Cloud Computing Capstone Task 1 Report

Roney Duílio Stein

Video: https://youtu.be/ex9K63YRLjc

Source code: https://github.com/roneystein/aviation-ontime-pig-processing

General Considerations

- The datasets used to answer the questions were:
  - Airline Origin and Destination Survey: used only for question 1.1;
  - Airline On-Time Performance Data: remaining questions.
- On-time performance is defined as the percentage of flights arrived with 15 or more minutes of delay.
- Questions selected: 1.1, 1.3, 2.2, 2.3, 2.4, 3.1, 3.2.
- Video doesn't cover loading all the scenarios because of the time constraint of 5 minutes.

Data extraction and cleaning

Used a shell script to verify the archive's ZIP files integrity, the existence and content of CSV files inside each. Two ZIP files had no valid content. Extraction was automated using two shell scripts that, when run in the directory containing the data files/directories, processed the content using pipes, not temporary files, as follows:

1. Extracted the CSV files from ZIP files;
2. Processed through the Perl script (one for Origin Destination and another for On-time tables) filtering the desired data and only required columns;
3. Compressed using BZip2;
4. Piped to proper HDFS location.

Origin-Destination input files were generated with the format:
Origin Destination
On-time input files were generated with the format:
Date Weed_day CarrierID Carrier Flight Origin Destination Departure_time Departure_delayed_in_minutes Arrival_time Arrival_delayed_flag Arrival_delayed_minutes

System construction and integration

The Hadoop cluster was setup using 5 EC2 c4.xlarge nodes. Cassandra was initially built on a t1.micro instance and later moved to the same Hadoop cluster nodes. Pig was configured to store data directly into Cassandra and it read the BZip2 files as the input. Each job was submitted using a command line, and one command line only. Additional commands where necessary to query the output as shown in the video. Intermediary map files were compressed with LZO.
Overall, the questions were processed **between 3 and 4 minutes each** when using the 5 node cluster with EC2 c4.2xlarge instances. Question 3.2 was processed using c4.xlarge instances and, in this case, it was a lengthy process with low resources consumptions**, taking 2h**, mostly streaming the results to the Cassandra cluster, which was running spread in the 4 data nodes. Maybe a case of setting database indexes.

Optimizations

Some optimizations made in the process or system:

- No intermediary or temporary data files were created during the cleaning/pre-processing phase and the data was read compressed from the Zip files and stored compressed in HDFS in BZip2 files, which are splittable.
- Some Pig/Hadoop parameters were tuned to increase the parallelism of map computation.
- Hadoop and Pig were configured to use LZO compression for temporary map files, as they are fast to compress/decompress and could lead to a better I/O and network transfer performance increase. Could have used Snappy as well.
- For the question 3.2 the four Hadoop data nodes were used to build a four nodes Cassandra cluster, using the idle compute power when the mapreduce was ending and Cassandra was asking for power.
- The outputs from the Pig jobs were transferred directly to Cassandra.

Algorithms

The algorithms were coded using Pig version 0.15.

Question 1.1: A word counting.

1. Load all origin-destination input files in the format;
2. For each line tokenize words (separate origin from destination);
3. Group by airport and count the grouped lines;
4. Sort by number of lines (arrivals and departures) in descending order;
5. Crop the top 10;
6. Output count (total of arrivals and departures) and airport.

Question 1.3: Calculate average value on a group of words.

1. Load all on-time input files;
2. Keep only the day of week and arrival delay flag (set when >15 min. delayed) fields;
3. Group by day of week;
4. For each day of week calculate the percentage of delayed arrivals;
5. Sort by the percentage in ascending order;
6. Output day of weed and percentage of delayed arrivals.

Question 2.2: Calculate average for a group then reorder and crop the results.

1. Load all on-time input files;
2. Keep origin, destination and departure delayed flag (set when >15 min. delayed) fields;
3. Group all records by origin-destination tuples;
4. For each origin-destination group calculate the percentage of delayed departures and generate: origin, destination and percentage;
5. Group by origin;
6. For each origin:
    a. Order by percentage of delayed departures in ascending order;
    b. Crop the top 10;
7. Generates: origin and list of destinations, comma-separated, in order of least delayed (left) to most delayed (right).

Question 2.3: Calculate average for a group, sort the carriers and list the top 10 as a comma separated list.

1. Load all on-time input files;
2. Keep origin, destination, carrier and arrival delayed flag (set when >15 min. delayed) fields;
3. Group by tuples of origin-destination-carrier;
4. For each tuple calculate the percentage of delayed arrivals (arrival performance);
5. Group by origin-destination pairs;
6. For each pair:
    a. Order by percentage of delayed arrivals in ascending order (or arrival performance in descending order);
    b. Crop the top 10;
7. Generates: origin, destinations and top 10 least delayed carriers, in a comma-separated list, in order of least delayed (left) to most delayed (right).

Question 2.4: Calculate average value for a group.

1. Load all on-time input files;
2. Keep origin, destination and arrival delayed minutes (0 or positive numbers);
3. Group all records by origin-destination pairs;
4. For each pair calculate the average of delayed minutes;
5. Outputs origin, destination and the calculated average.

Question 3.2: Split the data into two different groups of records; calculate a time difference and merge then based on the time difference and origin/destination.

1. Load all year 2008 on-time input files;
2. For each line keep/generate: date, "carrier + flight number", origin, destination, departure time, arrival delay minutes (>0), date +2 days
3. Filter the lines with departure time < 12:00h to a group "X", the first leg;
4. Filter the lines with departure time > 12:00h to a group "Y", the second leg;
5. For X and Y, each: for each date – origin – destination we keep only the most on-time flight;
6. Merge X and Y generating all the possible combinations for each date;
7. Output: X departure date, X departure time, X flight, X origin, X destination or intermediate airport, Y destination of final destination, Y departure date, Y departure time, Y flight, total delay in minutes.

Results

Question 1.1: Top 10 most popular airports.

Popularity	Airport
58187766	ATL
49596357	ORD
44373231	DFW
31219499	DEN
25452018	LAX
24668293	MSP
24276984	CLT
22927673	DTW
21864094	PHX
21501371	IAH

Question 1.3: Rank of the days of weed by on-time arrival performance.

Day of Week	% of delayed arrivals
Saturday	17.170948184159514
Tuesday	18.837742475105646
Monday	19.60122803398963
Sunday	19.787507762028206
Wednesday	20.43301101641691
Thursday	22.98366271134333
Friday	23.879051559476984

Question 2.2: For each X airport rank the top 10 airports in decreasing order of on-time departure performance from X.

Origin	Least Delayed (left to right)
CMI	PIT, DAY, STL, PIA, DFW, CVG, ATL, ORD
BWI	MLB, IAD, DAB, CHO, UCA, SRQ, SJU, OAJ, BGM, GSP
MIA	BUF, SAN, HOU, ISP, PSE, SLC, TLH, MEM, GNV, SJC
LAX	SDF, LAX, BZN, PMD, MAF, VIS, MEM, IYK, HDN, SNA
IAH	MSN, MLI, HOU, AGS, EFD, JAC, RNO, MDW, VCT, CLL
SFO	SCK, MSO, SDF, FAR, LGA, PIE, BNA, OAK, MKE, MEM

Question 2.3: For each source-destination rank the top 10 carriers in decreasing order of on-time arrival performance.

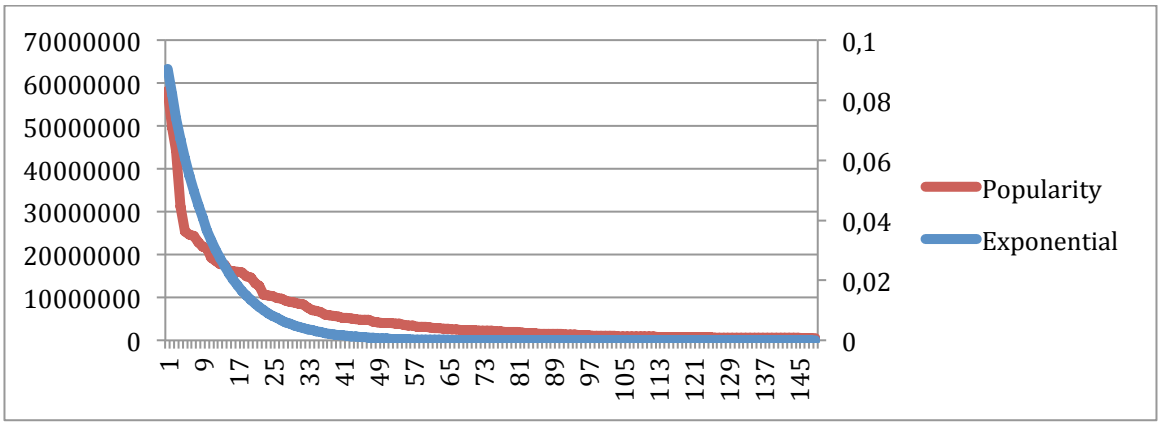
Origin	Destination	Least Delayed Carriers (left to right)
CMI	ORD	MQ
IND	CMH	AA, CO, HP, US, NW, DL, EA
DFW	IAH	CO, OO, UA, XE, EV, DL, AA, MQ

LAX	SFO	TZ, F9, PS, EV, AA, US, MQ, CO, WN, UA
JFK	LAX	UA, AA, HP, DL, TW
ATL	PHX	FL, US, HP, EA, DL

Question 2.4: For each source-destination determine the mean arrival delay in minutes.

Origin	Destination	Mean Delay (minutes)
CMI	ORD	15
IND	CMH	6
DFW	IAH	11
LAX	SFO	13
JFK	LAX	14
ATL	PHX	13

Question 3.1: Does the popularity distribution of airports follow a Zipf distribution? If not, what distribution does it follow?
The popularity is **not** a Zipf distribution as it doesn't make a straight line in a logarithmic chart. Without extensive research we could conclude that **it follows an exponential distribution** which is both drawn below for comparison using a linear chart using Lambda=0,1.



Question 3.2:

1st leg departure date	Origin Airp.	Interm. Airport	Dest. Airport	1st leg flight	1st leg dep. time	2nd leg flight	2nd leg dep. date	2nd leg dep. time
04/03/2008	CMI	ORD	LAX	MQ 4401	0810	AA 607	06/03/2008	1950
09/09/2008	JAX	DFW	CRP	AA 845	0725	MQ 3627	11/09/2008	1645
01/04/2008	SLC	BFL	LAX	OO 3755	1100	OO 5429	03/04/2008	1455
12/07/2008	LAX	SFO	PHX	UA 889	0759	WN 2645	14/07/2008	2030
10/06/2008	DFW	ORD	DFW	AA 2332	0915	AA 2333	12/06/2008	1515
01/01/2008	LAX	ORD	JFK	UA 944	0705	B6 918	03/01/2008	1900