Analyzing Air Quality Improvement During COVID-19

CS591K1 Midterm Demo

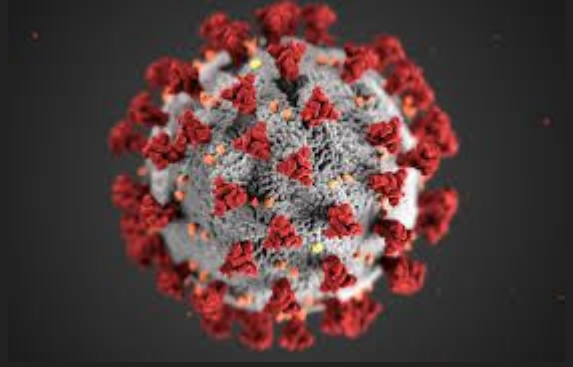
Group 8: Andy Lyu, Mina Morcos, Snigdha Kalathur

Project Overview

The outbreak of COVID-19 in 2020 has created numerous social, economic, and environmental repercussions. As governments announced lockdowns to minimize the spread of the virus, many businesses closed their doors and regular commuters stayed at home, resulting in a reduction in the amount of air pollution caused by traffic and business operations.

The DEBS Grand Challenge is a series of competitions in which participants compete with the goal of building faster and more scalable distributed and event-based systems to solve practical problems. This year, the the goal of the DEBS Grand Challenge is to detect geographical areas where the air quality has changed significantly due to lockdown measures during the first wave of covid infections.

This data analysis will be important for climate groups and governments to detect which countries are most impacted by traffic pollution which will also be helpful for mitigating the inevitable rise of air pollution once the COVID-19 pandemic subsides and lockdowns end.



Project Overview

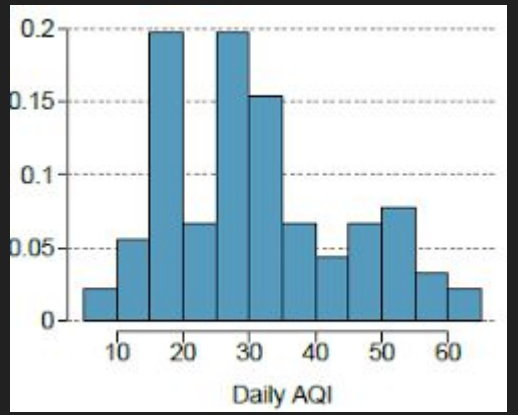
Our goal is the implement the following two queries accurately and with low latency. We hope to to obtain a high ranking on the leaderboard for the DEBS Grand Challenge based on correctness of results and processing performance.

Query 1

The first query returns the top 50 cities in terms of air quality improvement as well as their current air quality indices. The AQI for each city will be calculated from the average particles coming from geo-distributed air quality sensors over a sliding window of 24 hours. This query will result in a ranking of the top 50 cities by their improvement over a 5-Day Average AQI compared to the previous year.

Query 2

The second query results in a histogram of the longest streaks of good air quality for the last 7 days, defined as the time span in seconds since a city flipped to a “good” AQI value. The histogram will have 14 buckets of equal length from 0 to the maximum length. Both query 1 and query 2 will run in parallel.



Status Update - Completed and Current Progress

Completed

- Implemented server-client handshake protocol required by the competition
  - Passed test benchmarks with latency adjustments
  - Able to read challenge data including city location coordinates, AQI measurements and their corresponding locations.
- Converted protobuf files to java stubs and reimplemented provided python logic
- Setup project workflow with Apache Maven and webhooks
- Integration of source layers into the Flink application

Work in Progress

- Integrating the API communicator to a Flink source
- Designing operators to produce correct results (passing a test run)

Status Update - Division of Work

Snigdha

- Operator design
  - In charge of fully understanding the required queries and designing their implementation in Flink
- Operator Implementation
  - WIll work with Mina to implement operators

Baiqing

- Source Layer
  - Completed Flink source layer implementation and protobuf conversion
- Project Leader
  - Overseeing all parts of the project

Mina

- Integration
  - Integration of source layers into the Flink application
- Operator Implementation
  - Will work with Snigdha to implement operators

Demo

Live demo of working components

Query 1 Implementation Design (Work in Progress)

keyBy to partition stream by location

Define a sliding window over 24 hours using *SlidingEventTimeWindows*

Get snapshots taken every 5 minutes into a separate window using *TumblingEventTimeWindows*

At each snapshot, compute current and historic AQI averages with snapshot - 5 days and snapshot - 365 days - 5 days

Once entire batch is processed, summarize results and rank cities by the 5 day average.

```
message TopKCities {
    int32 position = 1;
    string city = 2;
    int32 averageAQIImprovement = 3;
    int32 currentAQIP1 = 5;
    int32 currentAQIP2 = 6;
}
```

Challenges

- Understanding gRPC, using it within the project
  - Figuring out what each data message entry means within the larger program
- Generating correct stubs and performing proper handshakes
- Comprehending challenge goals, how the different queries should be calculated
- Analyzing the data that the server gives us

Immediate Future Work

1. Finishing operator designs
    a. Effort: Reading through documentation to fully understand how data is coming in and designing operators in Flink
    b. Time: approx. 4 hours
2. First attempt at implementation of operators
    a. Effort: Snigdha and Mina will meet to successfully filter and analyze data in the operators (without focusing on accuracy or efficiency at first)
    b. Time: approx. 4 hours
3. Implementation of operators to the point of successful passing of tests
    a. Effort: Snigdha and Mina will meet to continue implementing the operators (focusing on accuracy, efficiency, and passing tests)
    b. Time: approx: 10 hours
4. Improving accuracy and minimizing latency
    a. Effort: Playing around with VM, batch sizes, and other parameters
    b. Time: approx 4 hours