Student: Giang Bui

ID: 37306207

Date: 19/04/2023

DATA 420- ASSIGNMENT 1

PREFACE

For this assignment, our focus will be on analysing climate summaries data obtained from The Global Historical Climatology Network. This extensive dataset contains daily climate records from more than 100,000 land surface stations in 180 countries and territories. Every 30 days, about 20,000 updates are added, covering a diverse range of variables including precipitation, daily maximum and minimum temperatures, snowfall, snow depth, among others. To ensure data consistency, the information was gathered from various sources and underwent a standard set of quality assurance reviews.

The objective of this report is to outline the methodology and present the findings based on the code developed.

PROCESSING

Question 1. Defining source of metadata table and '*daily*' data

a. Data structure

The entirety of the data is located within the HDFS directory *hdfs:///data/ghcnd/*, which contains four metadata files in .txt format (ghcnd-countries.txt, ghcnd-inventory.txt, ghcnd-states.txt, ghcnd-stations.txt) and a folder called "*daily*." This folder contains 263 gzip csv files , each of which contains climate summaries for a given year from 1750 to 2023. The gzip format is a compressed file format which is used to reduce the size of the file without losing any of the original data. The directory tree is structured as follows:

```
/data/ghcnd/
  |--- daily
  |----- 1750.csv.gz
  |----- 1763.csv.gz
  |-----  ...
  |----- 2021.csv.gz
  |----- 2022.csv.gz
  |----- 2023.csv.gz
  |--- ghcnd-countries.txt
  |--- ghcnd-inventory.txt
  |--- ghcnd-states.txt
  |--- ghcnd-stations.txt
```

b. The evolution of file sizes from 1750 to 2023:

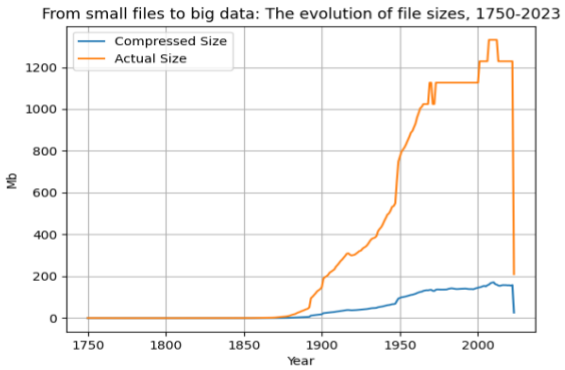


Figure 1. The evolution of file sizes from 1750 to 2023

The plot in Figure 1 clearly indicates that file size is increasing as additional stations and variables are incorporated. During the period from 1750 to 1877, file sizes remained below 1 megabyte. However, after this point, there was a significant surge in file size. By 1965, file sizes had surpassed 1 gigabyte and continued to increase steadily until they reached their peak of 1.3 gigabytes during the 2007-2012 period. Since then, the file sizes have remained relatively stable.

c. The size of '*daily*' folder and metadata files:

File	Size
Daily folder (includes 263 gzip files)	97.6 Gb
ghcnd-countries.txt	28.6 K
ghcnd-inventory.txt	259.0 M
ghcnd-states.txt	8.5 K
ghcnd-stations.txt	81.5 M
Total	98.0 Gb

Figure 2. The ghcnd files size summary

It is apparent from the figure 2 that the "*daily*" folder, which contains 263 gzip files, accounts for 99.6% of the total size, equating to 97.6 gigabytes. Among the four metadata text files, the largest is *ghcnd-inventory.txt*, which occupies 259.0 megabytes, followed by *ghcnd-stations.txt* at 81.6 megabytes. The sizes of *ghcnd-countries.txt* and *ghcnd-states.txt* are comparatively minor, at only 28.6 and 8.5 kilobytes, respectively.

Question 2. Exploring briefly metadata tables

a. Daily's schema definition based on the description:

The daily's schema in description is described as figure 3 below:

Name	Type in description
ID	StringType()
DATE	StringType()
ELEMENT	StringType()
VALUE	DoubleType()
MEASUREMENT FLAG	StringType()
QUALITY FLAG	StringType()
SOURCE FLAG	StringType()
OBSERVATION TIME	StringType()

Figure 3. Daily's schema description in assignment

b. Daily's data description accuracy:

The 1000 rows of '*daily*' data have been loaded into Spark. The data description provided in "DATA420-23S1 Assignment 1 (GHCN Data Analysis in Spark).pdf" is accurate, except for the "DATE" and "OBSERVATION TIME" fields, which are not formatted correctly for automatic parsing by DateType() and TimestampType(). To address this, these fields should be manually converted from StringType() to DateType() and TimestampType(), respectively, after being loaded. Additionally, it is recommended to optimize the "VALUE" field by defining it as an IntegerType() rather than RealType(), as mentioned in the readme.txt.

c. Metadata tables summary:

The metadata tables are stored in a fixed-width text format, so specific substrings within the text records must be selected to create their corresponding schemas. To accomplish this, the text files are first loaded into Spark as dataframes containing a single column consisting of the text string for each table. Using the pyspark.sql.functions.substring and trim() functions, the relevant text is extracted and placed into its respective column. The schema's data types are then applied based on the column name.

Stations metadata table

Figure 4 below is the schema for the "stations" metadata table:

Name	Type in description
Station_ID	StringType()
Latitude	DoubleType()
Longitude	DoubleType()
Elevation	DoubleType()
State	StringType()
Name	StringType()
GSN_Flag	StringType()
HCN_CRN_Flag	StringType()
WMO_ID	StringType()

Figure 4. 'stations' metadata table schema

The "stations" metadata table contains 124,247 rows, indicating that there are 124,247 distinct stations. A staggering 93.6% of all stations, or a whopping 116,297 stations, do not possess a WMO (World Meteorological Organization) ID.

States metadata table

Below is the figure 5 showing the schema for the "states" metadata table:

Name	Type in description
StateCode	StringType()
StateName	StringType()

Figure 5. 'states' metadata table schema

The "states" metadata table consists of 74 rows, representing the 74 states/territories of the United States and provinces of Canada.

Countries metadata table

Below is the figure 6 showing the schema for the "countries" metadata table:

Name	Type in description
Country_Code	StringType()
Country_Name	StringType()

Figure 6. 'countries' metadata table schema

The "countries" metadata table comprises 219 rows, indicating that it contains Code and Name on 219 countries or territories worldwide.

Inventory metadata table

Below is the figure 7 showing the schema for the "*inventory*" metadata table:

Name	Type in description
Inventory_ID	StringType()
Latitude	DoubleType()
Longitude	DoubleType()
Element	StringType()
FirstYear	IntegerType()
LastYear	IntegerType()

Figure 7. 'inventory' metadata table schema

The "*inventory*" metadata table consists of 737,925 rows, which correspond to the various elements that were recorded by each station and the time periods during which those elements were recorded.

In the metadata "*inventory*" data table, we have 124,239 distinct Inventory_ID, while the total number of stations according to 'stations' metadata table is 123,247 which means there are 8 stations were inactive OR we have missing values from '*inventory'* metadata table.

Question 3. Combining Metadata Tables into a Single Enriched Metadata Table

We merge the metadata tables 'stations', 'countries', 'states', and *'inventory'* into a single table, which allows for easy filtering and sorting of station-level attributes.

Metadata tables transformation

To include information on the country and state of each station, we combine the 'stations' table with the 'countries' and 'states' metadata tables, using matching IDs between the tables.

Examining the '*inventory'* metadata table":

We grouping the '*inventory*' metadata table by 'Element' and by 'Inventory_ID' separately. And aggregate the information needed.

Below is the information that we collected from the transformed tables:

- There are 144 element types used as the weather measurements. Among them, we have 5 core elements as mentioned in the *readme.txt* file including: *PRCP - Precipitation (tenths of mm); SNOW - Snowfall (mm); SNWD - Snow depth (mm); TMAX - Maximum temperature (tenths of degrees C); TMIN - Minimum temperature (tenths of degrees C)*.
- The figure 8 below displays the top 10 most commonly used elements. *PRCP* emerges as the most popular element with 122,229 stations recording it. *SNOW* and *MDPR* rank second and third with 75,563 and 67,146, respectively. The remaining elements follow suit in descending order of popularity.

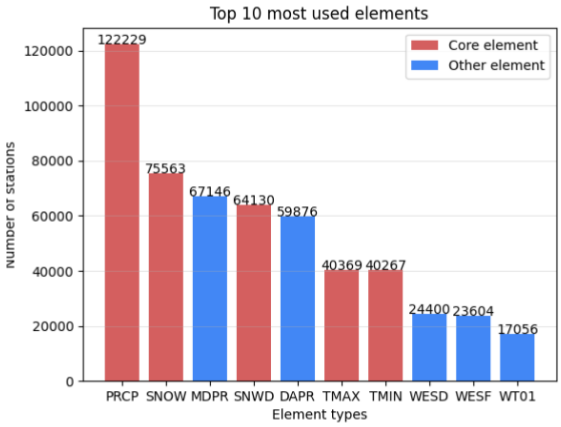


Figure 8. Top 10 most frequent used elements by stations

- The figure 9 below displays the top 5 stations that collect the largest number of elements. As we can see, 70 is the maximum number of elements recorded which belong to stations with ID USW00014607 and USW00013880. Upon inspecting the top 1000 stations in terms of the number of elements recorded, it is evident that a majority of them are located in the US.

Inventory_ID	FirstYear	LastYear	Total_Element	Core_Element	Other_Element
USW00014607	1939	2023	70	5	65
USW00013880	1937	2023	70	5	65
USW00023066	1900	2023	67	5	62
USW00013958	1938	2023	66	5	61
USW00024121	1988	2023	65	5	60

Figure 9. top 5 stations that colect the largest number of elements

- A total of 20,449 stations have collected all five core elements

Finding total number of stations only selected one core element named ‘PRCP’ and without selecting any other element:”

To achieve this task, first, we create the table *‘core_inventory’* with filter ‘*inventory’* metadata table which have ‘PRCP’ element. Then, we create a table named ‘inventory_stations_summary_prcp’ with filter ‘*inventory_stations_summary’* table which have ‘Total_Element’ is equal to 1. Finally, we create ‘*inventory_stations_summary’* which is a result from performing the Inner Join of the two tables above. Then we figure out there are 16,272 stations have only selected one core element, specifically 'PRCP', without selecting any other element.

Combine the metadata table into a single enriched table:

We merge the two table ‘*stations’* and ‘*inventory_stations_summary’* and save it as the ‘*stations_enriched’* metadata table which have all new columns included. The enriched table has schema as Figure 10 below:

Name	Type in description
Country_Code	StringType()
Station_ID	StringType()
Latitude	DoubleType()
Longitude	DoubleType()
Elevation	DoubleType()
Name	StringType()
GSN_Flag	StringType()
HCN_CRN_Flag	StringType()
WMO_ID	StringType()
Country_Name	StringType()
StateCode	StringType()
StateName	StringType()
FirstYear	IntegerType()
last year	IntegerType()
Total_Element	IntegerType()
Core_Element	IntegerType()
Other_Element	IntegerType()

Figure 10. stations_enriched's schema

The ‘*stations’* table enriched with additional information is more valuable for subsequent data analysis. It has been formatted as CSV, which is a widely supported text-based format that can be easily parsed into a structured format by most programming languages.

Perform a join operation between the '*daily*' data and the *'stations_enriched*' metadata table:
By performing a LEFT JOIN operation between the enriched stations data and a subset of 1000 rows from the '*daily'* data the identify columns of two tables are matched, and then filtering out the NULL values in the 'Station_ID' column, we can observe that there are no stations in the subset of '*daily'* data that are not present in the enriched stations data.

When we do the similar method with the full '*daily'* data, the computational time is expensive because of the shuffling cost when joining the huge '*daily'* data with the much smaller 'station_enriched' data. To avoid this, we can employ a BROADCAST JOIN, the *stations* data, which is significantly smaller than the '*daily'* data, can be broadcasted to every executor and joined locally, eliminating the need for data shuffling.

We can count the total number of distinct stations in '*daily'* data and compare with total number of distinct stations in the '*stations_enriched'*. If the number of stations in '*daily'* data is smaller than the number of stations in '*stations_enriched'*, then we can conclude that all of the stations in '*daily*' are listed in the '*station_enriched'* metadata table. However, it is still have a chance that the stations in '*daily'* might be not listed in the '*stations_enriched'*, because we can not check if the ID of stations are matched or not without joining the tables. With this way, we can see that total number of distinct stations in '*full_daily'* is 124,240, while the total number of stations in '*stations_enriched'* is 124,247. Which suggest there is high chance that all the stations in daily data is listed in the '*stations_enriched'*

Beside LEFT JOIN, we can employ the INNER JOIN the '*daily'* table and '*stations_enriched*' metadata table. The '*stations_enriched'* once again is suggested to broadcast to every executor and joined locally. After joining, we count the distinct stations recorded in the inner joined table and in the '*stations_enriched'*. If the number of the stations in the inner-joined table is fewer than that of the '*stations_enriched'*, we can conclude that all the stations in the '*daily'* are listed in the '*stations_enriched'* metadata table. From the result of the performance, there are 124,240 distinct stations in the inner-joined table which is fewer than that of the '*stations_enriched'* which 124,247. We can conclude that there are no stations in the subset of daily data that are not present in the enriched stations data.

ANALYSIS

Question 1. Thorough Examination of '*Stations'* Metadata

Stations and their networks:

Out of the 124,247 stations in total, 41,467 were active in the year 2022. To determine this, we counted all unique stations that were first active in or before 2002 and last active in or after 2022.

The GCOS Surface Network (GSN) consists of 991 stations, the US Historical Climatology Network (HCN) comprises 1,218 stations, and the US Climate Reference Network (CRN) has 234 stations.

Furthermore, 15 stations are found in more than one network, which are GSN and HCN, and all of them are located in the United States. These stations have been active for at least 60 years until 2023, and they collect a broad range of elements. The detail of stations are shown as Figure 11 below:

Country Code	Station_ID	GSN Flag	HCN_CRN Flag	State Code	First Year	Last Year	Total Element
US	USW00013782	GSN	HCN	SC	1893	2023	26
US	USW00024128	GSN	HCN	NV	1877	2023	52
US	USW00023044	GSN	HCN	TX	1938	2023	54
US	USW00093729	GSN	HCN	NC	1957	2023	50
US	USW00003870	GSN	HCN	SC	1962	2023	52
US	USW00023051	GSN	HCN	NM	1896	2023	49
US	USW00094008	GSN	HCN	MT	1942	2023	52
US	USW00012921	GSN	HCN	TX	1946	2023	52
US	USW00014922	GSN	HCN	MN	1938	2023	57
US	USW00014771	GSN	HCN	NY	1938	2023	53
US	USW00024144	GSN	HCN	MT	1938	2023	54
US	USW00014742	GSN	HCN	VT	1940	2023	54
US	USW00024213	GSN	HCN	CA	1941	2023	41
US	USW00012836	GSN	HCN	FL	1948	2023	54
US	USW00093193	GSN	HCN	CA	1941	2023	54

Figure 11. Stations found in more than one network

Total number of stations per country

The United States ranks first in terms of the total number of stations per country, with 70,627 stations, leaving behind Australia in second place with 17,088 stations and Canada in third with 9,146 stations. The Figure 12 below illustrates the top 10 countries with the largest number of stations.

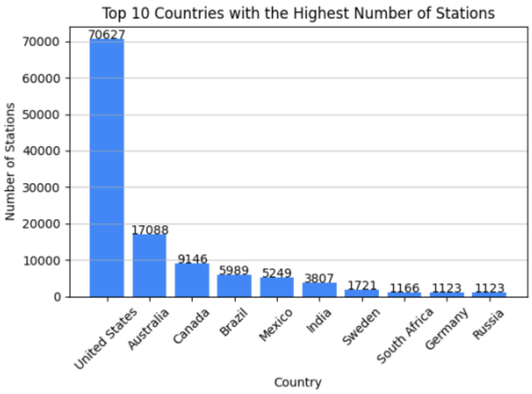


Figure 12. Top 10 countries with highest number of stations

Total number of stations per state

The Figure 13 below shows the top 10 states in terms of the number of stations they have. Texas has the highest number of stations with 5,957, followed by Colorado with 4,570 and California with 3,024.

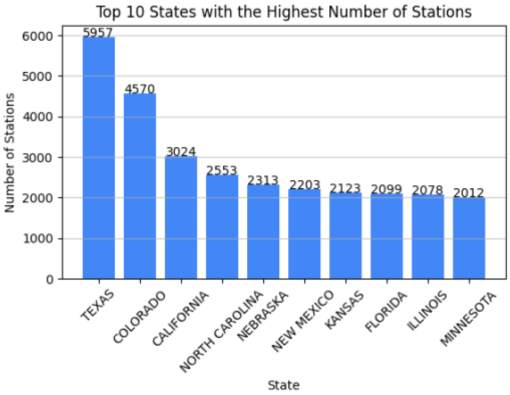


Figure 13. Top 10 states with highest number of stations

Total number of stations in the southern hemisphere

There are a total of 25,337 stations located in the southern hemisphere, which represents 20.39% of all the stations worldwide.

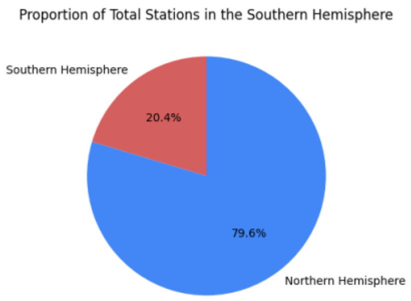


Figure 14. Proportion of the stations in the Sothern hemisphere

The total number of stations in the territories of the United States around the world is 374. Puerto Rico has the highest number of stations, with a total of 223. The figure 15 below displays the top 5 US territories with the highest number of stations:

Country_Code	Country_Name	Total_Stations
RQ	Puerto Rico	233
VQ	Virgin Islands	68
GQ	Guam	30
AQ	American Samoa	21
CQ	Northern Mariana Islands	11

Figure 15.Top 5 terriotories of United State with highest number of stations

Question 2 The geographical distance between two stations.

Set up function to calculate the distance between two points on a spherical object

The Haversine formula is utilized to determine the shortest distance between two points on a spherical object, such as the Earth. This mathematical equation considers the curvature of the Earth's surface and uses the latitude and longitude coordinates of two points to compute the distance between them.

For this task, a Spark UDF (user-defined function) called 'geo_distance' is utilized to create a function that calculates the distance between two stations.

The stations in United Arab Emirates (UAE) are used to test the created function. First, we filt the stations from UAE and save it in the new table named ‘ae_stations’. There are only 4 stations in UAE. Second, we use the CROSS JOIN function to pair the stations in UAE together, and remove the duplicate pairs, results in 6 pairs. The table name is ‘ae_stt_pairs’. Finally, we use the “geo_distance” function to calculate the distance of each pair of stations and save it under the new column name ‘distance’ in the ae_stt_pairs. The 6 distances of 6 pairs is shown as the figure 16 below:

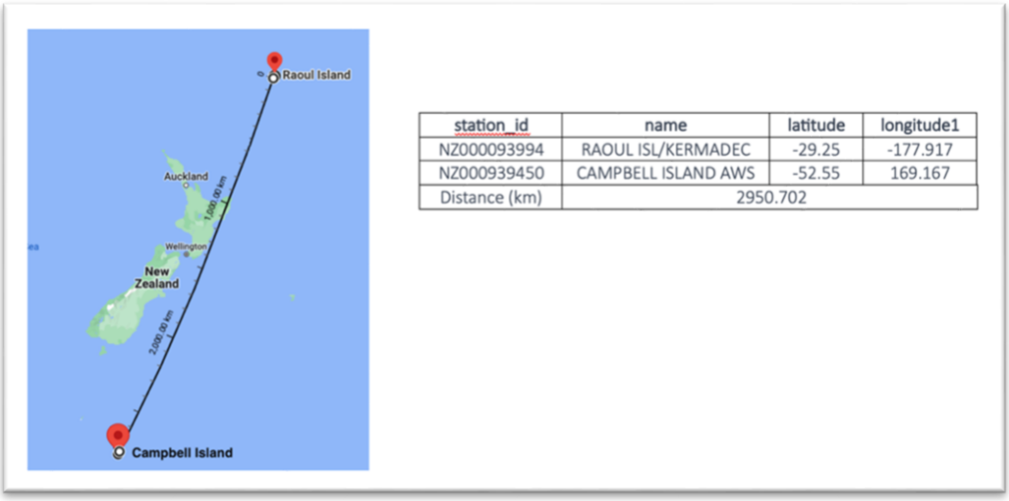
Station A				Station B				
station_id	name	latitude	longitude	station_id	name	latitude	longitude	Distance (km)
AE000041196	SHARJAH INTER. AIRP	25.333	55.517	AEM00041218	AL AIN INTL	24.262	55.609	68.117
AE000041196	SHARJAH INTER. AIRP	25.333	55.517	AEM00041194	DUBAI INTL	25.255	55.364	17.71
AE000041196	SHARJAH INTER. AIRP	25.333	55.517	AEM00041217	ABU DHABI INTL	24.433	54.651	112.041
AEM00041194	DUBAI INTL	25.255	55.364	AEM00041218	AL AIN INTL	24.262	55.609	68.235

AEM00041194	DUBAI INTL	25.255	55.364	AEM00041217	ABU DHABI INTL	24.433	54.651	95.041
AEM00041217	ABU DHABI INTL	24.433	54.651	AEM00041218	AL AIN INTL	24.262	55.609	107.078

Figure 16. The distance (km) between UAE stations

Apply this function to compute the pairwise distances between all stations in New Zealand.

New Zealand has a total of 15 stations, resulting in 105 unique pairs. Using the 'geo_distance' function to calculate the distances between each pair of stations and save the result to the output directory. From the result, we found that the pair with the furthest distance is RAOUL ISL/KERMADEC and CAMPBELL ISLAND AWS, with a distance of 2950.702 km.



station_id	name	latitude	longitude1
NZ000093994	RAOUL ISL/KERMADEC	-29.25	-177.917
NZ000939450	CAMPBELL ISLAND AWS	-52.55	169.167
Distance (km)	2950.702		

Figure 17.The pair of stations with furthest distance (km)

Question 3. Examining transformation efficiency

Explore the default block size, files' sizes and the block required for each file

By default, the block size of Hadoop Distributed File System (HDFS) is 134,217,728 bytes, which is equivalent to 128 MB.

The log messages about the health and status of two files 2023.csv.gz and 2022.csv.gz from a Hadoop Distributed File System (HDFS) cluster indicate that:

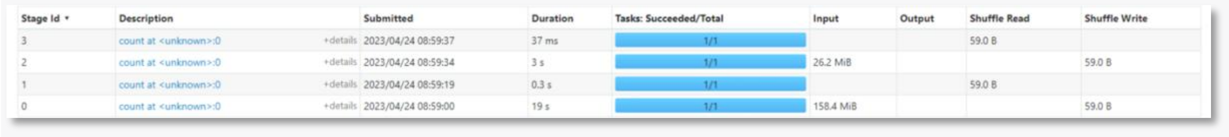
- The file 2023.csv.gz is made up of a single block with a size of 27,521,531 bytes and being replicated across 8 different nodes in the cluster.
- The file 2022.csv.gz has a total size of 166,075,423 bytes, and it consists of two blocks with sizes of 134,217,728 bytes and 31,857,695 bytes respectively.

  Regarding to the file 2023.csv.gz, Spark cannot load and execute transformations in parallel since there's only one block available. Spark requires multiple blocks, residing on different data nodes and executed by separate executors, to process in parallel. However, in the file 2022.csv.gz, there are two blocks, allowing Spark to load and apply transformations in parallel.

Count the number of observations in 2022 and then separately in 2023. Explore the number of tasks executed

In the year 2022, there are 37,375,779 observations, while in the year 2023, there are 6,031,842 observations. It's worth noting that the year 2023 has not completed yet.

The job "Count the number of observations in 2022" consists of 2 stages with stage id 0 and 1, and the job "Count the number of observations in 2023" consists of 2 stages with stage id 2 and 3. The detail of stages is shown as the figure 18 below. As we can see that there is one task per stage.



Stage Id ▾	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
3	count at <unknown>:0	+details	2023/04/24 08:59:37	37 ms	1/1			59.0 B	
2	count at <unknown>:0	+details	2023/04/24 08:59:34	3 s	1/1	26.2 MiB			59.0 B
1	count at <unknown>:0	+details	2023/04/24 08:59:19	0.3 s	1/1			59.0 B	
0	count at <unknown>:0	+details	2023/04/24 08:59:00	19 s	1/1	158.4 MiB			59.0 B

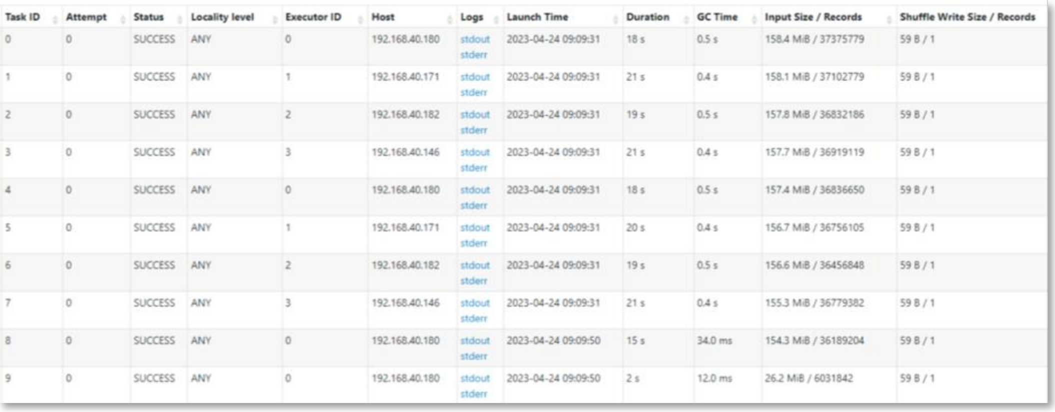
Figure 18. Detail of job "count number of observations'

Conclusion: The number of tasks executed was not not correspond to the number of blocks in each input. Each block of input data is processed by one task, but the number of tasks that are executed for a job can be different from the number of blocks in the input data. The number of tasks is determined by the Spark application's configuration settings, such as number of partitions, which dictate how many tasks are created to process the data.

Load and count the number of observations from 2014 to 2023 (inclusive).

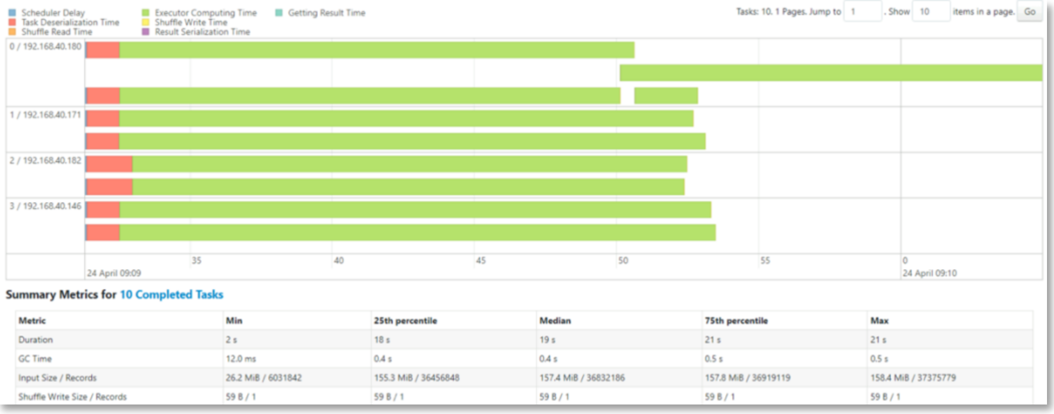
Across the span of 10 years, from 2014 to 2023, there are 337,279,894 observations. The job to process this data is comprised of two stages. The first stage, which involves loading the CSV files, contains 10 tasks, and the second stage only has one "counting" task. The detail of first and stage is shown as below.

Stage 0 as figure 19 and 20: there are 10 compressed files, resulting in 10 partitions and 10 corresponding tasks in stage 0. As we can see from the figure below, there are 4 worker nodes executing these 10 tasks. there are 8 tasks out of 10 tasks are launched simultaneously.



Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Input Size / Records	Shuffle Write Size / Records
0	0	SUCCESS	ANY	0	192.168.40.180	stdout stderr	2023-04-24 09:09:31	18 s	0.5 s	158.4 MiB / 37375779	59 B / 1
1	0	SUCCESS	ANY	1	192.168.40.171	stdout stderr	2023-04-24 09:09:31	21 s	0.4 s	158.1 MiB / 37102779	59 B / 1
2	0	SUCCESS	ANY	2	192.168.40.182	stdout stderr	2023-04-24 09:09:31	19 s	0.5 s	157.8 MiB / 36832186	59 B / 1
3	0	SUCCESS	ANY	3	192.168.40.146	stdout stderr	2023-04-24 09:09:31	21 s	0.4 s	157.7 MiB / 36919119	59 B / 1
4	0	SUCCESS	ANY	0	192.168.40.180	stdout stderr	2023-04-24 09:09:31	18 s	0.5 s	157.4 MiB / 36836650	59 B / 1
5	0	SUCCESS	ANY	1	192.168.40.171	stdout stderr	2023-04-24 09:09:31	20 s	0.4 s	156.7 MiB / 36756105	59 B / 1
6	0	SUCCESS	ANY	2	192.168.40.182	stdout stderr	2023-04-24 09:09:31	19 s	0.5 s	156.6 MiB / 36456848	59 B / 1
7	0	SUCCESS	ANY	3	192.168.40.146	stdout stderr	2023-04-24 09:09:31	21 s	0.4 s	155.3 MiB / 36779382	59 B / 1
8	0	SUCCESS	ANY	0	192.168.40.180	stdout stderr	2023-04-24 09:09:50	15 s	34.0 ms	154.3 MiB / 36189204	59 B / 1
9	0	SUCCESS	ANY	0	192.168.40.180	stdout stderr	2023-04-24 09:09:50	2 s	12.0 ms	26.2 MiB / 6031842	59 B / 1

Figure 19. The details tasks from stage 0 - "loading csv file"

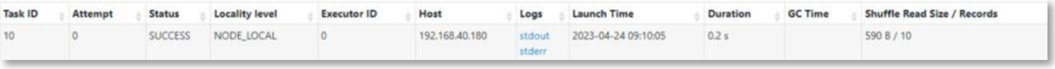


Summary Metrics for 10 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	2 s	18 s	19 s	21 s	21 s
GC Time	12.0 ms	0.4 s	0.4 s	0.5 s	0.5 s
Input Size / Records	26.2 MiB / 6031842	155.3 MiB / 36456848	157.4 MiB / 36832186	157.8 MiB / 36919119	158.4 MiB / 37375779
Shuffle Write Size / Records	59 B / 1	59 B / 1	59 B / 1	59 B / 1	59 B / 1

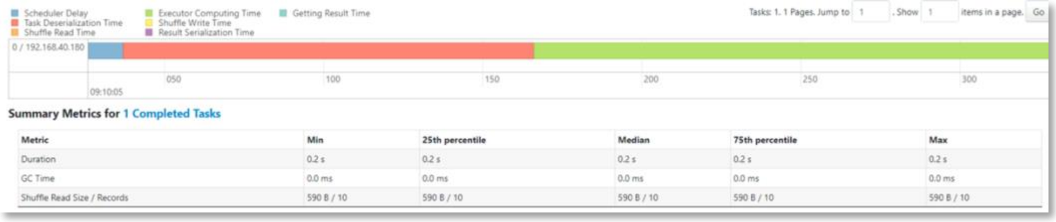
Figure 20. Event timeline from stage 0 - "loading CSV file"

Stage 1: In this stage, there is one count task executed by the worker node 0 as figure 21 and 22.



Task ID	Attempt	Status	Locality level	Executor ID	Host	Logs	Launch Time	Duration	GC Time	Shuffle Read Size / Records
10	0	SUCCESS	NODE_LOCAL	0	192.168.40.180	stdout stderr	2023-04-24 09:10:05	0.2 s		590 B / 10

Figure 21. The details tasks from stage 1 - "counting observations"



Summary Metrics for 1 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	0.2 s	0.2 s	0.2 s	0.2 s	0.2 s
GC Time	0.0 ms	0.0 ms	0.0 ms	0.0 ms	0.0 ms
Shuffle Read Size / Records	590 B / 10	590 B / 10	590 B / 10	590 B / 10	590 B / 10

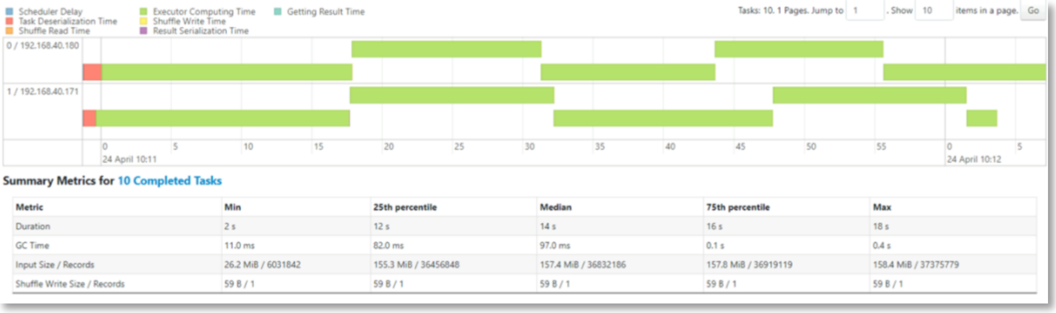
Figure 22. Event timeline from stage 1 - "counting observations"

How Spark partitions input files that are compressed

Spark is unable to split compressed input files like gzip or bzip2 into smaller parts as it can with uncompressed files. Therefore, Spark creates a single partition for each compressed file, causing each compressed file to be processed by a single executor.

Parallelism for stage 0:

Looking at Figure 20 - the Event Timeline of stage 0, it is evident that utilizing 4 executor instances, each with 2 cores, resulted in a parallelism level of 8, allowing for the simultaneous launch of 8 tasks. Next, we analyze the parallelism level achieved when using 2 executor instances, each with 1 core per executor, as shown in the figure 23 below.



Summary Metrics for 10 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	2 s	12 s	14 s	16 s	18 s
GC Time	11.0 ms	82.0 ms	97.0 ms	0.1 s	0.4 s
Input Size / Records	26.2 MiB / 6031842	155.3 MiB / 36456848	157.4 MiB / 36832186	157.8 MiB / 36919119	158.4 MiB / 37375779
Shuffle Write Size / Records	59 B / 1	59 B / 1	59 B / 1	59 B / 1	59 B / 1

Figure 23. Parallelism using two worker nodes and one core each node

It is evident that the achieved level of parallelism was 2, with a total processing time of roughly 67 seconds. This is in contrast to the previous run where 4 executor instances, each with 2 cores, resulted in a parallelism level of 8. In the previous run, the first 8 tasks were completed in just 20 seconds, with all 10 tasks completed in around 32 seconds, which is approximately 35 seconds faster than this run.

To enhance the task processing, we can increase the level of parallelism by adding more worker nodes and increasing the number of cores per worker node. For instance, if we increase our resources to include 4 executor instances, each with 3 cores per executor, the level of parallelism will increase to 12. If we use those resources when loading and applying transformations to *'daily'*, there are 12 tasks will run parallelly.

Question 4. Exploring and visualising ‘*daily*’ data

Count the number of observations

There are 3,064,620,240 observations in the ‘*daily’* .

The figure 24 below shows the number of observations that have a core element. It is evident that among the five core elements, PRCP- Precipitation (tenths of mm) is the most commonly observed with 1,057,396,673 observations.

Element	count
SNWD	294,454,702
SNOW	348,203,650
TMIN	450,155,708
PRCP	1,057,396,673
TMAX	451,364,119

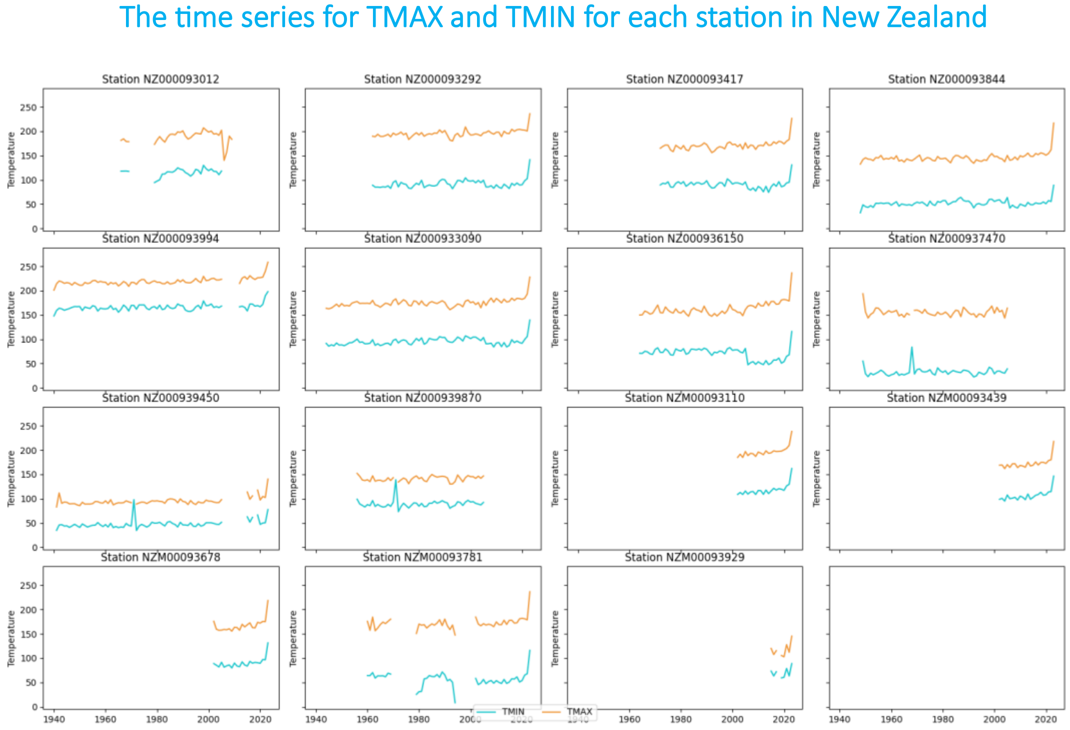
Figure 24. Total daily observations from five core elements

A total of 9,118,440 observations with TMIN lack a corresponding TMAX observation, and these observations are present in 27,876 distinct stations.

Over a span of 84 years, there are 478,712 observations of both TMIN and TMAX for all stations located in New Zealand.

The program's output is first stored in HDFS and then copied to the local disk. The number of rows in the part files was counted using the "wc-l" bash command, which showed that there were 84 files and 478,712 rows, matching the 84 years and 478,712 observations of data processed in Spark.

The time series plot for TMAX and TMIN for each station in New Zealand



The time series for TMAX and TMIN for each station in New Zealand

Figure 25. The time series for TMAX and TMIN for each station in New Zealand

From figure 25, we can see that the period of time recording are different among the stations.
In 1940, two New Zealand stations, namely NZ000093994 and NZ000939450, began recording TMAX and TMIN temperature elements, making them the earliest recorders of these elements. In contrast, some stations like NZM00093929, NZM00093110, NZM00093439, and NZM00093678 began recording these elements much later, with NZM00093929 starting around 2014 and the others in the early 2000s. By 2023, 12 out of the 15 stations with these elements were still active. Nonetheless, there are gaps in the recorded data between 1940 and 2023 due to missing TMAX and TMIN values from some stations.

In particular, station NZ000093012 experienced an unusual pattern in 2006 and 2007, with a sudden drop in maximum temperature. As for TMIN, some unusual highs were recorded at station NZ000937470 in 1968, NZ000939450 in 1971, and NZ000939870 in 1971.
The gap between TMAX and TMIN are different among the stations. While some stations have a big gap such as NZM00093781, NZ000937470, Some stations such as NZM00093929, NZ000939450 has a much more narrow gaps.

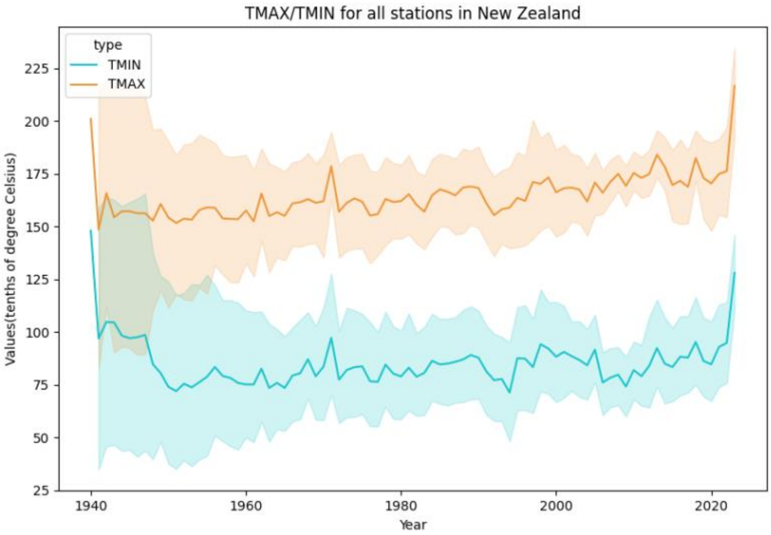


Figure 26. TMAX and TMIN for all stations in New Zealand

It is evident that TMAX and TMIN in New Zealand have fluctuated with an upward trend, indicating that the country may be experiencing the effects of global warming. Furthermore, the gap between TMIN and TMAX has widened over time.
By grouping the precipitation observations by year and country, we have the top 10 average rainfall as table below. As we can see the highest average rainfall was witnessed in Equatorial Guinea in the year 2000 with 4361.0 mm, followed by Dominican Rupublic in 1775 with 3414.0 mm and Lao in 1974 with 2480.5 mm.

year	Country_Code	Country_Name	Average rainfall (mm)
2000	EK	Equatorial Guinea	4361.0
1975	DR	Dominican Republic	3414.0
1974	LA	Laos	2480.5
1978	BH	Belize	2244.7
197	NN	Sint Maarten	1967.0
1974	CS	Costa Rica	1820.0
1979	BH	Belize	1755.5
1973	NS	Suriname	1710.0
1978	UC	Curacao	1675.0
1977	BH	Belize	1541.7

Figure 27. Top 10 countries got largest average rainfall in 2022

By examining the observation with the highest average rainfall in Equatorial Guinea in 2000, it was found that there are only two weather stations in Equatorial Guinea: EKM00064810 and EKM00064820. Further analysis of all observations from these two stations with the element 'PRCP' in the year 2000 showed that there was only one record available for the whole year 2000 as figure 28 below. However, this observation had a Qflag value of "G", indicating that it failed the gap check, and an Sflag value of "S", suggesting that the value may differ significantly from the true *'daily'* data, particularly for precipitation, as mentioned in the Readme file. As a result, this finding should be approached with caution and may not be a reliable representation of actual precipitation levels in Equatorial Guinea in 2000.

Daily_ID	Date	Element	Value	Mflag	Qflag	Sflag	Observation_Time
EKM00064810	20000622	PRCP	4361	null	G	S	null

Figure 28. Daily rainfall in Equatorial Guinea in 2000

Before plotting the World average rainfall in 2022, we examine the distribution of these figure. The distribution is illustrated as the figure 29 below:

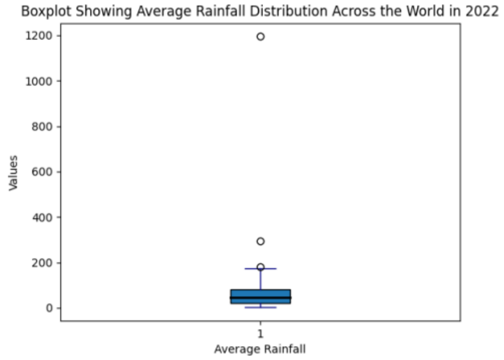


Figure 29.World average rainfall distribution in 2022

As we can see that there is one extreme outlier which is from Sierra Leone with the average rainfall of 1195.0 mm in 2022. By examining all *'daily'* records from this country as the figure 30 below, we only observed 2 observations in the whole year 2022 for PRCP element, which might introduce bias. Noticeably, those values were labelled 'S' in Sflag which means the data is label "to be used with caution, especially with precipitation". Because those observations makes this country data significantly different from other countries, which will skew the result and cause the color in the choropleth map to become very bright. For those reasons, we should remove this value to improve the quality of the choropleth map.

Daily_ID	Date	Element	Value	Mflag	Qflag	Sflag	Observation_Time
SL000061856	20220918	PRCP	2390	null	null	S	null
SL000061856	20220919	PRCP	0	null	null	S	null

Figure 30. 'daily' records of Sierra Leone in 2022

The choropleth map below illustrates the average rainfall for each country in the year 2022. After removing the outlier Sierra Leone, country Angola, located in Africa, received the highest amount of rainfall with approximately 293mm. The map (Figure 31) shows that regions closer to the equator such as Africa, South Asia, and South America received higher average rainfall compared to continents farther away such as North Asia, North America, Europe, Australia, and Antarctica. It is worth noting that some countries in the middle of Africa and Asia have missing values.

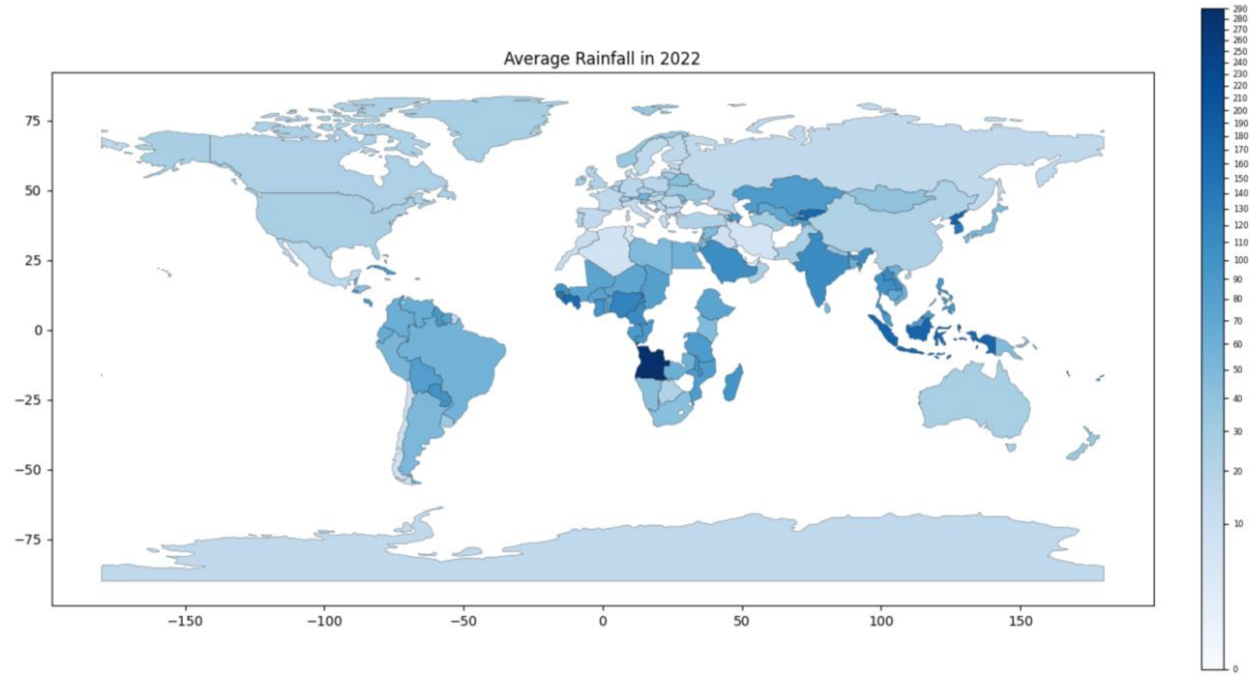


Figure 31. The average rainfall for each country in the year 2022