**EXP 3: Map Reduce program to process a weather dataset.**

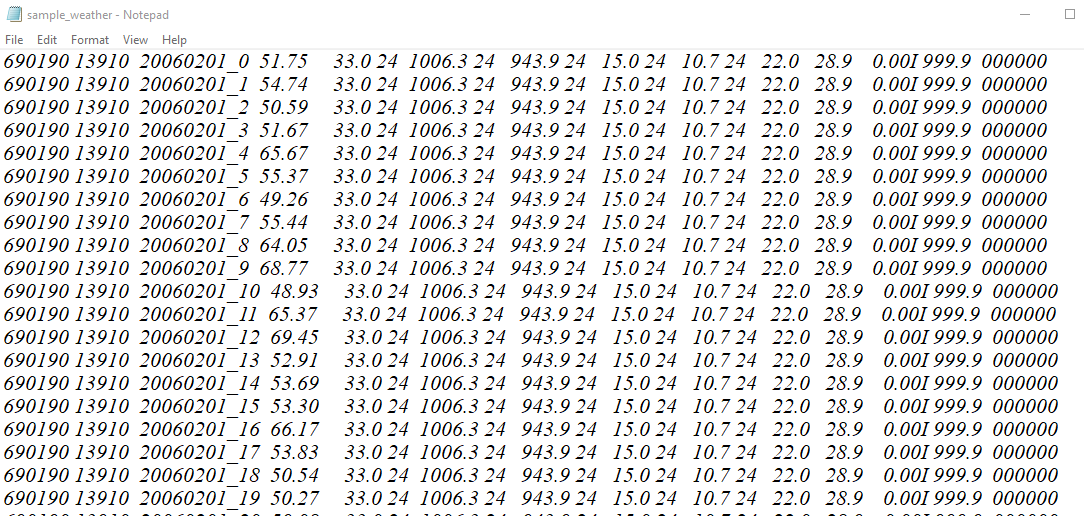
**AIM:**

To implement MapReduce program to process a weather dataset.

**PROCEDURE:**

**Step 1: Create Data File:**

Create a file named "sample\_weather.txt" and populate it with text data that you wish to analyse.



**Step 2: Mapper Logic - mapper.py:**

Create a file named "mapper.py" to implement the logic for the mapper. The mapper will read input data from STDIN, split lines into words, and output each word with its count.

**mapper.py:**

#!/usr/bin/python3

import sys

def map1():

for line in sys.stdin:

tokens = line.strip().split()

if len(tokens) < 13:

continue

station = tokens[0]

if "STN" in station:

continue

date\_hour = tokens[2]

temp = tokens[3]

dew = tokens[4]

wind = tokens[12]

if temp == "9999.9" or dew == "9999.9" or wind == "999.9":

continue

hour = int(date\_hour.split("\_")[-1])

date = date\_hour[:date\_hour.rfind("\_")-2]

if 4 < hour <= 10:

section = "section1"

elif 10 < hour <= 16:

section = "section2"

elif 16 < hour <= 22:

section = "section3"

else:

section = "section4"

key\_out = f"{station}\_{date}\_{section}"

value\_out = f"{temp} {dew} {wind}"

print(f"{key\_out}\t{value\_out}")

if \_\_name\_\_ == "\_\_main\_\_":

map1()

**Step 3: Reducer Logic - reducer.py:**

Create a file named "reducer.py" to implement the logic for the reducer. The reducer will aggregate the occurrences of each word and generate the final output.

**reducer.py:**

#!/usr/bin/python3

import sys

def reduce1():

current\_key = None

sum\_temp, sum\_dew, sum\_wind = 0, 0, 0

count = 0

for line in sys.stdin:

key, value = line.strip().split("\t")

temp, dew, wind = map(float, value.split())

if current\_key is None:

current\_key = key

if key == current\_key:

sum\_temp += temp

sum\_dew += dew

sum\_wind += wind

count += 1

else:

avg\_temp = sum\_temp / count

avg\_dew = sum\_dew / count

avg\_wind = sum\_wind / count

print(f"{current\_key}\t{avg\_temp} {avg\_dew} {avg\_wind}")

current\_key = key

sum\_temp, sum\_dew, sum\_wind = temp, dew, wind

count = 1

if current\_key is not None:

avg\_temp = sum\_temp / count

avg\_dew = sum\_dew / count

avg\_wind = sum\_wind / count

print(f"{current\_key}\t{avg\_temp} {avg\_dew} {avg\_wind}")

if \_\_name\_\_ == "\_\_main\_\_":

reduce1()

**Step 4: Prepare Hadoop Environment:**

|  |  |
| --- | --- |
| Start the Hadoop daemons and create a directory in HDFS to store your data. Run the following commands to store the data in the WeatherData Directory.  *start-all.cmd*  *cd C:/Hadoop/sbin*  *hdfs dfs -mkdir /WeatherData*  *hdfs dfs -put C:/Users/user/Documents/DataAnalytics2/input.txt /WeatherData*  *hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-2.7.0.jar ^*  *-input /user/input/sample\_weather.txt ^*  *-output /user/output ^*  *-mapper "python C:/ Users/user/Documents/DataAnalytics2/mapper.py" ^*  *-reducer "python C:/ Users/user/Documents/DataAnalytics2/reducer.py"*  **Step 5: Check Output:**   |  | | --- | | Check the output of the Word Count program in the specified HDFS output directory.  *hdfs dfs -cat /WeatherData/output/part-00000*  **OUTPUT:**      C:\Users\mukhi\AppData\Local\Packages\5319275A.51895FA4EA97F_cv1g1gvanyjgm\TempState\A305C17F85B99588CE35CB4E9CD6134F\WhatsApp Image 2024-08-27 at 01.47.03_036f0359.jpg | |

**RESULT:**

Thus, the program for weather dataset using Map Reduce has been executed successfully.