1. [What is the difference between spark context and spark session?](https://www.quora.com/What-is-the-difference-between-spark-context-and-spark-session)

Prior to version 2.0, [**SparkContext**](http://data-flair.training/blogs/sparkcontext-in-apache-spark-tutorial/) was the entry point for spark jobs and Starting from **Apache Spark** 2.0, Spark Session is the new entry point for Spark applications.

Prior to version 2.0 in order to use APIs of SQL,HIVE , and Streaming, separate contexts need to be created like below:

val conf=newSparkConf()

val sc = new SparkContext(conf)

val hc = new hiveContext(sc)

val ssc = new streamingContext(sc).

Spark 2.0 onwards in order to use APIs of SQL, HIVE, and Streaming, no need to create separate contexts as sparkSession includes all the APIs.

**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.appName('hack\_find').getOrCreate()

**from** **pyspark.sql** **import** SparkSession \\python

**import org.apache.spark.sql.SparkSession \\Scala**

val spark: SparkSession = SparkSession.builder

.appName("My Spark Application") // optional and will be autogenerated if not specified

.master("local[\*]") // only for demo and testing purposes, use spark-submit instead

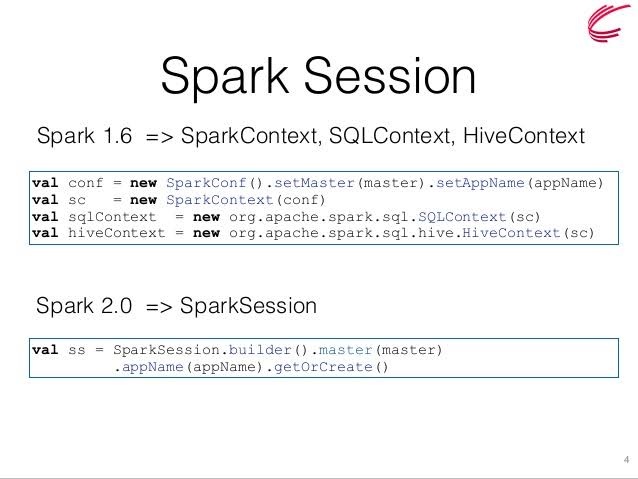
.enableHiveSupport() // self-explanatory, isn't it?

.config("spark.sql.warehouse.dir", "target/spark-warehouse")

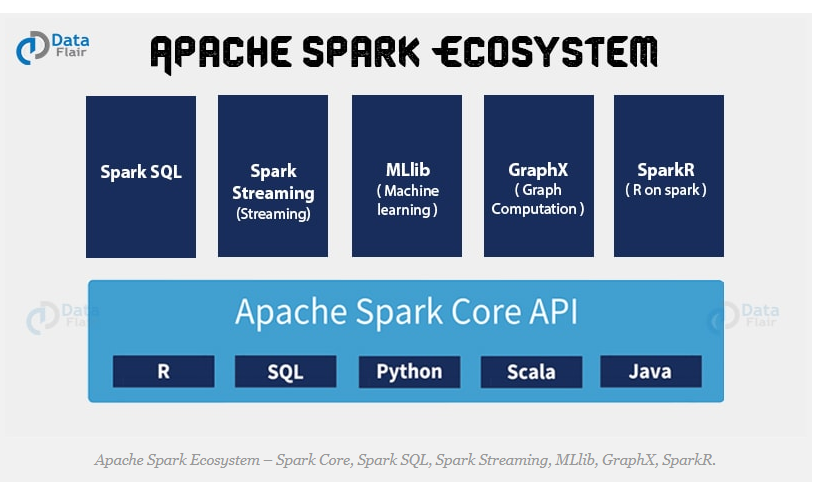
.getOrCreate

And stop the current SparkSession using stop method.

spark.stop



1. What are components of SPARK?



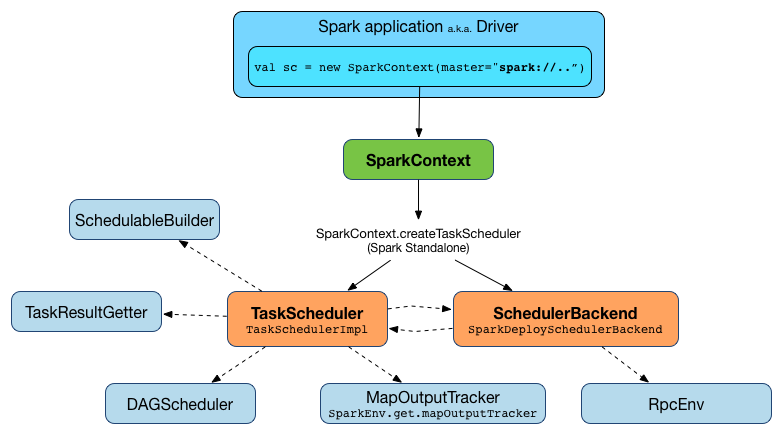
1. What are functions of Spark core?

* Distributed Task Dispatching
* Job Scheduling
* I/O Functions
* Fault recovery.
* In-memory computation (Overcomes the snag of **MapR**)
* Observing Spark Cluster.

1. What is Spark Scheduler?

TaskScheduler — Spark Scheduler

TaskScheduler is responsible for submitting tasks for execution in a Spark application (per scheduling policy).



Spark comes with the following custom TaskSchedulers:

**TaskSchedulerImpl** — the default TaskScheduler (that the following two YARN-specific TaskSchedulers extend).

**YarnScheduler** for Spark on YARN in client deploy mode.

**YarnClusterScheduler** for Spark on YARN in cluster deploy mode.

The source of TaskScheduler is available in org.apache.spark.scheduler.TaskScheduler.

1. [What is the difference between spark scheduling mode and application queue in spark?](https://stackoverflow.com/questions/38874777/what-is-the-difference-between-spark-scheduling-mode-and-application-queue-in-sp)

When you submit a Scala code to spark, spark client will interact with yarn and launch a yarn application. This application will be duty on all the jobs in your Scala code. In most cases, each job corresponds to an Spark Action like reduce (), collect (). Then, the problem comes, how to schedule different jobs in this application, for example, in your application, there a 3 concurrent jobs comes out and waiting for execution? To deal with it, Spark make the scheduler rule for job, including FIFO and Fair. That is to say , spark scheduler ,including FIFO and Fair, is on the level of job, and it is the spark Application Master which is do the scheduling work.

To set a [Fair Scheduler](https://spark.apache.org/docs/latest/job-scheduling.html#fair-scheduler-pools) pool for a JDBC client session, users can set the spark.sql.thriftserver.scheduler.pool variable:

**SET** spark.**sql**.thriftserver.scheduler.pool=accounting;

1. Why Spark is faster than Hadoop MapR?

* MapReduce is good for processing data in the offline mode but when it comes to online data processing in a streaming manner Spark has some distinct advantages since it does not persists the data in the disk but rather does it in the memory itself for faster read and write
* Spark offers in-memory caching abstraction that is ideal for workloads wherein multiple operations need to access the same input data
* MapReduce starts a new Java Virtual Machine for each of the task while Spark has an executor Java Virtual Machine deployed on each node due to this it is a simple task of making a Remote Procedure Call and thus it is extremely fast
* Spark utilizes Direct Acyclic Graph that helps to do all the optimization and computation in a single stage rather than multiple stages in the MapReduce model
* The DAG also helps the Spark engine to leave the reducing task behind.
* The core processing methodology of Spark is the RDD which is resilient distributed dataset which is an immutable distributed collection of objects for computing data on different nodes with logical partitions

1. Spark "long running" tasks low performance, how to investigate?

<https://community.hortonworks.com/questions/85451/spark-long-running-tasks-low-performance-how-to-in.html>

1. What is Driver Program in Spark? Give Example?

The spark driver is the program that declares the transformations and actions on RDDs of data and submits such requests to the master.

In practical terms, the driver is the program that creates the SparkContext, connecting to a given Spark Master. In the case of a local cluster, like is your case, the master\_url=spark://<host>:<port>

Its location is independent of the master/slaves. You could co-located with the master or run it from another node. The only requirement is that it must be in a network addressable from the Spark Workers.

This is how the configuration of your driver looks like:

val conf = new SparkConf()

.setMaster("master\_url") // this is where the master is specified

.setAppName("SparkExamplesMinimal")

.set("spark.local.ip","xx.xx.xx.xx") // helps when multiple network interfaces are present. The driver must be in the same network as the master and slaves

.set("spark.driver.host","xx.xx.xx.xx") // same as above. This duality might disappear in a future version

val sc = new spark.SparkContext(conf)

// etc...

1. What is Pair RDD?

Pair RDDs can be created by running a map() function that returns key or value pairs.

val pairs = lines.map(x => (x.split(" ")(0), x))

Actions Available on Pair RDDs are as follows :

**countByKey() :** Count the number of elements for each key pair.

**collectAsMap() :** Collect the result outputs as a map to provide easy lookup.

**lookup(key) :** Return all values associated with the provided key pair.

1. How to create broadcast variable in Spark?

The return variable of sc.broadcast is of type Broadcast<String[]> and not String[]. When you want to access the value, you simply call value() on the variable. From your example it would be like:

Broadcast<String[]> broadcastedFieldNames = sc.broadcast(fieldNames)

DataFrame df = sourceFrame.toDF(broadcastedFieldNames.value())

Note, that if you are writing this in Java, you probably want to wrap the SparkContext within the JavaSparkContext. It makes everything easier and you can then avoid having to pass a ClassTag to the broadcast function.

1. Why Lazy evaluation by Spark?

In Spark, lazy evaluation comes when Spark transformation occurs. **Transformations** are lazy in nature meaning when we call some operation in RDD, it does not execute immediately. Spark maintains the record of which operation is being called(Through [**DAG**](http://data-flair.training/blogs/directed-acyclic-graph-dag-in-apache-spark/)).

Since transformations are lazy in nature, so we can execute operation any time by calling an action on data. Hence, in lazy evaluation data is not loaded until it is necessary.

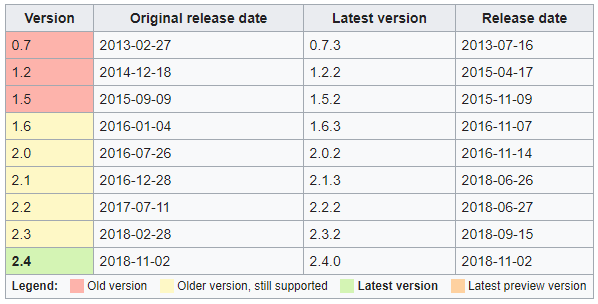
1. **How to start pyspark interactive shell?**

Type *pyspark* at the CLI

1. How to Start Spark Scala interactive shell?

Type *spark-shell* at CLI

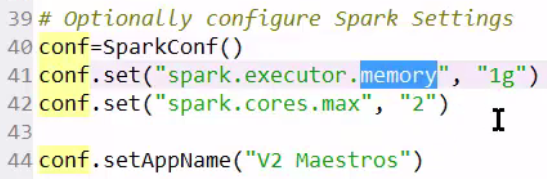
1. Which version of Spark have you worked on?
2. Which version of Spark have you worked on?



1. Which version of python is supported by Spark latest?

Python 2.7 – 3.4

1. How to set memory and cores for my pyspark script?



1. **How to create local sparkContext?**



1. Cache vs Persists in Saprk?

<https://stackoverflow.com/questions/26870537/what-is-the-difference-between-cache-and-persist>

* We can mark an RDD to be persisted using the persist() or cache() methods on it.
* Each persisted RDD can be stored using a different storage level
* The cache() method is a shorthand for using the default storage level, which is StorageLevel.MEMORY\_ONLY (store deserialized objects in memory).
* Use persist() if you want to assign a storage level other than MEMORY\_ONLY to the RDD

<http://spark.apache.org/docs/latest/rdd-programming-guide.html>

1. Map vs Flat Map function in Spark?

<https://stackoverflow.com/questions/22350722/what-is-the-difference-between-map-and-flatmap-and-a-good-use-case-for-each>

map: It returns a new RDD by applying given function to each element of the RDD. Function in map returns only one item.

flatMap: Similar to map, it returns a new RDD by applying a function to each element of the RDD, but output is flattened

If you are asking the difference between RDD.map and RDD.flatMap in Spark, map transforms an RDD of size N to another one of size N . eg.

myRDD.map(x => x\*2)

for example, if myRDD is composed of Doubles .

While flatMap can transform the RDD into anther one of a different size: eg.:

myRDD.flatMap(x =>new Seq(2\*x,3\*x))

which will return an RDD of size 2\*N or

myRDD.flatMap(x =>if x<10 new Seq(2\*x,3\*x) else new Seq(x)

1. Can Spark be run without Hadoop? Yes
2. Spark performance optimization?

<https://medium.com/teads-engineering/spark-performance-tuning-from-the-trenches-7cbde521cf60>

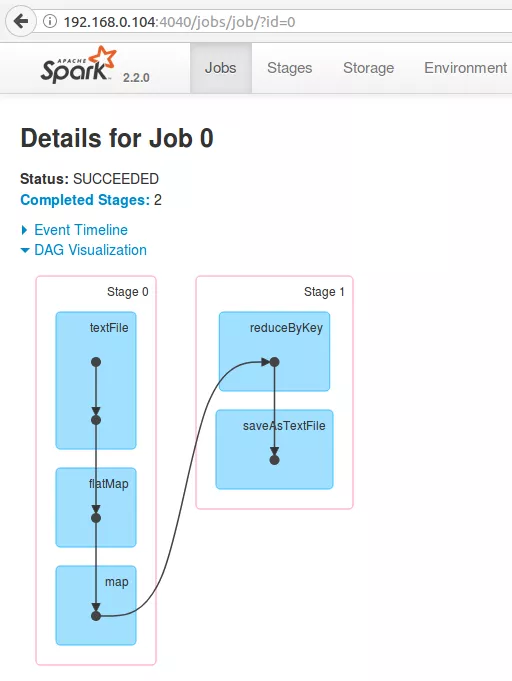
* Avoid User-Defined Functions (UDFs) as much as possible.

def currency = udf(  
(currencySub: String, currencyParent: String) ⇒  
 Option(currencyParent) match {  
 case Some(curr) ⇒ curr  
 case \_ ⇒ currencySub  
 }  
)

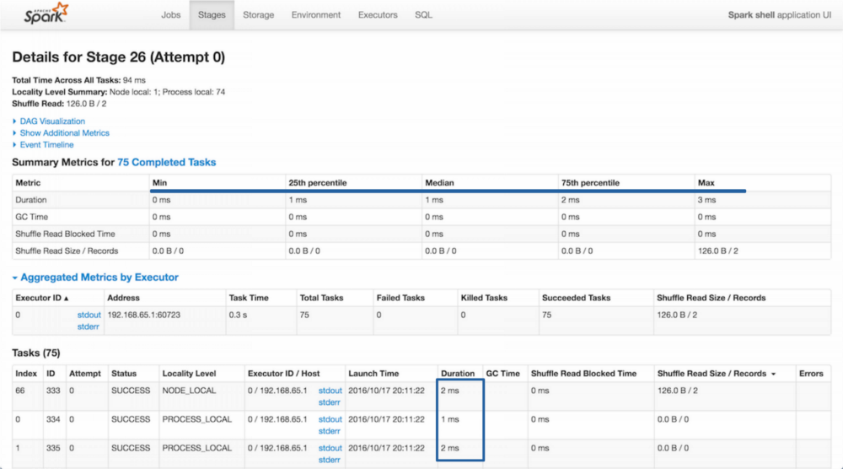
Avoid User-Defined Aggregate Functions (UDAFs)

df.stat.approxQuantile(“value”, Array(0.5), 0)

* Analysing Spark’s execution plan
* Shuffles are expensive
* Reducing the number of stages is a obvious
* The Directed Acyclic Graph (DAG) in Spark UI can also be used to visualize the task repartition in each stage.



* Highly imbalanced datasets:
* It’s fairly easy to check min, max and median duration in Spark UI. Here is a balanced example:



* Inappropriate use of caching
* Broadcasting
* Cloud Related Optimizaiton
  + -- driver-memory 1g
  + -- driver-cores 1
  + -- executor-memory 20g = executor heap size
  + -- executor-cores 4
  + -- num-executors $executorCount

1. DataSets Vs DataFrames in Spark?

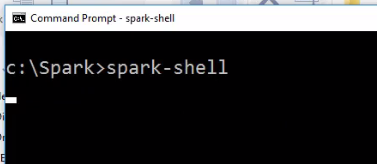
<https://techvidvan.com/tutorials/apache-spark-dataframe-vs-datasets/>

1. How to write SQL in pySpark?
2. What is SparkSQL?
3. What is Serialization and Deserialization? When do we use it in Spark?
4. How to run code on SPARK 1 or 2

Export SPARK\_MAJOR\_VERSON=2

1. What is difference between hiveCtx.sql vs pySpark.sql
2. What transformations and Actions have you done on data frame?
3. What do we mean by schema in database?
4. **How to build a sparkSession in Spark 2.0 using pyspark?**
5. **Multiline comment in python?**
6. **How to create pipeline in Spark?**
7. How to Start Spark?

Type spark-shell from spark directory, or from anywhere.



1. What is Scala REPL?
2. How to load scala programming file in Spark?

: load my\_first\_program

1. How to control messages displaying on spark console? How to set logging level in Spark?

Edit your conf/log4j.properties file and change the following line:

log4j.rootCategory=INFO, console

to  
log4j.rootCategory=ERROR, console

Another approach would be to:

Start spark-shell and type in the following:

import org.apache.log4j.Logger

import org.apache.log4j.Level

Logger.getLogger("org").setLevel(Level.OFF)

Logger.getLogger("akka").setLevel(Level.OFF)

1. How to unzip file in Ubuntu?

tar xvf file\_name

1. What is difference between Spark Submit and Load method to run codes in Spark?
2. What is **libsvm** format file?
3. What are some sectors using SparkML? List SparkML projects.

• Fraud detection.

• Web search results.

• Real-time ads on web pages

• Credit scoring and next-best offers.

• Prediction of equipment failures.

• New pricing models.

• Network intrusion detection.

• Recommendation Engines

• Customer Segmentation

• Text Sentiment Analysis

• Predicting Customer Churn

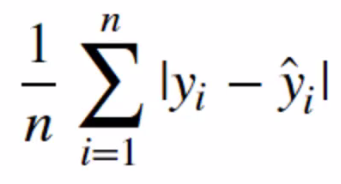
• Pattern and image recognition.

• Email spam filtering.

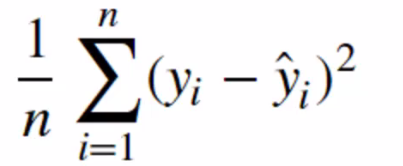
• Financial Modeling

1. What are different types of errors and their significance in Data Science?

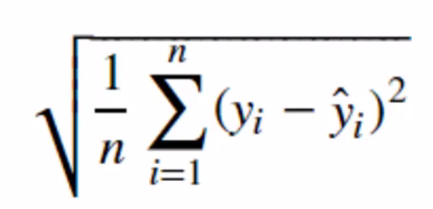
* Mean Absolute Error



* Mean Square Error: More popular because it punishes larger error by squaring



* Root Mean Squared Error: More popular compared to MSE, because units are interpretable.



1. What is difference between list and tuple in Scala?

tup has type (Int, String, Double), so you can get data back with its correct type: tup.\_1 has type Int.

list has type List[Any], so you've lost all type information: list(0)'s type is Any.

Don't use Any (or List[Any], etc.) unless you have to; certainly don't use it when a tuple will do.

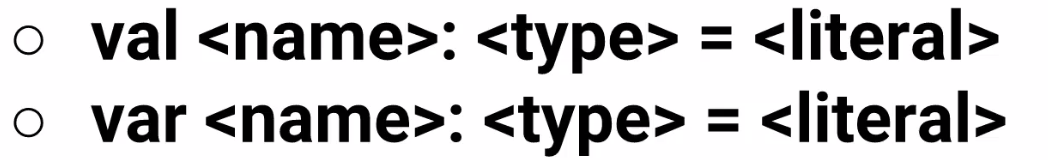
Lists are immutable.

1. What is difference between range and Range?
2. **Difference between Method and Function?**
3. Difference between val and var in Scala?

Values (val) and Variables (var):

* Values (val) are immutable, once they are assigned they cannot be changed. Err: reassignment to Val
* Variables (var) can be reassigned.
* Note, when reassigning you must use the same Data Type! Err: type mismatch

General Format when creating an object:

* 
* Later on we will see that Scala can also infer the data type, so we will skip defining it.
* Let's get started with some examples!

1. What is registerTempTable in Spark? What is createOrReplaceTempView in Spark?
2. Sliding window and batch in Spark? What is **Window length** and **Sliding interval** in Spark?
3. Why we need to import pyspark.SQL to work with DataFrames?

Because DataFrame is based on SparkSQL.

1. What is use of tilde operator in pySpark dataframes?

df.filter( (df["Close"] < 200) & ~(df['Open'] < 200) ).show()

1. How to Start a simple Spark Session?

**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.appName("walmart").getOrCreate()

1. How to Load the Walmart Stock CSV File, have Spark infer the data types?

df = spark.read.csv('walmart\_stock.csv',header=**True**,inferSchema=**True**)

1. What does the Schema look like?

df.printSchema()

1. How to Print out the first 5 columns?

*# Didn't strictly need a for loop, could have just then head()*

**for** row **in** df.head(5):

print(row)

print('**\n**')

1. What is significance of R2 in regression model?
2. What is significance of Serialization and Deserialization in data processing?
3. How to start Spark thrift serve from pyspark?

.config('spark.sql.hive.thriftServer.singleSession', True)

1. How to create RDD?
2. Map vs Flatmap in Spark?
3. What is Speculative Execution in Spark?

spark. Speculation = false

If set to "true", performs speculative execution of tasks

spark. speculation. interval = 10ms

spark. speculation. interval = 100ms

spark. speculation.multiplier 1.5

spark. speculation.muttiplier 1.5

spark. speculation. quantile 0.75

Fraction of tasks which must be complete before speculation is enabled for a particular stage

1. Differnece between Hash, Range and Custom Partitioner?

<https://acadgild.com/blog/partitioning-in-spark>