Διαχείριση Δεδομένων Μεγάλης Κλίμακας - Αναφορά Εργασίας

Διολέτης Μηνάς

Προμπονάς Αντώνιος

Ιούνιος 2024

1 Δ ιευχρινίσεις

Μπορείτε να βρείτε τα πηγαία αρχεία για των υλοποιήσεων για όλα τα σχετικά ζητήματα στον ακόλουθο σύνδεσμο στο github:

https://github.com/minasd1/big_data_management

Παρακάτω παρουσιάζονται οι κατάλληλες επεξηγήσεις για όσα ζητούμενα απαιτώνται.

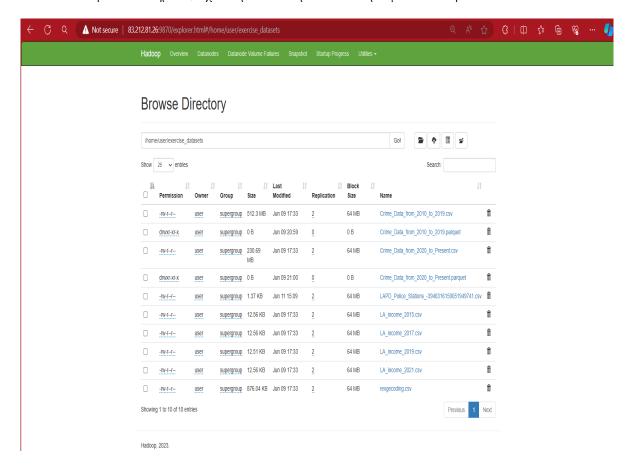
2 Ζητούμενο 2

Αρχικά, φορτώθηκαν τοπικά τα δοθέντα αρχεία. Έπειτα, με χρήση του WinScp μεταφέρθηκαν στον κόμβο master. Στη συνέχεια, εκτελέστηκαν διαδοχικά οι ακόλουθες εντολές.

```
hadoop fs -mkdir -p ~/exercise_datasets/
hadoop fs -put *.csv ~/exercise_datasets
hadoop fs -ls ~/exercise_datasets/
```

Βάσει των παραπάνω, δημιουργήθηκε ένας φάκελος hdfs, τοποθετήθηκαν μέσα σε αυτόν όλα τα αρχεία ενδιαφέροντος, και εξετάστηκε η ορθή τους τοποθέτηση.

Η κατάσταση του συστήματος αχείων με τα δεδομένα διαθέσιμα φαίνεται στην εικόνα 1.



Σχήμα 1: Στιγμιότυπο οθόνης με την κατάσταση του συστήματος αρχείων

Ο κώδικας μετατροπής των ζητούμενων αρχείων από csv σε parquet είναι επίσης αναρτημένος στο σύνδεσμο του github.

3 Ζητούμενο 3

Παρακάτω παρουσιάζονται τα αποτελέσματα για τις διαφορετικές εκτελέσεις του ζητουμένου 3.

+	+	
year month		tallranki
+	+	++
2010 3	17595	j1 j
2010 7	17520	12
2010 5	17338	[3]
2011 8	17139	11 1
2011 5	17050	12
2011 3	116951	[3]
2012 8	17696	j1 j
2012 10	17477	12
2012 5	17391	[3]
2013 8	17329	jı j
2013 7	116714	12
2013 5	16671	[3]
2014 7	17456	j1 j
2014 10	17300	12
2014 12	17076	[3]
2015 8	19134	j1 j
2015 10	19065	12
2015 7	18755	[3]
2016 8	119834	jı j
2016 10	119678	[2]
2016 7	19343	[3]
2017 10	20436	jı j
2017 8	20127	[2]
2017 7	20034	[3]
2018 5	20277	[1]
2018 7	19998	[2]
2018 10	119851	[3]
2019 7	19349	[1]
2019 8	19094	[2]
2019 3	18967	[3]
[2020]1	18512	[1]
2020 2	17443	[2]
2020 7	17257	[3]
2021 10	19191	jı j
2021 7	18954	[2]
[2021]11	18666	[3]
2022 5	20784	[1]
[2022]8	20585	12 1
[2022]6	20418	[3]
[2023]10	20350	[1]
[2023]8	20343	[2]
[2023]1	20257	[3 [
[2024]1	20000	[1]
[2024]2	18008	[2]
[2024]3	17179	[3]
+	+	++

Σχήμα 2: Αποτέλεσμα υλοποίησης ερωτήματος 1 με χρήση SQL και csv αρχείων

+	+	++
year month		
+		++
[2010]3	17595	[1]
[2010]7	17520	[2]
[2010]5	17338	[3]
[2011]8	17139	[1]
2011 5	17050	[2]
2011 3	16951	[3]
2012 8	17696	[1]
2012 10	17477	[2]
2012 5	17391	[3]
2013 8	17329	[1]
2013 7	16714	[2]
2013 5	16671	[3]
2014 7	17456	[1]
2014 10	17300	[2]
2014 12	17076	[3]
2015 8	19134	[1]
[2015]10	19065	[2]
2015 7	18755	[3]
2016 8	19834	[1]
2016 10	19678	[2]
[2016]7	19343	[3]
[2017]10	20436	[1]
2017 8	20127	[2]
2017 7	20034	[3]
2018 5	20277	[1]
2018 7	19998	[2]
2018 10	19851	[3]
2019 7	19349	[1]
2019 8	19094	[2]
[2019]3	18967	[3]
[2020]1	18512	[1]
[2020]2	17443	[2]
[2020]7	17257	[3]
[2021]10	19191	[1]
2021 7	18954	[2]
2021 11	18666	[3]
2022 5	20784	1
2022 8	20585	[2]
2022 6	20418	[3]
2023 10	20350	1
2023 8	20343	[2]
2023 1	20257	[3 [
2024 1	20000	1
2024 2	18008	[2]
2024 3	17179	[3 [
+	-+	++

 Σ χήμα 3: Αποτέλεσμα υλοποίησης ερωτήματος 1 με χρήση SQL και parquet αρχείων

+		
year mont	h crime_to	tal rank
++	-+ 17595	++
2010 3		1
2010 7 2010 5	17520 17338	2 3
2010 3	117139	
2011 5	117050	11
2011 3	116951	3
2012 8	117696	11 1
2012 10	17477	12
2012 5	117391	13
2013 8	17329	11
2013 7	116714	12
2013 5	16671	13
2014 7	117456	11
2014 10	117300	12
2014 12	17076	13
2015 8	119134	11
2015 10	119065	12 1
2015 7	18755	13 1
2016 8	19834	ii i
2016 10	19678	12 1
2016 7	19343	i3 i
2017 10	20436	iı i
2017 8	20127	i2 i
2017 7	20034	i3 i
2018 5	20277	jı j
2018 7	19998	[2]
2018 10	19851	[3]
2019 7	19349	11
2019 8	19094	[2]
[2019]3	18967	[3]
2020 1	18512	[1]
[2020]2	17443	[2]
[2020]7	17257	[3]
[2021]10	19191	[1]
[2021]7	18954	[2]
2021 11	18666	[3]
2022 5	20784	[1]
2022 8	20585	[2]
[2022]6	20418	13 1
[2023]10	20350	1
2023 8	20343	[2]
2023 1	20257	[3]
2024 1	120000	1
2024 2	18008	2
2024 3	17179	[3]
+	-+	+

Σχήμα 4: Αποτέλεσμα υλοποίησης ερωτήματος 1 με χρήση dataframes και csv αρχείων

+	+	+
year month	crime to	tal rank
+	· +	++
[2010]3	17595	[1]
	17520	12
	17338	[3]
	17139	jı j
	17050	12
	116951	[3]
	17696	jı j
	17477	12
2012 5	17391	[3]
	17329	jı j
	16714	12
	16671	[3]
	17456	ji j
	17300	12
	17076	13
	19134	11 1
	19065	12
	18755	[3]
	119834	ii i
	19678	12
	19343	13 j
	20436	iı i
	20127	12
	20034	i3 i
	20277	iı i
	19998	i2 i
	19851	3
	19349	iı i
	19094	12
	18967	із і
	118512	iı i
	17443	12
	17257	[3]
	19191	ii i
	18954	2
	18666	[3]
	20784	ii i
	20585	2
	20418	[3]
	20350	j1 j
	20343	2
	20257	[3]
	20000	j1 j
	18008	12
2024 3	17179	[3]
+	+	+

Σχήμα 5: Αποτέλεσμα υλοποίησης ερωτήματος 1 με χρήση dataframes και parquet αρχείων

Οι χρόνοι εκτέλεσης για τις διάφορες μεθόδους υλοποίησης του ερωτήματος παρουσιάζονται παρακάτω:

Table 1: Execution time for the different methods

Method	Execution time
SQL-CSV	1.4 min.
SQL-Parquet	46 sec.
Dataframe-CSV	$1.3 \mathrm{min}.$
Dataframe-Parquet	40 sec.

Όπως είναι ευκόλα αντιληπτό ο χρόνος εκτέλεσης όταν χρησιμοποιούμε αρχεία εισόδου parquet είναι πολύ μικρότερος σε σύγκριση με τον χρόνο υλοποίησης με είσοδο αρχειών μορφής csv. Το γεγονός αυτό συμβαίνει για αρκετούς λόγους. Πρώτα απόλα, τα αρχεία Parquet χρησιμοποιούν μια δομή δεδομένων που είναι προσαρμοσμένη για ανάγνωση και εγγραφή από διαφορετικά εργαλεία επεξεργασίας δεδομένων, όπως το Apache Spark ή το Apache Hive. Αυτή η δομή επιτρέπει την αποθήκευση των δεδομένων σε μορφή στήλης, η οποία είναι πιο αποδοτική για πολλούς τύπους επεξεργασίας. Χρησιμοποιούν συμπίεση δεδομένων, που μπορεί να μειώσει το μέγεθος των αρχείων και αυτό έχει ως αποτέλεσμα να αποθηκεύουν περισσότερα δεδομένα στον ίδιο χώρο αποθήκευσης. Ακόμα, επειδή τα δεδομένα αποθηκεύονται σε συμπιεσμένη μορφή και με δομή στήλης, η ανάγνωση τους από εφαρμογές επεξεργασίας δεδομένων μπορεί να είναι πολύ πιο γρήγορη σε σχέση με τα αρχεία CSV.

4 Ζητούμενο 4

Οι χρόνοι εκτέλεσης για τις διάφορες υλοποιήσεις του ζητουμένου 4 φαίνονται παρακάτω.

Table 2: Execution Times for Different Methods

Method	Execution Time
RDD API	1.4 min
SQL API	$52 \mathrm{\ s}$

Τα παραπάνω αποτελέσματα μπορούν να εξηγηθούν, αν διακρίνει κανείς τα χαρακτηριστικά των δύο APIs.

4.1 RDD API

- Χαμηλότερο Επίπεδο Αφαίρεσης: Η RDD είναι μια βασιχή δομή δεδομένων στο Spark, που αντιπροσωπεύει μια αμετάβλητη συλλογή αντιχειμένων που μπορούν να επεξεργαστούν παράλληλα σε έναν συγχεχριμένο cluster.
- Χειροχίνητη Βελτιστοποίηση: Με τις RDD, οι προγραμματιστές έχουν περισσότερο έλεγχο στις εργασίες επεξεργασίας δεδομένων, αλλά πρέπει επίσης να ασχοληθούν με βελτιστοποιήσεις χαμηλού επιπέδου όπως η κατανομή δεδομένων.
- Έλλειψη Βελτιστοποίησης Ερωτήματος: Οι λειτουργίες RDD δεν επωφελούνται από τεχνικές υψηλού επιπέδου βελτιστοποίησης ερωτημάτων, που οδηγεί σε λιγότερο αποτελεσματικά σχέδια εκτέλεσης.

4.2 SQL API

- Υψηλότερο Επίπεδο Αφαίρεσης: Οι SQL-like APIs στο Spark, όπως τα DataFrame APIs, παρέχουν μια πιο καλή προσέγγιση στην επεξεργασία δεδομένων, επιτρέποντας στους προγραμματιστές να κάνουν τροποποιήσεις και διάφορες άλλες ενέργειες χρησιμοποιώντας SQL-likeσυντακτικό.
- Βελτιστοποίηση Ερωτήματος: Αυτά τα ΑΡΙς εκμεταλλεύονται το Catalyst, τον βελτιστοποιητή ερωτημάτων του Spark, για να βελτιστοποιήσουν και να εκτελέσουν ερωτήματα αποτελεσματικά.

Οφέλη στην Απόδοση: Λόγω των δυνατοτήτων βελτιστοποίησης ερωτημάτων και των υψηλότερων επιπέδων αφαίρεσης, τα APIs που βασίζονται σε SQL μπορούν συχνά να δημιουργήσουν πιο αποδοτικά σχέδια εκτέλεσης, οδηγώντας σε ταχύτερους χρόνους εκτέλεσης σε σύγκριση με υλοποιήσεις που βασίζονται σε RDD.

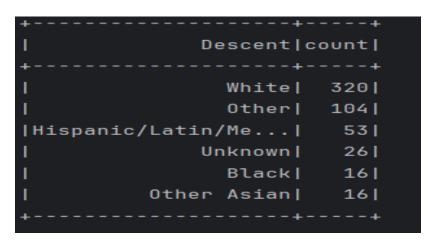
5 Ζητούμενο 5

Παρακάτω παρουσιάζονται οι χρόνοι εκτέλεσης των διαφόρων υλοποιήσεων του ερωτήματος, όπως αυτοί προέκυψαν για το ζητούμενο 5.

Table 3: Execution Times for Different Join Methods

Join Method	Execution Time
Simple Join	$2.0 \min$
Broadcast Join	$2.0 \min$
Shuffle Hash Join	$1.9 \min$

Κάθε υλοποιήση πραγματοποιήθηκε με ένα διαφορετικού είδους join. Το συμπέρασμα που μπορούμε να βγάλουμε με το συγκεκριμένο πίνακα είναι ότι και τα 3 είδη join εκτελούνται με σχεδόν ίδιο χρόνο, άλλα η υλοποίηση με Shuffle Hash Join είναι ελαφρώς ταχύτερη. Υπό αυτή τη συνθήκη θεωρούμε πιο απόδοτική υλοποίηση, εκείνη που χρησιμοποιήσε ως μέθοδο join το Shuffle Hash. Με βάση το τελευταίο συμπέρασμα και χρησιμοποιώντας γενικές γνώσεις, καταλήγουμε πως αν επρέπε να επιλέξουμε μία μέθοδο join, θα επιλέγαμε το Shuffle Hash Join. Το Shuffle Hash Join είναι κατάλληλο για μεγάλα datasets όπου απαιτείται η ανακατανομή (shuffling) των δεδομένων για να επιτευχθεί το join και μπορεί και να κατακερματίζει (hashes) τα δεδομένα και τα διανέμει στους κόμβους για πιο αποδοτικό join. Ένα ακόμη επιχείρημα που ενισχύει τη θέση μας, είναι ότι τα άλλα 2 είδη join δεν συνιστώνται για μεγάλα datasets.



Σχήμα 6: Αποτέλσμα εκτέλεσης για εκείνους τους πολίτες με με το υψηλότερο εισόδημα

Σχήμα 7: Αποτέλσμα εκτέλεσης για εκείνους τους πολίτες με με το χαμηλότερο εισόδημα

6 Ζητούμενο 6

Παρακάτω παρουσιάζονται τα αποτελέσματα για τις απαιτούμενες υλοποιήσεις του ζητήματος, στην υποδεδειγμένη μορφή.

```
Division: 77TH STREET, Average Distance: 2.6879445529419006, Total Incidents: 17021
Division: SOUTHEAST, Average Distance: 2.1054930763583433, Total Incidents: 12948
Division: NEWTON, Average Distance: 2.0189044028686967, Total Incidents: 9844
Division: SOUTHWEST, Average Distance: 2.6996409889496373, Total Incidents: 8912
Division: HOLLENBECK, Average Distance: 2.6522532236227696, Total Incidents: 6202
Division: HARBOR, Average Distance: 4.081918878967312, Total Incidents: 5622
Division: RAMPART, Average Distance: 1.5757419679342042, Total Incidents: 5116
Division: MISSION, Average Distance: 4.7152276045871, Total Incidents: 4503
Division: OLYMPIC, Average Distance: 1.8218137017944847, Total Incidents: 4424
Division: NORTHEAST, Average Distance: 3.904507046334755, Total Incidents: 3920
Division: FOOTHILL, Average Distance: 3.802602034758503, Total Incidents: 3775
Division: HOLLYWOOD, Average Distance: 1.4599717361140725, Total Incidents: 3643
Division: CENTRAL, Average Distance: 1.1380512738908837, Total Incidents: 3615
Division: WILSHIRE, Average Distance: 2.3139357549411534, Total Incidents: 3525
Division: NORTH HOLLYWOOD, Average Distance: 2.719192182766968, Total Incidents: 3465
Division: WEST VALLEY, Average Distance: 3.5289110070807195, Total Incidents: 2903
Division: VAN NUYS, Average Distance: 2.2216495747680365, Total Incidents: 2733
Division: PACIFIC, Average Distance: 3.72911487784952, Total Incidents: 2709
Division: DEVONSHIRE, Average Distance: 4.010338840870013, Total Incidents: 2472
Division: TOPANGA, Average Distance: 3.486742209540246, Total Incidents: 2285
Division: WEST LOS ANGELES, Average Distance: 4.2433943270626004, Total Incidents: 1541
```

Σχήμα 8: Αποτέλεσμα της υλοποίησης με broadcast join.

```
Division: 77TH STREET, Average Distance: 2.6879445529418926, Total Incidents: 17021
Division: SOUTHEAST, Average Distance: 2.1054930763583535, Total Incidents: 12948
Division: NEWTON, Average Distance: 2.0189044028687073, Total Incidents: 9844
Division: SOUTHWEST, Average Distance: 2.6996409889496364, Total Incidents: 8912
Division: HOLLENBECK, Average Distance: 2.6522532236227723, Total Incidents: 6202
Division: HARBOR, Average Distance: 4.081918878967318, Total Incidents: 5622
Division: RAMPART, Average Distance: 1.5757419679342066, Total Incidents: 5116
Division: MISSION, Average Distance: 4.715227604587099, Total Incidents: 4503
Division: OLYMPIC, Average Distance: 1.82181370179448, Total Incidents: 4424
Division: NORTHEAST, Average Distance: 3.904507046334745, Total Incidents: 3920
Division: FOOTHILL, Average Distance: 3.8026020347584994, Total Incidents: 3775
Division: HOLLYWOOD, Average Distance: 1.459971736114078, Total Incidents: 3643
Division: CENTRAL, Average Distance: 1.1380512738908841, Total Incidents: 3615
Division: WILSHIRE, Average Distance: 2.313935754941158, Total Incidents: 3525
Division: NORTH HOLLYWOOD, Average Distance: 2.7191921827669554, Total Incidents: 3465
Division: WEST VALLEY, Average Distance: 3.528911007080722, Total Incidents: 2903
Division: VAN NUYS, Average Distance: 2.221649574768035, Total Incidents: 2733
Division: PACIFIC, Average Distance: 3.7291148778495233, Total Incidents: 2709
Division: DEVONSHIRE, Average Distance: 4.0103388408700065, Total Incidents: 2472
Division: TOPANGA, Average Distance: 3.486742209540246, Total Incidents: 2285
Division: WEST LOS ANGELES, Average Distance: 4.243394327062591, Total Incidents: 1541
```

Σχήμα 9: Αποτέλεσμα της υλοποίησης με repartition join.

7 Ζητούμενο 7

Παραχάτω εμφανίζεται το αποτέλεσμα της υλοποίησης για το εν λόγω ζητούμενο.

+	+	++
DIVISION	average distance	incidents total
+	+	++
77TH STREET	2.687863878979968	17021
SOUTHEAST	2.10536542436028	12948
NEWTON	2.01890408171001	9844
SOUTHWEST	2.69955690695839	8912
HOLLENBECK	2.652183094886917	6202
HARBOR	4.081888808832712	5622
RAMPART	1.5756271891244937	5116
MISSION	4.715336137664879	4503
OLYMPIC	1.8217951174232616	4424
NORTHEAST	3.9044950000219503	3920
FOOTHILL	3.8025931752182	3775
HOLLYWOOD	1.4599420127798626	3643
CENTRAL	1.1381342343584135	3615
WILSHIRE	2.313930122144256	3525
NORTH HOLLYWOOD	2.7191304762900135	3465
WEST VALLEY	3.5289325229810755	2903
VAN NUYS	2.221636501342287	2733
PACIFIC	3.729126540793401	2709
DEVONSHIRE	4.01035846098293	2472
TOPANGA	3.48689778486279	2285
WEST LOS ANGELES	4.243530477603231	1541
+	+	++

 Σ χήμα 10: Αποτέλεσμα της υλοποίησης με dataframes.