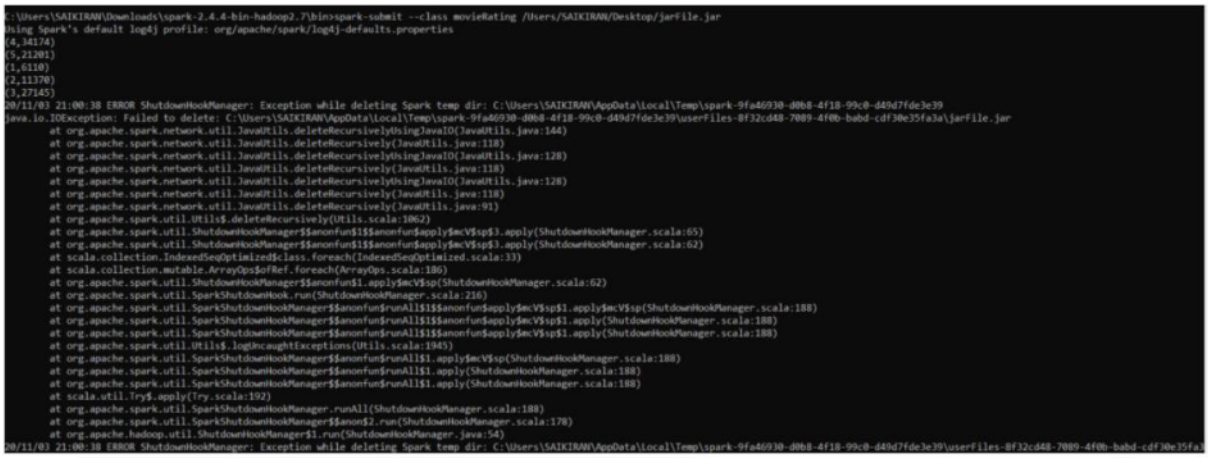
Week-11 FAQs

W11:1 is it necessary to write spark.stop() at the end of every spark job?

Ans: it's a good practise to close connections at the end.

W11:2 Rittu is getting the following error while executing spark on windows using jar - "Exception while deleting Spark temp dir". he had checked on stack overflow but people there are skeptic about compatibility of spark with windows and suggested to have some changes in log4j file. Please suggest any approach towards the issue.



Ans:

1. One reason, Raj found on stackoverflow is that if we have a take or count action after saveAsText or other format then this issue arises because that action will have a lock on tp directory and shutDownHookManager will not be able to delete that.
2. Else u can try that thread to sleep, so that while one thread is sleeping ,other will write to disc and till thread one wakes up 2nd thread will release the lock on that dir.
3. If nothing worked, leave it for now. as per this jira issue https://issues.apache.org/jira/browse/SPARK-12216 its a bug in spark with windows system or in java process as this issue does not occur on unix based systems

W11:3 In spark Dataframe code, how to remove header?

Ans: If the csv file has a header (column names in the first row) then set header=true. This will use the first row in the csv file as the dataframe's column names. Setting header=false (default option) will result in a dataframe with default column names: _c0, _c1, _c2, etc. Setting this to true or false should be based on your input file.

W11:4 what if we have more than one line as header. In my previous project we use to receive files with first line as filename with date n time n system info Second line with description of file Third line as column name Wanted to know how to deal with it

Ans: Below are two solutions

1. We first get the lines which we don't want from raw data using rdd.take(3) method (input 3 as in our example 3 lines of headers) and then apply filter and thus skipping 3 lines .
2. We can assign an index to the each row (using zipWithIndex in rdd or adding a column index with value as monotonically_increasing_id function in DF) and filtering rows with id >3

W11:5 when to use Map and MapPartitions transformation? Raj understands MapPartitions works faster when compared to Map and is useful for critical applications . However, are there any particular use cases?

Ans: There can be a case where we need to apply some complex operation to each row, then we create the helper objects. But initializing them for each row would be very costly so we create them per partition/worker node and can use it for all rows in that partition.
e.g.

1. Open db connection and close for each partition instead of each row,
2. Creating a CSV parser using new CSVParser() for each partition to parse line in RDD

W11:6 Dataframe doesn't show compile time error. But we miss .load while reading data from source and creating a Dataframe. Then when we try to execute .show() it gives a compile time error. Are there some specific errors which are not shown during compile time rather than run time?

Ans: -if you miss the .load like this

```
val inputDF = spark.read
.format("csv")
.option("path", "/E:/input.csv")
```

Then the datatype of inputDF variable would be a DataFrameReader, which does not have methods like show, write etc, hence we get errors. These methods are defined in DataFrame class and if we write .load which is defined in DataFrameReader and whose return type is DataFrame then only .show will work

Also when we say dataframes does not give compile time error, we mean that we would not be getting error at compile time even if we give column name which does not exists in DF as while giving input column names are just the strings. Whereas in case of DS which is strongly typed will give error at compile time.

W11:7 Why spark.read in dataframes is an action? Action means it is computed immediately however on Spark-shell, Sam found that spark.read is still a lazy operation and is executed only when an action is called, then why does read appear as a job in the DAG?

Ans: Dataframes work slightly differently. It wants to know the number of partitions (metadata) well in advance to optimize things better. Also lets say when you say infer schema. it has to scan the entire data to infer the data types

W11:8 For converting an rdd to dataframe we can use 2 methods. createdataframe and rdd.toDF, however createdataframe does not need this import spark.implicits._ to be imported. my question is which is a better option among the 2 and why?

Ans: Looking at syntax aspects toDF() syntax

is simpler than createdataframe method

W11:9 can we join two different formats (csv and json) in spark?

Ans: Yes, File formats are only related when we read files from disk. So once both are read it will be dataframes. we can join two such dataframes.

W11:10 Shristy have 2 questions :- 1. When defining an explicit schema using StructType, we can define the nullable property as true/false in the 3rd parameter.

However when she set it to false then schema should be changed to not allow nulls
but schema is unchanged for 1st column.
Below is the code:

```
  val ordersSchema = StructType(List(
    StructField("orderId",IntegerType, false), // false means NULLs
    are not allowed
    StructField("orderDate", TimestampType),
    StructField("oCustId",IntegerType),
    StructField("status",StringType)
))
 val ordersDF = spark.read
              .format("csv")
              .option("header", true)
              .schema(ordersSchema)
              .option("path", "E:/Big_Data/Week 11-Spark/orders.csv")
              .load
  ordersDF.printSchema()
```

Schema output is given as screenshot
2.When defining explicit schema using StructType and StructField , we specify Spark provided datatypes but why convert them back to scala datatypes if spark datatypes are more optimized?

```
root
 |-- orderId: integer (nullable = true)
 |-- orderDate: timestamp (nullable = true)
 |-- oCustId: integer (nullable = true)
 |-- status: string (nullable = true)
```

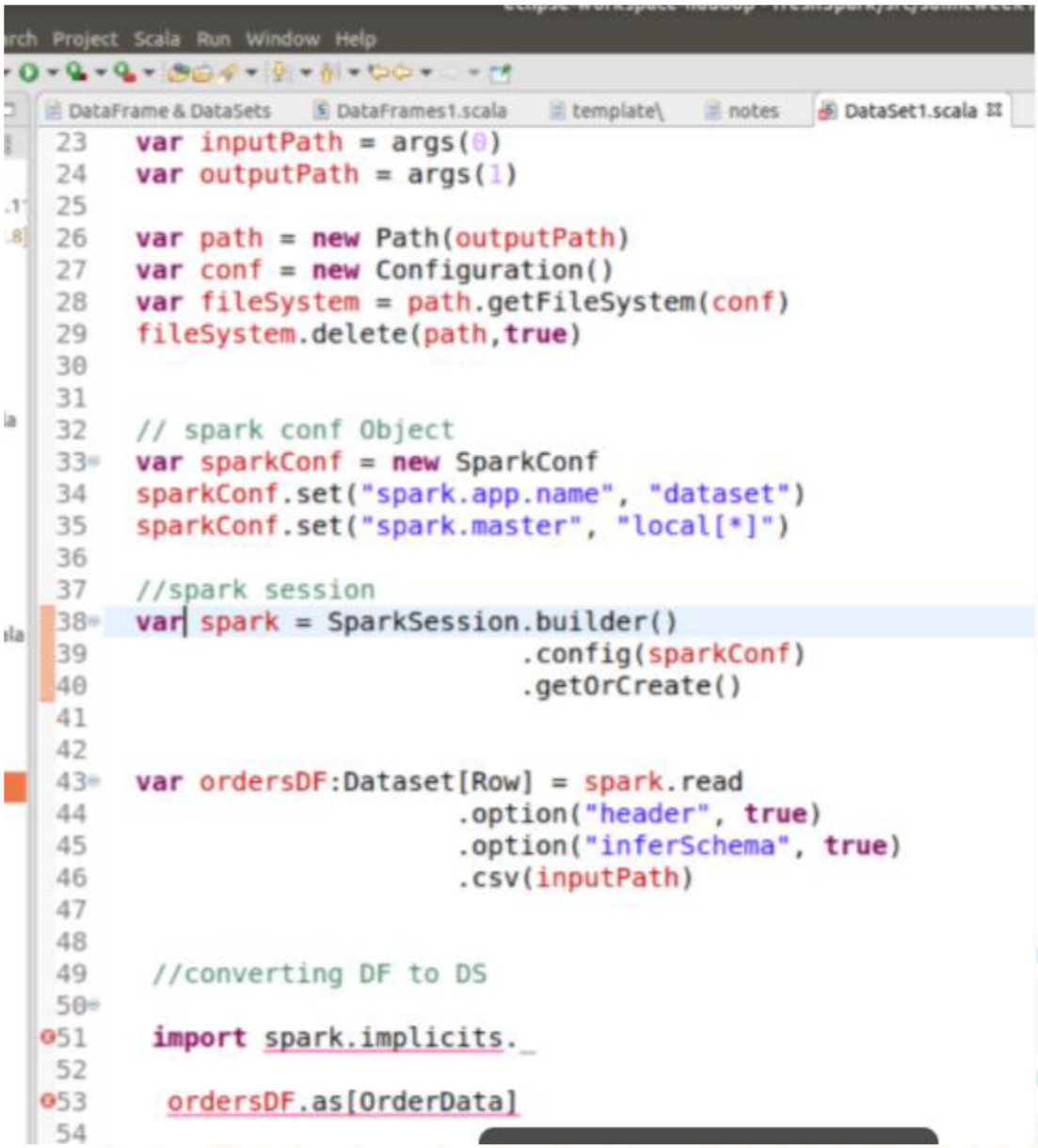
AnsThis is a open issue by apache , I guess
[SPARK-10848] Applied JSON Schema Works for json RDD but not when loading json file - ASF JIRA (apache.org)
There are other ways also to change the nullable property. can you try once to see if they are working or not?https://stackoverflow.com/questions/33193958/change-nullable-property-of-column-in-spark-dataframe
python - PySpark: StructField(..., ..., False) always returns `nullable=true` instead of `nullable=false` - Stack Overflow

Ffor question 2: Spark is a framework, in the end everything should run on jvm. So definitely it has to be a programming language that run's that's why eventually in an optimized manner the scala code is built

W11:11 Why is a new stage formed when there is shuffling of data involved? [Or] What advantages do we get by breaking tasks into stages?

Ans: Whenever shuffling and sort involved data moved from memory to disk after that wide transformation will start to work by loading back the data to memory. Whenever you're loading the data from HDFS a new stage will be created. Also to do wide transformation like reducebykey or groupbykey you need same keys in same reducer machine so to get same keys shuffle and sort required hence data will send to hdfs from spark memory

W11:12 In below pic, if Shyam create spark session using var instead val, he is getting an error while i try to change from DF to DS



Ans: this import requires a stable identifier (here spark session constant) which only val can give, hence we don't get error for val

W11:13 what is a stable identifier ?

Ans: Basically, a ***stable identifier*** is simply a name which is bound statically to a value. They are required for certain tasks (like pattern matching) so the compiler can make sense of the code it is generating and the types it is inferring.
So basically it needs something like constant.

W11:14 Is there a way to sort a column which is in RDD form in descending order ?

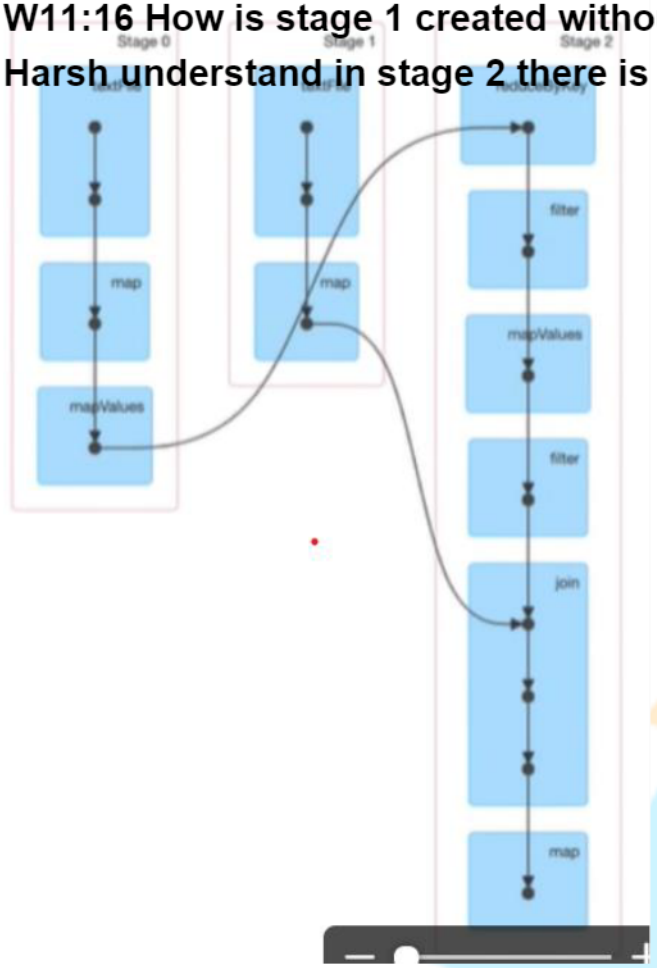
Ans: In rdd, we have <sortByKey(._1, false)
.sort(desc("colName"))

W11:15 When we use show or take or write actions in spark will all the data be sent to driver? If not, then why when we use collect does all the data go to the driver?

Ans: When we use show, take or collect, the resulting data always sent to Driver. The difference is: show or take scans limited no of partitions of an RDD to satisfy the limit supplied in the method parameter. So less no of elements sent to the driver.
Whereas collect scans all the partitions of an RDD and sends all the elements to the Driver.
For Example:

- Suppose we have an RDD containing 100 records and it is divided into 10 partitions across the cluster. Here each partition contains 10 records. take(10) will scan only the 1st partition and will send only 10 records to the Driver.
- show() by default sends 20 records to the driver that means it will scan only 2 partitions.
- collect() will return a list that contains all 100 records in the RDD to the driver. That means it has to scan all 10 partitions of the RDD. That's why collect() is not recommended because it will take more time to execute and may cause outOfMemory error if your driver node doesn't have enough memory particularly when your data set is huge.
- w.r.t write behaviour - we have a RDD with 10 partitions and we write it to HDFS then the executors operating on respective partition will write them to HDFS and we will get 10 part files, but if we do repartition(1) or coalesce (1) on the RDD and write it to HDFS then the data will be collected to driver and then written

W11:16 How is stage 1 created without any Wide Transformation? Harsh understand in stage 2 there is reduceByKey.



Ans: Seems you have loaded 2 text files. Stage 0 and 1 for 2 different text files loading, loading text files or parallelizing collections will form a new stage automatically. Here you loaded 2 text files for joining and hence it started two independent stages.

W11:17 Can we create multiple spark contexts in one spark application ?

Ans: No, 1 spark context is used for one application.

W11:18 how to load file with double pipe || as delimiter and also check for bad data? kindly suggest!!

Ans: Spark 2 does not support double delimiter as fields separator however, this is supported from spark 3. But you can read the file through RDD approach as complete string and later on split fields using || delimiter.
On bad data, there are several modes when reading a file like PERMISSIVE, DROPMALFORMED and FAILFAST. Specify above modes when reading a file through dataframe reader API as shown below
.option("mode", DROPMALFORMED)

W11:19 what is the difference between SparkContext and SparkSession? (please elaborate on practical differences)

Ans: SparkContext is basically a context or way to write and run your code on the spark cluster. If you just want to write spark code then spark context is okay but in real time you will be integrating spark with external tools like Hive, SQL, Nifi or another other tool, you need to manually import their context to use them with spark.
SparkSession automatically encapsulates all those contexts so you don't need to manually import context for different tools, which ends up saving developer code and time.

W11:20 import org.apache.log4j.Level

```
import org.apache.log4j.Logger
import
org.apache.spark.SparkConf
import
org.apache.spark.sql.SparkSessio
n import
org.apache.spark.sql.Dataset
import org.apache.spark.sql.Row
import java.sql.Timestamp case
class

orderData(order_id:Int,order_date:Timestamp,order_customer_id:I
nt,order_status:
String)
object DF1 extends App{
 Logger.getLogger("org").setLevel(Level.ERROR)
```

```
 val sparkConf = new SparkConf
 sparkConf.set("spark.app.name", "df")
 sparkConf.set("spark.master", "local[*]")
 val spark = SparkSession.builder()
                .config(sparkConf)
                .getOrCreate()
  val df1:Dataset[Row] = spark.read
           .format("csv")
           .option("path","/home/spark/work/inputSpark/sparkSumit
           Week11/order
s.csv")
           .option("inferschema", true)
           .option("mode" , "PERMISSIVE")
           .option("header", true)
           .load()
       import spark.implicits._
         val ds1 = df1.as[orderData]
     val df2 = ds1.filter(x => x.order_id>1000 )
     //ds1.filter(x => x.order_id>1000 ).filter(x=>x.order_id
     <1020).show
}
```

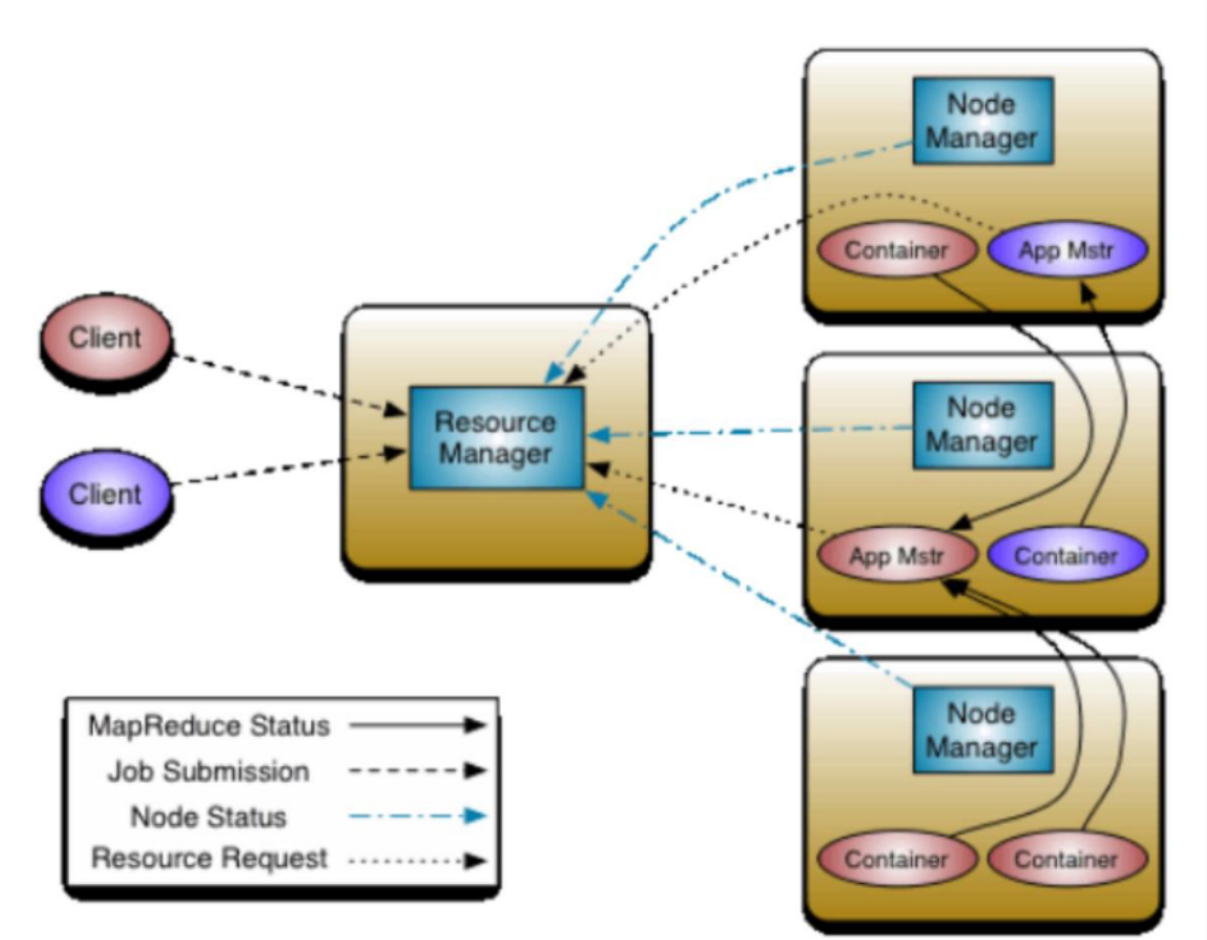
Ans: This error is raised because the last row if we see in orders.csv it has no value for order id column. So the value for order_id will be null in that row in DF/DS.

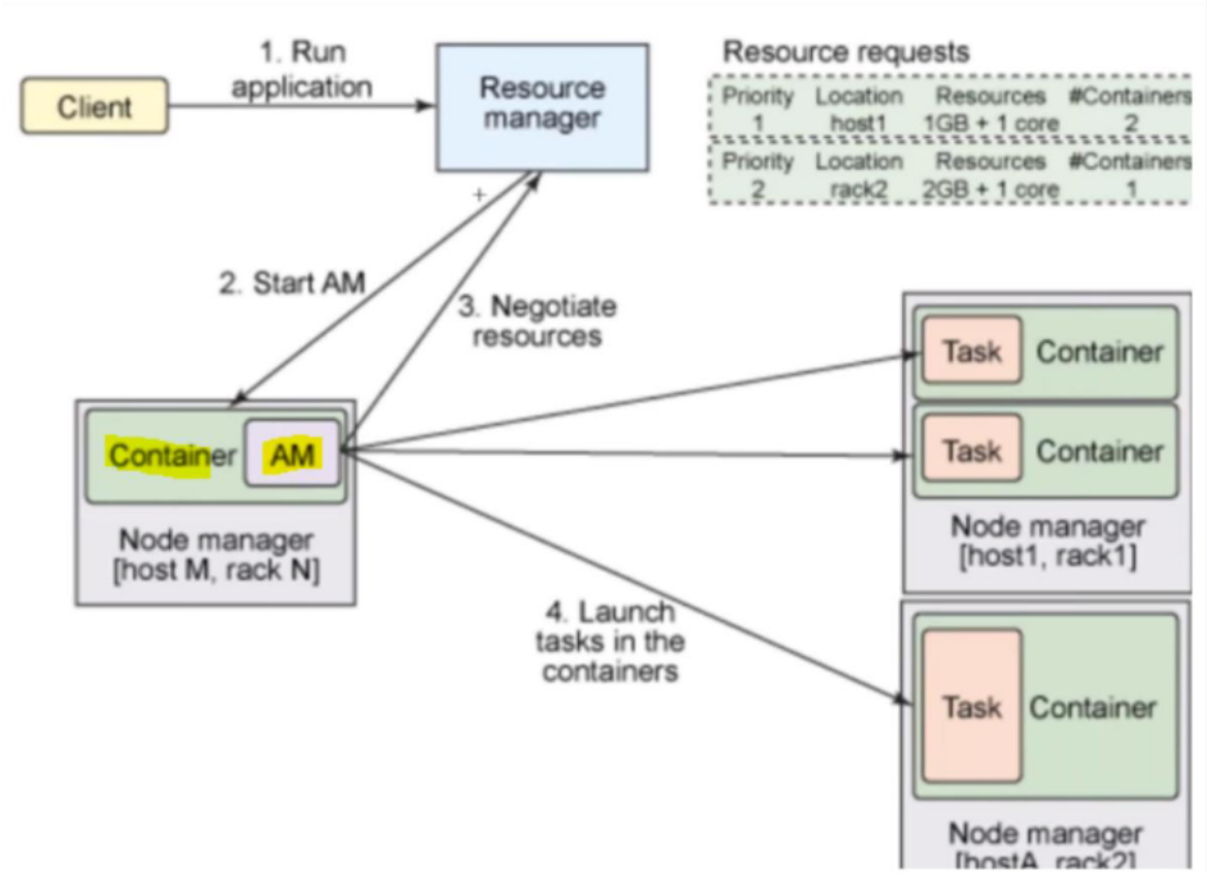
- Now when an action is called spark will start to deserialize it and start making an object out of each row (as we choose to use the DataSet approach). - It has to cast each order_id column value to the data type IntegerType (in our case) and when it reaches the last row it would have got error because it attempted to cast the NULL to Int.
- In order to solve the problem, while defining the case class , set the datatype of order_id to the Option[Int] and change filtration to ⇒ ordersDS.filter(x.order_id.getOrElse(0) > 1000)
- Also if we would have used the spark sql or DF approach instead of DataSets , you would not get this error at all. - because those do not need deserialization.

W11:21 Anderson has doubts about cache and persist, As we know that if we do not pass any storage level in persist then it will by default act like a cache. So now the question is when we can accomplish the same task using persist and persist also supports other storage options as well, then why we're having cache.

Ans: Both are same it's just that persistent as a term means permanent however nothing inside the memory(RAM) is permanent like Disk therefore Spark developers created the similar method with the name cache to avoid the confusion.. As cache represents something stored in memory temporarily

W11:22 In YARN architecture , this diagram shows AM is part of container. Then how come AM replies to RM and asks for resources. when there is no direct communication. confused in this two diagrams


Node Manager
Container
App Mstr
Client
Client
Resource Manager
Node Manager
App Mstr
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request


1. Run application
Client
Resource manager
Resource requests
Priority Location Resources #Containers
1 host1 1GB + 1 core 2
Priority Location Resources #Containers
2 rack2 2GB + 1 core 1
2. Start AM
3. Negotiate resources
Container AM
Node manager [host M, rack N]
Task Container
Task Container
Node manager [host1, rack1]
4. Launch tasks in the containers
Task Container
Node manager

Ans: ApplicationMaster is not a part of the container but it is launched inside the container to manage resources by ResourceManager.
When a job is submitted, ResourceManager launches a container with the help of Node Manager. Once the initial container is ready ResourceManager launches ApplicationMaster. When ApplicationMaster starts, it registers 1st with the ResourceManager.
Here to understand that ResourceManager is nothing but a combination of two components 1)ApplicationsManager 2)Scheduler. Once an ApplicationMaster launches it reports back to ApplicationsManager and negotiates for resources.
The ApplicationsManager manages all the ApplicationMasters launched in a cluster.

W11:23 Erick has one query. Suppose a single partition is there and we are using reduceByKey() transformation. Will it shuffle the data? if yes where the shuffled data will get stored as there is no other node in this scenario.

Ans: Yes shuffle will occur, shuffle job is to collect multiple same keys as a single key to reducer. Hence partition is one or many doesn't matter. Shuffle data is stored in memory.

W11:24 Jack is trying to achieve Spark-Hive Integration on my system but the code is failing with error msg "in thread "main" org.apache.spark.sql.AnalysisException: java.lang.RuntimeException: java.lang.RuntimeException: The root scratch dir:/tmp/hive on HDFS should be writable. Current permissions are: rwx------;"... Already write permissions given to

the mentioned directory [cloudera@quickstart ~]$ hdfs dfs -ls /tmp/hive

drwxrwxrwx - cloudera supergroup 0 2021-02-13 02:12 /tmp/hive/cloudera
drwxrwxrwx - hive supergroup 0 2021-02-13 05:11 /tmp/hive/hive
drwx-wx-wx - cloudera supergroup 0 2021-02-13 03:40 /tmp/hive ....Attached
screenshots of the error message and permissions given to /tmp/hive directory and code, Kindly help and do let me know if any further info is needed
Ans:
cp /usr/lib/hive/conf/hive-site.xml /usr/lib/spark/conf/hive-site.xml

W11:25 Can anyone explain what is aggregateByKey and combineByKey?

Ans: aggregateByKey() is similar as reduceByKey() but you can return result in different type. For example : Input is
("hello",4)
("hello",4)
output is
("hello","eight")
another difference is : aggregateByKey is similar to reduceByKey but you can give initial values when performing aggregation.
There is no automatic mapping. We need to write code for it. It's a flexibility given, if you need different input and output type then use aggregateByKey

W11:26 Amanda read at multiple places that we can have multiple spark sessions but underneath spark context remain the same?

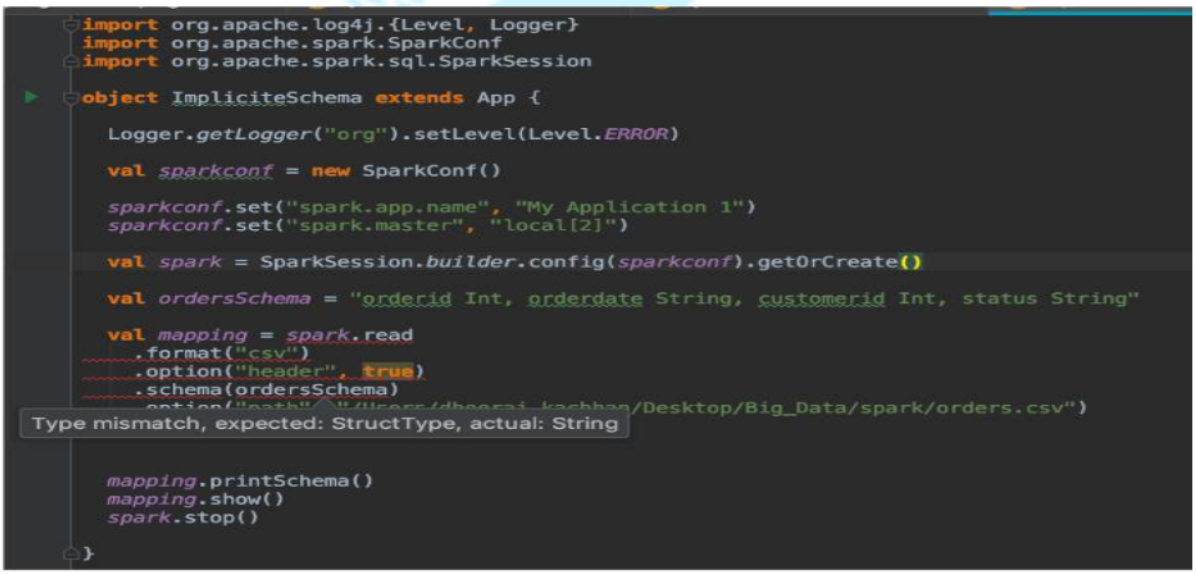
Ans: Yes you can create multiple spark sessions but spark context is one. spark session is made easy for developer to get all context in one basket

W11:27 In a Spark video, Sumit sir mentioned that Spark is not matured enough yet to take up Ingestion work like Sqoop. Can anyone help to understand what are the performance issues or challenges in using Spark for ingesting data from RDBMS?

Ans: Yes sqoop on spark is possible recently .,
https://stackoverflow.com/questions/33771776/apache-sqoop-and-spark

W11:28 Samme is getting this error while using DDL Schema approach in spark

Ans: Scala version 2.11 is compatible with spark sql 2.4.4



W11:29 Ronita is just running the code provided in our session.But She found below error

```scala
import java.sql.Timestamp
import org.apache.log4j.Level
import org.apache.log4j.Logger
import org.apache.spark.SparkConf
import org.apache.spark.sql.Dataset
import org.apache.spark.sql.Row
import org.apache.spark.sql.SparkSession

case class OrdersData(order_id: Int,order_date: Timestamp,order_customer_id: Int,order_status: String)
object DataFrametoSet extends App{
  Logger.getLogger("org").setLevel(Level.ERROR)
  val sparkConf = new SparkConf()
      sparkConf.set("spark.app.name","My first dataset")
      sparkConf.set("spark.master","local[2]")
  val spark = SparkSession.builder()
     .config(sparkConf)
     .getOrCreate()
  val OrdersDf: Dataset[Row] = spark.read
  .option("header",true)
  .option("inferSchema",true)
  .csv("file:///E:/Trendytech/Week-11/orders.csv")

  import spark.implicits._
 val OrdersDs = OrdersDf.as[OrdersData]

 OrdersDs.filter(x => x.order_id < 20).show()
// OrdersDf.show()
 //OrdersDf.printSchema()

 spark.stop()
```

```
Problems  Tasks  Console  Type Hierarchy
<terminated> DataFrametoSet$ [Scala Application] C:\Program Files\Java\jre1.8.0_281\bin\javaw.exe (Apr 13, 2021, 11:01:07 PM)
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
21/04/13 23:01:54 ERROR Executor: Exception in task 0.0 in stage 2.0 (TID 2)
java.lang.NullPointerException: Null value appeared in non-nullable field:
- field (class: "scala.Int", name: "order_customer_id")
- root class: "OrdersData"
If the schema is inferred from a Scala tuple/case class, or a Java bean, please try to use scala.Option[ ] or other nullable types (e.g.
        at org.apache.spark.sql.catalyst.expressions.GeneratedClass$GeneratedIteratorForCodegenStage1.processNext(Unknown Source)
        at org.apache.spark.sql.execution.BufferedRowIterator.hasNext(BufferedRowIterator.java:43)
        at org.apache.spark.sql.execution.WholeStageCodegenExec$$anonfun$13$$anon$1.hasNext(WholeStageCodegenExec.scala:636)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$2.apply(SparkPlan.scala:255)
        at org.apache.spark.sql.execution.SparkPlan$$anonfun$2.apply(SparkPlan.scala:247)
        at org.apache.spark.rdd.RDD$$anonfun$mapPartitionsInternal$1$$anonfun$apply$24.apply(RDD.scala:836)
        at org.apache.spark.rdd.RDD$$anonfun$mapPartitionsInternal$1$$anonfun$apply$24.apply(RDD.scala:836)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:123)
        at org.apache.spark.executor.Executor$TaskRunner$$anonfun$10.apply(Executor.scala:408)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1360)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:414)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
        at java.lang.Thread.run(Unknown Source)
```


Ans: Check your input file , seems there is null in order-customer-Id column.
Solution 1: remove nulls from input file
Open file in excel → filter by null , select all null rows , delete it.
Solution 2: Allow null values in the schema.

WEEK 11 FAQS

In case class you should define order_customer_id as "order_customer_id: Option[Int]" which means the column is nullable.
If you don't want to include nullable then you should be using "order_customer_id: Int"

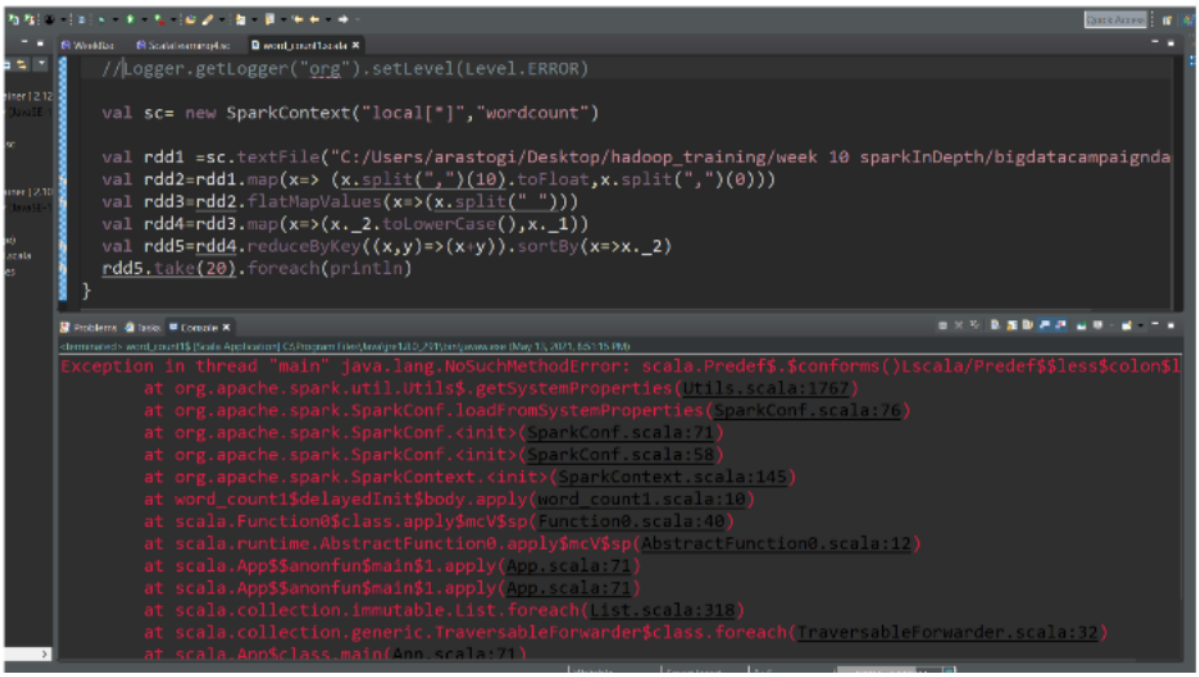
W11:30 Does anyone know how to read and an excel file using spark and export spark dataframe to excel

Ans:https://stackoverflow.com/questions/47442333/how-to-read-excel-data-into-a-dataframe-in-spark-scala

W11:31 how to run the spark jars? Manasa is using itversity labs

Ans: https://m.youtube.com/watch?v=5uHG0aqir5s

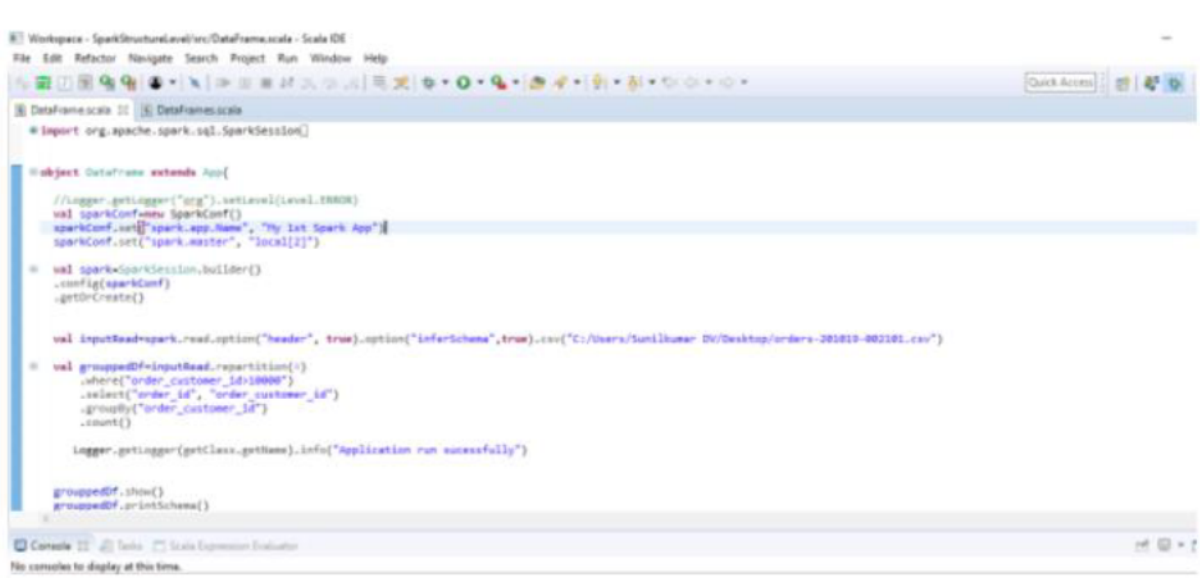
W11:32 Palkesh is getting this error, Can anyone pls help Palkesh out here



Ans: Set your scala compiler version to 2.11

WEEK 11 FAQS

W11:33 My workplace files are not hidden , can any one help me how do resolve



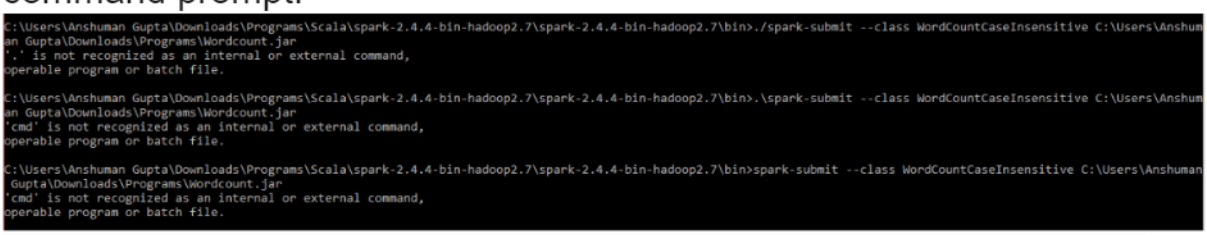
Ans: Top right corner you can see smalls button, click on it
OR double tab on your Object (Ex:- DataFrame.scala)

W11:34 While converting Dataframe to Dataset, we are declaring datatypes in case class, Also In dataframe, we are having datatypes (either structtype or DDL). Why we need to have Datatype declaration in two places ?

Ans : It depends on your code, if you are doing transformation on df then you need schema. if you are doing transformation on ds then you need schema on DS and not mandate on df.

```
//In the below code , there is no schema assigned for DF
val orders = spark.sqlContext.read.format("csv")
    .option("header", "true")
    .option("inferSchema", "false")
    .schema(schema)
    .load("Orders.csv") // DataFrame
    .as[Order] // DataSet // Order is a case class
```

W11:35 I am facing below issue while using spark-submit command in command prompt.



Ans : There is a space in folder hierarchy, in your name basically. Anshuman Gupta, remove the space and it should work fine.