Details - Duration 20 minutes

* Data is available in HDFS /public/randomtextwriter
* Get word count for the input data using space as delimiter (for each word, we need to get how many types it is repeated in the entire input data set)
* Number of executors should be 10
* Executor memory should be 3 GB
* Executor cores should be 20 in total (2 per executor)
* Number of output files should be 8
* Avro dependency details: groupId -> com.databricks, artifactId -> spark-avro\_2.10, version -> 2.0.1
* Target Directory: /user/<YOUR\_USER\_ID>/solutions/solution05/wordcount
* Target File Format: Avro
* Target fields: word, count
* Compression: N/A or default

[saurabhbhanwala@gw03 ~]$ export SPARK\_MAJOR\_VERSION=2

[saurabhbhanwala@gw03 ~]$ pyspark --master yarn --conf spark.ui.port=12181 --num-executors 10 --executor-cores 2 --executor-memory 3G --packages com.databricks:spark-avro\_2.10:2.0.1

SPARK\_MAJOR\_VERSION is set to 2, using Spark2

[saurabhbhanwala@gw03 ~]$ export SPARK\_MAJOR\_VERSION=2

[saurabhbhanwala@gw03 ~]$ pyspark --master yarn --conf spark.ui.port=12181 --num-executors 10 --executor-cores 2 --executor-memory 3G --packages com.databricks:spark-avro\_2.11:4.0.0

SPARK\_MAJOR\_VERSION is set to 2, using Spark2

>>> from pyspark.sql import SparkSession

>>> from pyspark.sql import functions

>>> from pyspark.sql.functions import \*

>>> import os

>>> os.system("hdfs dfs -tail /public/randomtextwriter")

tail: `/public/randomtextwriter': Is a directory

256

>>> os.system("hdfs dfs -ls /public/randomtextwriter")

Found 31 items

-rw-r--r-- 3 training hdfs 0 2017-01-18 20:24 /public/randomtextwriter/\_SUCCESS

-rw-r--r-- 3 training hdfs 1102230331 2017-01-18 20:24 /public/randomtextwriter/part-m-00000

-rw-r--r-- 3 training hdfs 1102241707 2017-01-18 20:24 /public/randomtextwriter/part-m-00001

-rw-r--r-- 3 training hdfs 1102239201 2017-01-18 20:24 /public/randomtextwriter/part-m-00002

-rw-r--r-- 3 training hdfs 1102229785 2017-01-18 20:23 /public/randomtextwriter/part-m-00003

-rw-r--r-- 3 training hdfs 1102239085 2017-01-18 20:24 /public/randomtextwriter/part-m-00004

-rw-r--r-- 3 training hdfs 1102245185 2017-01-18 20:24 /public/randomtextwriter/part-m-00005

-rw-r--r-- 3 training hdfs 1102242635 2017-01-18 20:24 /public/randomtextwriter/part-m-00006

-rw-r--r-- 3 training hdfs 1102227833 2017-01-18 20:24 /public/randomtextwriter/part-m-00007

-rw-r--r-- 3 training hdfs 1102244768 2017-01-18 20:24 /public/randomtextwriter/part-m-00008

-rw-r--r-- 3 training hdfs 1102244831 2017-01-18 20:24 /public/randomtextwriter/part-m-00009

-rw-r--r-- 3 training hdfs 1102233053 2017-01-18 20:24 /public/randomtextwriter/part-m-00010

-rw-r--r-- 3 training hdfs 1102228278 2017-01-18 20:24 /public/randomtextwriter/part-m-00011

-rw-r--r-- 3 training hdfs 1102239114 2017-01-18 20:24 /public/randomtextwriter/part-m-00012

-rw-r--r-- 3 training hdfs 1102235849 2017-01-18 20:24 /public/randomtextwriter/part-m-00013

-rw-r--r-- 3 training hdfs 1102238892 2017-01-18 20:24 /public/randomtextwriter/part-m-00014

-rw-r--r-- 3 training hdfs 1102237321 2017-01-18 20:23 /public/randomtextwriter/part-m-00015

-rw-r--r-- 3 training hdfs 1102236889 2017-01-18 20:24 /public/randomtextwriter/part-m-00016

-rw-r--r-- 3 training hdfs 1102244070 2017-01-18 20:24 /public/randomtextwriter/part-m-00017

-rw-r--r-- 3 training hdfs 1102228803 2017-01-18 20:23 /public/randomtextwriter/part-m-00018

-rw-r--r-- 3 training hdfs 1102231608 2017-01-18 20:24 /public/randomtextwriter/part-m-00019

-rw-r--r-- 3 training hdfs 1102228641 2017-01-18 20:24 /public/randomtextwriter/part-m-00020

-rw-r--r-- 3 training hdfs 1102235104 2017-01-18 20:24 /public/randomtextwriter/part-m-00021

-rw-r--r-- 3 training hdfs 1102227744 2017-01-18 20:24 /public/randomtextwriter/part-m-00022

-rw-r--r-- 3 training hdfs 1102226515 2017-01-18 20:24 /public/randomtextwriter/part-m-00023

-rw-r--r-- 3 training hdfs 1102224814 2017-01-18 20:24 /public/randomtextwriter/part-m-00024

-rw-r--r-- 3 training hdfs 1102224114 2017-01-18 20:24 /public/randomtextwriter/part-m-00025

-rw-r--r-- 3 training hdfs 1102242859 2017-01-18 20:24 /public/randomtextwriter/part-m-00026

-rw-r--r-- 3 training hdfs 1102252598 2017-01-18 20:24 /public/randomtextwriter/part-m-00027

-rw-r--r-- 3 training hdfs 1102244434 2017-01-18 20:24 /public/randomtextwriter/part-m-00028

-rw-r--r-- 3 training hdfs 1102232634 2017-01-18 20:24 /public/randomtextwriter/part-m-00029

>>> WordDF = spark.read.text("/public/randomtextwriter")

WordDF.show()

>>> wordDFSplit = WordDF.select(explode(split('value'," ")))

>>> wordDFSplit.show()

+--------------------+

| col|

+--------------------+

|SEQorg.apache.h...|

| steprelationship|

| pleasurehood|

| abusiveness|

| seelful|

| unstipulated|

| winterproof|

| �gmericarp|

| pentosuria|

| airfreighter|

| orthopedical|

| symbiogenetically|

| Hu|

| unrepealably|

| Gothish|

| unachievable|

| saguran|

| extraorganismal|

| defensibly|

| taver|

+--------------------+

only showing top 20 rows

>>> wordDFColumn = wordDFSplit.withColumnRenamed('col', 'Word')

>>> wordDFColumn.show()

+--------------------+

| Word|

+--------------------+

|SEQorg.apache.h...|

| steprelationship|

| pleasurehood|

| abusiveness|

| seelful|

| unstipulated|

| winterproof|

| �gmericarp|

| pentosuria|

| airfreighter|

| orthopedical|

| symbiogenetically|

| Hu|

| unrepealably|

| Gothish|

| unachievable|

| saguran|

| extraorganismal|

| defensibly|

| taver|

+--------------------+

only showing top 20 rows

>>> WordDFCount = wordDFColumn.groupBy('Word').count().withColumnRenamed('count','Count')

WordDFCount.show()

>>> WordDFCount.show()

+--------------------+-------+

| Word| Count|

+--------------------+-------+

| corbel|2925970|

| Spencerism|2926090|

| lebensraum|2927965|

| ��trailmaking| 6618|

|X����g

�L�...| 2|

| ��arsenide| 6478|

| �:9critically| 67|

| �ONswangy| 99|

|T?>quadrennial| 1|

|gWVbenzothi...| 1|

|X����g

�L�...| 1|

|2XWScorpaen...| 2|

| $XWBishareen| 1|

| �gflicitness| 39|

| 0/rebilling| 2|

| '?>sequacity| 2|

| ��porriginous| 6564|

| �DWhilkut| 48|

|�dcbacterio...| 69|

| cEDrede| 2|

+--------------------+-------+

only showing top 20 rows

WordDFCount.coalesce(8).write.format('com.databricks.spark.avro').save("/user/saurabhbhanwala/solutions/solution/wordcount")

<https://gpdb.docs.pivotal.io/5180/pxf/hdfs_avro.html>

exercise 6

Duration: 20 to 30 minutes

* Tables should be in hive database - <YOUR\_USER\_ID>\_retail\_db\_txt
  + orders
  + order\_items
  + customers
* Time to create database and tables need not be counted. Make sure to go back to Spark SQL module and create tables and load data
* Get details of top 5 customers by revenue for each month
* We need to get all the details of the customer along with month and revenue per month
* Data need to be sorted by month in ascending order and revenue per month in descending order
* Create table top5\_customers\_per\_month in <YOUR\_USER\_ID>\_retail\_db\_txt
* Insert the output into the newly created table

<https://stackoverflow.com/questions/30664008/how-to-save-dataframe-directly-to-hive>

<https://stackoverflow.com/questions/11269203/when-creating-an-external-table-in-hive-can-i-point-the-location-to-specific-fil>

<https://stackoverflow.com/questions/50050016/how-to-create-an-external-spark-table-from-data-in-hdfs>

[saurabhbhanwala@gw03 ~]$ export SPARK\_MAJOR\_VERSION=2

[saurabhbhanwala@gw03 ~]$ pyspark --master yarn --conf spark.ui.port=12181 --num-executors 4 --executor-cores 2 --executor-memory 2G

>>> ordersDF = spark.read.csv("/user/saurabhbhanwala/retail\_db/orders")

>>> orderItemsDF = spark.read.csv("/user/saurabhbhanwala/retail\_db/order\_items")

>>> customersDF = spark.read.csv("/user/saurabhbhanwala/retail\_db/customers")

>>> ordersDF.show()

+---+--------------------+-----+---------------+

|\_c0| \_c1| \_c2| \_c3|

+---+--------------------+-----+---------------+

| 1|2013-07-25 00:00:...|11599| CLOSED|

| 2|2013-07-25 00:00:...| 256|PENDING\_PAYMENT|

| 3|2013-07-25 00:00:...|12111| COMPLETE|

| 4|2013-07-25 00:00:...| 8827| CLOSED|

| 5|2013-07-25 00:00:...|11318| COMPLETE|

| 6|2013-07-25 00:00:...| 7130| COMPLETE|

| 7|2013-07-25 00:00:...| 4530| COMPLETE|

| 8|2013-07-25 00:00:...| 2911| PROCESSING|

| 9|2013-07-25 00:00:...| 5657|PENDING\_PAYMENT|

| 10|2013-07-25 00:00:...| 5648|PENDING\_PAYMENT|

| 11|2013-07-25 00:00:...| 918| PAYMENT\_REVIEW|

| 12|2013-07-25 00:00:...| 1837| CLOSED|

| 13|2013-07-25 00:00:...| 9149|PENDING\_PAYMENT|

| 14|2013-07-25 00:00:...| 9842| PROCESSING|

| 15|2013-07-25 00:00:...| 2568| COMPLETE|

| 16|2013-07-25 00:00:...| 7276|PENDING\_PAYMENT|

| 17|2013-07-25 00:00:...| 2667| COMPLETE|

| 18|2013-07-25 00:00:...| 1205| CLOSED|

| 19|2013-07-25 00:00:...| 9488|PENDING\_PAYMENT|

| 20|2013-07-25 00:00:...| 9198| PROCESSING|

+---+--------------------+-----+---------------+

only showing top 20 rows

SyntaxError: invalid syntax

>>> from pyspark.sql import functions

>>> from pyspark.sql.functions import \*

>>> from pyspark.sql.functions import col

>>> ordersDFWithColumn = ordersDF.select( col("\_c0").alias("order\_id"), col("\_c1").alias("order\_date"), col("\_c2").alias("order\_customer\_id"), col("\_c3").alias("order\_status"))

>>> orderItemsDFColumn = sqlContext.sql("select \_c0 as order\_item\_id, \_c1 as order\_item\_order\_id, \_c2 as order\_item\_product\_id, \_c3 as order\_item\_productQty, \_c4 as order\_item\_subtotal, \_c5 as order\_item\_prd\_price from orderItmesDF")

>>> orderItemsDFColumn = sqlContext.sql("select \_c0 as order\_item\_id, \_c1 as order\_item\_order\_id, \_c2 as order\_item\_product\_id, \_c3 as order\_item\_productQty, \_c4 as order\_item\_subtotal, \_c5 as order\_item\_prd\_price from orderItemsDF")

>>> orderItemsDFColumn.show()

+-------------+-------------------+---------------------+---------------------+-------------------+--------------------+

|order\_item\_id|order\_item\_order\_id|order\_item\_product\_id|order\_item\_productQty|order\_item\_subtotal|order\_item\_prd\_price|

+-------------+-------------------+---------------------+---------------------+-------------------+--------------------+

| 1| 1| 957| 1| 299.98| 299.98|

| 2| 2| 1073| 1| 199.99| 199.99|

| 3| 2| 502| 5| 250.0| 50.0|

| 4| 2| 403| 1| 129.99| 129.99|

| 5| 4| 897| 2| 49.98| 24.99|

| 6| 4| 365| 5| 299.95| 59.99|

| 7| 4| 502| 3| 150.0| 50.0|

| 8| 4| 1014| 4| 199.92| 49.98|

| 9| 5| 957| 1| 299.98| 299.98|

| 10| 5| 365| 5| 299.95| 59.99|

| 11| 5| 1014| 2| 99.96| 49.98|

| 12| 5| 957| 1| 299.98| 299.98|

| 13| 5| 403| 1| 129.99| 129.99|

| 14| 7| 1073| 1| 199.99| 199.99|

| 15| 7| 957| 1| 299.98| 299.98|

| 16| 7| 926| 5| 79.95| 15.99|

| 17| 8| 365| 3| 179.97| 59.99|

| 18| 8| 365| 5| 299.95| 59.99|

| 19| 8| 1014| 4| 199.92| 49.98|

| 20| 8| 502| 1| 50.0| 50.0|

+-------------+-------------------+---------------------+---------------------+-------------------+--------------------+

only showing top 20 rows

>>> customersDF.show()

+---+-----------+---------+---------+---------+--------------------+-------------+---+-----+

|\_c0| \_c1| \_c2| \_c3| \_c4| \_c5| \_c6|\_c7| \_c8|

+---+-----------+---------+---------+---------+--------------------+-------------+---+-----+

| 1| Richard|Hernandez|XXXXXXXXX|XXXXXXXXX| 6303 Heather Plaza| Brownsville| TX|78521|

| 2| Mary| Barrett|XXXXXXXXX|XXXXXXXXX|9526 Noble Embers...| Littleton| CO|80126|

| 3| Ann| Smith|XXXXXXXXX|XXXXXXXXX|3422 Blue Pioneer...| Caguas| PR|00725|

| 4| Mary| Jones|XXXXXXXXX|XXXXXXXXX| 8324 Little Common| San Marcos| CA|92069|

| 5| Robert| Hudson|XXXXXXXXX|XXXXXXXXX|10 Crystal River ...| Caguas| PR|00725|

| 6| Mary| Smith|XXXXXXXXX|XXXXXXXXX|3151 Sleepy Quail...| Passaic| NJ|07055|

| 7| Melissa| Wilcox|XXXXXXXXX|XXXXXXXXX|9453 High Concession| Caguas| PR|00725|

| 8| Megan| Smith|XXXXXXXXX|XXXXXXXXX|3047 Foggy Forest...| Lawrence| MA|01841|

| 9| Mary| Perez|XXXXXXXXX|XXXXXXXXX| 3616 Quaking Street| Caguas| PR|00725|

| 10| Melissa| Smith|XXXXXXXXX|XXXXXXXXX|8598 Harvest Beac...| Stafford| VA|22554|

| 11| Mary| Huffman|XXXXXXXXX|XXXXXXXXX| 3169 Stony Woods| Caguas| PR|00725|

| 12|Christopher| Smith|XXXXXXXXX|XXXXXXXXX|5594 Jagged Ember...| San Antonio| TX|78227|

| 13| Mary| Baldwin|XXXXXXXXX|XXXXXXXXX|7922 Iron Oak Gar...| Caguas| PR|00725|

| 14| Katherine| Smith|XXXXXXXXX|XXXXXXXXX|5666 Hazy Pony Sq...| Pico Rivera| CA|90660|

| 15| Jane| Luna|XXXXXXXXX|XXXXXXXXX| 673 Burning Glen| Fontana| CA|92336|

| 16| Tiffany| Smith|XXXXXXXXX|XXXXXXXXX| 6651 Iron Port| Caguas| PR|00725|

| 17| Mary| Robinson|XXXXXXXXX|XXXXXXXXX| 1325 Noble Pike| Taylor| MI|48180|

| 18| Robert| Smith|XXXXXXXXX|XXXXXXXXX|2734 Hazy Butterf...| Martinez| CA|94553|

| 19| Stephanie| Mitchell|XXXXXXXXX|XXXXXXXXX|3543 Red Treasure...| Caguas| PR|00725|

| 20| Mary| Ellis|XXXXXXXXX|XXXXXXXXX| 4703 Old Route|West New York| NJ|07093|

+---+-----------+---------+---------+---------+--------------------+-------------+---+-----+

only showing top 20 rows

from functools import reduce

>>> oldColumns = customersDF.schema.names

>>> newColumns = ["cust\_id", "cust\_fname", "cust\_lname", "cust\_email", "cust\_password", "cust\_street", "cust\_street", "cust\_state", "cus\_zip"]

>>> customersDFColumn = reduce(lambda cd, id: cd.withColumnRenamed(oldColumns[id], newColumns[id]), range(len(oldColumns)), customersDF)

>>>

>>> customersDFColumn.show()

+-------+-----------+----------+----------+-------------+--------------------+-------------+----------+--------+

|cust\_id| cust\_fname|cust\_lname|cust\_email|cust\_password| cust\_street| cust\_city|cust\_state|cust\_zip|

+-------+-----------+----------+----------+-------------+--------------------+-------------+----------+--------+

| 1| Richard| Hernandez| XXXXXXXXX| XXXXXXXXX| 6303 Heather Plaza| Brownsville| TX| 78521|

| 2| Mary| Barrett| XXXXXXXXX| XXXXXXXXX|9526 Noble Embers...| Littleton| CO| 80126|

| 3| Ann| Smith| XXXXXXXXX| XXXXXXXXX|3422 Blue Pioneer...| Caguas| PR| 00725|

| 4| Mary| Jones| XXXXXXXXX| XXXXXXXXX| 8324 Little Common| San Marcos| CA| 92069|

| 5| Robert| Hudson| XXXXXXXXX| XXXXXXXXX|10 Crystal River ...| Caguas| PR| 00725|

| 6| Mary| Smith| XXXXXXXXX| XXXXXXXXX|3151 Sleepy Quail...| Passaic| NJ| 07055|

| 7| Melissa| Wilcox| XXXXXXXXX| XXXXXXXXX|9453 High Concession| Caguas| PR| 00725|

| 8| Megan| Smith| XXXXXXXXX| XXXXXXXXX|3047 Foggy Forest...| Lawrence| MA| 01841|

| 9| Mary| Perez| XXXXXXXXX| XXXXXXXXX| 3616 Quaking Street| Caguas| PR| 00725|

| 10| Melissa| Smith| XXXXXXXXX| XXXXXXXXX|8598 Harvest Beac...| Stafford| VA| 22554|

| 11| Mary| Huffman| XXXXXXXXX| XXXXXXXXX| 3169 Stony Woods| Caguas| PR| 00725|

| 12|Christopher| Smith| XXXXXXXXX| XXXXXXXXX|5594 Jagged Ember...| San Antonio| TX| 78227|

| 13| Mary| Baldwin| XXXXXXXXX| XXXXXXXXX|7922 Iron Oak Gar...| Caguas| PR| 00725|

| 14| Katherine| Smith| XXXXXXXXX| XXXXXXXXX|5666 Hazy Pony Sq...| Pico Rivera| CA| 90660|

| 15| Jane| Luna| XXXXXXXXX| XXXXXXXXX| 673 Burning Glen| Fontana| CA| 92336|

| 16| Tiffany| Smith| XXXXXXXXX| XXXXXXXXX| 6651 Iron Port| Caguas| PR| 00725|

| 17| Mary| Robinson| XXXXXXXXX| XXXXXXXXX| 1325 Noble Pike| Taylor| MI| 48180|

| 18| Robert| Smith| XXXXXXXXX| XXXXXXXXX|2734 Hazy Butterf...| Martinez| CA| 94553|

| 19| Stephanie| Mitchell| XXXXXXXXX| XXXXXXXXX|3543 Red Treasure...| Caguas| PR| 00725|

| 20| Mary| Ellis| XXXXXXXXX| XXXXXXXXX| 4703 Old Route|West New York| NJ| 07093|

+-------+-----------+----------+----------+-------------+--------------------+-------------+----------+--------+

only showing top 20 rows

>>> ordersDFWithColumn.createOrReplaceTempView("ItemdDFCl")

>>> orderItemsDFColumn.createOrReplaceTempView("orderItemsDFCl")

>>> customersDFColumn.createOrReplaceTempView("customersDFCl")

>>> sqlContext.sql("create external table ordersH LOCATION 'retail\_db\_text' AS select \* from ItemdDFCl")

DataFrame[]

>>> sqlContext.sql("create external table ordersHV LOCATION '/user/saurabhbhanwala/retail\_db\_txt/orders' AS select \* from ItemdDFCl")

DataFrame[]

>>> sqlContext.sql("create external table orderItemsHV LOCATION '/user/saurabhbhanwala/retail\_db\_txt/orderItems' AS select \* from orderItemsDFCl")

DataFrame[]

>>> sqlContext.sql("create external table customersHV LOCATION '/user/saurabhbhanwala/retail\_db\_txt/customers' AS select \* from customersDFCl")

>>> sqlContext.sql("select o.order\_id, sum(oi.order\_item\_subtotal) as revenueclient\_loop: send disconnect: Connection reset by peerte,1,7), '\\-', '') as MONTH from ordersHV o inner join orderItemsHV oi on o.order\_id = oi.order\_item\_order\_id inner join customersDF c on o.order\_customer\_id = c.cust\_id group by o.order\_customer\_id,REGEXP\_REPLACE(substr(o.order\_date,1,7), '\\-', ''), o.order\_id").show()

>>> sqlContext.sql("select o.order\_id, sum(oi.order\_item\_subtotal) as revenue, o.order\_customer\_id,REGEXP\_REPLACE(substr(o.order\_date,1,7), '\\-', '') as MONTH from ordersHV o inner join orderItemsHV oi on o.order\_id = oi.order\_item\_order\_id group by o.order\_customer\_id,REGEXP\_REPLACE(substr(o.order\_date,1,7), '\\-', ''), o.order\_id").show()

+--------+------------------+-----------------+------+

|order\_id| revenue|order\_customer\_id| MONTH|

+--------+------------------+-----------------+------+

| 34606| 359.91| 10173|201402|

| 34623| 409.98| 11653|201402|

| 34904| 64.98| 9574|201402|

| 35663| 1099.83| 10744|201403|

| 36293| 545.89| 7118|201403|

| 36340| 229.95| 4221|201403|

| 36423| 649.89| 9453|201403|

| 36646| 789.86| 6664|201403|

| 36703| 399.99| 9596|201403|

| 36767| 719.88| 508|201403|

| 36843|489.96000000000004| 11662|201403|

| 38146|424.83000000000004| 695|201403|

| 38162|1099.8899999999999| 7097|201403|

| 38220| 689.89| 4004|201403|

| 38336|1034.7800000000002| 6611|201403|

| 38458| 379.95| 6291|201403|

| 38925| 873.8900000000001| 7726|201403|

| 38970| 1089.92| 3281|201403|

| 39212| 849.9300000000001| 11843|201403|

| 39577| 799.9000000000001| 10302|201403|

+--------+------------------+-----------------+------+

only showing top 20 rows

>>> sqlContext.sql("select c.cust\_fname, c.cust\_lname , REGEXP\_REPLACE(substr(o.order\_date,1,7), '\\-', '') as MONTH, round(sum(oi.order\_item\_subtotal), 2) as revenue from ordersHV o inner join orderItemsHV oi on o.order\_id = oi.order\_item\_order\_id inner join customersHV c on c.cust\_id = o.order\_customer\_id group by o.order\_customer\_id,REGEXP\_REPLACE(substr(o.order\_date,1,7), '\\-', ''), c.cust\_fname, c.cust\_lname order by revenue desc").show()