**EX.NO.10** **24/09/2024**

# IMPLEMENT A MAPREDUCE PROGRAM TO PROCESS A WEATHER DATASET

## AIM:

To implement a MapReduce python program to process a weather dataset in Hadoop.

## PROCEDURE:

1. Open command prompt as administrator and start the Hadoop by using the command:

```
start-all.cmd
```

2. Create a new directory in the Hadoop file systems using the command:

```
hadoop fs -mkdir /weather
```

3. Upload the input text file into the weather directory using the command:

```
hadoop fs -put

C:/Users/manoj/OneDrive/Documents/DataAnalytics/WeatherPrediction/sample_weather.txt /weather
```

4. Create the mapper and reducer files.

5. To execute the files with Hadoop streaming run the following command:

```
hadoop jar C:/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar ^ -file
C:/Users/manoj/Documents/DataAnalytics/WeatherPrediction/mapper.py
^ -file
C:/Users/manoj/Documents/DataAnalytics/WeatherPrediciton/reducer.py
^ -input /weather/sample_weather.txt ^ -output /weather/output ^ -mapper
"python mapper.py" ^ -reducer "python reducer.py"
```

## MAPPER.PY:

```
#!C:/ProgramData/chocolatey/bin/python3.exe

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__":
```

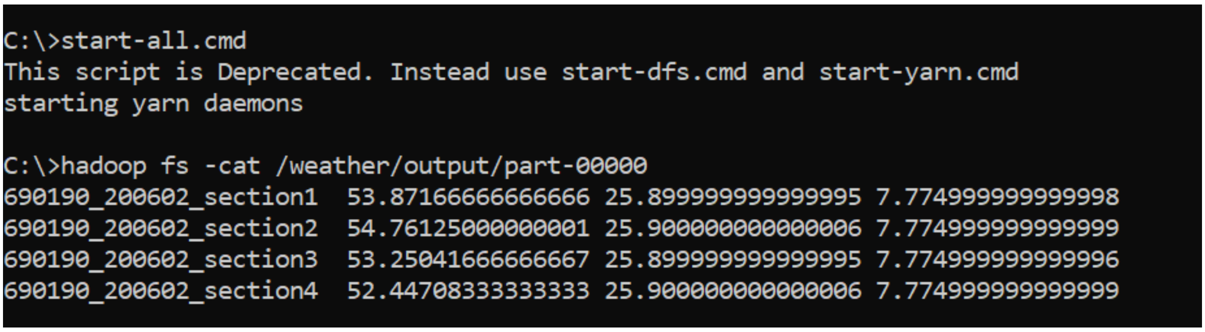
```
map1()
```

**REDUCER.PY:**

```
#!C:/ProgramData/chocolatey/bin/python3.exe
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()
```

**OUTPUT:**



```
C:\>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons

C:\>hadoop fs -cat /weather/output/part-00000
690190_200602_section1  53.87166666666666 25.899999999999995 7.774999999999998
690190_200602_section2  54.76125000000001 25.900000000000006 7.774999999999999
690190_200602_section3  53.25041666666667 25.899999999999995 7.774999999999996
690190_200602_section4  52.44708333333333 25.900000000000006 7.774999999999999
```

**RESULT:**

Thus the implementation of the MapReduce python program to process a weather dataset in Hadoop is executed successfully.