Hadoop and Spark Developer - CCA 175

Problem Scenario 2

PLEASE READ THE INTRODUCTION TO THIS SERIES. CLICK ON HOME LINK AND READ THE INTRO BEFORE ATTEMPTING TO SOLVE THE PROBLEMS

Video walk through of the solution to this problem can be found here ***[Click here]***

Click here for the video version of this series. This takes you to the youtube playlist of videos.

Problem 2:

1. Using sqoop copy data available in mysql products table to folder **/user/cloudera/products** on hdfs as text file. columns should be delimited by pipe '|'
2. move all the files from **/user/cloudera/products** folder to **/user/cloudera/problem2/products** folder
3. Change permissions of all the files under **/user/cloudera/problem2/products** such that owner has read,write and execute permissions, group has read and write permissions whereas others have just read and execute permissions
4. read data in **/user/cloudera/problem2/products** and do the following operations using **a)** dataframes api **b)** spark sql **c)** RDDs aggregateByKey method. Your solution should have three sets of steps. Sort the resultant dataset by category id
    - filter such that your RDD\DF has products whose price is lesser than 100 USD
    - on the filtered data set find out the higest value in the product_price column under each category
    - on the filtered data set also find out total products under each category
    - on the filtered data set also find out the average price of the product under each category
    - on the filtered data set also find out the minimum price of the product under each category
5. store the result in avro file using snappy compression under these folders respectively
    - /user/cloudera/problem2/products/result-df
    - /user/cloudera/problem2/products/result-sql
    - /user/cloudera/problem2/products/result-rdd

Solution:
Try your best to solve the above scenario without going through the solution below. If you could then use the solution to compare your result. If you could not then I strongly recommend that you go through the concepts again (this time in more depth). Each step below provides a solution to the points mentioned in the Problem Scenario.

Step 1:

```
sqoop import \
--connect "jdbc:mysql://quickstart.cloudera:3306/retail_db" \
--username retail_dba \
--password cloudera \
--table products \
--as-textfile \
--target-dir /user/cloudera/products \
--fields-terminated-by '|';
```

Step 2:

```
hadoop fs -mkdir /user/cloudera/problem2/
hadoop fs -mkdir /user/cloudera/problem2/products
hadoop fs -mv /user/cloudera/products/* /user/cloudera/problem2/products/
```

Step 3:

```
//Read is 4, Write is 2 and execute is 1.
//ReadWrite,Execute = 4 + 2 + 1 = 7
//Read,Write = 4+2 = 6
//Read ,Execute=4+1=5

hadoop fs -chmod 765 /user/cloudera/problem2/products/*
```

Step 4:

```
scala> var products = sc.textFile("/user/cloudera/products").map(x=> {var d = x.split('|');
(d(0).toInt,d(1).toInt,d(2).toString,d(3).toString,d(4).toFloat,d(5).toString)});


scala>case class Product(productID:Integer, productCatID: Integer, productName: String, productDesc:String, productPrice:Float,
productImage:String);


scala> var productsDF = products.map(x=> Product(x._1,x._2,x._3,x._4,x._5,x._6)).toDF();
```

Step 4-Data Frame Api:

```
scala> import org.apache.spark.sql.functions._
scala> var dataFrameResult = productsDF.filter("productPrice <
100").groupBy(col("productCategory")).agg(max(col("productPrice")).alias("max_price"),countDistinct(col("productID")).alias("tot_products"),ro
und(avg(col("productPrice")),2).alias("avg_price"),min(col("productPrice")).alias("min_price")).orderBy(col("productCategory"));
scala> dataFrameResult.show();
```

Step 4 - Spark SQL:

```
productsDF.registerTempTable("products");
var sqlResult = sqlContext.sql("select product_category_id, max(product_price) as maximum_price, count(distinct(product_id)) as
total_products, cast(avg(product_price) as decimal(10,2)) as average_price, min(product_price) as minimum_price from products where
product_price <100 group by product_category_id order by product_category_id desc");
sqlResult.show();
```

Step 4 - RDD aggregateByKey:

Search This Blog

Search

CCA 175 Hadoop and Spark Developer Preparation

A leader with a unique blend of deep technology expertise and strong management skil

Arun Kumar Pasuparthi
G+ Follow 212
View my complete profile

Report Abuse

Blog Archive

April 2017 (1)

```
var rddResult = productsDF.map(x=>(x(1).toString.toInt,x(4).toString.toDouble)).aggregateByKey((0.0,0.0,0,9999999999999.0))((x,y)=>
(math.max(x._1,y),x._2+y,x._3+1,math.min(x._4,y)),(x,y)=>(math.max(x._1,y._1),x._2+y._2,x._3+y._3,math.min(x._4,y._4))).map(x=>
(x._1,x._2._1,(x._2._2/x._2._3),x._2._3,x._2._4)).sortBy(_._1, false);
rddResult.collect().foreach(println);
```

Step 5:

```
-> import com.databricks.spark.avro._;
-> sqlContext.setConf("spark.sql.avro.compression.codec","snappy")
->dataFrameResult.write.avro("/user/cloudera/problem2/products/result-df");
->sqlResult.write.avro("/user/cloudera/problem2/products/result-sql");
->rddResult.toDF().write.avro("/user/cloudera/problem2/products/result-rdd");;
```

20 comments:

Unknown May 18, 2017 at 2:05 PM

Hi Arun,

First of all thank you for such a confidence boosting blog on CCA175.
I would like to highlight a small correction in aggregateByKey transformation.
The default value of the MIN_PRICE should not be 0.0. If it is 0.0, for every MATH.MIN(a,b) the output would be 0.0. As a workaround this could be replaced by a higher value(10000.0) which would ultimately be swapped by the MIN values in the process.

Sample output:
(58,241.0,170.0,4,115.0)
(57,189.99,154.99,6,109.99)
(56,159.99,159.99,2,159.99)
(54,299.99,209.99,6,129.99)

Regards,
--Lax Dash

Reply

▾ Replies

Arun Kumar Pasuparthi May 18, 2017 at 2:48 PM

Sorry about that. I actually did it in the right way in the video. Check video tutorial for this between 31.08 and 31.10. The correct solution already exists. I think i did not update the blog. Thanks for bringing to my notice.

Reply

Sayali Mahajan May 28, 2017 at 2:32 PM

In the exam is it asked whether to solve any problem with RDD or DataFrames.
Or what matters is the output?
Please respond!!

Reply

▾ Replies

Arun Kumar Pasuparthi May 28, 2017 at 4:42 PM

there is a very very good possibility that you get questions where you are you asked to complete missing lines of code in a spark program written in scala or python. While the exam looks at final output, for you fill the missing lines of code is the fastest option. otherwise, you will have to rewrite the code in your preferred way which will not always result in the right answer. And most importantly will be time consuming. I hope i answered your question

Sayali Mahajan May 28, 2017 at 8:28 PM

Thank you Arun Sir!!
I am following you video and have noticed that you are also using Scala.
I also was wondering if the code snippet in the exam is in python or Scala. Or do we get a choice to at the first place which language are we going to write the exam. I have been practicing CCA175 with Scala.

Arun Kumar Pasuparthi June 4, 2017 at 5:13 PM

According to exam website you will have to understand both scala and python. Majority of the problem solution consists of using api. the function names are same in python on scala. I recommend that you solve all these problems using python as well so that you are well prepared answering any questions related to completing a portion of code that is written either in scala or python.

Reply

adarsh pratap singh June 25, 2017 at 6:43 AM

var d = x.split('|'); -----> var d = x.split('\\|');

Reply

▾ Replies

Arun Kumar Pasuparthi June 25, 2017 at 8:42 AM

Adarsh,

thank you for following my posts. You don't have to escape when supplying a character literal for pipe character. Here are all the variations. I hope below helps you remember what works and what does not for your exam. All the very best.

-- WITH ESCAPE BUT PASSING A STRING THAT IS A LITERAL INSIDE A DOUBLE QUOTES

```
scala> sc.textFile("/user/cloudera/products").map(x=> {var d =x.split("\\|"); (d(0),d(1),d(2))}).take(1).foreach(println);
Output - (1,2,Quest Q64 10 FT. x 10 FT. Slant Leg Instant U)
```

--WITHOUT ESCAPE PASSING A STRING I.E LITERAL WITHING DOUBLE QUOTES

```
scala> sc.textFile("/user/cloudera/products").map(x=> {var d =x.split("|"); (d(0),d(1),d(2))}).take(1).foreach(println);
Output - (,1,|)
```

--WITHOUT ESCAPE CHAR AND SENDING A CHAR LITERAL THAT IS in single quotes.

```
scala> sc.textFile("/user/cloudera/products").map(x=> {var d =x.split('|'); (d(0),d(1),d(2))}).take(1).foreach(println);
Output - (1,2,Quest Q64 10 FT. x 10 FT. Slant Leg Instant U)
```

adarsh singh June 25, 2017 at 11:11 AM

Thanks a lot Arun.. it really helped me ..

Unknown September 24, 2017 at 3:09 PM

This comment has been removed by the author.

Lakshmi Thiagarajan September 24, 2017 at 3:12 PM

Thanks Arun for the very thorough problem stmts, am preparing for CCA175 using ur blog. Here I have question on your answer to adarsh.

So the safe way to use for split is this
x.split("\\|") ??

I have always used x.split(",") for my comma delimited fields so far , so why not x.split("|") work for us , pls explain.

Reply

Pn July 18, 2017 at 8:48 AM

This comment has been removed by the author.

Reply

Pn July 18, 2017 at 8:49 AM

```
scala> var productsDF = products.map(x=> Product(x._1,x._2,x._3,x._4,x._5,x._6)).toDF();
```

gives error on quickstart 5.10

```
"Failed to start database 'metastore_db' with class loader org.apache.spark.sql.hive.client.IsolatedClientLoader$$anon$1@5fa9ef3d, see the next exception for details.
at org.apache.derby.iapi.error.StandardException.newException(Unknown Source)
at org.apache.derby.impl.jdbc.SQLExceptionFactory.wrapArgsForTransportAcrossDRDA(Unknown Source)
... 135 more
Caused by: ERROR XSDB6: Another instance of Derby may have already booted the database /home/cloudera/metastore_db.
at org.apache.derby.iapi.error.StandardException.newException(Unknown Source)
at org.apache.derby.iapi.error.StandardException.newException(Unknown Source)
at org.apache.derby.impl.store.raw.data.BaseDataFileFactory.privGetJBMSLockOnDB(Unknown Source)
at org.apache.derby.impl.store.raw.data.BaseDataFileFactory.run(Unknown Source)
at java.security.AccessController.doPrivileged(Native Method)
"
```

I you help to resolve it.

Reply

Replies

Pn July 20, 2017 at 12:22 PM

Hi Team......can anyone help me how to fix the below error in quick start

```
"Caused by: ERROR XSDB6: Another instance of Derby may have already booted the database /home/cloudera/metastore_db.
at org.apache.derby.iapi.error.StandardException.newException(Unknown Source)
```

Reply

Bala September 5, 2017 at 1:22 AM

Hi Arun ! the question asks the user to sort the data by product_category_id (which bydefault i understand to be in ascending order). Whereas in your solution under "SPARK SQL" you have ordered the result set by descending order.

" product_category_id desc"

In the DF result set, you have sorted in ascending order - "orderBy(col("productCategory"));"

am i missing anything here or should your answer be updated in this blog ?

Reply

kiran tej November 9, 2017 at 1:26 PM

hi arun,
while am executing the step-4, am getting the below error as managed memory leak. And my spark version is spark 1.6.0 . Am doing this in cloudera quickstart vm.
As it is error, it is not showing the result also.

Can you please tell me how to change this ERROR to WARN so that i can see the result.

```
dataFrameResult.show();
```

```
17/11/09 13:19:01 WARN memory.TaskMemoryManager: leak 8.3 MB memory from org.apache.spark.unsafe.map.BytesToBytesMap@129e18f
17/11/09 13:19:01 ERROR executor.Executor: Managed memory leak detected; size = 8650752 bytes, TID = 7
17/11/09 13:19:01 ERROR executor.Executor: Exception in task 0.0 in stage 10.0 (TID 7)
```

Reply

簡小宇 November 26, 2017 at 7:25 PM

Hi Arun,

Thank you for the sharing, I found some issues which looks typo when trying your solutions.

Please help to correct me if I am wrong, thanks.

STEP 4 :

```
scala> var products = sc.textFile("/user/cloudera/problem2/products/").map(x=> {var d = x.split('|'); (d(0).toInt,d(1).toInt,d(2).toString,d(3).toString,d(4).toFloat,d(5).toString)});
```

Step 4-Data Frame Api:

```
scala> var dataFrameResult = productsDF.filter("productPrice < 100").groupBy(col("productCatID")).agg(max(col("productPrice")).alias("max_price"),countDistinct(col("productID")).alias("tot_products"),round(avg(col("productPrice")),2).alias("avg_price"),min(col("productPrice")).alias("min_price")).orderBy(col("productCatID"));
```

Step 4- spark sql:

```
var sqlResult = sqlContext.sql("SELECT productCatID, max(productPrice) AS maximum_price, count(distinct(productID)) AS total_products, cast(avg(productPrice) as DECIMAL(10,2)) AS average_price, min(productPrice) AS minimum_price FROM products WHERE productPrice < 100 GROUP BY productCatID ORDER BY productCatID asc");
```

Reply

Aayush Desai November 29, 2017 at 5:25 PM

In the second question, if you are making both the directories at that moment itself, what dta is getting transferred when you are using -mv command?

Reply

Anand January 2, 2018 at 9:21 PM

Hi Arun, one important query, for calculating the avg. price you are rounding it off to two decimal places even though it's not explicitly mentioned in the problem statement. So, will it be mentioned explicitly during the exam? or do we need to take care of it implicitly?

Reply

Mochamad Eka Pramudita January 9, 2018 at 6:26 PM

Hi Arun, in the real exam, can I use pyspark?
Or it's only allowed use spark-shell ?

Reply

Home

Subscribe to: Posts (Atom)

If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

(no title)

If you have landed on this page then you are most likely aspiring to learn Hadoop ecosystem of technologies and tools. Why not make you...

Simple theme. Powered by Blogger.