Proposal Solution

1. Introduction

As I see it, it’s quite clear that there is a need to differentiate between daily calculations and query calculations. While a daily calculation should occur every day, query calculation can be on-demand with specific parameters. Moreover, query calculation depends on daily calculations.

Having said that, my solution provides two flavors: one for daily calculations, other for query calculations.

1. Architecture

2.1 Properties:

1. Spark: 2.4.4
2. Scala: 2.12.10
3. Java: 1.8
   1. Assumptions:
4. Input: as described in the assignment, while in every document there are possibly some JSONs, each one in a different line
5. Output­: Avro Parquet format
6. Every field is potentially nullable
   1. Daily Calculation:

I will suggest two flavors (daily & on-demand), and I think the first one is more suitable for solving the given problem:

1. Daily:

Calculating the day before statistics in a time of a specific time zone that guarantees having yesterday data from all time zones.

After calculating the day before results, summing up all results from all days according to the requirement.

1. On-Demand: only when needed we will calculate the day before results.
   1. Observations: since we calculate every day results we will be able to calculate 365 days ago results since we store stats results from every day and in order to answer the specification, we only need to sum up these results. In addition, if we have results of last 30 days, we don’t need to calculate them again!
2. Code Example:

the code illustrated the calculations in general, the is a docker for S3 to test the main function, and there is a unit test for it.