ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΣΤΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

Συστήματα Διαχείρισης Δεδομένων Μεγάλης Κλίμακας – Πλήρους Φοίτησης 2017-2018

ΔΙΔΑΣΚΩΝ: Ιωάννης Κωτίδης (kotidis@aueb.gr)

Βοηθός : Μπλέτσος Ντριτάν (dritanbleco@aueb.gr)

Μέλη της ομάδας

ΔΙΟΜΗΔΗΣ ΠΑΝΑΓΙΩΤΗΣ ΑΜ: Μ317003

ΦΙΛΤΙΣΑΚΟΣ ΣΠΥΡΙΔΩΝ ΑΜ: Μ317016

ΦΟΥΡΑΚΗΣ ΣΠΥΡΙΔΩΝ ΑΜ: Μ317017

Εργασία Hadoop/Spark

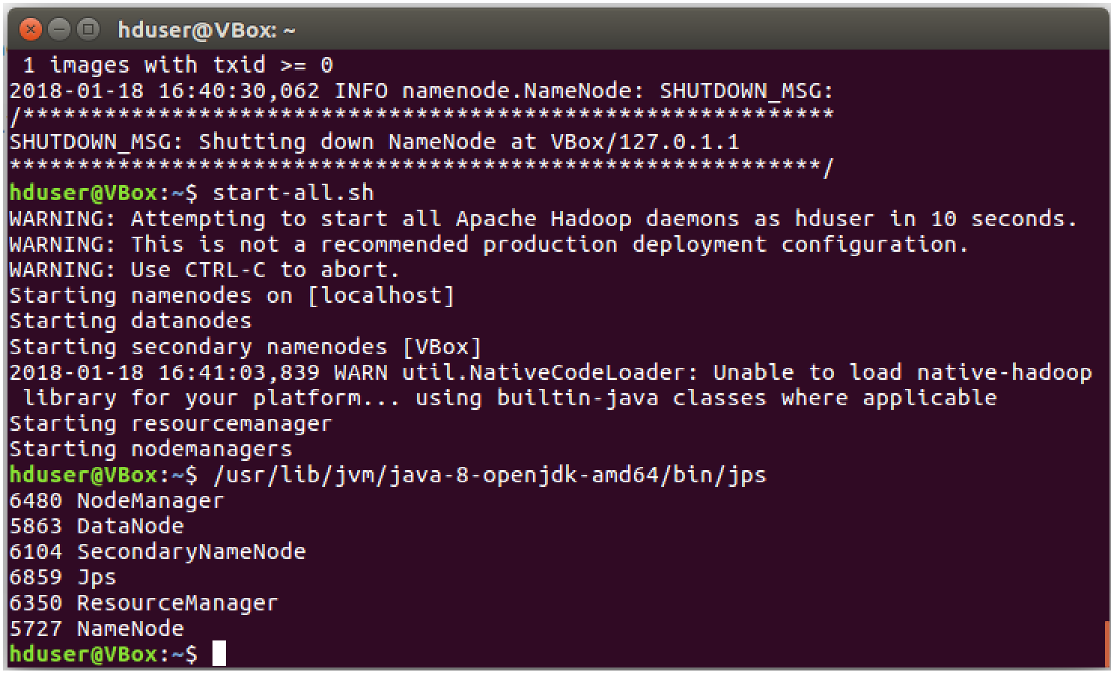
Αναφορά:

Αφού κατεβάσουμε τα Hadoop και Spark από τους ιστότοπούς τους , και κάνουμε όλες τις διαδικασίες για να μπορούν να εκκινήσουν σωστά

Πριν ξεκινήσουμε όλα τα services δίνουμε την εντολή

hdfs namenode -format

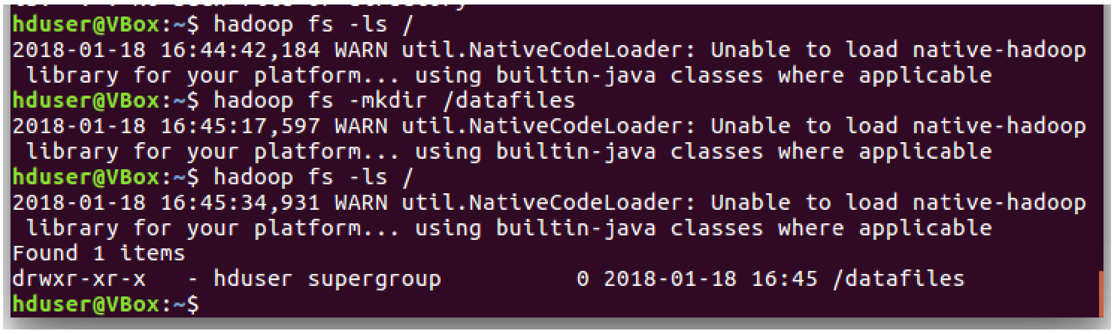
Ελέγχουμε ότι τρέχουν και τα έξι



```
 1 images with txid >= 0
2018-01-18 16:40:30,062 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at VBox/127.0.1.1
************************************************************/
hduser@VBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [VBox]
2018-01-18 16:41:03,839 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Starting resourcemanager
Starting nodemanagers
hduser@VBox:~$ /usr/lib/jvm/java-8-openjdk-amd64/bin/jps
6480 NodeManager
5863 DataNode
6104 SecondaryNameNode
6859 Jps
6350 ResourceManager
5727 NameNode
hduser@VBox:~$ 
```

Φτιάχνουμε ένα directory για να εισάγουμε τα αρχεία με τα δεδομένα

hadoop fs -mkdir /datafiles

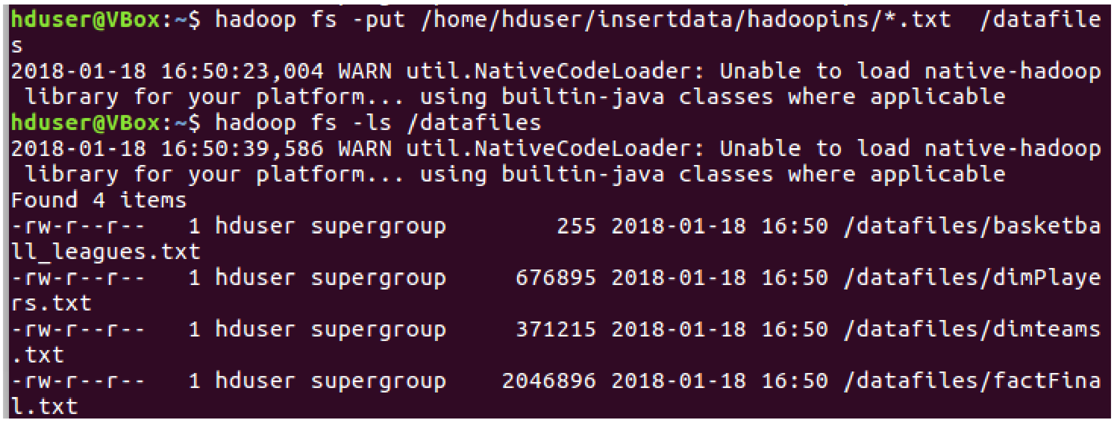


```
hduser@VBox:~$ hadoop fs -ls /
2018-01-18 16:44:42,184 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
hduser@VBox:~$ hadoop fs -mkdir /datafiles
2018-01-18 16:45:17,597 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
hduser@VBox:~$ hadoop fs -ls /
2018-01-18 16:45:34,931 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2018-01-18 16:45 /datafiles
hduser@VBox:~$
```

και τα εισάγουμε

hadoop fs -put /home/hduser/insertdata/hadoopins/*.txt /datafiles

στην συνέχεια κάνουμε -ls να δούμε ένα έχουν εισαχθεί

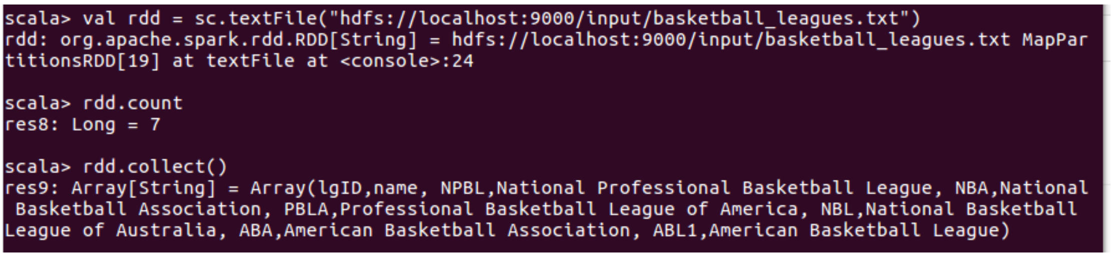


```
hduser@VBox:~$ hadoop fs -put /home/hduser/insertdata/hadoopins/*.txt  /datafile
s
2018-01-18 16:50:23,004 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
hduser@VBox:~$ hadoop fs -ls /datafiles
2018-01-18 16:50:39,586 WARN util.NativeCodeLoader: Unable to load native-hadoop
 library for your platform... using builtin-java classes where applicable
Found 4 items
-rw-r--r--   1 hduser supergroup        255 2018-01-18 16:50 /datafiles/basketba
ll_leagues.txt
-rw-r--r--   1 hduser supergroup     676895 2018-01-18 16:50 /datafiles/dimPlaye
rs.txt
-rw-r--r--   1 hduser supergroup     371215 2018-01-18 16:50 /datafiles/dimteams
.txt
-rw-r--r--   1 hduser supergroup    2046896 2018-01-18 16:50 /datafiles/factFina
l.txt
```

(λόγω ενός bug το interface δεν λειτουργεί)

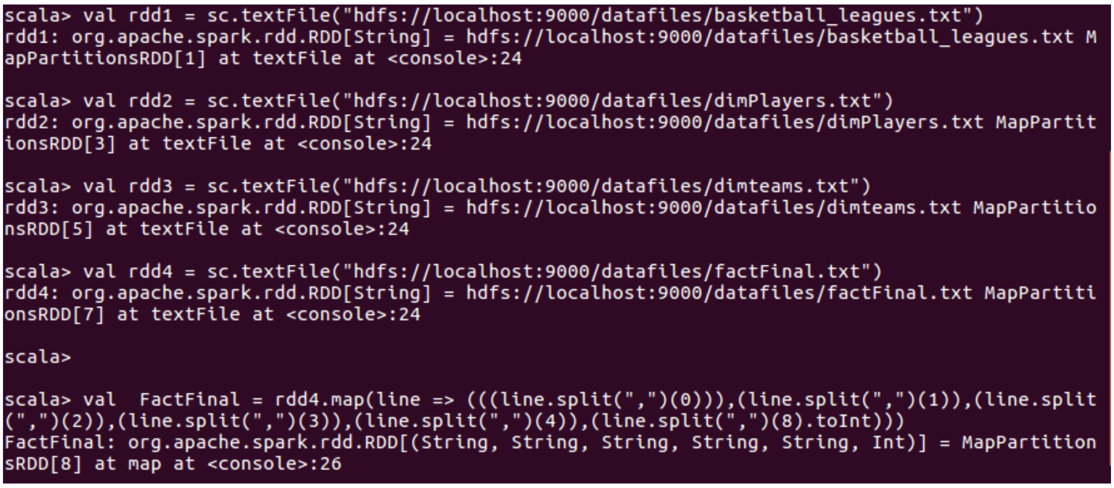
Στην συνέχεια δοκιμάζω να κάνω μια μεταβλητή για να τεστάρω ότι έχει περάσει σωστά.

val rdd = sc.textFile("hdfs://localhost:9000/input/basketball_leagues.txt")



```
scala> val rdd = sc.textFile("hdfs://localhost:9000/input/basketball_leagues.txt")
rdd: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/input/basketball_leagues.txt MapPar
titionsRDD[19] at textFile at <console>:24

scala> rdd.count
res8: Long = 7

scala> rdd.collect()
res9: Array[String] = Array(lgID,name, NPBL,National Professional Basketball League, NBA,National
 Basketball Association, PBLA,Professional Basketball League of America, NBL,National Basketball
League of Australia, ABA,American Basketball Association, ABL1,American Basketball League)
```

Συνεχίζω και δημιουργω 4 rdds

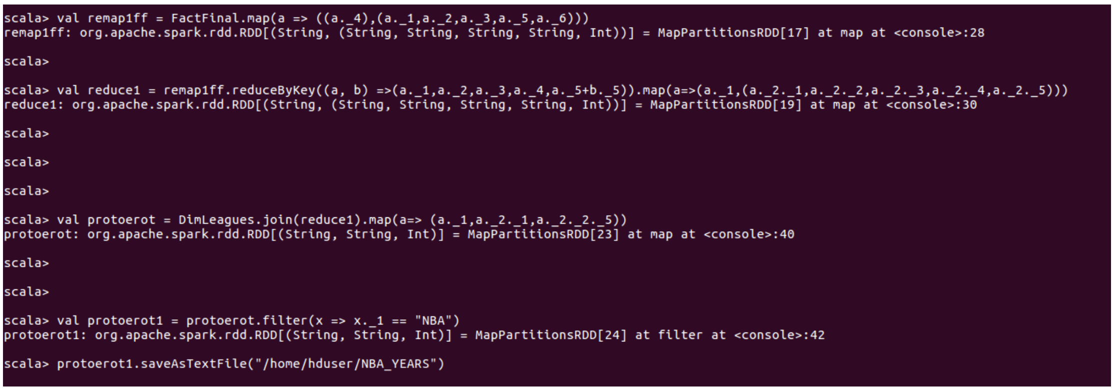


```
scala> val rdd1 = sc.textFile("hdfs://localhost:9000/datafiles/basketball_leagues.txt")
rdd1: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/datafiles/basketball_leagues.txt M
apPartitionsRDD[1] at textFile at <console>:24

scala> val rdd2 = sc.textFile("hdfs://localhost:9000/datafiles/dimPlayers.txt")
rdd2: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/datafiles/dimPlayers.txt MapPartit
ionsRDD[3] at textFile at <console>:24

scala> val rdd3 = sc.textFile("hdfs://localhost:9000/datafiles/dimteams.txt")
rdd3: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/datafiles/dimteams.txt MapPartitio
nsRDD[5] at textFile at <console>:24

scala> val rdd4 = sc.textFile("hdfs://localhost:9000/datafiles/factFinal.txt")
rdd4: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/datafiles/factFinal.txt MapPartiti
onsRDD[7] at textFile at <console>:24

scala>

scala> val  FactFinal = rdd4.map(line => (((line.split(",")(0))),(line.split(",")(1)),(line.split
(",")(2)),(line.split(",")(3)),(line.split(",")(4)),(line.split(",")(8).toInt)))
FactFinal: org.apache.spark.rdd.RDD[(String, String, String, String, String, Int)] = MapPartition
sRDD[8] at map at <console>:26
```

Επομεν βήμα είναι να θέσουμε ερωτηματα

Το πρωτο ερώτημα θα είναι ποσά καλαθια έχουν μπει στο πρωταθλημα NBA όλα τα χρόνια

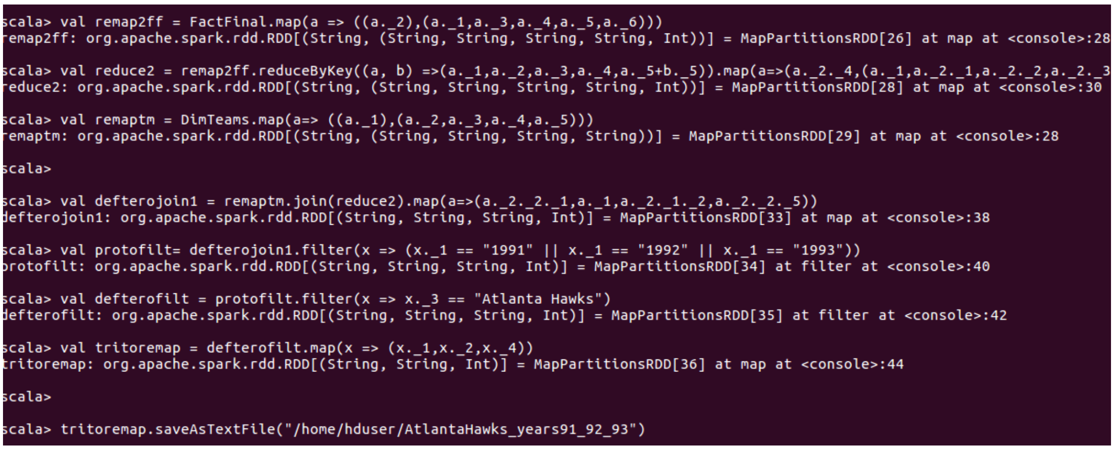
Για να το βρούμε θα ανακατασκευάσουμε το rdd του fact table ώστε να έχει πρωτευον κλειδι τα πρωταθλήματα ώστε να κανουμε reduce by key



```
scala> val remap1ff = FactFinal.map(a => ((a._4),(a._1,a._2,a._3,a._5,a._6)))
remap1ff: org.apache.spark.rdd.RDD[(String, (String, String, String, String, Int))] = MapPartitionsRDD[17] at map at <console>:28

scala>

scala> val reduce1 = remap1ff.reduceByKey((a, b) =>(a._1,a._2,a._3,a._4,a._5+b._5)).map(a=>(a._1,(a._2._1,a._2._2,a._2._3,a._2._4,a._2._5)))
reduce1: org.apache.spark.rdd.RDD[(String, (String, String, String, String, Int))] = MapPartitionsRDD[19] at map at <console>:30

scala>

scala>

scala>

scala> val protoerot = DimLeagues.join(reduce1).map(a=> (a._1,a._2._1,a._2._2._5))
protoerot: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[23] at map at <console>:40

scala>

scala>

scala> val protoerot1 = protoerot.filter(x => x._1 == "NBA")
protoerot1: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[24] at filter at <console>:42

scala> protoerot1.saveAsTextFile("/home/hduser/NBA_YEARS")
```

Το επόμενο ερώτημα θα είναι όλα τα καλάθια που εβαλαν οι «Atlanta Hawks» τις χρονιες 91-92-93.

Πάλι ανακατασκευάζω τα rdd ώστε να μπορέσω να κάνω reduce by key



```
scala> val remap2ff = FactFinal.map(a => ((a._2),(a._1,a._3,a._4,a._5,a._6)))
remap2ff: org.apache.spark.rdd.RDD[(String, (String, String, String, String, Int))] = MapPartitionsRDD[26] at map at <console>:28

scala> val reduce2 = remap2ff.reduceByKey((a, b) =>(a._1,a._2,a._3,a._4,a._5+b._5)).map(a=>(a._2._4,(a._1,a._2._1,a._2._2,a._2._3
reduce2: org.apache.spark.rdd.RDD[(String, (String, String, String, String, Int))] = MapPartitionsRDD[28] at map at <console>:30

scala> val remaptm = DimTeams.map(a=> ((a._1),(a._2,a._3,a._4,a._5)))
remaptm: org.apache.spark.rdd.RDD[(String, (String, String, String, String))] = MapPartitionsRDD[29] at map at <console>:28

scala>

scala> val defterojoin1 = remaptm.join(reduce2).map(a=>(a._2._2._1,a._1,a._2._1._2,a._2._2._5))
defterojoin1: org.apache.spark.rdd.RDD[(String, String, String, Int)] = MapPartitionsRDD[33] at map at <console>:38

scala> val protofilt= defterojoin1.filter(x => (x._1 == "1991" || x._1 == "1992" || x._1 == "1993"))
protofilt: org.apache.spark.rdd.RDD[(String, String, String, Int)] = MapPartitionsRDD[34] at filter at <console>:40

scala> val defterofilt = protofilt.filter(x => x._3 == "Atlanta Hawks")
defterofilt: org.apache.spark.rdd.RDD[(String, String, String, Int)] = MapPartitionsRDD[35] at filter at <console>:42

scala> val tritoremap = defterofilt.map(x => (x._1,x._2,x._4))
tritoremap: org.apache.spark.rdd.RDD[(String, String, Int)] = MapPartitionsRDD[36] at map at <console>:44

scala>

scala> tritoremap.saveAsTextFile("/home/hduser/AtlantaHawks_years91_92_93")
```