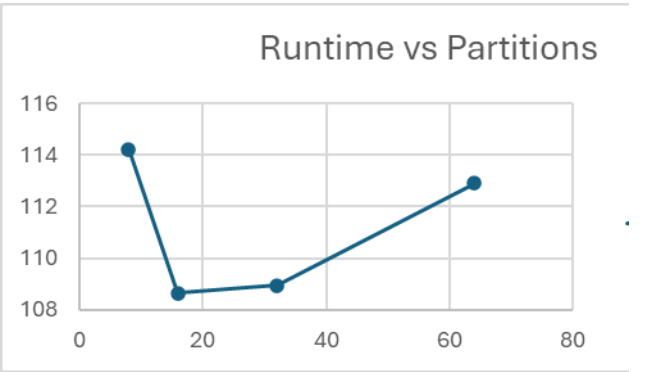


Εικόνα 1Χρόνοι εκτέλεσης A ερωτήματος

Ανάλυση: Παρατηρήθηκε εκθετική αύξηση του πλήθους των ζευγών καθώς αυξάνεται η απόσταση ε, γεγονός που επιβεβαιώνει τη σημασία αυτής της παραμέτρου. Ο χρόνος εκτέλεσης αυξάνεται αναλόγως, αλλά με σχετικά γραμμική συμπεριφορά λόγω αποδοτικής παραλληλοποίησης.

3.3Μεταβλητή: Αριθμός Partitions

Partitions	Χρόνος Εκτέλεσης (sec)
8	114.22
16	108.65
32	108.95
64	112.90

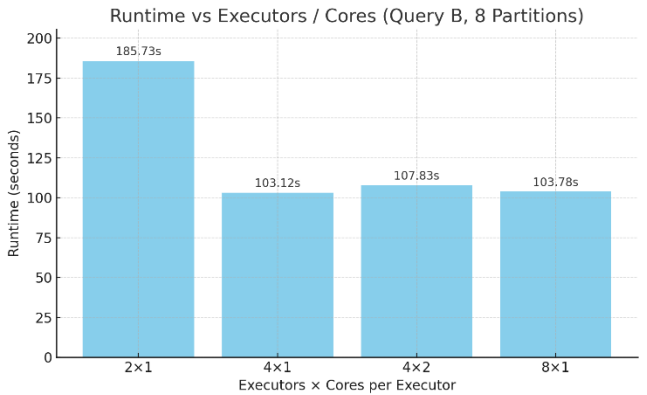


Εικόνα 2Διαφοροποίηση χρόνου εκτέλεσης για διαφορετικό αριθμό partitions

Ανάλυση: Η καλύτερη επίδοση σημειώθηκε στα 16 partitions. Για πολύ μικρό αριθμό partitions δεν αξιοποιούνται επαρκώς οι πυρήνες, ενώ η υπερβολική αύξησή τους (64) οδήγησε σε overhead και ελαφρά υποβάθμιση της απόδοσης.

3.4Μεταβλητή: Executors και Cores

Executors	Cores/Executor	Partitions	Χρόνος Εκτέλεσης (sec)
2	1	8	185.73
4	1	8	103.12
4	2	8	107.83
8	1	8	103.78



Εικόνα 3Διαφοροποίηση χρόνου εκτέλεσης με διαφορετικές παραμέτρους

Ανάλυση: Η καλύτερη απόδοση (103.12 δευτερόλεπτα) καταγράφηκε με 4 executors και 1 core ανά executor. Αυτή η ρύθμιση αποδείχθηκε πιο αποτελεσματική διότι:

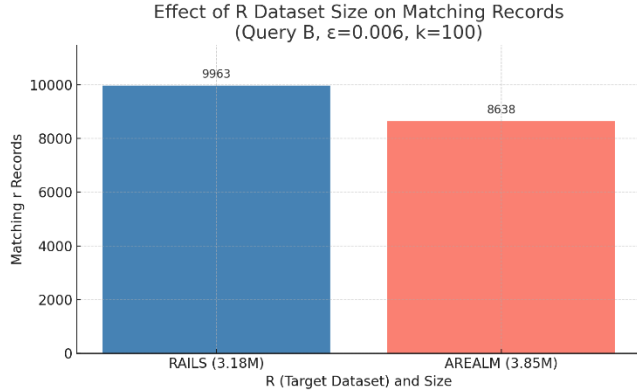
- Οι 4 executors κατανέμονται ομοιόμορφα στους 4 διαθέσιμους κόμβους του cluster, αποφεύγοντας υπερφόρτωση σε οποιονδήποτε κόμβο.
- Η χρήση ενός μόνο πυρήνα ανά executor ελαχιστοποιεί το contention μεταξύ threads και αποτρέπει το overhead από εναλλαγές περιεχομένων CPU cache (context switching).
- Ο αριθμός των partitions (8) αντιστοιχεί σε δύο tasks ανά executor, γεγονός που εξασφαλίζει λογική παραλληλία χωρίς αχρείαστο scheduling overhead.

Παρόλο που οι υπόλοιπες παραμετροποιήσεις είχαν το ίδιο πλήθος αποτελεσμάτων, δεν κατάφεραν να επιτύχουν καλύτερους χρόνους, είτε λόγω μειωμένου παραλληλισμού (2 executors), είτε

λόγω αυξημένου overhead επικοινωνίας (8 executors), είτε λόγω sharing πόρων εντός του ίδιου node (2 cores ανά executor).

### 3.5 Επιδράσεις του Swapping (Ασυμμετρία)

R	S	Volume of data records	$\epsilon$	k	Matching r Records
RAILS	AREALM	3184948	0.006	100	9963
AREALM	RAILS	3848971	0.006	100	8638



#### Εικόνα 4Μη συμμετρικότητα

Ανάλυση: Η αλλαγή κατεύθυνσης R και S στο Query B είχε σαφή επίδραση στο πλήθος των αποτελεσμάτων. Το γεγονός αυτό υποδηλώνει την ασυμμετρία στον χωρικό καταμερισμό των δεδομένων.

Παρόλο που το σύνολο AREALM, όταν χρησιμοποιείται ως R (το σύνολο που ελέγχεται για γείτονες), περιέχει περισσότερες εγγραφές (3.849.871) από το RAILS (3.184.948), το αποτέλεσμα του Query B περιλαμβάνει λιγότερες αντιστοιχίες (8.638 έναντι 9.963).

Αυτό το φαινομενικά αντιφατικό εύρημα καταδεικνύει τη σημασία της χωρικής κατανομής των δεδομένων και όχι απλώς του πλήθους τους. Το Query B εξαρτάται από το πόσοι γείτονες του S βρίσκονται εντός της απόστασης  $\epsilon$  για κάθε εγγραφή του R. Εάν τα σημεία του S είναι αραιά ή διασκορπισμένα, ακόμα και ένας μεγάλος αριθμός εγγραφών μπορεί να αποδώσει λίγες αντιστοιχίες.

Αντίθετα, το RAILS εμφανίζει πιθανώς μεγαλύτερη τοπική πυκνότητα, επιτρέποντας σε περισσότερες εγγραφές του να πληρούν το κριτήριο του  $k=100$ .

Συμπερασματικά, το πείραμα της αντιστροφής των συνόλων ανέδειξε ότι η γεωγραφική πυκνότητα και η δομή του dataset είναι καθοριστικοί παράγοντες για την απόδοση χωρικών ερωτημάτων και όχι απλώς το μέγεθός τους.

### 3.6 Μεταβλητή: k στο Query B

k	Matching r Records	Χρόνος Εκτέλεσης (sec)
10	300,825	86.51
50	47,493	87.40
100	9,963	84.11
200	1,216	89.73
500	131	93.72

Ανάλυση: Καθώς το k αυξάνεται, ο αριθμός των αποτελεσμάτων μειώνεται— κάτι το αναμενόμενο καθώς γίνονται πιο αυστηρά τα κριτήρια. Ο χρόνος εκτέλεσης δεν επηρεάζεται γραμμικά από αυτή τη μείωση, κάτι που αποδεικνύει ότι το computational κόστος

εξαρτάται από την πλήρη ανάλυση όλων των εγγραφών, όχι μόνο των τελικών αποτελεσμάτων.

### 3.7 Βέλτιστη Παραμετροποίηση

Ο συνδυασμός 4 executors  $\times$  1 core και 16 partitions προσέφερε την καλύτερη συνολική απόδοση, με χρόνο εκτέλεσης μόλις 98.93 δευτερόλεπτα για 9963 εγγραφές. Η παραμετροποίηση αυτή εξισορροπεί αποτελεσματικά την παράλληλη εκτέλεση με την αποδοτική χρήση υπολογιστικών πόρων:

- Τα 16 partitions κατανέμονται σε 4 executors με 1 core έκαστο, επιτρέποντας στα tasks να εκτελούνται σε «λογικά κύματα», μεγιστοποιώντας την εκμετάλλευση των διαθέσιμων vCPUs του cluster (4 nodes  $\times$  4 vCPUs = 16 cores συνολικά).
- Αποφεύγεται το overhead κατανομής υπερβολικών partitions και διατηρείται η μνήμη ανά executor εντός ασφαλών ορίων.
- Δεν παρατηρείται υποβάθμιση από excessive shuffling ή task queuing, καθώς κάθε executor χειρίζεται 4 partitions χωρίς συμφόρηση.

Συμπερασματικά, η εν λόγω παραμετροποίηση αποτελεί τη βέλτιστη επιλογή για το συγκεκριμένο cluster και φορτίο εργασίας.

## 4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα εργασία ανέδειξε τη σημασία της βελτιστοποίησης παραμέτρων για την αποδοτική εκτέλεση χωρικών συζεύξεων σε μεγάλα σύνολα δεδομένων με τη χρήση του Apache Spark. Μέσω της υλοποίησης και αξιολόγησης των ερωτημάτων Query A και Query B, επιβεβαιώθηκε ότι οι παράμετροι όπως η απόσταση  $\epsilon$ , το πλήθος γειτόνων  $k$ , ο αριθμός των διαμερίσεων (partitions), καθώς και η κατανομή των executors και cores, επηρεάζουν σημαντικά τον χρόνο εκτέλεσης.

Η καλύτερη απόδοση επιτεύχθηκε με παραμετροποίηση 4 executors, 1 core ανά executor και 16 partitions, η οποία εκμεταλλεύεται πλήρως τους διαθέσιμους πόρους ενός Spark cluster 4 κόμβων με 16 συνολικά vCPUs. Επιπλέον, παρατηρήθηκε ότι οι αυξομειώσεις του  $k$  στο Query B επηρεάζουν κυρίως τον όγκο των αποτελεσμάτων, ενώ το  $\epsilon$  επηρεάζει άμεσα τον αριθμό των υπολογισμών. Η ανάλυση της ασυμμετρίας των δεδομένων μεταξύ R και S υπογράμμισε τη σημασία της χωρικής κατανομής στην ερμηνεία των αποτελεσμάτων.

Συνολικά, η αλγοριθμική προσέγγιση με grid-based χωρική διαμέριση αποδείχθηκε αποδοτική και επεκτάσιμη για την επίλυση χωρικών συζεύξεων σε πραγματικά δεδομένα μεγάλης κλίμακας

## 5 ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Μελλοντική εργασία θα μπορούσε να επικεντρωθεί στην εφαρμογή άλλων τεχνικών χωρικής κατανομής, όπως μικρότερα grid ή περιοχές μεταβλητού μεγέθους, ώστε να περιοριστεί ο αριθμός των άχρηστων συγκρίσεων. Επίσης, θα μπορούσε να εξεταστεί η αυτόματη ρύθμιση παραμέτρων (όπως ο αριθμός partitions και executors) ανάλογα με το μέγεθος των δεδομένων και τους διαθέσιμους πόρους.

Μια ακόμα κατεύθυνση είναι η αξιολόγηση της ίδιας προσέγγισης σε άλλα σύνολα γεωχωρικών δεδομένων ή και σε πραγματικό χρόνο (π.χ. streaming δεδομένα), για να φανεί κατά πόσο μπορεί να επεκταθεί η μεθοδολογία σε άλλες περιπτώσεις χρήσης.

## 6 ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] J. M. Patel, D. J. DeWitt, J. F. Naughton, and B. Tierney. 1996. A partition based spatial-merge join. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

<https://pages.cs.wisc.edu/~dewitt/includes/paradise/spjoin.pdf>

- [2] C. Doulkeridis, A. S. Nimatullah, and N. Pelekis. 2025. Parallel spatial join processing with adaptive replication. Distributed and Parallel Databases (to appear). [Χρησιμοποιήθηκε ως έμπνευση για τον μηχανισμό αναπαραγωγής γειτονικών κελιών και την παράλληλη επεξεργασία.]
- [3] J. Tang, J. Sun, C. Zhang, and T. Yu. 2016. Efficient processing of spatial joins using Spark. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), 2265–2270. DOI: <https://doi.org/10.1109/BigData.2016.7840823>
- [4] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI).
- [5] Apache Spark Documentation. Available at: <https://spark.apache.org/docs/latest/>
- [6] Ευρωπαϊκή Υποδομή Okeanos GRNET. Available at: <https://okeanos.grnet.gr/> Sheng Wang, Zhifeng Bao, J. Shane Culpepper, and Gao Cong. 2021. A Survey on Trajectory Data Management, Analytics, and Learning. ACM Comput. Surv. 54, 2, Article 39 (March 2022), 36 pages. <https://doi.org/10.1145/3440207>