Hot Tub - DIC Project Report

Kai Fleischman & Martin Schuck

October 25, 2020

1 Introduction

Tackling crisis of global scale is an huge problem in today's societies. It seems that humans have a hard time at grasping phenomena which exceed their regular event horizon. This was particularly evident during the spread of the recent Corona pandemic. However, while the Corona crisis appears to be under control, an even greater thread looms in the coming years. Global warming has been one of the gravest and yet most abstract dangers to human society in recent decades. Since graphic visualizations and daily updates proved to be most effective in getting to the public during the pandemic, we proposed a real time global warming monitor as a warning. The project aimed to create a world map which shows the temperature difference to the common global reference in real time.

2 Methodology

The central idea of the project was to create a real time temperature difference map. This map consists of two components: a reference model from pre-industrial times, and a live model of the current temperatures around the globe.

For the historical data, we chose to use data from the Berkeley Earth dataset [1]. It contains not only historical data, but also a reference temperature for each longitude and latitude in each dataset. Our final reference model is synthesized with these samples and bicubic interpolation for higher resolution.

In order to generate a live model, we employ the *OpenWeatherMap* API [2]. A set of locations is constantly updated and a record of the last 24 hours kept in a Cassandra database. From these data points we again interpolate the global temperature map. Since live data is sparse compared to the reference model samples, nearest neighbour interpolation and gaussian filtering is used on the global map instead of bicubic interpolation.

With these two models, our webserver is able to constantly display the most recent estimated temperature difference map.

3 Architecture

The project architecture adopts the partitioning induced by our two distinct maps. For the reference map we use a *model_creator* script which extracts relevant data points from the Berkeley Earth dataset and saves them into a separate file. This routine only has to be executed once. A *reference_model_loader* class then takes care of interpolation and provides a live interface to the specific model for each day.

In order to make the streaming part of our pipeline expandable to more weather stations, we use Kafka with Zookeeper as coordinator to pass messages from our API request workers. Each worker publishes to a central Kafka topic. A message contains the city's longitude/latitude, sampled time and current temperature. This message stream gets consumed by a *stream_processing* node using SparkStreaming. Messages are decoded and saved to Cassandra. We decided to use the database instead of SparkStreaming windowing because of the additional persistence and ease of access to the data. Decoupled from the stream processing, a *batch_processing* node continuously uses pyspark and CQL to average each city's temperature entries and collect these samples. It synthesizes the current estimate of the global temperature map, generates the reference model for the current day with the *reference_model_loader* and saves the heat map of their differences to our webserver's resources.

The webserver is therefore decoupled from our pipeline and only has to load the updated heat

map. To support portability and increase ease of execution, each node is started within its own container. A single Docker-compose file takes care of both the build and startup procedures, and ensures correct staging to avoid connection errors. Figure 1 provides a flowchart of the complete architecture.

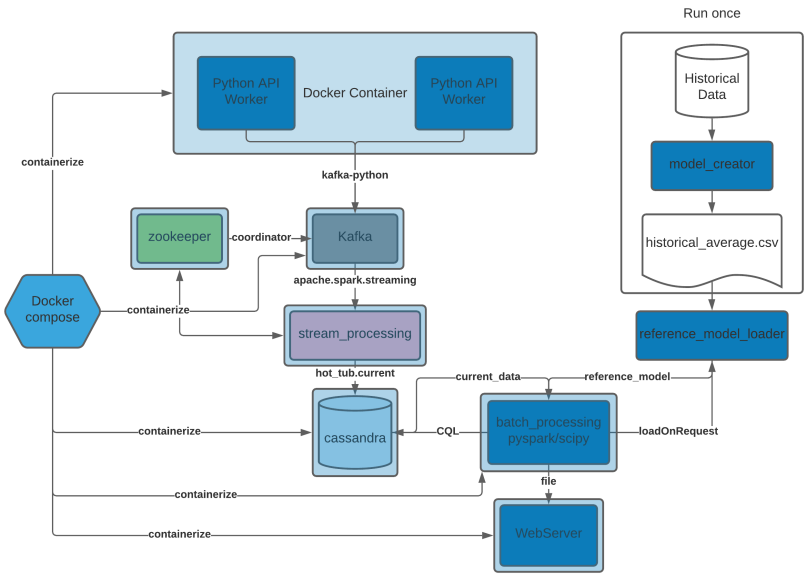

Figure 1: Hot Tub architecture. Python nodes are in dark blue, Scala nodes in pink, Docker containers surrounding their components in light blue.

4 Results

We were able to implement both reference and live temperature model synthesizing using scalable frameworks with Kafka, Spark and Cassandra [3, 4, 5]. As a result, we obtain a high resolution difference map of the current temperature compared to a pre-industrial level. Figure 2 shows this heat map. We observe significantly higher temperatures in parts of Europe, Russia, Africa and South America, whereas the US currently seems to experience temperatures below the pre-industrial average for this day. For reasons discussed in section 5, results for Greenland and Antarctica may be misleading.

5 Discussion

Verification of this map proves to be difficult because ground truth data is not available to us, and our method is inherently susceptible to fluctuations. Although intentional to provide users with a live monitor, this susceptibility prevents most methods of testing. It does however produce plausible results considering our sampling density. Exceptions are regions with extremely few data points such as Antarctica, where the nearest neighbour interpolation approach fails to account for the latitude dependency of the temperature and therefore fails to generalize correctly.

6 Usage

The project's only requirement is a Docker/Docker-compose installation. From the project's root folder, execute

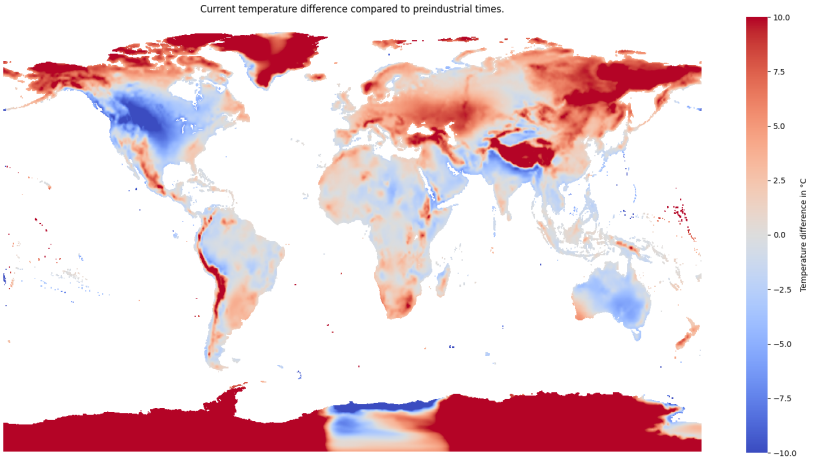

Figure 2: Hot Tub heat map result for the 25th of October 2020.

```
$ docker-compose up
```

This starts all nodes. The webserver will become accessible at *localhost:5000*. It should be noted that the system needs to run at least 24 hours to populate its Cassandra database in order to obtain a meaningful interpolation. Of course an inspection of the model during this process is possible, but a runtime smaller than 24 hours potentially leads to the observation of a significant bias due to day/night temperatures.

7 Future work

During the project, we were severely limited by memory (16GB) as all services ran on a single machine. Since the system is designed with horizontal scaling in mind, distributing nodes across different machines would be the next logical thing. Also, in order to fully utilize the capabilities of our architecture and frameworks while also significantly increasing accuracy, it would be desirable to add orders of magnitude more API workers/requests to the system. However, this requires a premium subscription to the *OpenWeatherMap* API. Furthermore, the addition of data sources for remote and low sample density regions such as Greenland, the Arctic and Antarctica would be highly desirable in order to address the main weakness of the current model.

References

[1] Berkeley Earth dataset, *http://berkeleyearth.org/archive/data/*, Berkeley Earth Surface Temperature-Projekt. Daily Land (Experimental; 1880 – 1890) Average Temperature (TAVG) 1º x 1º Latitude-Longitude Grid

[2] OpenWeatherAPI, *https://openweathermap.org/api*

[3] Kafka, *https://kafka.apache.org/*

[4] Spark, *https://spark.apache.org/*

[5] Cassandra, *https://cassandra.apache.org/*