Flight Dataset

Some insights on the U.S. Airline data using Apache Pig

Understanding Data

The U.S. Department of Transportation's (DOT) Bureau of Transportation Statistics (BTS) tracks the on-time performance of domestic flights operated by large air carriers. Summary information on the number of on-time, delayed, cancelled, and diverted flights appears in DOT's monthly Air Travel Consumer Report, published about 30 days after the month's end

The data format is comma separated values These are 2 different datasets, i.e., Flight_details.csv and Airports.csv. Let us understand one at a time.

1. Flight_details.csv
There are 29 columns in flight_details.csv dataset. Please check flight_description file for details about schema/fields.

2. Airports.csv
- iata: the international airport abbreviation code
- name of the airport
- city and country in which airport is located.
- lat and long: the latitude and longitude of the airport

Problem Statement 1

- **Find out the top 5 most visited destinations.**

1) LOAD the flight data set.

```
grunt> load_flightdata = LOAD
'file:///home/cloudera/khasimbabu/PIG_Excercise/CaseStudy3/flights_details.csv' USING
PigStorage(',');
```

2) Extract the selective columns data

```
grunt> flight_details = FOREACH load_flightdata GENERATE (int)$1 as year, (int)$10 as flightnu,
(chararray)$17 as origin, (chararray)$18 as dest;
grunt> describe flight_details;
```

```
flight_details: {year: int,flightnu: int,origin: chararray,dest: chararray}
```

3) Extract Destination not null data

```
grunt> flight_dest_notnull = FILTER flight_details by dest is not null;
```

4) Group by destination data

```
grunt> grp_flight_dest = GROUP flight_dest_notnull by dest;
grunt> describe grp_flight_dest;
```

```
grp_flight_dest: {group: chararray,flight_dest_notnull: {(year: int,flightnu: int,origin: chararray,dest:
chararray)}}
```

5) Count the flights by destination and arrange them in desending order.
grunt> grp_flightcount_dest = FOREACH grp_flight_dest GENERATE group,
COUNT(flight_dest_notnull.flightnu);
grunt> order_flightcount_dest = ORDER grp_flightcount_dest by $1 DESC;

6) limit – Top 5 records
grunt> limit_flightcount_dest = LIMIT order_flightcount_dest 5;

7) LOAD Airport data
grunt> load_airportdata = LOAD
'file:///home/cloudera/khasimbabu/PIG_Excercise/CaseStudy3/airports.csv' USING PigStorage(',');

8) Airport selective columns
grunt> airport_details = FOREACH load_airportdata GENERATE (chararray)$0 as dest,(chararray)$2 as city, (chararray)$4 as country;

grunt> describe airport_details;

airport_details: {dest: chararray,city: chararray,country: chararray}

9) Join flight, airport
grunt> join_flight_airport = JOIN limit_flightcount_dest by $0, airport_details by dest;

grunt> describe join_flight_airport;

join_flight_airport: {limt_flightcount_dest::group: chararray,long,airport_details::dest: chararray,airport_details::city: chararray,airport_details::country: chararray}

grunt> dump join_flight_airport;

Problem Statement 2

- **Which month has seen the most number of cancellations due to bad weather?**

1) Flight cancel details
grunt> flight_cancel_details = FOREACH load_flightdata GENERATE (int)$2 as month, (int)$10 as flightnum, (chararray)$22 as cancelled, (chararray)$23 as cancelcode;

2) List the cancelled flights due to bad weather
grunt> flight_cancelfilter = FILTER flight_cancel_details by cancelled=='1' AND cancelcode=='B';

grunt> describe flight_cancelfilter;

flight_cancelfilter: {month: int,flightnum: int,cancelled: chararray,cancelcode: chararray}

3) Group the cancelled flights by month
grunt> grp_cancelfilter = GROUP flight_cancelfilter by month;

grunt> describe grp_cancelfilter;

grp_cancelfilter: {group: int,flight_cancelfilter: {(month: int,flightnum: int,cancelled: chararray,cancelcode: chararray)}}

4) Count, Order and LIMIT the cancelled flights
grunt> count_cancelflight = FOREACH grp_cancelfilter GENERATE group,
COUNT(flight_cancelfilter.cancelcode);

grunt> order_cancelflight = ORDER count_cancelflight by $1 DESC;

grunt> limit_cancelflight = LIMIT order_cancelflight 1;
grunt> dump limit_cancelflight;

Problem Statement 3

- **Top ten origins with the highest AVG departure delay**

1) flight_origindata
grunt> flight_origindata = FOREACH load_flightdata GENERATE (int)$16 as dep_delay,(chararray)$17 as origin;

2) flight orgin not null
grunt> grp_flightbyorigin_notnull = FILTER flight_origindata by dep_delay is not null AND origin is not null;

3) group by origin data
grunt> grp_flightbyorigin = GROUP grp_flightbyorigin_notnull by origin;
grunt> describe grp_flightbyorigin;

grp_flightbyorigin: {group: chararray,grp_flightbyorigin_notnull: {(dep_delay: int,origin: chararray)}}

4) AVG of depdelay
grunt> flight_depdelay_avg = foreach grp_flightbyorigin generate group,
AVG(grp_flightbyorigin_notnull. dep_delay);

5) Order Origin data
grunt> flight_depdelay_order = ORDER flight_depdelay_avg by $1 DESC;

grunt> flight_depdelay_limit = LIMIT flight_depdelay_order 10;

6) Load Airport data
grunt> load_airportdata = LOAD
'file:///home/cloudera/khasimbabu/PIG_Excercise/CaseStudy3/airports.csv' USING PigStorage(',');

7) Airport origin details
grunt> airport_origindata = foreach load_airportdata generate (chararray)$0 as origin, (chararray)$2 as city, (chararray)$4 as country;

grunt> describe airport_origindata;

airport_origindata: {origin: chararray,city: chararray,country: chararray}

8) join_flight_airportorigin
grunt> join_flight_airportorigin = JOIN flight_depdelay_limit by $0 , airport_origindata by origin;

grunt> describe join_flight_airportorigin;

join_flight_airportorigin: {flight_depdelay_limit::group: chararray,{(dep_delay: int)},airport_origindata::origin: chararray,airport_origindata::city: chararray,airport_origindata::country: chararray}

9) Order and Limit the origin
grunt> flight_origin_final = FOREACH join_flight_airportorigin GENERATE $0,$1,$2,$4;

grunt> order_flight_originfinal = ORDER flight_origin_final by $3 DESC;

grunt> dump order_flight_originfinal;

Problem Statement 4

- **Which route (origin & destination) has seen the maximum diversion?**

1) flight_diversiondata
grunt> flight_diversiondata = FOREACH load_flightdata GENERATE (chararray)$17 as origin,(chararray)$18 as dest,(int)$24 as diversion;

2) flight_diversion_notnull
grunt> flight_diversion_notnull = FILTER flight_diversiondata by (origin is not null) AND (dest is not null) AND (diversion==1);

3) grp_flight_diversion
grunt> grp_flight_diversion = GROUP flight_diversion_notnull by (origin,dest);

grunt> describe grp_flight_diversion;

grp_flight_diversion: {group: (origin: chararray,dest: chararray),flight_diversion_notnull: {(origin: chararray,dest: chararray,diversion: int)}}

4) Count, Order, Limit flight diversion
grunt> count_flight_diversion = FOREACH grp_flight_diversion GENERATE group,COUNT(flight_diversion_notnull.diversion);

grunt> order_flight_diversion = ORDER count_flight_diversion by $1 DESC;

grunt> describe count_flight_diversion;

count_flight_diversion: {group: (origin: chararray,dest: chararray),long}

grunt> limit_flight_diversion = LIMIT order_flight_diversion 10;

grunt> dump limit_flight_diversion;