Apache Spark Handbook

Sasikumar Venkatesh

May 31, 2020

1 Verify Spark Installation

Let us verify the following software are installed?

- Java 8
- Spark 2.3.0 and above versions

Step - 1: To verify the java installed, run the below command in Terminal.

```
$ java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed
    mode)
```

Step - 2: To verify the spark installed, follow the below instructions

(a) Make sure $SPARK_HOME variable is set and $SPARK_HOME/bin to the PATH variable

```
$ echo $SPARK_HOME
/Users/s0v00eo/hpc/spark
```

(b) Running Spark-Shell and Spark UI

```
$ spark-shell --master local[*]
```

The spark-shell REPL interactive mode looks as shown in Figure 1.

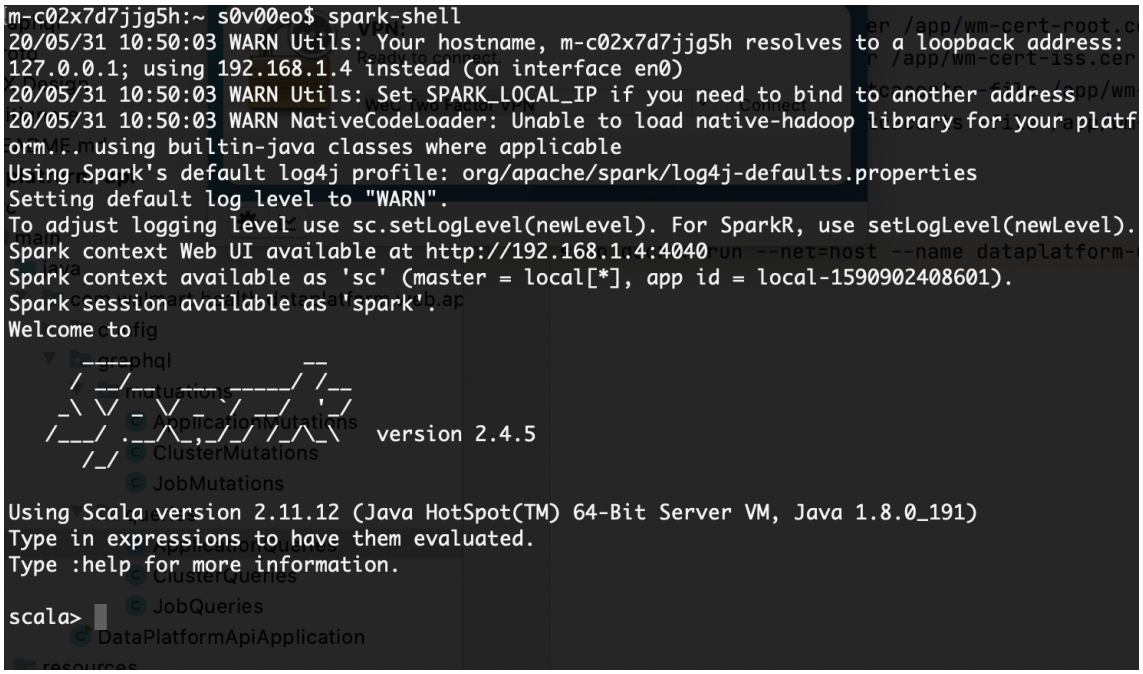

Figure 1: Spark-Shell Interactive Scala Shell

(c) Now open the browser and check the following URL for spark UI. `http://localhost:4040` (You can see the exact port in the previous step screen shot. In this case You can see Spark Context Web UI available at http://192.168.1.4:4040)

The Spark Web UI is available as shown in Figure 2.

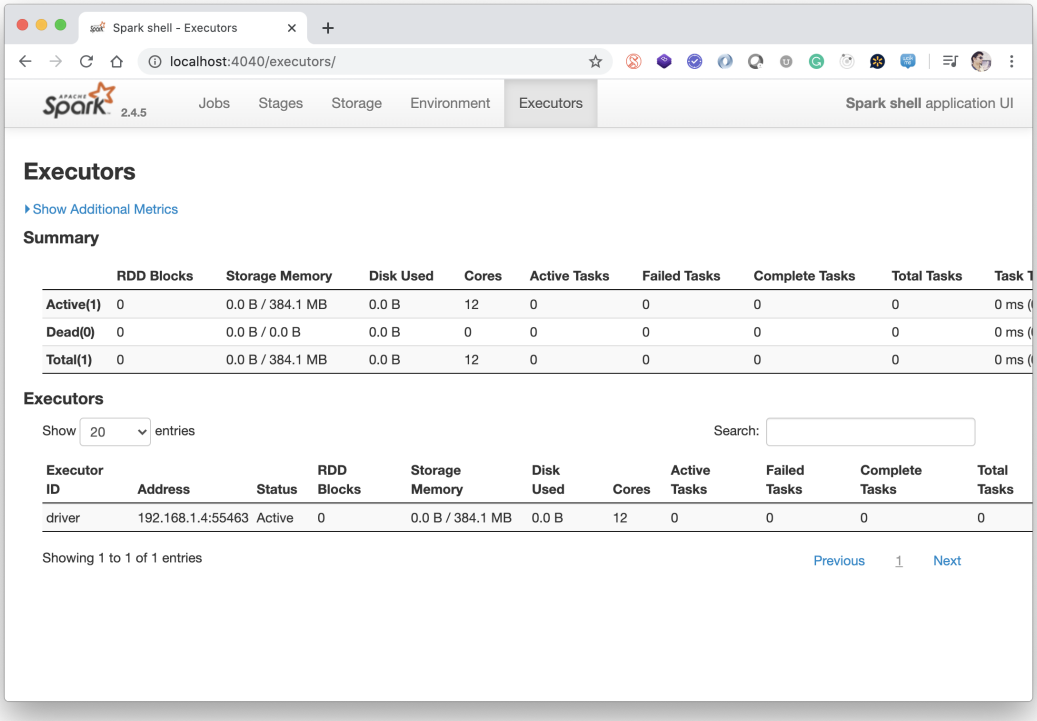

Figure 2: Spark Web UI

2 Excercises

Let's run the simple first exercise in Spark Scala.

This exercise is to show how spark can use multiple cores and process the data parallelly.

Problem: Consider we have an Array of numbers in the range from 1 to 1 million. Let us find how many numbers are divisible by 13 in the above range. In single processor program looks as below.

```
int countOf(int[] array, int number) {
   int count = 0;
   for(int i=0;i<array.length; i++) {
      if(array[i] % number == 0)
         count++;
   }
   return count;
}
```

Let us see, how can we run the same exercise with Multiple processors. To see the Number of processors available in your computer, run the below command in the terminal. (For Linux OS only)

```
$ nproc
   12
```

Run the below scala commands in spark-shell.

```
var list = 1 to 1000000

val listRdd = sc.parallelize(list)

listRdd.getNumPartitions

val count=listRdd.filter(_ % 13 == 0).collect.size
count: Int = 76923
```

We got the count. How do we make sure that, this is something ran on multiple processors? Let us verify that, Open the Spark Web UI: `http://localhost:4040`

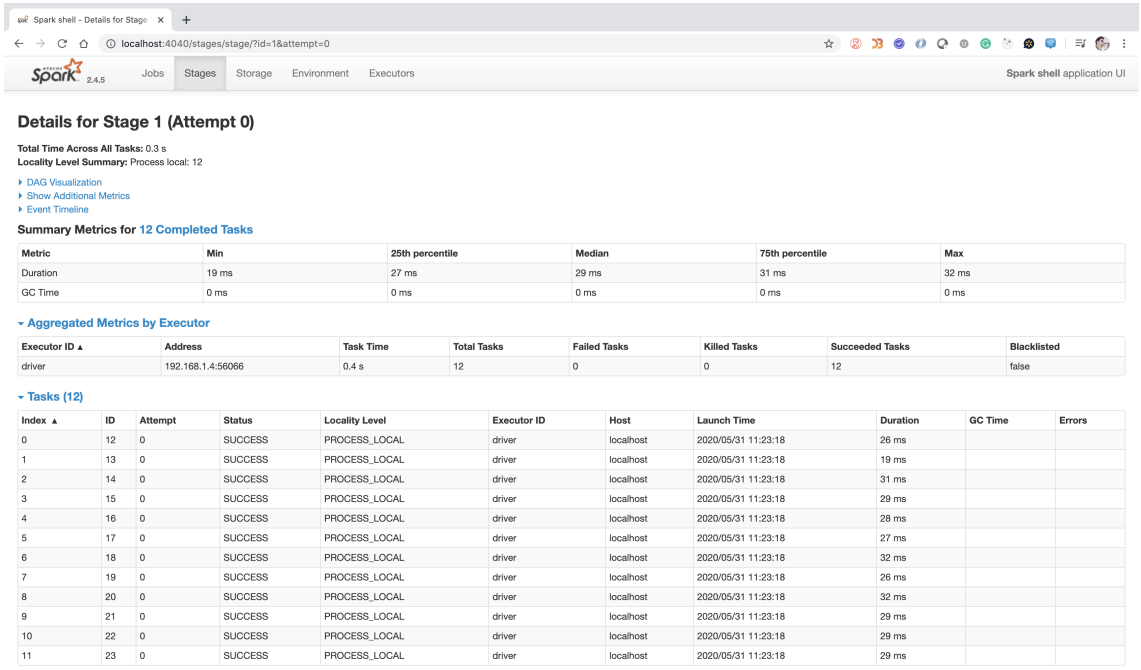

Figure 3: Spark UI for Executor Logs

Click on the event timeline and see the way it executed in parallel. You can increase the number of partitions to create in an RDD.

```
val listRdd = sc.parallelize(list, 24) //(some arbitrary number)
```

Similarly experiment with increasing the number of partitions, changing the used cores for processing while opeing the spark-shell

```
$ spark-shell --master local[2] //Two Cores are used
```

2.1 Word Counting Problem

Consider we have an E-book (The Adventures of Sherlock Holmes) text file which contains 10000+ lines of text and we wanted to find each word and its count.

1. Copy the book.txt from dataset folder to your preferable location. In this Example, /workshop/Dataset/book.txt
2. Run the below Scala commands. (spark-shell)

```
var file=sc.textFile("~/workshop/Dataset/book.txt")

file.collect

var words = file.flatMap(line => line.split(" "))

words.collect

var wordMap = words.map(word => (word,1))

wordMap.collect

var results = wordMap.reduceByKey(_ + _)

results.collect

//write the reults into another file
results.saveAsTextFile("~/workshop/Dataset/book-out")
```

For every exercise to understand how Spark it executes, visit `http://localhost:4040` or whichever the port is running to check the spark logs.

2.2 Sample Data Analysis using Spark

In the Dataset folder we have covid-19 India dataset. File-Name (covid_19_india.csv). This is a csv file with comma separated values. The below are the description of the dataset.

```
|-- Sno: string (nullable = true)
|-- Date: string (nullable = true)
|-- Time: string (nullable = true)
|-- State: string (nullable = true)
|-- ConfirmedIndianNational: string (nullable = true)
|-- ConfirmedForeignNational: string (nullable = true)
|-- Cured: string (nullable = true)
|-- Deaths: string (nullable = true)
|-- Confirmed: string (nullable = true)
```

There are 2486 rows of data given at different period of time. Lets do some analyses on this data.

1. Find the state wise count of confirmed cases of covid-19.
2. Using the IndividualDetails.csv, Find the no. of patients whose age >50
3. Using the IndividualDetails.csv, Find the no. of patients who travelled from Wuhan
4. Using the IndividualDetails.csv, Find the no. of patients who got transmitted by locally or by family members
5. Using the IndividualDetails.csv, Find the no. of patients who are affected by COVID who travelled from US/UK

3 Spark SQL

This section contains the hand-on for Spark SQL.

3.1 Read/Write File Data Source

Lets read a file using data Apache Spark Source APIs.

```
var covidDf = spark.read.option("header","true")
              .csv("~/workshop/Dataset/covid-19/IndividualDetails.csv")

covidDf.printSchema

covidDf.createOrReplaceTempView("covid")

var res=spark.sql("select count(*) from covid where age>=50")

res.show

// Write data to a file
res.saveAsTextFile("~/workshop/output/out-1")
```

3.2 Read/Write MySQL Datasource

When it comes to MySQL database, we need to place the mysql-jdbc jar into the spark jars folder, so that the spark can pick it and run.

1. Download the MySQL jar or copy from the Jars folder and paste into $SPARK_HOME/jars folder
2. If Spark-Shell is already open close it with Ctrl + C and run spark-shell again.

```
val jdbcDF = spark.read
.format("jdbc")
.option("url", "jdbc:mysql://localhost:3306/sample_db")
.option("dbtable", "test")
.option("user", "username")
.option("password", "password")
.load()

//Write to JDBC
import java.util.Properties
val connectionProperties = new Properties()

connectionProperties.put("user", "")
connectionProperties.put("password", "")

jdbcDF.withColumnRenamed("table", "table_number")
.write
.jdbc(jdbcUrl, "table", connectionProperties)
```

3.3 Excercises

1. Calculate the number of testing labs available in india by state. Using covid-19 dataset
2. Calculate the different number of lab types by state and all over india
3. Find the count of male and female patients in tamilnadu got affected due to travelled to Delhi
4. Find some intersting pattern or anamoly from the dataset for Tamil Nadu State
5. Find some intersting pattern or anamoly from the dataset for Mathya Pradesh State.