**Running the program:**

We have created a jar file for easy execution of the program

Please use the following command to execute:

hadoop jar weather.jar com.dbms.weather.Weather locations recordings output

recordings – folder having the recordings files (\*.txt)

locations – folder having the location file (\*.csv)

output – folder to be created to hold the output.

During the execution program will create two folders for intermediate output storage during chaining jobs:

Temp\_one\_output

Temp\_two\_output

**MapReduce Jobs:**

For the problem we have used two map-combine-reduce jobs and an additional map job in the end for sorting the results.

* Map-Reduce1:

Two maps are used one for the recordings and other for the locations file.

MapText.java: This Map extracts the stationID, temperature, frequency of temp recordings and the precipitation from the recordings. The key of stationID, month has a tag of “TXT” to recognize in the combiner.

Value field consists of temperature, frequency of temp recordings and the precipitation separated by a comma.

Eg: key – TXT0100020, 01 value – 34.5, 3, 0.04

MapCsv.java: This map extracts the stationID and its corresponding country-state. The key of stationID has a tag of “CSV” to recognize in the combiner. Value field consists of country and state concatenated. Only US country is considered in this phase.

Eg: key – CSV0100020 value – USCA

CombineText.java: In the combiner after recognizing the tag of the key, if it is “TXT” it calculates the average monthly temperature and sum of precipitation for the month and emits output. If it is “CSV” it just takes distinct value of country-state. Here the key field is just the stationID.

Eg: key – 0100020 value – 01, 33.2, 0.06

key – 0100020 value – USCA

ReducerWeather.java: The reducer picks up the values for each key (stationID). It extracts the country-state and makes it the key and concatenates every month temperature and precipitation values using an “X” and emits it to a temporary output file to be chained to the next Map-reduce job. Here a check is also made to filter out records that do not belong to US country.

Eg: key – USCA value – 01, 33.8, 0.06 X 02, 56.8, 0.09…..

* Map-Reduce2:

One map, one combiner and one reducer is used here

FinalMap.java: Here the map simply picks up the key and value from the temporary output above and sends it to the combiner

Eg: key – USCA value – 01, 33.8, 0.06 X 02, 56.8, 0.09…..

FinalCombiner.java: The combiner here calculates the months with the minimum temperature and maximum temperature for each state and also the difference of the maximum and minimum. The key here is the difference between temperatures

Eg: key – 23.5 value – 01, 33.8, 0.06 X 07, 57.3, 84.8, 0.09, USCA

FinalReducer.java: This reducer simply emits the output of the combiner. This reducer can be eliminated.

Eg: key – 23.5 value – 01, 33.8, 0.06 X 07, 57.3, 84.8, 0.09, USCA

* Map3:

The purpose of this map is to sort the output from the earlier reducer by the key(difference in temperature). It is also used to format the output to the required format

5.77 VI Temp 77.55 Prec 1.25,February-MinMonth Temp 83.32 Prec 2.45,July-MaxMonth

6.4 PR Temp 76.79 Prec 0.0,January-MinMonth Temp 83.19 Prec 0.0,August-MaxMonth

8.08 HI Temp 69.69 Prec 2.41,February-MinMonth Temp 77.78 Prec 1.29,August-MaxMonth

The join between the location and the resources happens in the combiner CombineText.java.

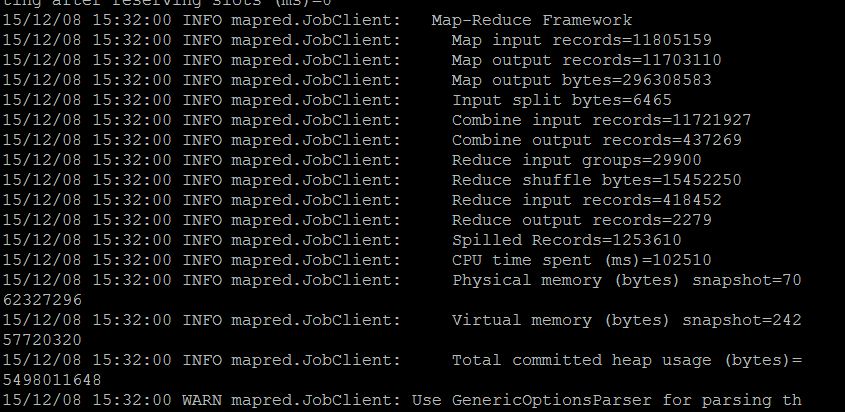
The join is preferred in the combiner rather than the mapper because the locations file is 6MB and we felt it was not efficient for doing the join in the Mapper.

The precipitation values are also calculated in the project. The frequency of temperature recordings is also taken into consideration to give appropriate weightage for the recordings.

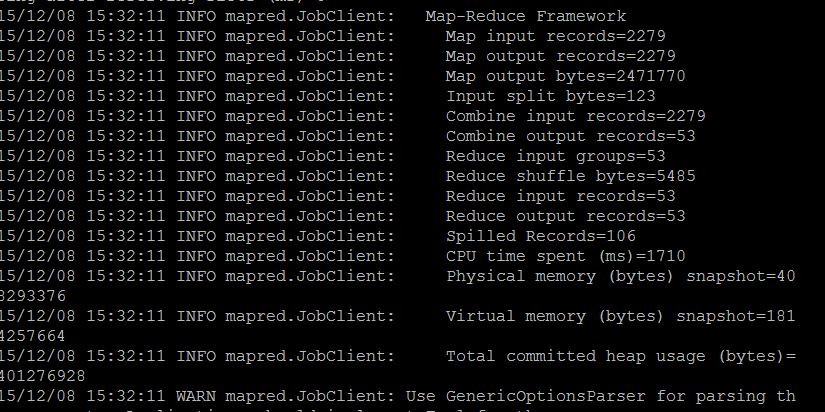
**OUTPUT:**

The file outputFile has the output of the program.

First job:



Second job:



Third job:

