

ETL with Apache Pig

Apache Pig Basic

Running a word count pig script below:

Source codes for word count

Source code for the program is shown below. It's only 5 lines of codes. You can find it in /home/data/CSC534BDA/Pig/ directory

```
tdend2@node00:~  
[tdend2@node00 ~]$ cat /home/data/CSC534BDA/Pig/wordcount.pig  
-- filename: wordcount.pig  
-- Load input from the file named Mary, and call the single  
-- field in the record 'line'.  
lines = LOAD 'WordCount/MaryHadALittleLamb.txt' AS (line:chararray);  
  
-- TOKENIZE splits the line into a field for each word.  
-- FLATTEN will take the collection of records returned by  
-- TOKENIZE and produce a separate record for each one,  
-- calling the single field in the record word.  
words = FOREACH lines GENERATE FLATTEN(TOKENIZE(line)) as word;  
  
-- Now group them together by each word  
grouped = GROUP words BY word;  
  
-- Count them  
wordcount = FOREACH grouped GENERATE group, COUNT(words);  
  
-- Print out the results  
dump wordcount;  
[tdend2@node00 ~]$
```

Loading data to HDFS

We already have the data in your user directory in HDFS.

```
tdend2@node00:~  
[tdend2@node00 ~]$ hadoop fs -ls WordCount  
Found 2 items  
-rw-r--r-- 3 tdend2 hadoop 109 2024-09-07 13:01 WordCount/MaryHadALittleLamb.txt  
drwxr-xr-x - tdend2 hadoop 0 2024-09-07 14:20 WordCount/output  
[tdend2@node00 ~]$
```

Run the Pig program

We are ready to run first Pig program. We don't need to compile it. So ran it. It read the data from user directory in HDFS.

```
tdend2@node00:~  
[tdend2@node00 ~]$ pig -f /home/data/CSC534BDA/Pig/wordcount.pig  
WARNING: Use "yarn jar" to launch YARN applications.  
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).  
log4j:WARN Please initialize the log4j system properly.  
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.  
656 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : LOCAL  
658 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : MAPREDUCE  
658 [main] INFO org.apache.pig.ExecTypeProvider - Picked MAPREDUCE as the ExecType  
2024-09-22 16:46:40,857 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0-cdh6.3.2 (rUr  
2024-09-22 16:46:40,858 [main] INFO org.apache.pig.Main - Logging error messages to: /home/tdend2  
2024-09-22 16:46:41,299 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file /home/td  
2024-09-22 16:46:41,356 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job  
2024-09-22 16:46:41,356 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngin  
2024-09-22 16:46:42,101 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG  
2024-09-22 16:46:42,101 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timelin  
2024-09-22 16:46:42,822 [main] INFO org.apache.pig.impl.util.SpillableMemoryManager - Selected he  
hold = 489580128
```

```

Input(s):
Successfully read 4 records (503 bytes) from: "hdfs://node00.sun:8020/user/tdend2/WordCount/MaryHadALittleLamb.txt"

Output(s):
Successfully stored 19 records (182 bytes) in: "hdfs://node00.sun:8020/tmp/temp1383483782/tmp-724238377"

Counters:
Total records written : 19
Total bytes written : 182
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1722897143033_0332

2024-09-22 16:47:05,715 [main] INFO org.apache.hadoop.yarn.client.RMPProxy - Connecting to ResourceManager at node00
2024-09-22 16:47:05,718 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed
2024-09-22 16:47:05,737 [main] INFO org.apache.hadoop.yarn.client.RMPProxy - Connecting to ResourceManager at node00
2024-09-22 16:47:05,739 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed
2024-09-22 16:47:05,757 [main] INFO org.apache.hadoop.yarn.client.RMPProxy - Connecting to ResourceManager at node00
2024-09-22 16:47:05,761 [main] INFO org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed
2024-09-22 16:47:05,781 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher
2024-09-22 16:47:05,783 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - yarn.resourcemanager.system-
blisher.enabled

2024-09-22 16:47:05,784 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2024-09-22 16:47:05,797 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2024-09-22 16:47:05,797 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(a,1)
(as,1)
(to,1)
(and,1)
(go.,1)
(had,1)
(its,1)
(the,1)
(was,2)
(Mary,2)
(lamb,2)
(snow,1)
(sure,1)
(that,1)
(went,1)
(white,1)
(fleece,1)
(little,1)
(everywhere,1)
2024-09-22 16:47:05,867 [main] INFO org.apache.pig.Main - Pig script completed in 25 seconds and 109 milliseconds (25109 ms)
[tdend2@node00 ~]$

```

After producing lots of logs, saw some statistics about job (program ran)

It showed Input(s) shows input files' path/names and number of records.

1. ETL Real Dataset using Apache Pig

Created Pig scripts to carry out essential data operations and tasks. Implemented Pig Latin scripts to process, analyze, and manipulate data files of truck drivers' statistics.

Trucking IoT Data

- Dataset: located in /home/data/CSC534BDA/datasets/Truck-IoT
- Related GitHub project: <https://github.com/hortonworks-gallery/iot-truck-streaming>

Looking at a use case where we have a truck fleet. Each truck has been equipped to log location and event data. These events are streamed back to a datacenter where we will be

processing the data. The company wants to use this data to better understand risk. Our goal is to

find the sum of hours and miles logged for each truck driver.

The dataset, Trucking IoT, contains the following files:

- drivers.csv

- o This has driver information. It contains records showing driverId, name, ssn, location, certified, and wage-plan.

- timesheet.csv

- o This contains records showing driverId, week, hours-logged, and miles-logged.

- truck_event_text_partition.csv

- o This contains records showing driverId, truckId, eventTime, eventType, longitude, latitude, eventKey, CorrelationId, driverName, routeId, routeName, and eventDate

The dataset is located in /home/data/CSC534BDA/datasets/Truck-IoT of Linux file system (Not in the HDFS) of the cluster.

ls -alFh /home/data/CSC534BDA/datasets/Truck-IoT

```
[tdend2@node00 ~]$ ls -alFh /home/data/CSC534BDA/datasets/Truck-IoT
total 2.2M
drwxrwxr-x 2 sslee777 sslee777 84 Sep 15 2020 ./
drwxrwxr-x 6 sslee777 sslee777 76 Oct 22 2021 ../
-rw-rw-r-- 1 sslee777 sslee777 2.0K Sep 15 2020 drivers.csv
-rw-rw-r-- 1 sslee777 sslee777 26K Sep 15 2020 timesheet.csv
-rw-rw-r-- 1 sslee777 sslee777 2.2M Sep 15 2020 truck_event_text_partition.csv
[tdend2@node00 ~]$
```

To see the first 10 rows of **drivers.csv**, used 'head' Linux commands.

head /home/data/CSC534BDA/datasets/Truck-IoT/drivers.csv

```
tdend2@node00:~
[tdend2@node00 ~]$ head /home/data/CSC534BDA/datasets/Truck-IoT/drivers.csv
driverId,name,ssn,location,certified,wage-plan
10,George Vetticaden,621011971,244-4532 Nulla Rd.,N,miles
11,Jamie Engesser,262112338,366-4125 Ac Street,N,miles
12,Paul Coddin,198041975,Ap #622-957 Risus. Street,Y,hours
13,Joe Niemiec,139907145,2071 Hendrerit. Ave,Y,hours
14,Adis Cesir,820812209,Ap #810-1228 In St.,Y,hours
15,Rohit Bakshi,239005227,648-5681 Dui- Rd.,Y,hours
16,Tom McCuch,363303105,P.O. Box 313- 962 Parturient Rd.,Y,hours
17,Eric Mizell,123808238,P.O. Box 579- 2191 Gravida. Street,Y,hours
18,Grant Liu,171010151,Ap #928-3159 Vestibulum Av.,Y,hours
[tdend2@node00 ~]$
```

To see the first 10 rows of **timesheet.csv**, used 'head' Linux commands. It has a lot of fields about trucks and gas/miles.***head /home/data/CSC534BDA/datasets/Truck-IoT/timesheet.csv***

```
tdend2@node00:~
[tdend2@node00 ~]$ head /home/data/CSC534BDA/datasets/Truck-IoT/timesheet.csv
driverId,week,hours-logged,miles-logged
10,1,70,3300
10,2,70,3300
10,3,60,2800
10,4,70,3100
10,5,70,3200
10,6,70,3300
10,7,70,3000
10,8,70,3300
10,9,70,3200
[tdend2@node00 ~]$
```

To see the first 10 rows of **truck_event_text_partition.csv**, use 'head' Linux commands.
head /home/data/CSC534BDA/datasets/TruckIoT/truck_event_text_partition.csv

```
tdend2@node00:~$ head /home/data/CSC534BDA/datasets/Truck-IoT/truck_event_text_partition.csv
driverId,truckId,eventTime,eventType,longitude,latitude,eventKey,CorrelationId,driverName,routeId,routeName,eventDate
14,25,59:21.4,Normal,-94.58,37.03,14|25|9223370572464814373,3.66E+18,Adis Cesir,160405074,Joplin to Kansas City Route 2,2016-05-27-22
18,16,59:21.7,Normal,-89.66,39.78,18|16|9223370572464814089,3.66E+18,Grant Liu,1565885487,Springfield to KC Via Hanibal,2016-05-27-22
27,105,59:21.7,Normal,-90.21,38.65,27|105|9223370572464814070,3.66E+18,Mark Lochbihler,1325562373,Springfield to KC Via Columbia Route 2,2016-05-27-22
11,74,59:21.7,Normal,-90.2,38.65,11|74|9223370572464814123,3.66E+18,Jamie Engesser,1567254452,Saint Louis to Memphis Route2,2016-05-27-22
22,87,59:21.7,Normal,-90.04,35.19,22|87|9223370572464814101,3.66E+18,Nadeem Asghar,1198242881,Saint Louis to Chicago Route2,2016-05-27-22
22,87,59:22.3,Normal,-90.37,35.21,22|87|9223370572464813486,3.66E+18,Nadeem Asghar,1198242881,Saint Louis to Chicago Route2,2016-05-27-22
23,68,59:22.4,Normal,-89.91,40.86,23|68|9223370572464813450,3.66E+18,Adam Diaz,160405074,Joplin to Kansas City Route 2,2016-05-27-22
11,74,59:22.5,Normal,-89.74,39.1,11|74|9223370572464813355,3.66E+18,Jamie Engesser,1567254452,Saint Louis to Memphis Route2,2016-05-27-22
20,41,59:22.5,Normal,-93.36,41.69,20|41|9223370572464813344,3.66E+18,Chris Harris,160779139,Des Moines to Chicago Route 2,2016-05-27-22
tdend2@node00 ~]$
```

Column names in truck_event_text_partition.csv:

driverId,truckId,eventTime,eventType,longitude,latitude,eventKey,correlationId,driverName,routeId,routeName,eventDate

Load the data to HDFS

Created a 'Pig' directory in CSC534BDA directory in Linux filesystem. Then changed working directory to Pig.

```
mkdir CSC534BDA/pig
cd CSC534BDA/pig/
```

```
tdend2@node00:~/CSC534BDA/pig
[tdend2@node00 ~]$ mkdir CSC534BDA/pig
[tdend2@node00 ~]$ cd CSC534BDA/pig/
[tdend2@node00 pig]$ ls
[tdend2@node00 pig]$
```

Loaded the dataset (three files) to HDFS from /home/data/CSC534BDA/datasets/Truck-IoT/ as shown below:

```
[tdend2@node00 pig]$ ls /home/data/CSC534BDA/datasets/Truck-IoT
drivers.csv timesheet.csv truck_event_text_partition.csv
[tdend2@node00 pig]$
```

Created a directory in HDFS to store dataset and loaded the file into it

```
tdend2@node00:~/CSC534BDA/pig
[tdend2@node00 pig]$ hadoop fs -mkdir /user/tdend2/Pig
[tdend2@node00 pig]$ hadoop fs -put /home/data/CSC534BDA/datasets/Truck-IoT/*.csv /user/tdend2/Pig/
[tdend2@node00 pig]$ hadoop fs -ls Pig
Found 3 items
-rw-r--r-- 3 tdend2 hadoop 2043 2024-09-22 17:22 Pig/drivers.csv
-rw-r--r-- 3 tdend2 hadoop 26205 2024-09-22 17:22 Pig/timesheet.csv
-rw-r--r-- 3 tdend2 hadoop 2272077 2024-09-22 17:22 Pig/truck_event_text_partition.csv
[tdend2@node00 pig]$
```

After loading the data, we see the dataset in HDFS. **hadoop fs -ls Pig**

hadoop fs -put /home/data/CSC534BDA/datasets/Truck-IoT/*.csv /user/tdend2/Pig/

```
[tdend2@node00 pig]$ hadoop fs -ls /user/tdend2/Pig
Found 3 items
-rw-r--r--  3 tdend2 hadoop      2043 2024-09-22 17:22 /user/tdend2/Pig/drivers.csv
-rw-r--r--  3 tdend2 hadoop     26205 2024-09-22 17:22 /user/tdend2/Pig/timesheet.csv
-rw-r--r--  3 tdend2 hadoop    2272077 2024-09-22 17:22 /user/tdend2/Pig/truck_event_text_partition.csv
[tdend2@node00 pig]$

[tdend2@node00 pig]$ hadoop fs -ls /user/$USER/Pig
Found 3 items
-rw-r--r--  3 tdend2 hadoop      2043 2024-09-22 17:22 /user/tdend2/Pig/drivers.csv
-rw-r--r--  3 tdend2 hadoop     26205 2024-09-22 17:22 /user/tdend2/Pig/timesheet.csv
-rw-r--r--  3 tdend2 hadoop    2272077 2024-09-22 17:22 /user/tdend2/Pig/truck_event_text_partition.csv
[tdend2@node00 pig]$
```

Extract, Transform, and Load (ETL) data

Created a Pig Script

First, created a new file named `sum_of_hours_miles.pig` ***touch sum_of_hours_miles.pig*** then use vim to edit it ***touch sum_of_hours_miles.pig***

To enter insert mode in vim press i (means insert) and to exit press esc and :wq (write and quit) to save. We may also exit without saving by pressing :q! (means quit)

```
tdend2@node00:~/CSC534BDA/pig
[tdend2@node00 pig]$ touch sum_of_hours_miles.pig
[tdend2@node00 pig]$ vim sum_of_hours_miles.pig
```

Writing a Pig script

Load drivers.csv Data

The first thing we need to do is load the data. We use the load statement for this. The PigStorage

function is what does the loading and we pass it a comma as the data delimiter. Our code is:
drivers = LOAD 'Pig/drivers.csv' USING PigStorage(',');

Filter Out Data

First line of the dataset has header (column names). To filter out the first row of the data (header), we have to add this line:

Define a Relation with a Schema

The next thing we want to do is name the fields. We will use a FOREACH statement to iterate through the raw_drivers data object.

So, the FOREACH statement will iterate through the raw_drivers data object and GENERATE pulls out selected fields and assigns them names. The new data object we are creating is then named driver_details. Our code will now be:

Perform these operations for timesheet data as well

Load the timesheet data and then filter out the first row of the data to remove column headings and then use FOREACH statement to iterate each row and GENERATE to pull out selected fields and assign them names.

Filter The Data (all hours and miles for each driverId)

The next line of code is a GROUP statement that groups the elements in timesheet_logged by the

driverId field. So, the grp_logged object will then be indexed by driverId. In the next statement as we iterate through grp_logged we will go through driverId by driverId. Type in the code:

Find the Sum of Hours and Miles Logged by each Driver

In the next FOREACH statement, we are going to find the sum of hours and miles logged by each driver. The code for this is:

Join driverId, Name, Hours and Miles Logged

Now that we have the sum of hours and miles logged, we need to join this with the driver_details

data object so we can pick up the name of the driver. The result will be a dataset with driverId, name, hours logged and miles logged. At the end we DUMP the data to the output.

```
raw_drivers = FILTER drivers BY $0>1;
drivers_details = FOREACH raw_drivers GENERATE $0 AS driverId, $1 AS name;
timesheet = LOAD 'Pig/timesheet.csv' USING PigStorage(',');
raw_timesheet = FILTER timesheet by $0>1;
timesheet_logged = FOREACH raw_timesheet GENERATE $0 AS driverId, $2 AS
hours_logged,
$3 AS miles_logged;
```

Full Pig Latin script

Execute the Pig script

Ran the Pig Latin code by running 'pig -f source-file.pig'. Pig use YARN as a default like MapReduce Java program.

If you have any errors, you will see the messages in pig_xxxx.log file, shown above.

```
join_sum_logged = JOIN sum_logged by driverId, drivers_details by driverId;
join_data = FOREACH join_sum_logged GENERATE $0 as driverId, $4 as name, $1 as
hours_logged, $2 as miles_logged;
dump join_data;
drivers = LOAD 'Pig/drivers.csv' USING PigStorage(',');
raw_drivers = FILTER drivers BY $0>1;
drivers_details = FOREACH raw_drivers GENERATE $0 AS driverId, $1 AS name;
timesheet = LOAD 'Pig/timesheet.csv' USING PigStorage(',');
raw_timesheet = FILTER timesheet by $0>1;
timesheet_logged = FOREACH raw_timesheet GENERATE $0 AS driverId, $2 AS
hours_logged,
$3 AS miles_logged;
grp_logged = GROUP timesheet_logged by driverId;
sum_logged = FOREACH grp_logged GENERATE group as driverId,
SUM(timesheet_logged.hours_logged) as sum_hourslogged,
SUM(timesheet_logged.miles_logged) as sum_mileslogged;
join_sum_logged = JOIN sum_logged by driverId, drivers_details by driverId;
join_data = FOREACH join_sum_logged GENERATE $0 as driverId, $4 as name, $1 as
hours_logged, $2 as miles_logged;
dump join_data;
```



```

tdend2@node00:~/CSC534BDA/pig
drivers = LOAD 'Pig/drivers.csv' USING PigStorage(',');
raw_drivers = FILTER drivers BY $0>1;
drivers_details = FOREACH raw_drivers GENERATE $0 AS driverId, $1 AS name;
timesheet = LOAD 'Pig/timesheet.csv' USING PigStorage(',');
raw_timesheet = FILTER timesheet by $0>1;
timesheet_logged = FOREACH raw_timesheet GENERATE $0 AS driverId, $2 AS hours_logged,
$3 AS miles_logged;
grp_logged = GROUP timesheet_logged by driverId;
sum_logged = FOREACH grp_logged GENERATE group as driverId,
SUM(timesheet_logged.hours_logged) as sum_hourslogged,
SUM(timesheet_logged.miles_logged) as sum_mileslogged;
join_sum_logged = JOIN sum_logged by driverId, drivers_details by driverId;
join_data = FOREACH join_sum_logged GENERATE $0 as driverId, $4 as name, $1 as
hours_logged, $2 as miles_logged;
dump join_data;

```

Execute the Pig script

Ran the Pig Lation code by running 'pig -f source-file.pig'. Pig use YARN as a default like MapReduce Java program.

```

tdend2@node00:~/CSC534BDA/pig
[tdend2@node00 pig]$ touch sum_of_hours_miles.pig
[tdend2@node00 pig]$ vim sum_of_hours_miles.pig
[tdend2@node00 pig]$ [tdend2@node00 pig]$ pig -f sum_of_hours_miles.pig
WARNING: Use "yarn jar" to launch YARN applications.
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
660 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : LOCAL
661 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : MAPREDUCE
661 [main] INFO org.apache.pig.ExecTypeProvider - Picked MAPREDUCE as the ExecType

```

HadoopVersion	PigVersion	UserId	StartedAt	FinishedAt	Features
3.0.0-cdh6.3.2	0.17.0-cdh6.3.2	tdend2	2024-09-22 17:42:21	2024-09-22 17:43:05	HASH_JOIN, GROUP_BY, FILTER

Success!

```

tdend2@node00:~/CSC534BDA/pig
Input(s):
Successfully read 1769 records (26584 bytes) from: "hdfs://node00.sun:8020/user/tdend2/Pig/timesheet.csv"
Successfully read 35 records from: "hdfs://node00.sun:8020/user/tdend2/Pig/drivers.csv"

Output(s):
Successfully stored 34 records (1277 bytes) in: "hdfs://node00.sun:8020/tmp/temp1294404681/tmp757692174"

Counters:
Total records written : 34
Total bytes written : 1277
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1722897143033_0355 -> job_1722897143033_0356,
job_1722897143033_0356

```

Last half of output:

```

(10,George Vetticaden,3232.0,147150.0)
(11,Jamie Engesser,3642.0,179300.0)
(12,Paul Coddin,2639.0,135962.0)
(13,Joe Niemiec,2727.0,134126.0)
(14,Adis Cesir,2781.0,136624.0)
(15,Rohit Bakshi,2734.0,138750.0)
(16,Tom McCuch,2746.0,137205.0)
(17,Eric Mizell,2701.0,135992.0)
(18,Grant Liu,2654.0,137834.0)
(19,Ajay Singh,2738.0,137968.0)
(20,Chris Harris,2644.0,134564.0)
(21,Jeff Markham,2751.0,138719.0)
(22,Nadeem Asghar,2733.0,137550.0)
(23,Adam Diaz,2750.0,137980.0)
(24,Don Hilborn,2647.0,134461.0)
(25,Jean-Philippe Playe,2723.0,139180.0)
(26,Michael Aube,2730.0,137530.0)
(27,Mark Lochbihler,2771.0,137922.0)
(28,Olivier Renault,2723.0,137469.0)
(29,Teddy Choi,2760.0,138255.0)
(30,Dan Rice,2773.0,137473.0)
(31,Rommel Garcia,2704.0,137057.0)
(32,Ryan Templeton,2736.0,137422.0)
(33,Sridhara Sabbella,2759.0,139285.0)
(34,Frank Romano,2811.0,137728.0)
(35,Emil Siemes,2728.0,138727.0)
(36,Andrew Grande,2795.0,138025.0)
(37,Wes Floyd,2694.0,137223.0)
(38,Scott Shaw,2760.0,137464.0)
(39,David Kaiser,2745.0,138788.0)
(40,Nicolas Maillard,2700.0,136931.0)
(41,Greg Phillips,2723.0,138407.0)
(42,Randy Gelhausen,2697.0,136673.0)
(43,Dave Patton,2750.0,136993.0)
2024-09-22 17:43:05,930 [main] INFO org.apache.pig.Main - Pig script completed in 46 seconds and
[tdend2@node00 pig]$

```

So, we have created a simple Pig script that reads in some comma separated data

2. Write Pig scripts for finding truck drivers who exceeded the speed limit, 'Overspeed'.

Pig operates on data flows. We consider each group of rows together and we specify how we operate on them as a group. As the datasets get larger and/or add fields our Pig script will remain pretty much the same because it is concentrating on how we want to manipulate the data.

a. Dataset: Truck IoT dataset

i. Dataset location (Linux filesystem):

/home/data/CSC534BDA/datasets/Truck-IoT/

ii. Filenames: truck_event_text_partition.csv

b. Wrote and ran Pig scripts

i. (20pts) Found all truck drivers who exceeded the speed limit, 'Overspeed'

ii. (10pts) Defined schema when you load the data (and not used \$0, \$1, etc.)

iii. (10pts) Showed the driver's events grouped if the drivers exceeded the speed limit multiple times.

c. (10pts) Showed outputs: all of the 'overspeed' drivers

i. Output example (shown first row only)

```
(29,{(29,66,00:47.8,Overspeed,-
94.57,35.37,29|66|9223370572464728016,3660000000000000000,Teddy
Choi,803014426,Wichita to Little Rock Route 2,2016-05-27-22))}
```

Inspecting the dataset headband found the column names from the header as follows:

Column names in truck_event_text_partition.csv:

driverId,truckId,eventTime,eventType,longitude,latitude,eventKey,correlationId,driverName,routeId,routeName,eventDate

Pig script:

-- Load the dataset with schema

```
data = LOAD '/user/tdend2/Pig/truck_event_text_partition.csv'
```

```
USING PigStorage(',')
```

```
AS (
```

```
    driverId:int,
```

```
    truckId:int,
```

```
    eventTime:chararray,
```

```
    eventType:chararray,
```

```
    longitude:float,
```

```
    latitude:float,
```

```
    eventKey:chararray,
```

```
    correlationId:chararray,
```

```
    driverName:chararray,
```

```
    routeId:chararray,
```

```
    routeName:chararray,
```

```
    eventDate:chararray
```

```
);
```

```

-- Filter out rows where driverId is null (this would typically be the header row)
data_filtered = FILTER data BY driverId IS NOT NULL;

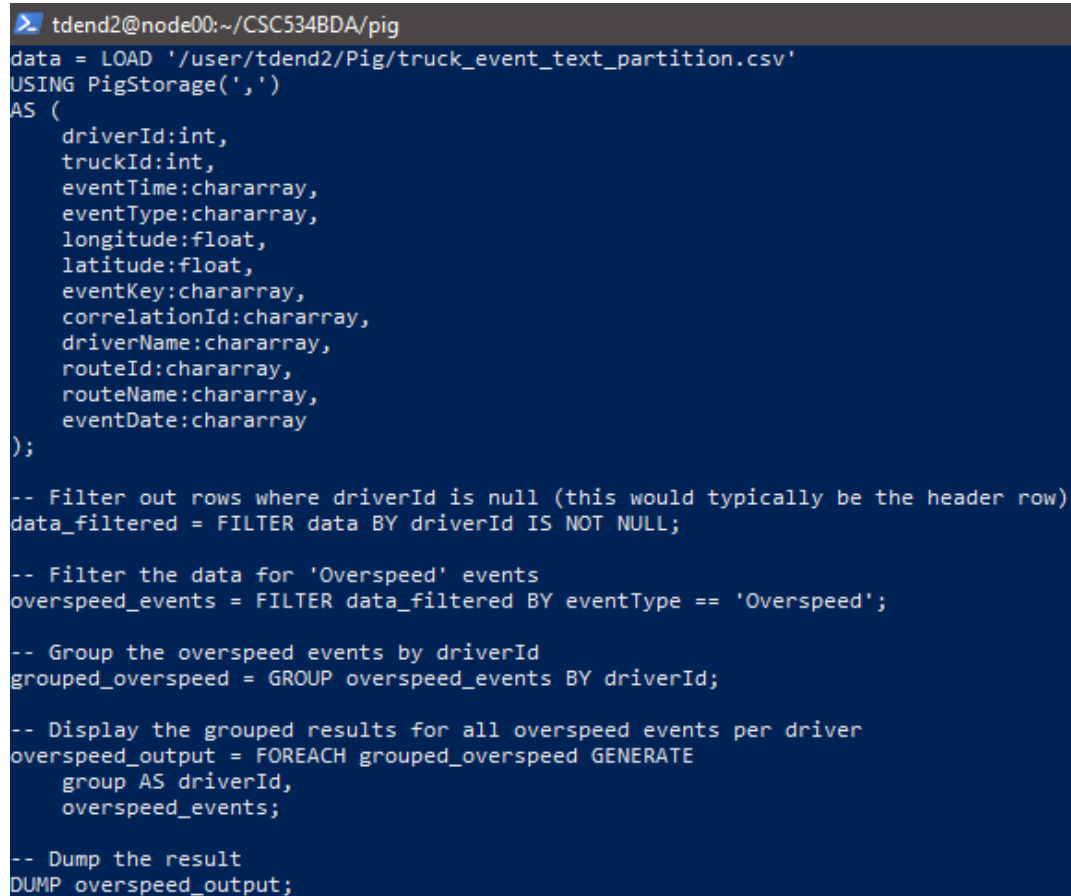
-- Filter the data for 'Overspeed' events
overspeed_events = FILTER data_filtered BY eventType == 'Overspeed';

-- Group the overspeed events by driverId
grouped_overspeed = GROUP overspeed_events BY driverId;

-- Display the grouped results for all overspeed events per driver
overspeed_output = FOREACH grouped_overspeed GENERATE
    group AS driverId,
    overspeed_events;

-- Dump the result
DUMP overspeed_output;

```



```

tdend2@node00:~/CSC534BDA/pig
data = LOAD '/user/tdend2/Pig/truck_event_text_partition.csv'
USING PigStorage(',')
AS (
    driverId:int,
    truckId:int,
    eventTime:chararray,
    eventType:chararray,
    longitude:float,
    latitude:float,
    eventKey:chararray,
    correlationId:chararray,
    driverName:chararray,
    routeId:chararray,
    routeName:chararray,
    eventDate:chararray
);

-- Filter out rows where driverId is null (this would typically be the header row)
data_filtered = FILTER data BY driverId IS NOT NULL;

-- Filter the data for 'Overspeed' events
overspeed_events = FILTER data_filtered BY eventType == 'Overspeed';

-- Group the overspeed events by driverId
grouped_overspeed = GROUP overspeed_events BY driverId;

-- Display the grouped results for all overspeed events per driver
overspeed_output = FOREACH grouped_overspeed GENERATE
    group AS driverId,
    overspeed_events;

-- Dump the result
DUMP overspeed_output;

```

Executed the above script named '**overspeed_drivers.pig**':

```
> tdend2@node00:~/CSC534BDA/pig
```

```
[tdend2@node00 pig]$ vim overspeed_drivers.pig
[tdend2@node00 pig]$ pig -f overspeed_drivers.pig
WARNING: Use "yarn jar" to launch YARN applications.
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
664 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : LOCAL
665 [main] INFO org.apache.pig.ExecTypeProvider - Trying ExecType : MAPREDUCE
665 [main] INFO org.apache.pig.ExecTypeProvider - Picked MAPREDUCE as the ExecType
```

```
> tdend2@node00:~/CSC534BDA/pig
```

Success!

Job Stats (time in seconds):

JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceTime	AvgReduceTime
job_1722897143033_0367	1	1	4	4	4	3	3	3	3

MedianReduceTime Alias Feature Outputs

job_1722897143033_0367 1 1 4 4 4 4 3 3 3 3 data,data_filtered,grouped_overspeed,overspeed_output

GROUP BY hdfs://node00.sun:8020/tmp/temp-1203137391/tmp944629792,

Input(s):

Successfully read 17076 records (2272473 bytes) from: "/user/tdend2/Pig/truck_event_text_partition.csv"

Output(s):

Successfully stored 8 records (1423 bytes) in: "hdfs://node00.sun:8020/tmp/temp-1203137391/tmp944629792"

Counters:

Total records written : 8

Total bytes written : 1423

Spillable Memory Manager spill count : 0

Total bags proactively spilled: 0

Total records proactively spilled: 0

Job DAG:

job_1722897143033_0367

Output:

```
tdend2@node00:~/CSC534BDA/pig
2024-09-22 18:33:00,412 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
(10,{{(10,85,00:39.7,Overspeed,-94.23,37.09,10|85|9223370572464736126,3.66E+18,George Vetticaden,1390372503,Saint Louis to Tulsa,2016-05-27-22),(10,85,59:46.9,Ov
0572464788896,3.66E+18,George Vetticaden,1390372503,Saint Louis to Tulsa,2016-05-27-22)}}))
(11,{{(11,74,59:29.1,Overspeed,-88.07,41.48,11|74|9223370572464806746,3.66E+18,Jamie Engesser,1567254452,Saint Louis to Memphis Route2,2016-05-27-22)}})
(18,{{(18,16,00:47.2,Overspeed,-94.28,39.53,18|16|9223370572464728575,3.66E+18,Grant Liu,1565885487,Springfield to KC Via Hanibal,2016-05-27-22)}})
(20,{{(20,41,00:46.9,Overspeed,-89.03,41.92,20|41|9223370572464728915,3.66E+18,Chris Harris,160779139,Des Moines to Chicago Route 2,2016-05-27-22)}})
(25,{{(25,96,00:40.1,Overspeed,-89.54,36.84,25|96|9223370572464735726,3.66E+18,Jean-Philippe Player,371182829,Memphis to Little Rock,2016-05-27-22)}})
(26,{{(26,57,00:48.8,Overspeed,-95.99,36.17,26|57|9223370572464727046,3.66E+18,Michael Aube,1325712174,Saint Louis to Tulsa Route2,2016-05-27-22)}})
(28,{{(28,39,00:47.5,Overspeed,-94.28,39.53,28|39|9223370572464728273,3.66E+18,Olivier Renault,137128276,Springfield to KC Via Hanibal Route 2,2016-05-27-22)}})
(29,{{(29,66,00:47.8,Overspeed,-94.57,35.37,29|66|9223370572464728016,3.66E+18,Teddy Choi,803014426,Wichita to Little Rock Route 2,2016-05-27-22)}})
2024-09-22 18:33:00,489 [main] INFO org.apache.pig.Main - Pig script completed in 30 seconds and 74 milliseconds (30074 ms)
[tdend2@node00 pig]$
```

8 drivers overspeeded.

One record from above output is:

```
(26,{{(26,57,00:48.8,Overspeed,-95.99,36.17,26|57|9223370572464727046,3.66E+18,Michael Aube,1325712174,Saint Louis to Tulsa Route2,2016-05-27-22)}})
```

=====THE END=====