Zafar Mahmood Distributed Lab Solution 5 17 Mai 2017

Hadoop 2.7.2

Subversion https://git-wip-us.apache.org/repos/asf/hadoop.git -r

b165c4fe8a74265c792ce23f546c64604acf0e41

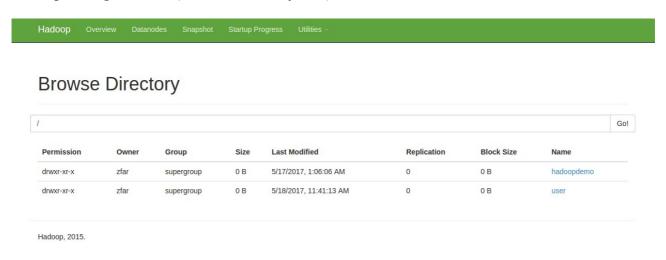
Compiled by jenkins on 2016-01-26T00:08Z

Compiled with protoc 2.5.0

From source with checksum d0fda26633fa762bff87ec759ebe689c

This command was run using /home/zfar/hadoop-2.7.2/share/hadoop/common/hadoop-common-2.7.2.jar

Hadoop Configurations: (Browse the file system)



Exercise 2 : Analysis Of Airport

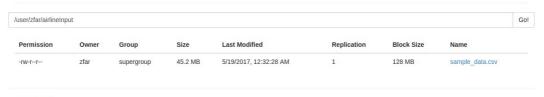
• Data Filtration Applied!

- Geography All
- Year 2017
- Filter Period January
- Time Period Flight Date
- o Origin Origin
- Destination Destination
- Departure Performance DepTime, DepDelay
- Arrival Performance: ArrTime, ArrDelay

• Computing the maximum, minimum, and average departure delay for each airport

- Hadoop Steps
 - Upload the data file to Data Node
 - hdfs dfs -put hadoop-2.7.2/hadoop/movieLen/output.dat /user/zfar/airLine/Input

Browse Directory



Hadoop, 2015.

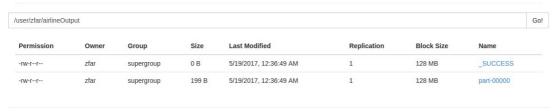
- Algorithm steps Steps Followed
 - Mapper file
 - Receives the input sys
 - Now strip and split the data ":: "
 - Now take the column that needed
 - Delay and Airport Names
 - Pass it reducer function
 - Now it will start checking the airports names, present and old as it come sorted
 - Now a list has been make for each airport, which will store its min, max and average delay time, in the end it get displayed in hadoop -output directory

```
-9 48 19.5
            -10 -7 -8.5
            -2 354 143.25
            -6 88 41.0
            -4 -3 -3.5
            -10 -6 -8.33333333333
-7 400 31.6986969471
ATW
AUS
            -7 -7 -7.0
-9 63 6.20574434807
-7 6 0.25
AVL
BDL
            -10 27 8.28193456615
            1 245 92.3333333333
            -11 -10 -10.5
-17 1447 110.95192718
BMI
BNA
           0 0 0.0
-11 160 8.43913718607
BOI
BOS
                    -6.0
                    -1.0
```

using hadoop run command

hadoop jar hadoop-2.7.2/share/hadoop/tools/lib/hadoop-streaming-2.7.2.jar -file hadoop-2.7.2/hadoop/airline/mapper.py -mapper "python2.7 mapper.py" -file hadoop-2.7.2/hadoop/airline/reducer.py -reducer "python2.7 reducer.py" -input /user/zfar/airline/input -output /user/zfar/airline/output

Browse Directory



Hadoop, 2015.

- Computing a ranking list that contains top 10 airports by their average Arrival delay
 - Algorithm is used same as above but in the end we have names of all airports but here its evaluated using their arrival delay
 - After when the its computer, sort arrival delay and display the top 10 airports
 - file mapper_arrival.py , reducer_arrival.py

```
['FNT', 'ERI', 'PIB', 'BQK', 'DHN', 'DAY', 'ABE', 'EWN', 'FSM', 'BOI']
[606.0, 640.0, 715.0, 722.0, 732.0, 749.0, 750.0, 825.0, 840.0, 851.0]
```

Exercise: Analysis of Movie dataset using Map and Reduce

Preprocessed: Merged two files *

Part 1: Highest Average Rating Movie Title

Algorithm

Mapper function: passed key, value [Movie title, rating]

Reducer function : Calculate the average of each movie and result back

Files Attached (mapper_title.py, reducer_title.py & all other files with title are outputs with different mapper and reducers)

| | Mapper_1 | Mapper_3 |
|-----------|----------------------------|-----------------------------|
| Reducer_1 | 172.080148935318 | 161.023507 |
| Reducer_3 | 83.188 , 125.1353 , 164.94 | 78.066 , 116.282 , 154.0162 |

Part 2: Find the user who has assign lowest average rating among all the users who rated more than 40 times

Algorithm:

As this is the simple algorithm

- Take a userID that is coming to reducers from the mapper,
- Now count the number of items (rated items) that it received from mapper store it in an array
- Now when the new user comes with different ID, we will check the length of stored ratings from array, If its > 40 then take mean and store it in final array
- For finalizing we just take the minimum rating corresponding to index for User ID

Files Attached (40_User_mapper.py, 40 User_reducer.py and all other files with name 40 user are output files with different mappers and reducers)

With Short Sample Data

Part 3: Find the highest average rated movie genre

Algorithm:

The algorithm is same is the old one is to keep track of genere (As this is used as a key in this question)

but before it has been merged (Movie and Ranking) based on their ID's

Tested On small Data

```
Best Ratio Genere Drama|Mystery
rating 3.91463414634
total program time : 0.9292268753051758
```

Files attached (mapper_t_3.py , reducer_t_3.py and all files with t_3 are output files)

| | Reducer _ 1 | Reducer _ 3 |
|------------|-------------|---------------------------|
| Mapper _ 1 | 118.14 | 38.103, 115.98 , 137.13 |
| Mapper _ 3 | 121.726 | 67.5201 , 81.92 , 133.260 |