Resource Used:

Running Apache pig Script on Hadoop Cluster [Activity] | Finding most popular movie in a dataset (BY : AmpCode), Link : https://www.youtube.com/watch?v=lh1bTEwxdBY

Week 3 : Exercise : Execute the pig script to find the “most popular moive in the dataset”. In this example we will be dealing with 2 files (ratings.data and movies.item).

Consider the dataset:

wget <https://raw.githubusercontent.com/ashaypatil11/hadoop/main/movies.item>,

wget https://raw.githubusercontent.com/ashaypatil11/hadoop/main/ratings.data

Pig : Is just like sql scripting language which is written in pig

Steps:

Creating a realtions: Relations are just like a data set or data frames when we deal with pig.

**In the program file:**

Creater realtion: 1 ratings

ratings: Is realtion which is created for uploading up the data for ratings.data file. (Use the LOAD method for loading the data file, save to HDFS (provide HDFS Path)), Need to specify “Main Editor” to do that use AS (we give the columns and the data types for the specified fields)

**CODE:**

ratings = LOAD ‘/pig1/ratings.data’ AS (userdID:int, movieID:int, rating:int, ratingTime:int);

metadata = LOAD ‘/pig1/movies.item’ USING PigStorage(‘ |’) AS (movieID:int, movieTitle:chararray, releaseDate:chararray, videoRelease:chararray, imdblink:chararray);

nameLookup = FOREACH metadata GENERATE movieID, movieTitle, ToUnixTime(ToDate(releaseDate, ‘dd-mm-yyyy’)) AS releaseTime;

ratingsByMovie = GROUP ratings BY movieID;

avgRatings = FOREACH ratingsByMovie GENERATE group as movieID, AVG(ratings.rating) as avgRating;

fiveStarMovies = FILTER avgRatings BY avgRating > 4.0;

fiveStarsWithData = JOIN fiveStarMovies By movieID, nameLookup By movieID;

oldestFiveStarMovies = ORDER fiveStarsWithData BY nameLookup::releaseTime;

DUMP oldestFiveStarMovies;

Create another relation: 2 metadata

metadata: Here we are loading the movies.item file but in this case the file is demilited by pipe character ( ‘|‘) for this purpose we have used PigStorage for specifying the different delimiter

for this file. We are providng metadata as movieid , movie title with datatypes as chararray (chararray is nothing but string for pig) – Indirectly these all fields are present in our movies.items file

Using these relations (ratings and metadata) we have loaded the dataset.

Create another relation: 3 nameLookup

nameLookup: FOREACH will be used for creating another relation from existing relation (using metadata relation we are creating nameLookup), in which we are generating the movieid and movietile to create new column (releasetime). But releasetime is generated by using release date. In data set the format of releasedate is dd-mm-yy. But we need to convert to required format (unix time stamp)by using ToUnixTIME. So we use orderby operation on top of this column

Create another relation: 4 ratingsByMovie

ratingsByMovie: Using GROUP BY function (we are grouping the ratings by movies id) : Grouping the ratings by the movies id (It may seem like reducer function if comapred it to mapreduce operations-more narrowed down the results)

Create another relation: 5 avgratings

avgratings: Calculate the average ratingd for each movie id. For that again use FOREACH (using the existing relation avgratingsByMovie and generate it as avgRating.

Create another relation: 6 fiveStarMovies

fiveStarMovies: (To get most popular movies – means appreciated by audience, movie will be having atleast 4 star or above rating). To do it use “FILTER”- Its just like where condition in SQL.

Here – Filtering out the average ratings by the ratings greater then 4

**Note :** Same thing if you do using MapReduce program – program will be lengthier and Time Consuming. Pig gives the capability to process our data more efficiently in simple manner.

Create another relation: 7 fiveStarsWithData

fiveStarsWithData: here joining the 2 relations (1st is fiveStarMovies and nameLookup). It means we are joining the fields used in nameLookup i.e. movieID, movieTitle,releaseTime on the basis of movieID because the movieID (movieID is common in both)

Create another relation: 8 oldestFiveStarMovies

oldestFiveStarMovies: ordering out the data by using the release time which is present in name lookup , so as you can see the releasetime is present in the name lookup, so this releasetime which is being converted to unixtimestamp can be used as an order by (we can order that by ascending order)

Dump: Dump operation is used to showup the relation on the console. It gives the oldest five star movies fromour dataset (rating.data and movies.items files)

EXECUTION:

1. hdoop@hadoop-master:~$ wget https://raw.githubusercontent.com/ashaypatil11/hadoop/main/ratings.data

**Output:**

**-**-2024-01-13 13:31:42-- https://raw.githubusercontent.com/ashaypatil11/hadoop/main/ratings.data

Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...

Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 2079229 (2.0M) [text/plain]

Saving to: ‘ratings.data’

ratings.data 100%[===================>] 1.98M 5.76MB/s in 0.3s

2024-01-13 13:31:43 (5.76 MB/s) - ‘ratings.data’ saved [2079229/2079229]

1. hdoop@hadoop-master:~$ wget https://raw.githubusercontent.com/ashaypatil11/hadoop/main/movies.item

**Output:**

--2024-01-13 13:32:58-- https://raw.githubusercontent.com/ashaypatil11/hadoop/main/movies.item

Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.133, 185.199.108.133, 185.199.110.133, ...

Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.109.133|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 236344 (231K) [text/plain]

Saving to: ‘movies.item’

movies.item 100%[===================>] 230.80K --.-KB/s in 0.09s

2024-01-13 13:32:59 (2.63 MB/s) - ‘movies.item’ saved [236344/236344]

1. hdoop@hadoop-master:~$ ls

**Output:**

apache-hive-3.1.2-bin Music pig\_1705056244118.log

Desktop -output pig\_1705056895272.log

Documents Pictures pigscript.pig

Downloads pig-0.17.0 Public

hadoop pig\_1705050213253.log ratings.data

hbase-2.5.7 pig\_1705051969113.log -reducer

-input pig\_1705052008135.log snap

input.txt pig\_1705053705405.log spark-3.5.0-bin-hadoop3

kafka pig\_1705054104077.log Templates

kafka\_2.13-3.6.1 pig\_1705054134641.log tmp

-mapper pig\_1705054708734.log Videos

1. movies.item pig\_1705056164092.log wordcount.pig
2. hdoop@hadoop-master:~$ hdfs dfs -copyFromLocal /home/hdoop/movies.item/pig1/movies.items
3. hdoop@hadoop-master:~$ hdfs dfs -copyFromLocal /home/hdoop/ratings.data /pig1/ratings.data
4. hdoop@hadoop-master:~$ pig

**Output:**

2024-01-13 14:24:55,001 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL

2024-01-13 14:24:55,002 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE

2024-01-13 14:24:55,002 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType

2024-01-13 14:24:55,031 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58

2024-01-13 14:24:55,031 [main] INFO org.apache.pig.Main - Logging error messages to: /home/hdoop/pig\_1705136095027.log

2024-01-13 14:24:55,043 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/hdoop/.pigbootup not found

2024-01-13 14:24:55,225 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address

2024-01-13 14:24:55,225 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://192.168.159.101:9000

2024-01-13 14:24:55,629 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-6c9e62df-52ed-4ba9-9910-7deab5ba8792

2024-01-13 14:24:55,629 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false

1. grunt> pig movie.pig

**Output:**

2024-01-13 14:25:17,199 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 1000: Error during parsing. Encountered " <IDENTIFIER> "pig "" at line 1, column 1.

Was expecting one of:

<EOF>

"cat" ...

"clear" ...

"fs" ...

"sh" ...

"cd" ...

"cp" ...

"copyFromLocal" ...

"copyToLocal" ...

"dump" ...

"\\d" ...

"describe" ...

"\\de" ...

"aliases" ...

"explain" ...

"\\e" ...

"help" ...

"history" ...

"kill" ...

"ls" ...

"mv" ...

"mkdir" ...

"pwd" ...

"quit" ...

"\\q" ...

"register" ...

"rm" ...

"rmf" ...

"set" ...

"illustrate" ...

"\\i" ...

"run" ...

"exec" ...

"%default" ...

"%declare" ...

"scriptDone" ...

"" ...

"" ...

<EOL> ...

";" ...

Details at logfile: /home/hdoop/pig\_1705136095027.log

1. grunt> run movies.pig