World's Best Cube Country

Pablo Aliaga
Eric Jonsson
Erick Lemus

June 21, 2019

1 Goal

The goal of the project is to find out which country in the world that is the best at solving the Rubik's Cube. To achieve this goal, it is first of all very important to answer the question: what does it mean to be the "best"? To answer this question various other questions came up along the way. Which country has the best average time? which country has the most speedsolvers (person solving cubes at a competition)? And how can these two factors, along with others, finally decide which country that is the best one?

2 Data

We used data from the World Cube Association, the official worldwide organisation that regulates and holds competitions for a lot of puzzles, where cubes are the most popular ones. On their website a lot of well-updated data can be found and downloaded. With the power and influence that WCA has as the official organization we found it to be appropriate as source of data.

The data we used were the following TSV files:

- *WCA_export_Results.tsv* contains all the results from all competitions since WCA started to store data. It has the size of *205 MB* and consists of approximately *2 150 000* lines.
- *WCA_export_Persons.tsv* contains all the speedcubers who ever competed in a WCA-held competition. It has the size of *5 MB* and consists of approximately *125 000* lines.
- *WCA_export_Countries.tsv* is a small file containing all the countries ever represented in competition. It contains *207 lines* consisting of countries and ways of expressing multiple countries (used in for example team competitions).

3 Methods

We used Apache Pig as our tool for the project. Given our circumstances - files of no extreme size and with some a bit more complicated queries, we found that Hadoop is sufficiently rapid and that Pig as platform and Pig Latin as language was a good choice to go with.

We decided to only use the results of the event 3x3x3, commonly known as the standard Rubik's Cube. In 3x3x3 competitions, in every round the competitors solves the cube five times. Out of these five times, the fastest and the slowest times are eliminated and an average result is taken out of the other three results. This result is simply called *average*. We wanted the best *average* from every speedsolver. In order to do this we needed to:

- *FILTER* the results file to get only results from 3x3x3 events.
- *GROUP BY personId* which is a personal id that represents a speedsolver.
- *GENERATE* a new table with the *person* and their *MIN* (fastest) *average*.

Having a table with all the speedsolvers and their best average result we then needed to connect these to their respective country. Given that we have one file with the results, one with persons and one with countries the following we needed to do was therefore to *JOIN* them.
With the countries and results now connected, we could then try to evaluate how good every country is in terms of speedcubing. First we simply looked into how many active competitors that every country has and an average of all of their results using:

```
average_per_country = FOREACH results_by_country GENERATE group AS
    countryId, AVG(results_person.time) AS time_country,
    COUNT(results_person) AS nPerson;
```

However, we thought that a ranking cannot simply be built on neither active cubers (unfair considering differences in populations) or a country's average (what if a country has only one (but quite decent) competitor?). After various attempts on trying to find a appropriate formula we at last found the following:

$$Score(Country) = \frac{AC + AA * \log_{10}(\frac{CA}{CC}) * \log_{10}(\frac{AC}{AA} + 1)}{2} \quad (1)$$

where AC = Average time of the Country, AA = Average time of All countries, CC = amount of Cubers of the Country, CA = amount of Cubers in All countries.

It it created to give a bonus to the countries with more competitors and to those with a good average compared to the total average of all competitors. The logarithms are used to keep the bonuses reasonable, i.e. to not let them affect all to much. The lower the score, the better.

4 Results

Applying our different methods we found various answers to our question. Looking at the countries with most speedcubers the got the following top 5:

Country	Amount
United States	23 074
China	18 402
India	10 671
Russia	4 939
Brazil	4 581

If we're instead looking at the fastest average these five following countries were the best:

Country	Time (in seconds)
Haiti	10.69
Brunei	12.12
Guyana	14.38
Togo	15.29
Nicaragua	16.71

Finally using our formula (where a low score is better):

Country	Score
United States	4035
China	4226
Philippines	4974
Vietnam	5268
Indonesia	5546
Malaysia	5979
Haiti	6025

To the last result we added to more lines just to show that, even though Haiti has very few competitors (in fact just one!), they still made it to the top list due to their very good average.

5 Conclusion

We learned that, considering the scale of our project, the use of a distributed system like *HDFS* was very important. Using RDMS could possibly work, but would probably be extremely slow and use a lot of machine's resources.

The use of Pig Latin to write our queries and to run them on the HDFS was quite easy. Even though Pig is a lot slower than, for example SPARK or

HADOOP, the simplicity of Pig Latin made the programmer's work a lot more pleasant.

To conclude, we still don't know what country is best at solving the Rubik's Cube. Finding the best out of something is quite a dilemma, it's not like a math problem that has only one solution, this subject required a lot of our own vision about who's better and who's not at solving cubes. Anyways we found a lot of ways to rank countries, some of them made a lot of sense to us and some of them not.

Appendix

All the files are uploaded to GitHub.

But here's the code of the last Pig code to generate our final result:

```
raw_results = LOAD
    'hdfs://cm:9000/uhadoop2019/elsueco/WCA_export_Results.tsv' USING
    PigStorage('\t') AS (competitionId, eventId, roundTypeId, pos,
    best, average, personName, personId, personCountryId, formatId,
    value1, value2, value3, value4, value5, regionalSingleRecord,
    regionalAverageRecord);
-- Later you can change the above file to
    'hdfs://cm:9000/uhadoop/shared/imdb/imdb-stars.tsv' to see the full
    output

raw_persons = LOAD
    'hdfs://cm:9000/uhadoop2019/elsueco/WCA_export_Persons.tsv' USING
    PigStorage('\t') AS (id, subid, name, countryId, gender);
-- Later you can change the above file to
    'hdfs://cm:9000/uhadoop/shared/imdb/imdb-ratings.tsv' to see the
    full output

raw_countries = LOAD
    'hdfs://cm:9000/uhadoop2019/elsueco/WCA_export_Countries.tsv' USING
    PigStorage('\t') AS (id, name, continentId, iso2);
-- Later you can change the above file to
    'hdfs://cm:9000/uhadoop/shared/imdb/imdb-ratings.tsv' to see the
    full output

-- Now to implement the script

--De WCA_export_Results.tsv

--filter eventId = 333
results_333 = FILTER raw_results BY eventId == '333';

results_cut = FOREACH results_333 GENERATE personId, average;

--groupBy personID
results_by_person = GROUP results_cut BY personId;

--Select min(average) para cada persona
results_grouped = FOREACH results_by_person GENERATE group AS person,
    MIN(results_cut.average) AS time;

filtered_grouped = FILTER results_grouped BY time > 0;

group_results = GROUP filtered_grouped all;
```

```
total_people = FOREACH group_results GENERATE COUNT(filtered_grouped) as
    totalPeople;

total_average = FOREACH group_results GENERATE
    AVG(filtered_grouped.time) as totalAverage;

--Generar tabla personID, min(average)

persons_cut = FOREACH raw_persons GENERATE id, name as personName,
    countryId, gender;

--join con WCA_export_Persons.tsv con personID = id
results_person = JOIN filtered_grouped BY person, persons_cut BY id;

--*** filter by gender

--groupBy countryId
results_by_country = GROUP results_person BY countryId;

--Generar tabla countryId, avg(average)
average_per_country = FOREACH results_by_country GENERATE group AS
    countryId, AVG(results_person.time) AS time_country,
    COUNT(results_person) AS nPerson;

average_per_country_2 = FILTER average_per_country BY nPerson > 0 AND
    nPerson IS NOT NULL;

--Join con WCA_export_Countries.tsv con id = countryId
results_country = JOIN average_per_country_2 BY countryId, raw_countries
    BY id;

--Generar tabla name(country), avg
prefinal_results = CROSS results_country, total_people;
prefinal_results2 = CROSS prefinal_results, total_average;

final_results = FOREACH prefinal_results2 GENERATE name, time_country,
    nPerson , (time_country +
    totalAverage*LOG(totalPeople/nPerson)*LOG(time_country/totalAverage
    + 1))/2 as pablo_score;

filtered_results = FILTER final_results BY time_country > 0;


--orderby average
results_ordered = ORDER filtered_results BY pablo_score ASC;

STORE results_ordered INTO '/uhadoop2019/elsueco/results_project13/';
```