Project 2

Corso di Sistemi e Architetture per Big Data

A.A. 2021/22

Valeria Cardellini, Matteo Nardelli

Laurea Magistrale in Ingegneria Informatica

Project delivery

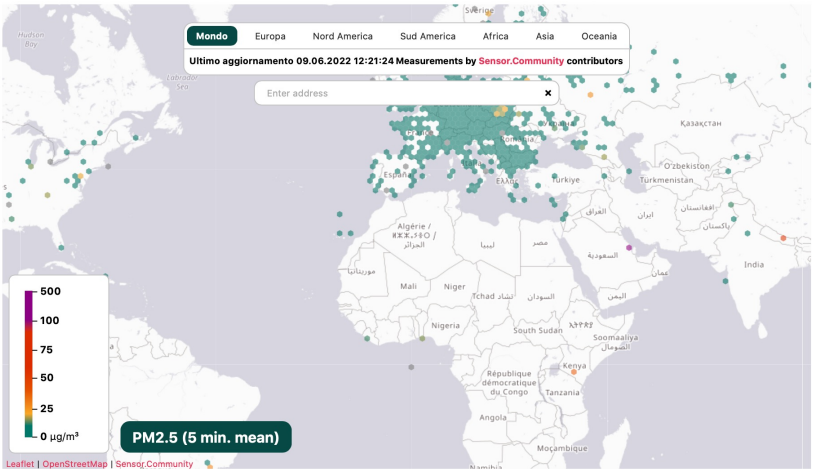
- Submission deadline
  - July 11, 2022
  - After the deadline, the maximum achievable score will be decreased by 1 point for each week of delay
- Your presentation
  - July 14, 2022 (to be confirmed)
- What to deliver
  - Link to cloud storage or repository containing the project code
  - Project report composed by 3-6 pages in ACM or IEEE proceedings format (only the report: by July 12)
  - Presentation slides (max. **15 minutes** per group), to be delivered after your presentation
- Team
  - Target: 2 students per team
  - Also possible 1 student or 3 students per team

Dataset

- You will use a real dataset related to temperature and pressure sensor streaming data
  - Goal: real-time analytics over streaming sensor data
  - Dataset, related to May 2022, is available in CSV format from https://archive.sensor.community/csv_per_month/2022-05/2022-05_bmp180.zip

Dataset

- Collection of sensor data from Sensors.Community
  - A contributors-driven global sensor network that creates Open Environmental Data
  - More than 13K active sensors in 73 countries



Dataset

- Tuple fields
  - `sensor_id`: sensor ID
  - `sensor_type`: sensor type, in our case BPM180 (pressure, altitude, temperature)
  - `location`: location ID
  - `lat`: latitude of the sensor location
  - `lon`: longitude of the sensor location
  - `timestamp`: timestamp of measurement (format: YYYY-MM-DDTHH:MM:SS)
  - `pressure`: pressure value (in Pa)
  - `altitude`: altitude value (in m)
  - `pressure_sealevel`: pressure at sea level
  - `temperature`: temperature (in ˚C)

Queries with Flink/Storm

- Use Apache Flink or alternatively Apache Storm
- Simulate the tuples arrival using the timestamps within the dataset
  - You can speed-up the time
- Include in your report/slides the queries' latency time and throughput on your reference architecture

Queries with Flink/Storm: Q1

1. For those sensors having `sensor_id` < 10000, find the number of measurements and the temperature average value

- Q1 output:
  `ts, sensor_id, count, avg_temperature`

- Using a tumbling window, calculate this query:
  - every 1 hour (event time)
  - every 1 week (event time)
  - from the beginning of the dataset

Queries with Flink/Storm: Q2

2. Find the real-time top-5 ranking of locations (location) having the highest average temperature and the top-5 ranking of locations (location) having the lowest average temperature

- Q2 output:
  `ts, location1, avg_temp1, ... location5, avg_temp5, location6, avg_temp6, ... location10, avg_temp10`

- Using a tumbling window, calculate this query:
  - every 1 hour (event time)
  - every 1 day (event time)
  - every 1 week (event time)

Queries with Flink/Storm: Q3

3. Consider the latitude and longitude coordinates within the geographic area which is identified from the latitude and longitude coordinates (38˚, 2˚) and (58˚, 30˚).

- Divide this area using a 4x4 grid and identify each grid cell from the top-left to bottom-right corners using the name "cell_X", where X is the cell id from 0 to 15. For each cell, find the average and the median temperature, taking into account the values emitted from the sensors which are located inside that cell
- Q3 output:
  `ts, cell_0, avg_temp0, med_temp0, ... cell_15, avg_temp15, med_temp15`

Queries with Flink/Storm: Q3

- Using a tumbling window, calculate this query:
  - every 1 hour (event time)
  - every 1 day (event time)
  - every 1 week (event time)

Optional part

- Use Kafka Streams or Spark Streaming to answer one query of your choice

Queries for the team

- 1 student in the team: queries 1 and 2
- 2 students in the team: all the three queries
- 3 students in the team: all the three queries plus queries 1 and 2 using Kafka Streams or Spark Streaming