Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
Τομέας Επικοινωνιών, Ηλεκτρονικής & Συστημάτων Πληροφορικής
ΑΚΑΔ. ΕΤΟΣ 2022-2023
ΑΘΗΝΑ

Προχωρημένα Θέματα Βάσεων Δεδομένων

Εργαστηριακή Άσκηση

Εργαστηριακή Ομάδα 16

Github repository: https://github.com/MarkRamosS/DataBases

Μάρκος Γεώργιος Ραμός (el18841)

Φαίδρα Αναστασία Ανθοπούλου (el18818)

Ζητούμενα:

1. Σύμφωνα με το tutorial κατεβάσαμε τα απαραίτητα πακέτα για να μπορέσουν να λειτουργήσουν τα Spark & HDFS. Δημιουργήσαμε 2 dataframes και δύο rdd, ένα για taxi trips και ένα για taxi zones. Ένα σύντομο tutorial για την εγκατάσταση hdfs και pyspark βρίσκεται στο τέλος της αναφοράς.

2.

Q1 output:

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

- 1 worker Execution Time: 21.513s
- 2 workers Execution Time: 15.094s

Q2 output:

```
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+
|VendorID|tpep_pickup_datetime|tpep_dropoff_datetime|passenger_count|trip_distance|RatecodeID|store_and_fwd_flag|
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+
|       1| 2022-01-22 11:39:07|  2022-01-22 12:31:09|            1.0|         33.4|       1.0|                 Y|
|       1| 2022-06-12 16:51:46|  2022-06-12 17:56:48|            9.0|         22.0|       1.0|                 N|
|       1| 2022-03-11 20:08:32|  2022-03-11 20:09:45|            1.0|          0.0|       1.0|                 N|
|       1| 2022-05-21 16:47:48|  2022-05-21 17:05:47|            1.0|          2.4|       3.0|                 N|
|       1| 2022-04-29 04:31:21|  2022-04-29 04:32:30|            2.0|          0.0|       1.0|                 N|
|       1| 2022-02-18 02:33:30|  2022-02-18 02:35:28|            1.0|          1.3|       1.0|                 N|
+--------+--------------------+--------------------+---------------+-------------+----------+------------------+
```

```
+------------+------------+------------+------------+-----+-------+----------+------------+
|PULocationID|DOLocationID|payment_type|fare_amount|extra|mta_tax|tip_amount|tolls_amount|
+------------+------------+------------+------------+-----+-------+----------+------------+
|          70|         265|           4|        88.0|  0.0|    0.5|       0.0|       193.3|
|         142|         132|           2|        67.5|  2.5|    0.5|       0.0|      800.09|
|         265|         265|           1|         2.5|  1.0|    0.5|      48.0|       235.7|
|         239|         246|           3|        31.5|  0.0|    0.0|       0.0|      813.75|
|         249|         249|           3|         3.0|  3.0|    0.5|       0.0|      911.87|
|         265|         265|           1|         3.0|  0.5|    0.5|     19.85|        95.0|
+------------+------------+------------+------------+-----+-------+----------+------------+
```

```
+---------------------+------------+--------------------+-----------+-----+
|improvement_surcharge|total_amount|congestion_surcharge|airport_fee|month|
+---------------------+------------+--------------------+-----------+-----+
|                  0.3|       282.1|                 0.0|        0.0|    1|
|                  0.3|      870.89|                 2.5|        0.0|    6|
|                  0.3|       288.0|                 0.0|        0.0|    3|
|                  0.3|      845.55|                 0.0|        0.0|    5|
|                  0.3|      918.67|                 2.5|        0.0|    4|
|                  0.3|      119.15|                 0.0|        0.0|    2|
+---------------------+------------+--------------------+-----------+-----+
```

- 1 worker Execution Time: 75.688s
- 2 workers Execution Time: 40.906s

3. Q3 output:

Με το DataFrame/SQL API:

```
+------------------------------------------+------------------+------------------+
|window                                    |Avg_distance      |Avg_cost          |
+------------------------------------------+------------------+------------------+
|{2022-01-01 00:00:00, 2022-01-16 00:00:00}|5.576410377852007 |19.903702637879007|
|{2022-01-16 00:00:00, 2022-01-31 00:00:00}|4.804840472309411 |19.03660791389491 |
|{2022-01-31 00:00:00, 2022-02-15 00:00:00}|5.950485844928086 |19.553891327978455|
|{2022-02-15 00:00:00, 2022-03-02 00:00:00}|6.1857672125677   |20.17207809365826 |
|{2022-03-02 00:00:00, 2022-03-17 00:00:00}|6.606992664131458 |20.692371844024798|
|{2022-03-17 00:00:00, 2022-04-01 01:00:00}|5.5247907317672045|21.118294613100417|
|{2022-04-01 01:00:00, 2022-04-16 01:00:00}|5.679221475787186 |21.513246092852775|
|{2022-04-16 01:00:00, 2022-05-01 01:00:00}|5.800096624033294 |21.43101017447181 |
|{2022-05-01 01:00:00, 2022-05-16 01:00:00}|6.25531698997756  |21.929327001976045|
|{2022-05-16 01:00:00, 2022-05-31 01:00:00}|8.000620246151973 |22.80847294458172 |
|{2022-05-31 01:00:00, 2022-06-15 01:00:00}|6.372734051706068 |22.444346976981922|
|{2022-06-15 01:00:00, 2022-06-30 01:00:00}|6.154208190020687 |22.352411132298936|
|{2022-06-30 01:00:00, 2022-07-15 01:00:00}|5.946051673803016 |22.242610840805348|
+------------------------------------------+------------------+------------------+
```

- 1 worker Execution Time: 29.046s
- 2 workers Execution Time: 21.672s

Με το RDD API:

```
('2022-06-15 01:00 - 2022-06-30 01:00', 5.670471776299459, 21.840972700932685)
('2022-02-15 00:00 - 2022-03-02 00:00', 5.698919426175817, 19.658923316421927)
('2022-03-02 00:00 - 2022-03-17 00:00', 6.1207327848835655, 20.179757081580487)
('2022-04-01 01:00 - 2022-04-16 01:00', 5.193223139921622, 21.000675728883905)
('2022-05-01 01:00 - 2022-05-16 01:00', 5.7703925689893305, 21.417107652537396)
('2022-04-16 01:00 - 2022-05-01 01:00', 5.314503701411956, 20.917793110614888)
('2022-05-16 01:00 - 2022-05-31 01:00', 7.515930761839871, 22.297428424099806)
('2022-05-31 01:00 - 2022-06-15 01:00', 5.888394673225383, 21.93306053408023)
('2022-06-30 01:00 - 2022-07-01 00:00', 5.4614052160373685, 21.731928766056832)
('2022-01-01 00:00 - 2022-01-16 00:00', 5.089510047904102, 19.393372269428813)
('2022-01-16 00:00 - 2022-01-31 00:00', 4.317320429524202, 18.522393584943963)
('2022-03-17 00:00 - 2022-04-01 01:00', 5.038649440113294, 20.605551779883456)
('2022-01-31 00:00 - 2022-02-15 00:00', 5.463405064437268, 19.040063051667797)
```

- 1 worker Execution Time: 363.786s
- 2 worker Execution Time: 201.216s

4. Q4 output:

day_of_week	hour_of_day	avg(passenger_count)
Wednesday	1	1.4088480212656305
Wednesday	0	1.4012291857176276
Wednesday	2	1.4011489645958584
Tuesday	0	1.4200313882151518
Tuesday	1	1.4175124740006593
Tuesday	2	1.4104520814693964
Friday	23	1.475576918073731
Friday	22	1.444813976205668
Friday	2	1.4230581143524386
Thursday	23	1.4053823152498932
Thursday	1	1.402590728520038
Thursday	0	1.4010382527988254
Saturday	23	1.522606766277207
Saturday	22	1.5068176194011382
Saturday	0	1.4993154284898547
Monday	0	1.4679887711672552
Monday	1	1.4442867916810471
Monday	2	1.4231993989051486
Sunday	0	1.5299456507188562
Sunday	1	1.527838567375201
Sunday	2	1.5080726185191242

- 1 worker Execution Time: 30.299s
- 2 workers Execution Time: 21.743s

Q5 output:

```
+-----+---+-------------------+
|month|day|           avg(tip)|
+-----+---+-------------------+
|    1|  9| 0.4578674775487535|
|    1| 31| 0.4393563580769872|
|    1|  1|0.29078036861366435|
|    1| 29| 0.2405951845436878|
|    1| 16|0.23377299918217617|
|    6| 13|0.38451369937243063|
|    6| 25|0.32913073292653017|
|    6| 10|0.27397637812780157|
|    6| 16| 0.2553497575787421|
|    6| 20|0.24242914593518236|
|    3| 18|0.29671341612657676|
|    3| 21|    0.2757992602492|
|    3| 26|0.22708845953721593|
|    3|  5|0.22555461372495167|
|    3| 12|0.22100859110807622|
|    5| 12|0.32402658973195914|
|    5| 20| 0.2603403609036704|
|    5| 16|0.23659110789277535|
|    5| 15|0.22052445247008398|
|    5|  6|0.21832006161882067|
|    4| 12| 0.4836884410450817|
|    4|  2|0.31175092883996536|
|    4| 21| 0.3044861250236238|
|    4|  3|0.24463727704754118|
|    4| 30|0.21996769659947238|
|    2| 21|0.25981657452765006|
|    2| 13| 0.2457206838940141|
|    2|  9|0.23904535643411468|
|    2| 10|0.23339615899346813|
|    2| 27|0.23300679951545766|
+-----+---+-------------------+
```

- 1 worker Execution Time: 25.405s
- 2 workers Execution Time: 18.104s

Σχολιασμός: Παρατηρούμε ότι σε όλες τις περιπτώσεις ο χρόνος εκτέλεσης με δύο workers είναι μικρότερος από αυτόν για έναν worker. Αυτό είναι λογικό, αφού ο φόρτος εργασίας μοιράζεται μεταξύ των δύο workers. Δεν μειώνεται στο μισό όμως, αφού προστίθεται ο χρόνος μεταφοράς των δεδομένων.

How to install Pyspark & Hdfs:

Αρχικά για master και slave εκτελέσαμε τα παρακάτω βήματα (μόνο τα 3,6,7 για τον slave).

3.Install python3.8:

a. sudo apt update

b. sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev wget libbz2-dev

c. wget https://www.python.org/ftp/python/3.8.0/Python-3.8.0.tgz

d. tar -xf Python-3.8.0.tgz e. cd Python-3.8.0

f. ./configure --enable-optimizations g. make -j 8 h. sudo make altinstall i. python3.8 --version (you should expect: Python 3.8.0)
j. Delete old links
sudo rm -rf /usr/bin/python3.5
sudo rm -rf /usr/bin/python3.5m
sudo rm -rf /usr/lib/python3.5
sudo rm -rf /etc/python3.5
sudo rm -rf /usr/local/lib/python3.5

4. Install pip

a. cd ../
b. wget https://bootstrap.pypa.io/get-pip.py
c. python3.8 get-pip.py

5. Install PySpark

a. pip3.8 install pyspark==3.1.3

6. Install Apache Spark

a. wget https://downloads.apache.org/spark/spark-3.1.3/spark-3.1.3-bin-hadoop2.7.tgz
b. tar -xzf spark-3.1.3-bin-hadoop2.7.tgz
c. nano ~/.bashrc
export SPARK_HOME=/home/user/spark-3.1.3-bin-hadoop2.7
export PATH=$PATH:$SPARK_HOME/sbin
export PYSPARK_PYTHON=python3.8
export PYSPARK_DRIVER_PYTHON=python3.8
export HADOOP_HOME=/home/user/spark-3.1.3-bin-hadoop2.7
d. source ~/.bashrc

7. Install Java

a. sudo apt-get install openjdk-8-jdk
b. java -version (you should expect: openjdk version "1.8.0_292")
8. Setup a Cluster (1 master and 1 worker)
a. Create a network at Okeanos and assign IPs to each VM
b. cd spark-3.1.3-bin-hadoop2.7/conf
c. touch spark-env.sh d. nano spark-env.sh
e. SPARK_MASTER_HOST='192.168.0.2' f. start-master.sh

Έπειτα στο path ~/spark-3.1.3-bin-hadoop2.7/conf/spark-env.sh γράψαμε το αρχείο
“SPARK_MASTER_HOST='192.168.0.2'
SPARK_LOCAL_IP='83.212.81.67'”
Στο ~/spark-3.1.3-bin-hadoop2.7/conf/masters γράψαμε:
192.168.0.2

Και στο ~/spark-3.1.3-bin-hadoop2.7/conf/workers γράψαμε:
192.168.0.2

192.168.0.1

Ώστε το spark να γνωρίζει ποιος είναι ο master και ποιοι είναι οι workers.

Για να ενεργοποιήσουμε έναν worker, εκτελούμε την παρακάτω εντολή:
start-worker.sh spark://192.168.0.2:7077

Για να μπορεί το σύστημα να τρέξει με δύο workers, χρειάζεται να φτιάξουμε ένα NAT δίκτυο. Για να το φτιάξουμε αυτό, χρησιμοποιήσαμε τα παρακάτω βήματα ενός περσινού tutorial που βρήκαμε εδώ:
“https://cdn.discordapp.com/attachments/775222267860090890/1071022394132672613/94783bab6577036a.docx”
Αρχικά μέσω
`sudo vim /etc/hosts`
Προσθέτουμε στο αρχείο hosts τις γραμμές και στον master και στον slave:
“192.168.0.2 master
192.168.0.1 slave”
Προσθέτουμε στην αρχή του αρχείου “/etc/resolv.conf” τη γραμμή:
“nameserver 8.8.8.8” για να μπορεί να συνδεθεί στο internet.

Έπειτα εκτελούμε στον master το bash script:
“
```
#!/bin/bash
echo "Enabling ipv4 forwarding (cleaning old rules)"
# flushing old rules -- USE WITH CARE
iptables --flush
iptables --table nat --flush
# MASQUERADE each request form the inside to the outer world
iptables -t nat -A POSTROUTING -j MASQUERADE
# enable IPv4 packet forwarding in the kernel
echo 1 > /proc/sys/net/ipv4/ip_forward
echo "Master is now operating as router"
```
“
Και στον slave το αντίστοιχο bash script:
“
```
#!/bin/bash
echo "Enabling ipv4 forwarding (cleaning old rules)"
# flushing old rules -- USE WITH CARE
iptables --flush
iptables --table nat --flush
# MASQUERADE each request form the inside to the outer world
iptables -t nat -A POSTROUTING -j MASQUERADE
```

```
# enable IPv4 packet forwarding in the kernel
echo 1 > /proc/sys/net/ipv4/ip_forward
echo "Master is now operating as router"
“
```