

Implement a MapReduce program to process a weather dataset**Steps:**

1. Open command prompt and run as administrator

Go to hadoop sbin directory

```
C:\Windows\system32>cd C:\Hadoop\sbin  
C:\Hadoop\sbin>_
```

Note:

1. Check hadoop/data/datanode and hadoop/data/namenode and if both folders are empty, type “hdfs namenode -format”.
2. Check python version with “python --version”.
3. Check “C:\Python39\” is added in Environment variables > System variables > Path, if not add your python path.
4. Check Environment variables > System variables > HADOOP_HOME is set as “C:\Hadoop”.

```
C:\Hadoop\sbin>echo %HADOOP_HOME%  
C:\Hadoop  
  
C:\Hadoop\sbin>python --version  
Python 3.11.4
```

2. Start Hadoop Services `start-dfs.cmd` `start-yarn.cmd`

```
C:\Windows\System32>start-all.cmd  
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd  
starting yarn daemons  
  
C:\Windows\System32>jps  
22208 NodeManager  
5808 ResourceManager  
19416 DataNode  
20888 Jps  
2492 NameNode  
  
C:\Windows\System32>_
```

3. Open the browser and go to the URL “localhost:9870”

Overview 'localhost:9000' (✓active)

Started:	Sun Aug 18 18:45:16 +0530 2024
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-a23ce25d-ee9d-4000-ac1f-044f436c4c8a
Block Pool ID:	BP-934656018-192.168.56.1-1723971050909

Summary

Security is off.
Safemode is off.
19 files and directories, 5 blocks (5 replicated blocks, 0 erasure coded block groups) = 24 total filesystem object(s).
Heap Memory used 74.86 MB of 193 MB Heap Memory. Max Heap Memory is 889 MB.
Non Heap Memory used 61.65 MB of 63.11 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	118.63 GB
----------------------	-----------

4. Create a Directory in HDFS

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

```
C:\hadoop\sbin>hadoop fs -mkdir /user/weather
mkdir: `/user/weather': File exists

C:\hadoop\sbin>
```

5. Copy the Input File to HDFS

```
hdfs dfs -put C:\Users\monid\OneDrive\Documents\DataAnalytics\sample_weather.txt /user/weather
```

```
C:\hadoop\sbin>hdfs dfs -put C:\Users\monid\OneDrive\Documents\DataAnalytics\sample_weather.txt /user/weather
put: `/user/weather/sample_weather.txt': File exists

C:\hadoop\sbin>
```

Note: mapper.py:

```
#!/usr/bin/env python 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:      #!
/usr/bin/env  python

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()

```

6. Run the Hadoop Streaming Job

```

hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.1.jar ^ -files
"/Users/monid/OneDrive/Documents/DataAnalytics/mapper2.py,/Users/monid/One

```

```
Drive/Documents/DataAnalytics/reducer2.py" ^ -input
/user/weather/sample_weather.txt ^ -output /user/output1 ^ -mapper "python
C:/Users/monid/OneDrive/Documents/DataAnalytics/mapper2.py" ^ -reducer
"python C:/Users/monid/OneDrive/Documents/DataAnalytics/reducer2.py "
```

```
C:\hadoop\sbin>hadoop jar C:\hadoop\share\hadoop\tools\lib\hadoop-streaming-3.3.1.jar ^ -files "/Users/monid/OneDrive
/Documents/DataAnalytics/mapper2.py,/Users/monid/OneDrive/Documents/DataAnalytics/reducer2.py" ^ -input /user/weather
/sample_weather.txt ^ -output /user/output1 ^ -mapper "python C:/Users/monid/OneDrive/Documents/DataAnalytics/mapp
er2.py" ^ -reducer "python C:/Users/monid/OneDrive/Documents/DataAnalytics/reducer2.py "
packageJobJar: [/C:/Users/monid/AppData/Local/Temp/hadoop-unjar5991909413546494244/] [] C:\Users\monid\AppData\Local\Tem
p\streamjob2531261441153576294.jar tmpDir=null
2024-09-14 08:23:08,511 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-14 08:23:08,736 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-14 08:23:08,915 ERROR streaming.StreamJob: Error Launching job : Output directory hdfs://localhost:9000/user/out
put1 already exists
Streaming Command Failed!
```

7. View the Output

```
hdfs dfs -cat /user/output1/part-00000
```

```
C:\hadoop\sbin>hdfs dfs -cat /weather/output/part-00000
cat: `/weather/output/part-00000': No such file or directory

C:\hadoop\sbin>hdfs dfs -cat /user/output1/part-00000
690190_200602_section1 53.87166666666666 25.899999999999995 7.774999999999999
690190_200602_section2 54.761250000000001 25.900000000000006 7.774999999999999
690190_200602_section3 53.250416666666667 25.899999999999995 7.774999999999999
690190_200602_section4 52.44708333333333 25.900000000000006 7.774999999999999

C:\hadoop\sbin>
```

8. Once the map reduce operations are performed successfully, the output will be present in the specified directory.

“/user/output1/part-00000”

The screenshot shows the Hadoop web interface with a modal window titled "File information - part-00000". The modal displays the following details:

- Block information: Block 0
- Block ID: 1073741904
- Block Pool ID: BP-399902486-192.168.228.238-1724038237583
- Generation Stamp: 1080
- Size: 312
- Availability:
 - Moni

Below the block information, the "File contents" section shows a table of data blocks:

Block ID	Block Size	Block Pool ID	Generation Stamp	Size	Availability
690190_200602_section1	53.87166666666666	25.899999999999995	7.7749999999999998		
690190_200602_section2	54.76125000000001	25.900000000000006	7.7749999999999999		
690190_200602_section3	53.25041666666667	25.899999999999995	7.7749999999999996		
690190_200602_section4	52.44708333333333	25.900000000000006	7.7749999999999999		

The background shows the "Browse Directory" page with the path "/user/output1" and a list of files.

9. Stop Hadoop Services `stop-dfs.cmd`

`stop-yarn.cmd`

```
C:\Hadoop\sbin>stop-dfs.cmd
SUCCESS: Sent termination signal to the process with PID 7964.
SUCCESS: Sent termination signal to the process with PID 13580.

C:\Hadoop\sbin>stop-yarn.cmd
stopping yarn daemons
SUCCESS: Sent termination signal to the process with PID 14412.
SUCCESS: Sent termination signal to the process with PID 7092.

INFO: No tasks running with the specified criteria.

C:\Hadoop\sbin>
```

RESULT:

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