Προχωρημένα Θέματα Βάσεων Δεδομένων

ΣΗΜΜΥ ΕΜΠ Εξάμηνο 9ο

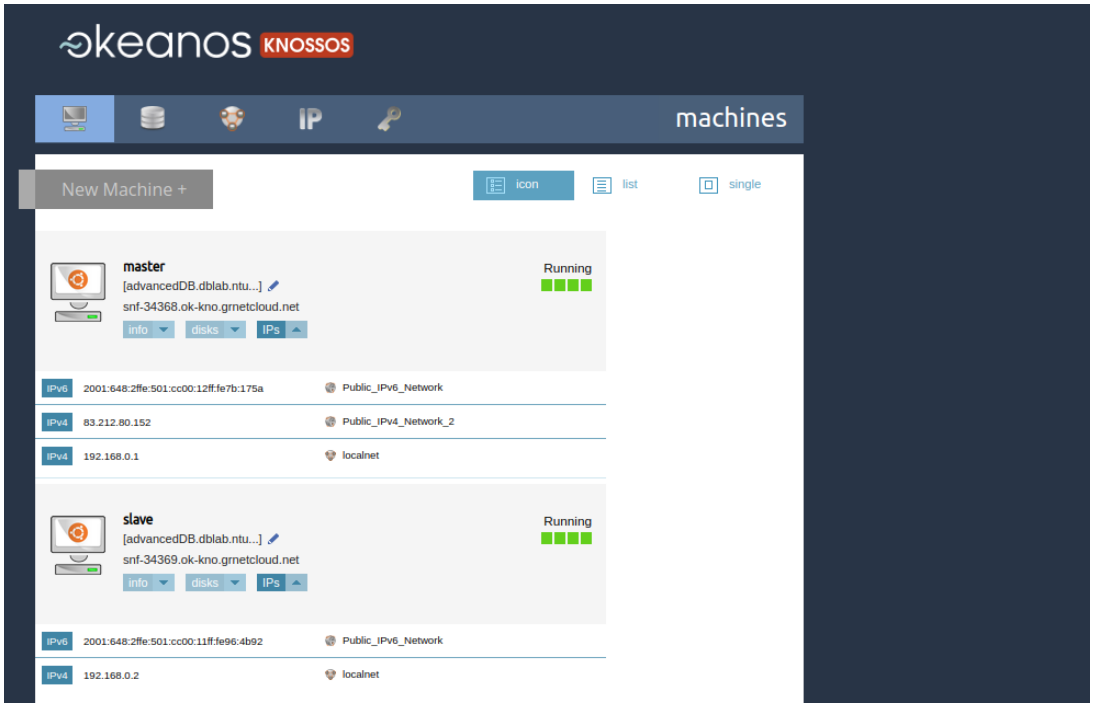
Εξαμηνιαία Εργασία στο Spark

2022-2023

Νικήτας Τσίννας, el18187
Κυριάκος Τσαρτσαράκος, el18054

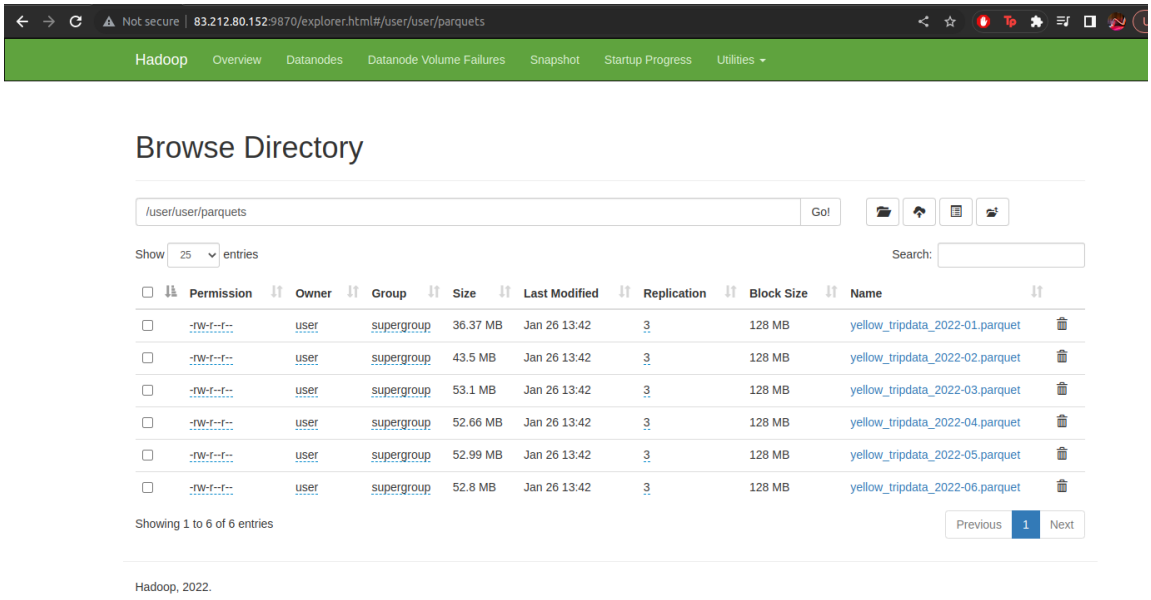
1. Εγκατάσταση του Hadoop και Spark & δημιουργία Dataframes και RDDs

Αφού έγινε το setup των εικονικών μηχανημάτων στον **okeanos** σύμφωνα με τις οδηγίες του εργαστηρίου δημιουργήθηκαν τα ακόλουθα 2 μηχανήματα master και slave όπως φαίνεται παρακάτω.

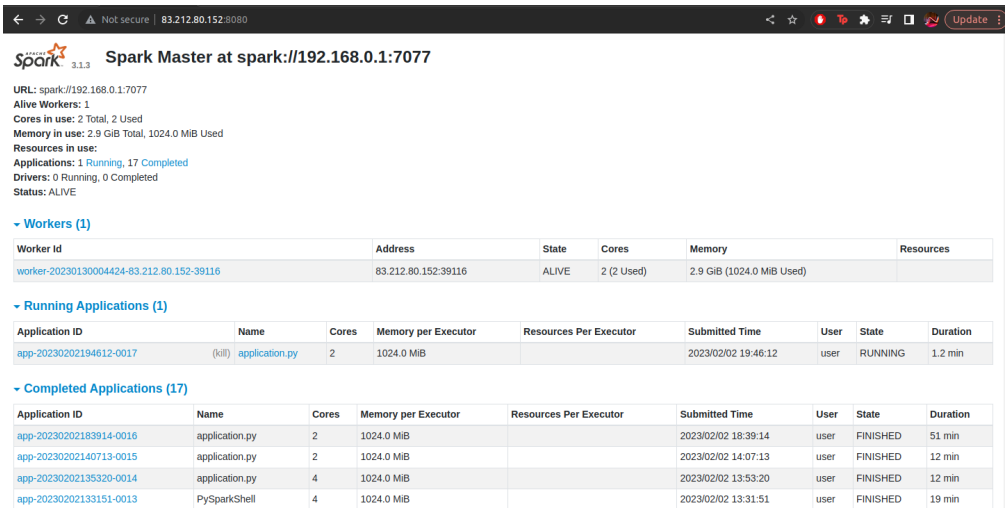


Έπειτα ακολουθήσαμε τις οδηγίες που βρίσκονται σε αυτήν την ιστοσελίδα https://sparkbyexamples.com/hadoop/apache-hadoop-installation/ για την εγκατάσταση και την εκκίνηση του hadoop και στα δύο μηχανήματα. Έγινε το format του hdfs δίσκου καθώς και η εκκίνηση του NameNode στο master και του DataNode στο slave. Η πρόσβαση στο Hadoop web UI του NameNode γίνεται πατώντας την διεύθυνση του master μηχανήματος μαζί με το port 9870 http://83.212.80.152:9870/ .

Έπειτα ανεβάσαμε στο hdfs cluster τα αρχεία parquets των yellow-taxi-trips για τους μήνες Ιανουάριο έως Ιούνιο του 2022 όπως φαίνεται παρακάτω.



Έπειτα, ακολουθήσαμε τις οδηγίες που μας δόθηκαν από το υλικό του εργαστηρίου του μαθήματος για την εγκατάσταση του Spark. Ξεκινήσαμε τον master στο μηχάνημα master και δύο workers (ένα στον master και ένα στον slave). Τα ερωτήματα τα τρέξαμε στο spark cluster με δύο και έναν worker. Για την παύση του ενός worker εκτελούμε την εντολή **./stop-worker.sh spark://192.168.0.1:7077** στον sbin του spark home directory ενός μηχανήματος από τα δύο. Το Spark web UI μπορεί να γίνει προσβάσιμο μέσω του link http://83.212.80.152:8080/ .



Για την δημιουργία των dataframes, αλλά και την εκτέλεση όλων των παρακάτω ερωτημάτων, δημιουργήσαμε ένα Python script. Χρησιμοποιήσαμε το PySpark API για να διαβάσουμε, πρώτα, τα αρχεία parquet από τον hdfs και έπειτα τα ενώσαμε σε ένα Dataframe.

Εδώ, πρέπει να σημειώσουμε πως παρατηρήσαμε μερικά dirty data και για αυτό εφαρμόσαμε φίλτρο για να κρατήσουμε δεδομένα που αφορούν τις ημερομηνίες που μας ενδιαφέρουν (Ιανουάριος - Ιούνιος 2022).

Για την δημιουργία των RDDs απλώς χρησιμοποιήσαμε την συνάρτηση rdd του Dataframe για να τα μετατρέψουμε σε rdds. Ο κώδικας που χρησιμοποιήθηκε μπορεί να βρεθεί στο GitHub repository του project https://github.com/WinRout/ntua_spark_project .

2. Εκτέλεση των Q1 και Q2

Τα αποτελέσματα του **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
2	2022-03-17 12:27:47	2022-03-17 12:27:58	1	0	1	N	12	12	1	2.5	0	0.5	40	0	0.3	45.8	2.5	0

Τα αποτελέσματα του **Q2**:

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
1	2022-01-22 11:39:07	2022-01-22 12:31:09	1	33.4	1	Y	70	265	4	88	0	0.5	0	193.3	0.3	282.1	0	0
1	2022-02-18 02:33:30	2022-02-18 02:35:28	1	1.3	1	N	265	265	1	3	0.5	0.5	19.85	95	0.3	119.15	0	0
1	2022-03-11 20:08:32	2022-03-11 20:09:45	1	0	1	N	265	265	1	2.5	1	0.5	48	235.7	0.3	288	0	0
1	2022-04-29 04:31:21	2022-04-29 04:32:30	2	0	1	N	249	249	3	3	3	0.5	0	911.87	0.3	918.67	2.5	0
1	2022-05-21 16:47:48	2022-05-21 17:05:47	1	2.4	3	N	239	246	3	31.5	0	0	0	813.75	0.3	845.55	0	0
1	2022-06-12 16:51:46	2022-06-12 17:56:48	9	22	1	N	142	132	2	67.5	2.5	0.5	0	800.09	0.3	870.89	2.5	0

Σημείωση: Όλοι οι χρόνοι εκτέλεσης παρατίθενται σε έναν πίνακα στο τέλος της αναφοράς.

3. Εκτέλεση του Q3 με Dataframe API και RDD API

Τα αποτελέσματα του **Q3** με **Dataframe** API:

start	end	average amount	average distance
2022-01-01 00:00:00	2022-01-16 00:00:00	19.903702637879	5.57641037785201
2022-01-16 00:00:00	2022-01-31 00:00:00	19.0366079138949	4.80484047230941
2022-01-31 00:00:00	2022-02-15 00:00:00	19.5538913279606	5.95048584492812
2022-02-15 00:00:00	2022-03-02 00:00:00	20.1720780936583	6.1857672125677
2022-03-02 00:00:00	2022-03-17 00:00:00	20.6923577131835	6.60698631990843
2022-03-17 00:00:00	2022-04-01 01:00:00	21.1182873078897	5.52478804839661

2022-04-01 01:00:00	2022-04-16 01:00:00	21.5132460928528	5.67922147578719
2022-04-16 01:00:00	2022-05-01 01:00:00	21.4310101744718	5.80009662403329
2022-05-01 01:00:00	2022-05-16 01:00:00	21.9293270019761	6.25531698997756
2022-05-16 01:00:00	2022-05-31 01:00:00	22.8084729445817	8.00062024615197
2022-05-31 01:00:00	2022-06-15 01:00:00	22.4443469769819	6.37273405170607
2022-06-15 01:00:00	2022-06-30 01:00:00	22.3524111322989	6.15420819002069
2022-06-30 01:00:00	2022-07-15 01:00:00	22.2426108408053	5.94605167380302

Τα αποτελέσματα **Q3** με **RDD** API:

fortnight of year	**average amount**	**average distance**
0	20.0284102241603	5.37590081254974
1	18.9541970498349	4.95562905973122
2	19.5560476912666	5.96893444319315
3	20.1137843727025	6.31243736438367
4	20.6522781741791	6.48048543405282
5	21.1080612367874	5.61365266723876
6	21.4972994608394	5.64988251048932
7	21.4767371257031	5.81309671442501
8	21.7861138519553	6.08056896645479
9	22.793742812774	7.99902920404029
10	22.4970248805684	6.436759370744
11	22.3811153371593	6.16659820596056
12	22.019707932065	5.93281039876143

Εδώ είναι σημαντικό να τονίσουμε πως τα αποτελέσματα των δύο ερωτημάτων έχουν μία μικρή απόκλιση η οποία οφείλεται στα ελαχίστως διαφορετικά χρονικά διαστήματα που έχουν οριστεί ως δεκαπενθήμερα.

Στην πρώτη περίπτωση χρησιμοποιήθηκε η συνάρτηση window της python που ορίζει ως δεκαπενθήμερο ακριβώς 15 24ωρα.

Στην δεύτερη περίπτωση (RDD) ωστόσο, χρησιμοποιούμε συνάρτηση που υπολογίζει τον αριθμό της ημέρας του χρόνου και παίρνουμε ως key την απόλυτη διαίρεση του με το 15. Αυτό σημαίνει πως το πρώτο δεκαπενθήμερο ορίζεται από την 1η Ιανουαρίου μέχρι και 14 Ιανουαρίου, δηλαδή το πρώτο διάστημα είναι 14ήμερο και έπειτα 15ήμερα. Όλα τα διαστήματα της δεύτερης περίπτωσης έχουν διαφορά μίας ημέρας από αυτά της πρώτης περίπτωσης, και μερικές φορές έχουν και διαφορά μίας ώρας λόγω της αλλαγής της θερινής ώρας του Μαρτίου.

4. Εκτέλεση των Q4 και Q5

Τα αποτελέσματα του **Q4**:

weekday	**hour**	**passengers**	**hour_rank**
1	0	1.52994565071886	1
1	1	1.5278385673752	2
1	2	1.50807261851912	3
2	0	1.46798877116726	1
2	1	1.44428679168105	2
2	2	1.42319939890515	3
3	0	1.42003138821515	1
3	1	1.41751247400066	2
3	2	1.4104520814694	3
4	1	1.40884802126563	1
4	0	1.40122918571763	2
4	2	1.40114896459586	3
5	23	1.40538231524989	1
5	1	1.40259072852004	2
5	0	1.40103825279883	3
6	23	1.47557691807373	1

6	22	1.44481397620567	2
6	2	1.42305811435244	3
7	23	1.52260676627721	1
7	22	1.50681761940114	2
7	0	1.49931542848985	3

Τα αποτελέσματα του **Q5**:

day	**tip_percentange**	**day_rank**
2022-01-09	45.7867477548721	1
2022-01-31	43.9356358077027	2
2022-01-01	29.0780368613684	3
2022-01-29	24.0595184543701	4
2022-01-16	23.3772999182201	5
2022-02-21	25.9816574527663	1
2022-02-13	24.5720683894025	2
2022-02-09	23.9045356434125	3
2022-02-10	23.3396158993487	4
2022-02-27	23.3006799515465	5
2022-03-18	29.6713416126597	1
2022-03-21	27.5799260249225	2
2022-03-26	22.7088459537216	3
2022-03-05	22.5554613724956	4
2022-03-12	22.1008591108086	5
2022-04-12	48.3688441045034	1
2022-04-02	31.175092883999	2
2022-04-21	30.4486125023628	3
2022-04-03	24.4637277047539	4

2022-04-30	21.9967696599467	5
2022-05-12	32.402658973198	1
2022-05-20	26.0340360903664	2
2022-05-16	23.65911078928	3
2022-05-15	22.0524452470095	4
2022-05-06	21.8320061618845	5
2022-06-13	38.4513699372461	1
2022-06-25	32.9130732926535	2
2022-06-10	27.3976378127807	3
2022-06-16	25.5349757578752	4
2022-06-20	24.2429145935191	5

5. Χρόνοι εκτέλεσης όλων των παραπάνω ερωτημάτων

Στο πάνω μέρος παρουσιάζονται οι χρόνοι εκτέλεσης με 1 worker, ενώ στο κάτω μέρος με 2 workers:

Query	**Type**	**Time**
1	SQL	6.50005960464478
2	SQL	26.2903051376343
3	DF	1.40668287277222
3	RDD	284.630479240417
4	SQL	1.42117123603821
5	SQL	1.30236942768097
Query	**Type**	**Time**
1	SQL	6.73260276317596
2	SQL	28.3596797943115
3	DF	1.52139577865601

3	RDD	287.312871599197
4	SQL	1.54345841407776
5	SQL	1.29263372421265

Να σημειωθεί σε αυτό το σημείο πως για μεγαλύτερη ακρίβεια χρησιμοποιήθηκε ο μέσος χρόνος 10 εκτελέσεων για κάθε ερώτημα.

Παρακάτω φαίνεται ένα screenshot από το ιστορικό των completed applications στο spark Web UI που επιβεβαιώνει τον αριθμό των πυρήνων που χρησιμοποιήθηκαν σε κάθε περίπτωση:

Completed Applications (19)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20230202204517-0018	application.py	4	1024.0 MiB		2023/02/02 20:45:17	user	FINISHED	55 min
app-20230202194612-0017	application.py	2	1024.0 MiB		2023/02/02 19:46:12	user	FINISHED	54 min