**Ex No 3**

**Map Reduce program to process a weather dataset.**

**AIM:**

To implement MapReduce program to process a weather dataset.

**PROCEDURE:**

# Start Hadoop Services

Make sure you are in the sbin folder of Hadoop. Start the Hadoop services by running the following commands:

cd /usr/local/Cellar/hadoop/3.4.0/libexec/sbin

./start-dfs.sh

./start-yarn.sh

# Prepare Your Files

Create the necessary files (mapper.py, reducer.py, and sample\_weather.txt) on your local machine or the server where Hadoop is installed.

**Create sample\_weather.txt:**

You can create this file in the current directory:

nano sample\_weather.txt

Example data for sample\_weather.txt:

STN001 2023-09-10\_04 15.0 12.0 5.0

STN001 2023-09-10\_10 25.0 20.0 8.0

STN002 2023-09-10\_16 30.0 25.0 10.0 STN002 2023-09-10\_22 22.0 18.0 7.0

**Create mapper.py:** nano mapper.py

Content for mapper.py:

#!/usr/bin/python3 import sys def map1():

for line in sys.stdin: tokens = line.strip().split() if len(tokens) < 4: continue

station = tokens[0] date\_hour = tokens[1] temp = tokens[2] dew = tokens[3] wind = tokens[4] if len(tokens) > 4 else "999.9" 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("\_")] 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()

**Create reducer.py:** nano reducer.py

Content for 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()

# Upload Files to HDFS

Next, move your data file to HDFS so that it can be processed by the Hadoop MapReduce job.

**Create HDFS Directory:**

hdfs dfs -mkdir /WeatherData

**Upload the Input Data (sample\_weather.txt) to HDFS:** hdfs dfs -put sample\_weather.txt /WeatherData

**Verify the file upload:**

hdfs dfs -ls /WeatherData

You should see something like:

Found 1 items

-rw-r--r-- 3 user group 1234 2024-09-11 12:00

/WeatherData/sample\_weather.txt

# Run the MapReduce Job

Now that your input file is in HDFS and your mapper.py and reducer.py are ready, you can run the MapReduce job.

Ensure you are still in the directory where your mapper.py and reducer.py scripts are located.

**Run the Hadoop Streaming Job:**

hadoop jar

/usr/local/Cellar/hadoop/3.4.0/libexec/share/hadoop/tools/lib/hadoop-s treaming-3.4.0.jar \

-input /WeatherData/sample\_weather.txt \

-output /WeatherData/output \

-mapper "python3 mapper.py" \ -reducer "python3 reducer.py"

This command tells Hadoop to:

* Take the input from /WeatherData/sample\_weather.txt on HDFS.
* Use mapper.py as the mapper script.
* Use reducer.py as the reducer script.
* Output the results to /WeatherData/output.

**Note:**

Ensure that both mapper.py and reducer.py have executable permissions. If not, make them executable by running:

chmod +x mapper.py reducer.py

# View the Output

After the job completes, you can check the output that was stored in HDFS.

**List the output directory:**

hdfs dfs -ls /WeatherData/output

You should see something like:

Found 1 items

-rw-r--r-- 3 user group 456 2024-09-11 12:20

/WeatherData/output/part-00000

**View the output data:**

hdfs dfs -cat /WeatherData/output/part-00000

This will print the final result of the MapReduce job. You should see output similar to:

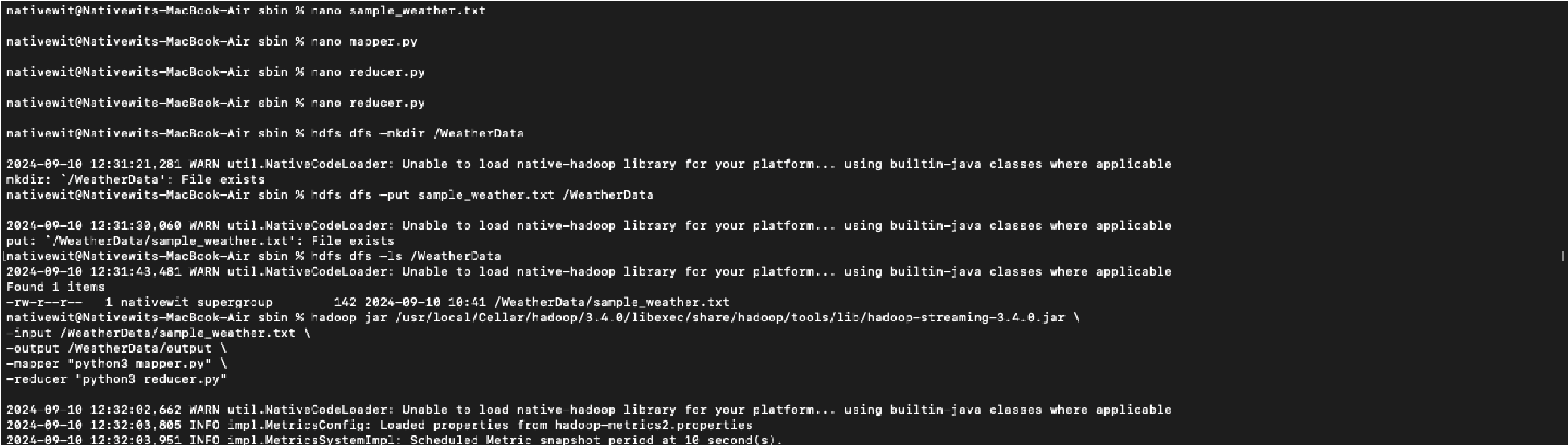
STN001\_2023-09-10\_section1 15.0 12.0 5.0

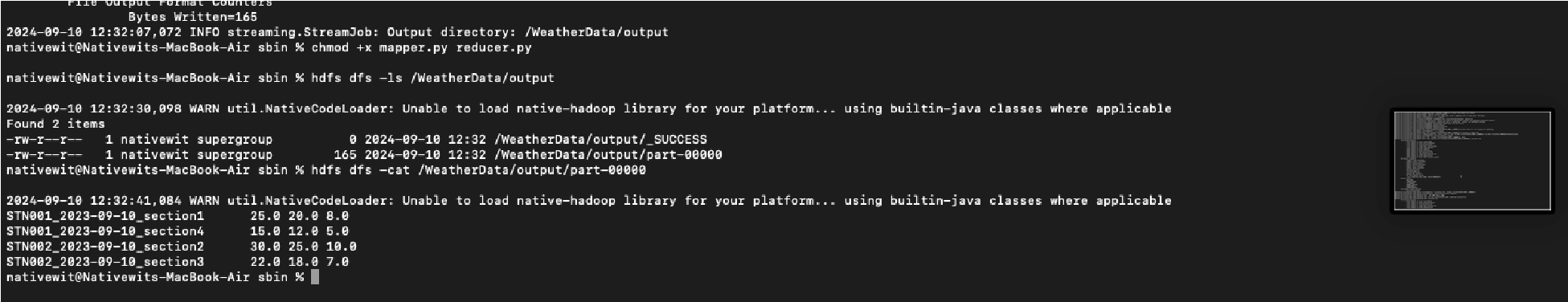
STN001\_2023-09-10\_section2 25.0 20.0 8.0

STN002\_2023-09-10\_section3 30.0 25.0 10.0

STN002\_2023-09-10\_section4 22.0 18.0 7.0

**OUTPUT:**





**RESULT:**

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