Apache Spark

Overview

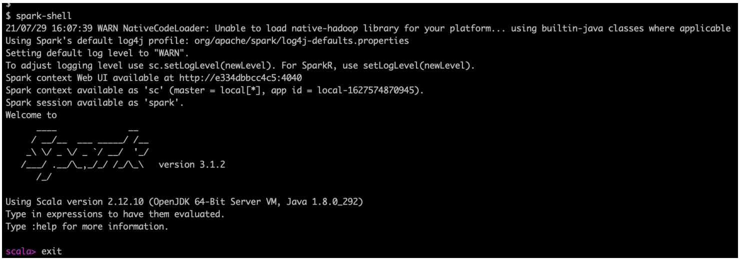
Apache spark is an open source lightning fast unified analytics engine for large scale processing and machine learning. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs.

Quick Start

Installation

- Download a docker image : `docker pull bitnami/spark`
- Once downloaded , run the image : `docker run -d --name spark bitnami/spark`
- Open the container in interactive mode: `docker exec -it spark shell`
- Create a text file using command: `cd /tmp && echo "apple orrage guava mango pineapple peaches apple mango oragne orange \n apple orrage guava mango pineapple peaches apple mango oragne orange " > words.txt`

- Run: spark-shell . It should open a scala based spark shell like below.



```
$
$ spark-shell
21/07/29 16:07:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Spark context Web UI available at http://e334dbbcc4c5:4040
Spark context available as 'sc' (master = local[*], app id = local-1627574870945).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.1.2
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala> exit
```

Write a word-count Program

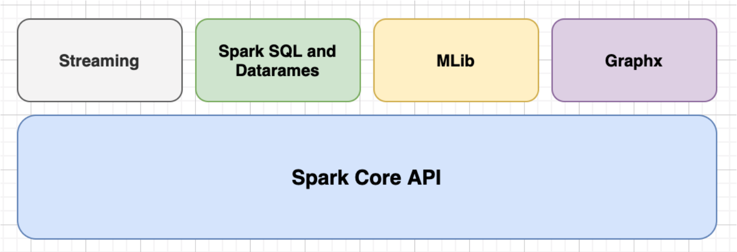
- Inside spark-shell Run this below code

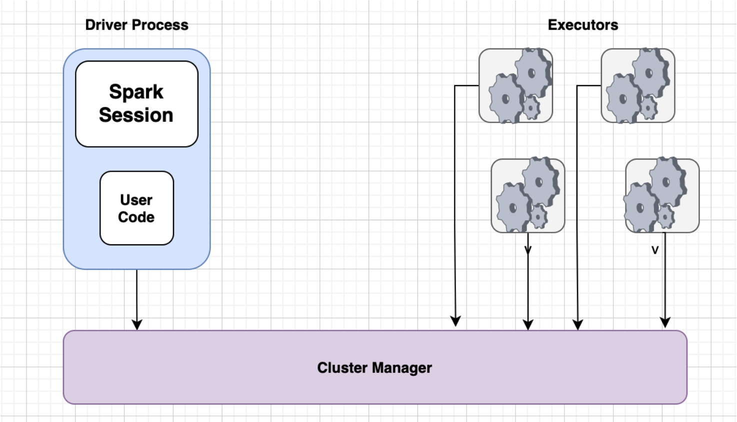
```
val textFile = spark.read.textFile("/tmp/words.txt")
textFile: org.apache.spark.sql.Dataset[String] = [value: string]

val wordCounts = textFile.flatMap(line => line.split(" ")).groupByKey(identity).count()
wordCounts: org.apache.spark.sql.Dataset[(String, Long)] = [key: string, count(1): bigint

wordCounts.collect()
res0: Array[(String, Long)] = Array((orange,2), (apple,4), (mango,4), (orrage,2), (pineap
```

Architecture

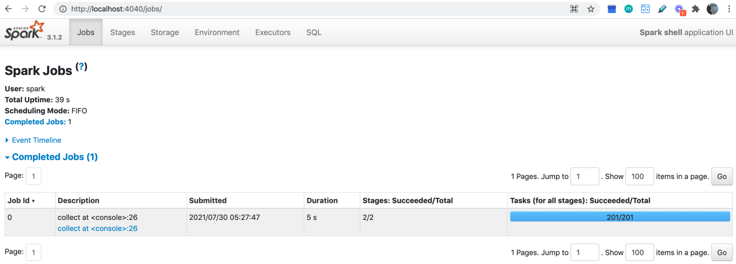




Highlights of Spark

- Speed
- Ease of Use
- A Unified Engine

- Spark Web UI (localhost:4040)



Spark Jobs (?)

User: spark
Total Uptime: 39 s
Scheduling Mode: FIFO
Completed Jobs: 1

Event Timeline

Completed Jobs (1)

Page: 1 · 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	collect at <console>:26 collect at <console>:26	2021/07/30 05:27:47	5 s	2/2	201/201

Page: 1 · 1 Pages. Jump to 1 . Show 100 items in a page. Go

Core Concepts

- DAG & Lazy Evaluation
- Transformations and actions
- RDD & Dataframe
- SparkSession

Reference

https://spark.apache.org/docs/latest/quick-start.html

https://www.oreilly.com/library/view/spark-the-definitive/9781491912201/

https://alvinalexander.com/scala/collection-scala-flatmap-examples-map-flatten/

https://spark.apache.org/docs/latest/configuration.html

https://spark.apache.org/docs/latest/api/java/index.html