09: Spark on Zeppelin – convert DataFrames to RDD and RDD to DataFrame

Posted on September 11, 2018

Pre-requisite: Docker is installed on your machine for Mac OS X (E.g. $ brew cask install docker) or Windows 10. Docker interview Q&As. This extends setting up Apache Zeppelin Notebook.

Important: It is not a best practice to mutate values or to use RDD directly as opposed to using Dataframes. Use of groupBy operation on RDD is also discouraged as it causes wide shuffling. There are alternative solutions like using “withColumn” on Dataframe to

300+ Java Interview FAQs

300+ Java FAQs 🔥

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs 🔥

Tutorials - Big Data

TUT - ⏯ Starting Big Data

TUT - Starting Spark & Scala

add a new column, UDF functions, window functions, explode function, etc to achieve the desired results more efficiently. There will be edge cases where use of the RDDs will give you more flexibility to achieve the desired outcome. The examples shown below is for the illustration purpose only.

Step 1: Pull this from the docker hub, and build the image with the following command.

```
$ docker pull apache/zeppelin:0.7.3

```

You can verify the image with the "docker images" command.

Step 2: Run the container with the above image.

```
$ docker run --rm -it -p 8080:8080 apache/zeppelin

```

Step 3: Open Zeppelin notebook via a web browser "http:localhost:8080". Create a note book with "spark" as a default interpreter.

This extends Spark convert DataFrames to RDD[Row] and RDD[Row] to DataFrame with use of a case class named Employee.

```
%spark


import org.apache.spark.sql.types._
import org.apache.spark.sql.Row
import org.apache.spark.rdd.RDD

case class Employee (id: Integer, name: String, lo

val employees = Seq(
    Employee(1, "John", "USA", 50000.0),
    Employee(2, "Peter", "AU",60000.0),
    Employee(3, "Sam", "AU", 60000.0),
    Employee(4, "Susan", "USA", 50000.0),
    Employee(5, "David", "USA", 70000.0),
    Employee(6, "Elliot", "AU", 50000.0)
)

val employee_df = spark.createDataFrame(
  spark.sparkContext.parallelize(employees)
)

employee_df.show()

val rdd  = employee_df.rdd
                     .groupBy(row => row(2)) //gr
                     .flatMap(x => {         //fl
                         //x is a Tuple, x._1 is 
                         // filter salary 60k or 
                         for (row <- x._2 if row.
                     })

val df2 = sqlContext.createDataFrame(rdd.map { ca
df2.show()

```

Output:

```
import org.apache.spark.sql.types._
import org.apache.spark.sql.Row
import org.apache.spark.rdd.RDD
defined class Employee
employees: Seq[Employee] = List(Employee(1,John,U
employee_df: org.apache.spark.sql.DataFrame = [id
+---+------+--------+-------+
| id|  name|location| salary|
+---+------+--------+-------+
|  1|  John|     USA|50000.0|
|  2| Peter|      AU|60000.0|
|  3|   Sam|      AU|60000.0|
|  4| Susan|     USA|50000.0|
|  5| David|     USA|70000.0|
|  6|Elliot|      AU|50000.0|
+---+------+--------+-------+
rdd: org.apache.spark.rdd.RDD[org.apache.spark.sq
df2: org.apache.spark.sql.DataFrame = [id: int, n
+---+-----+--------+-------+
| id| name|location| salary|
+---+-----+--------+-------+
|  2|Peter|      AU|60000.0|
|  3|  Sam|      AU|60000.0|
|  5|David|      USA|70000.0|
+---+-----+---------+-------+

```

‹ 08: Spark on Zeppelin – convert DataFrames to RDD[Row] and RDD[Row] to DataFrame

10: Spark on Zeppelin – union, udf and explode ›

Disclaimer