Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Ροή Λ: Προχωρημένα Θέματα Βάσεων Δεδομένων

(9ο Εξάμηνο)

Αναφορά Εξαμηνιαίας Εργασίας

Ομάδα 91:

Δημήτρης Καλαθάς 03118016
Δημήτρης Μπακάλης 03118163

GitHub: https://github.com/DKalathas/Advanced-Topics-in-Databases-NTUA

Εισαγωγή

Στην εργασία αυτή θα χρησιμοποιήσουμε αλγόριθμους και τεχνικές διαμοιρασμών πόρων, ώστε να επεξεργαστούμε μαζικά μεγάλες ποσότητες δεδομένων. Για να το πετύχουμε αυτό, έχουμε στην διάθεση μας πόρους στον Ωκεανό, ειδικότερα διαθέτουμε 2 VMs καθένα από τα οποία έχει 4 πυρήνες, 8192 MB RAM και 30GB hard drive (μνήμη). Για να αξιοποιηθούν κατάλληλα οι πόροι αυτοί, χρειάστηκε να τους συνδέσουμε σε ένα ιδιωτικό δίκτυο και σε αυτό να εγκαταστήσουμε το Hadoop και το Spark. Πιο συγκεκριμένα, θα γίνει χρήση του hdfs για την αποθήκευση του μεγάλου όγκου δεδομένων κατανεμημένα, ώστε να κάνουμε πιο εύκολους και ταχύτερους τους υπολογισμούς φέρνοντας τους κοντά στον εκάστοτε Worker που θα αναλάβει να τους εκτελέσει. Ταυτόχρονα, χρησιμοποιούμε το Spark για να γράψουμε κώδικα. Η γλώσσα προγραμματισμού που επελέγη για την συγκεκριμένη εργασία είναι η Python.

Ερώτημα 1

Για την εγκατάσταση του Hadoop και του Spark (των τελευταίων εκδόσεών τους) ακολουθήσαμε τα βήματα που υπάρχουν στο GitHub της εργασίας μας. Επίσης για να διαβάσουμε όλα τα αρχεία μαζί δημιουργήσαμε φάκελους στο file system του Hadoop, έναν με τα parquet και έναν για το csv, έτσι ώστε στο πρόγραμμα που γράψαμε, να διαβάσουμε το συγκεκριμένο directory, του κάθε φακέλου. Αξίζει να σημειωθεί ότι όλοι οι χρόνοι είναι μετρημένοι μετά από collects.

Ερώτημα 2

Χρησιμοποιώντας DataFrame εκτελούμε τα ζητούμενα queries και λαμβάνουμε τα ακόλουθα αποτελέσματα για το καθένα:

Για το Q1:

```
|VendorID|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|RatecodeID|store_and_fwd_flag|PULocationID|DOLocationID|payment_type|fare_amount|
extra|mta_tax|tip_amount|tolls_amount|improvement_surcharge|total_amount|congestion_surcharge|airport_fee| date_type|month|LocationID|  Borough|       Zone|service
_zone|
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+------------+------------+------------+-----------+
-----+-------+----------+------------+---------------------+------------+--------------------+-----------+----------+-----+----------+---------+-----------+-------
-----+
|       2| 2022-03-17 12:27:47|  2022-03-17 12:27:58|            1.0|          0.0|       1.0|                 N|         132|         132|           1|        2.5|
  0.0|    0.5|      40.0|         0.0|                  0.3|        45.8|                 2.5|        0.0|2022-03-17|    3|        12|Manhattan|Battery Park| Yellow
 Zone|
```

Εικόνα 1: Αποτελέσματα Q1.

Επομένως, το ζητούμενο tip amount είναι 40$.

Για το Q2:

Τα δεδομένα των ζητούμενων διαδρομών συνοψίζονται στον παρακάτω πίνακα:

```
|VendorID|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|RatecodeID|store_and_fwd_flag|PULocationID|DOLocationID|payment_type|fare_amount|
extra|mta_tax|tip_amount|tolls_amount|improvement_surcharge|total_amount|congestion_surcharge|airport_fee| date_type|month|month|max(tolls_amount)|
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+------------+------------+------------+-----------+
-----+-------+----------+------------+---------------------+------------+--------------------+-----------+----------+-----+-----+-----------------+
|       1| 2022-03-11 20:08:32|  2022-03-11 20:09:45|            1.0|          0.0|       1.0|                 N|         265|         265|           1|        2.5|
  1.0|    0.5|      48.0|       235.7|                  0.3|       288.0|                 0.0|        0.0|2022-03-11|    3|    3|            235.7|
|       1| 2022-05-21 16:47:48|  2022-05-21 17:05:47|            1.0|          2.4|       3.0|                 N|         239|         246|           3|       31.5|
  0.0|    0.0|       0.0|      813.75|                  0.3|      845.55|                 0.0|        0.0|2022-05-21|    5|    5|           813.75|
|       1| 2022-06-12 16:51:46|  2022-06-12 17:56:48|            9.0|         22.0|       1.0|                 N|         142|         132|           2|       67.5|
  2.5|    0.5|       0.0|      800.09|                  0.3|      870.89|                 2.5|        0.0|2022-06-12|    6|    6|           800.09|
|       1| 2022-04-29 04:31:21|  2022-04-29 04:32:30|            2.0|          0.0|       1.0|                 N|         249|         249|           3|        3.0|
  3.0|    0.5|       0.0|      911.87|                  0.3|      918.67|                 2.5|        0.0|2022-04-29|    4|    4|           911.87|
|       1| 2022-02-18 02:33:30|  2022-02-18 02:35:28|            1.0|          1.3|       1.0|                 N|         265|         265|           1|        3.0|
  0.5|    0.5|     19.85|        95.0|                  0.3|      119.15|                 0.0|        0.0|2022-02-18|    2|    2|             95.0|
|       1| 2022-01-22 11:39:07|  2022-01-22 12:31:09|            1.0|         33.4|       1.0|                 Y|          70|         265|           4|       88.0|
  0.0|    0.5|       0.0|      193.3|                  0.3|       282.1|                 0.0|        0.0|2022-01-22|    1|    1|            193.3|
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+------------+------------+------------+-----------+
-----+-------+----------+------------+---------------------+------------+--------------------+-----------+----------+-----+-----+-----------------+
```

Εικόνα 2: Αποτελέσματα Q2.

Σχετικά με τους χρόνους εκτέλεσης των δύο queries, προκύπτει το παρακάτω πινακάκι:

	1 worker	2 workers
Q1	29.43 s	12.32 s
Q2	61.43 s	22.33 s

Πίνακάς 1: Χρόνοι για Q1 και Q2.

Παρατηρούμε πως η διαφορά στους χρόνους εκτέλεσης με 1 και 2 workers είναι σημαντική, όπως ήταν αναμενόμενο. Πιο συγκεκριμένα, με 2 workers ο χρόνος εκτέλεσης στο Q1 υποδιπλασιάζεται, ενώ στο Q3 σχεδόν υποτριπλασιάζεται.

Ερώτημα 3

Στη συνέχεια, θα εκτελέσουμε το 3ο query, τόσο με DataFrame, όσο και με RDD API.

Με DataFrame API:

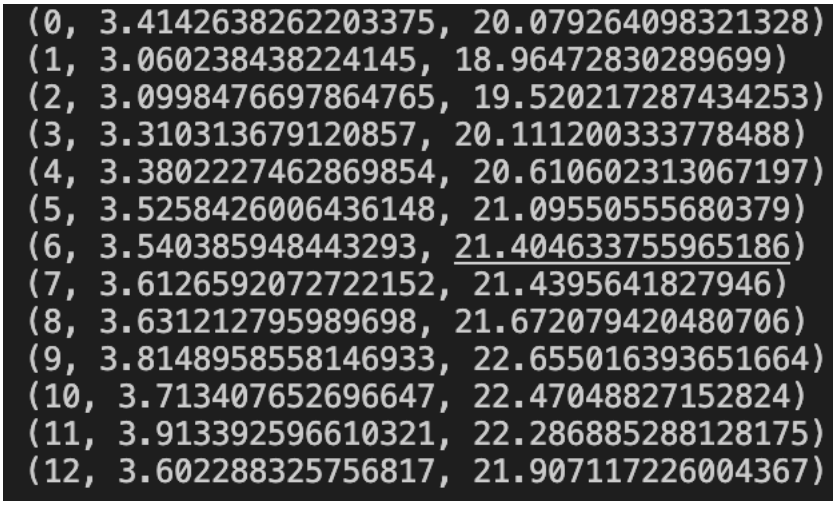
date_segment	average_trip_distance	average_total_amount
0	3.4142638262203375	20.079264098321328
1	3.060238438224145	18.96472830289699
2	3.0998476697864765	19.520217287434253
3	3.310313679120857	20.111200333778488
4	3.3802227462869854	20.610602313067197
5	3.5258426006436148	21.09550555680379
6	3.5403859484432934	21.404633755965186
7	3.6126592072722152	21.4395641827946
8	3.6312127959896974	21.672079420480706
9	3.8148958558146933	22.655016393651664
10	3.713407652696647	22.47048827152824
11	3.913392596610321	22.286885288128175
12	3.602288325756817	21.907117226004367

Εικόνα 3: Αποτελέσματα Q3 (DataFrame).

Στα παραπάνω αποτελέσματα το date_segment αναπαριστά το εκάστοτε δεκαπενθήμερο (0 ⇒ 01/01/2022 – 15/01/2022, 1 ⇒ 16/01/2022 – 30/01/2022 κλπ).

Με RDD API:

Κάνοντας collect το αποτέλεσμα του query, λαμβάνουμε τις παρακάτω τιμές, οι οποίες, όπως ήταν αναμενόμενο, συμπίπτουν με τις αντίστοιχες για DataFrame API (προφανώς και σε αυτή την περίπτωση τα δεδομένα αποθηκεύονται σε στήλες ως εξής: date_segment, average_trip_distance, average_total_amount).



Εικόνα 4: Αποτελέσματα Q3 (RDD).

Σχετικά με τους χρόνους εκτέλεσης των δύο queries, προκύπτει το παρακάτω πινακάκι:

	1 worker	2 workers
Q3 (DF)	9.88 s	6.47 s
Q3 (RDD)	304.01 s	129.16 s

Πίνακάς 2: Χρόνοι για Q3.

Σε αυτή την περίπτωση δεν παρατηρούμε μόνο τη διαφορά στον χρόνο εκτέλεσης που οφείλεται σε 1 παραπάνω worker, αλλά και την σημαντική διαφορά επίδοσης των DataFrames με τα RDD.

Παρατήρηση

Στο παραπάνω ερώτημα παρατηρήσαμε ότι υπάρχουν δύο συγκεκριμένα zones (Corona και Governor's Island/Ellis Island/Liberty Island) με δύο και τρία διαφορετικά IDs, αντίστοιχα, στο αρχείο csv. Κρίναμε ότι το να κάνουμε join τους δύο πίνακες (έναν με το csv και έναν με τα parquet) και να πάρουμε την συνθήκη με τα διαφορετικό zone αρχής και τέλους διαδρομής, θα ήταν πολύ χρονοβόρο και θα έριχνε κατά πολύ τον χρόνο εκτέλεσης του ερωτήματος. Ειδικά εφόσον παραπάνω IDs παρουσιάζουν μόνο 2 από τα 264 συνολικά zones θεωρήσαμε πως το συγκεκριμένο trade-off μεταξύ ακρίβειας και χρόνου εκτέλεσης δεν αξίζει. Επομένως, επιλέξαμε να μην κάνουμε το join και να φτιάξουμε την συνθήκη με τα IDs των περιοχών στα parquet αρχεία. Παρόλα αυτά, αν θέλαμε να είμαστε τυπικοί θα έπρεπε να γίνει με join και φιλτράρισμα, ώστε το zone αναχώρησης και το αντίστοιχο της άφιξης να είναι διαφορετικά, όπως έχει προαναφερθεί.

Ερώτημα 4

Για το Q4:

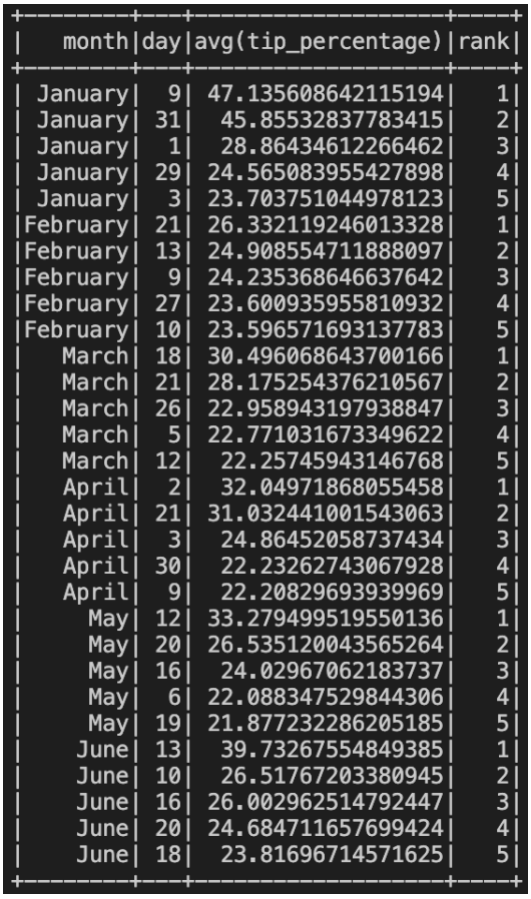
Στο συγκεκριμένο ερώτημα θα αναζητήσουμε για κάθε μέρα της εβδομάδας (πχ όλες τις Δευτέρες του εξεταζόμενου εξαμήνου) τα 3 καλύτερα διαστήματα μίας ώρας (πχ 7-8, 8-9 κλπ) σε σχέση με το σύνολο επιβατών σε ταξί. Τα αποτελέσματα του query φαίνονται στο παρακάτω πινακάκι, όπου day η μέρα της εβδομάδας, hour το διάστημα της μίας ώρας (0 ⇒ 00:00 – 01:00, 17 ⇒ 17:00 - 18:00 κλπ), sum_passengers το σύνολο των επιβατών και rank η κατάταξη της συγκεκριμένης ώρας.

day	hour	sum_passengers	rank
Sunday	0	227140	1
Sunday	17	225170	2
Sunday	19	225162	3
Monday	20	245860	1
Monday	21	236799	2
Monday	19	234960	3
Tuesday	20	274652	1
Tuesday	21	267519	2
Tuesday	19	256092	3
Wednesday	20	279977	1
Wednesday	21	274597	2
Wednesday	19	257587	3
Thursday	20	283853	1
Thursday	21	281513	2
Thursday	19	266583	3
Friday	21	287711	1
Friday	20	281312	2
Friday	22	254226	3
Saturday	21	272317	1
Saturday	20	271315	2
Saturday	19	260238	3

Εικόνα 5: Αποτελέσματα Q4.

Για το Q5:

Σε αυτή την περίπτωση θα αναζητήσουμε τις κορυφαίες 5 μέρες ανά μήνα με τον μεγαλύτερο μέσο όρο ποσοστών tips στις διαδρομές. Τα αποτελέσματα του query φαίνονται στο παρακάτω πινακάκι, όπου month ο μήνας, day η μέρα του μήνα, avg(tip_percentage) ο μέσος όρος των ζητούμενων ποσοστών και rank η κατάταξη της συγκεκριμένης μέρας.



month	day	avg(tip_percentage)	rank
January	9	47.135608642115194	1
January	31	45.85532837783415	2
January	1	28.86434612266462	3
January	29	24.565083955427898	4
January	3	23.703751044978123	5
February	21	26.332119246013328	1
February	13	24.908554711888097	2
February	9	24.235368646637642	3
February	27	23.600935955810932	4
February	10	23.596571693137783	5
March	18	30.496068643700166	1
March	21	28.175254376210567	2
March	26	22.958943197938847	3
March	5	22.771031673349622	4
March	12	22.25745943146768	5
April	2	32.04971868055458	1
April	21	31.032441001543063	2
April	3	24.86452058737434	3
April	30	22.23262743067928	4
April	9	22.20829693939969	5
May	12	33.279499519550136	1
May	20	26.535120043565264	2
May	16	24.02967062183737	3
May	6	22.088347529844306	4
May	19	21.877232286205185	5
June	13	39.73267554849385	1
June	10	26.51767203380945	2
June	16	26.002962514792447	3
June	20	24.684711657699424	4
June	18	23.81696714571625	5

Εικόνα 6: Αποτελέσματα Q5.

	1 worker	2 workers
Q4	14.22 s	8.51 s
Q5	39.01 s	25.65 s

Πίνακάς 3: Χρόνοι για Q4 και Q5.

Και σε αυτή την περίπτωση η διαφορά που οφείλεται στον extra worker είναι εμφανής, καθώς, όπως φαίνεται και στο παραπάνω πινακάκι, οδηγεί σε σημαντική μείωση του χρόνου εκτέλεσης.