**Ex no: 10**

# Hadoop - Map Reduce

## IMPLEMENT THE MAX TEMPERATURE MAPREDUCE PROGRAM TO IDENTIFY THE YEAR WISE MAXIMUM TEMPERATURE FROM SENSOR DATA

**Aim:**

To implement the max temperature mapreduce program to identify the year wise maximum temperature from sensor data.

**Procedure:**

1. Set up the Hadoop environment and configure necessary system paths.
2. Format the namenode using the "hdfs namenode -format" command if needed.
3. Start Hadoop services such as HDFS and YARN using `start-dfs.cmd` and `start-yarn.cmd`.
4. Open the Hadoop user interface in a browser at `localhost:9870` to check the cluster.
5. Create a directory in HDFS for storing the input sensor data.
6. Upload the sensor data text file to the HDFS directory created in the previous step.
7. Write a Mapper function to read each line of the sensor data, extract the year, temperature, and quality fields, and output valid year-temperature pairs.
8. Write a Combiner function to aggregate the temperatures for the same year from the Mapper output locally.
9. Write a Reducer function to find the maximum temperature for each year from the data output by the Combiner.
10. Compile and prepare the Python Mapper, Combiner, and Reducer scripts for execution in Hadoop Streaming.
11. Submit the MapReduce job using the Hadoop Streaming API, specifying the input directory, output directory, and the Mapper, Combiner, and Reducer scripts.
12. Monitor the job's execution and check for errors in the log.
13. Once the job is completed, view the output by accessing the result file in the HDFS output directory.
14. Verify that the output contains the maximum temperature for each year from the sensor data.
15. Stop Hadoop services after verifying the results by using `stop-dfs.cmd` and `stop-yarn.cmd`.

**Program:**

mapper.py:

```
#!/usr/bin/env python3
import sys
for line in sys.stdin:
    line = line.strip()
    if not line:
        continue
    try:
 # Extract relevant fields from the raw data
        year = line[15:19] # Extract year from the line
        temp_str = line[90:92] # Extract temperature from the line
        quality = line[92:93] # Extract quality indicator
 # Check if the temperature is valid and the quality is acceptable
        if temp_str != "+9999" and quality in ['0', '1', '4', '5', '9']:
            temp = int(temp_str)
            print(f"{year}\t{temp}")
    except Exception as e:
        sys.stderr.write(f"Error processing line: {line}\nException: {str(e)}\n")
```

combiner.py:

```
#!/usr/bin/env python
import sys
from collections import defaultdict
current_year = None
temp_set = set()
for line in sys.stdin:
    line = line.strip()
```

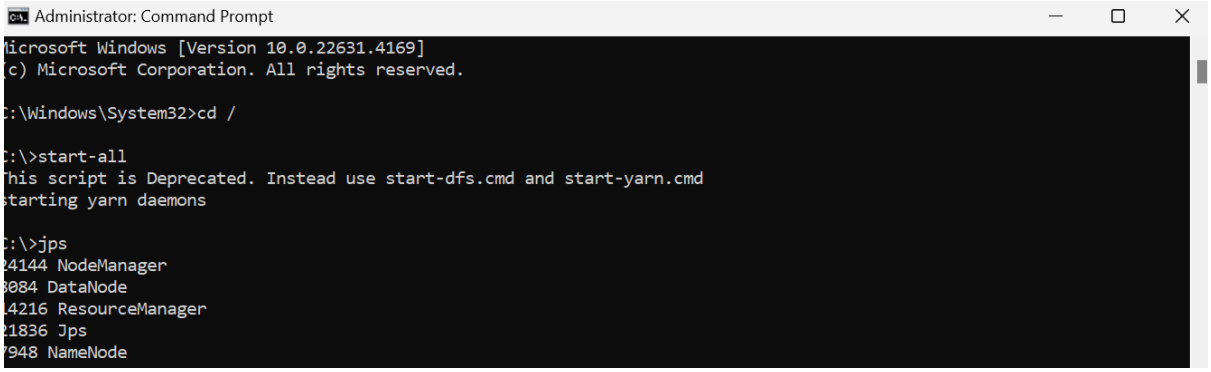
```
    if not line:
        continue
    try:
        year, temp_str = line.split('\t')
        temp = int(temp_str)
        if current_year == year:
            temp_set.add(temp)
        else:
            if current_year:
                # Print the year and the set of temperatures
                for t in temp_set:
                    print(f"{current_year}\t{t}")
            current_year = year
            temp_set = {temp}
    except Exception as e:
        sys.stderr.write(f"Error processing line: {line}\nException: {str(e)}\n")
# Output the set of temperatures for the last year
if current_year:
    for t in temp_set:
        print(f"{current_year}\t{t}")
```

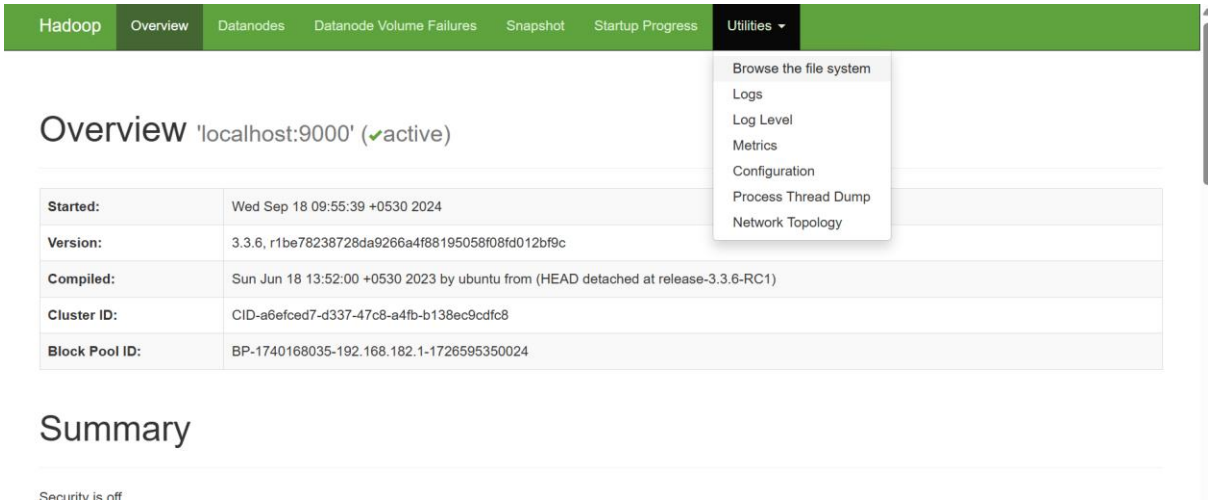
reducer.py:

```
#!/usr/bin/env python
import sys
current_year = None
max_temp = None
for line in sys.stdin:
    line = line.strip()
    if not line:
```

```
        continue
    try:
        year, temp_str = line.split('\t')
        temp = int(temp_str)
        if current_year == year:
            if temp > max_temp:
                max_temp = temp
            else:
                if current_year:
                    # Print the maximum temperature for the previous year
                    print(f"{current_year}\t{max_temp}")
                    current_year = year
                    max_temp = temp
    except Exception as e:
        sys.stderr.write(f"Error processing line: {line}\nException: {str(e)}\n")
# Output the maximum temperature for the last year
if current_year:
    print(f"{current_year}\t{max_temp}")
```

**Output:**



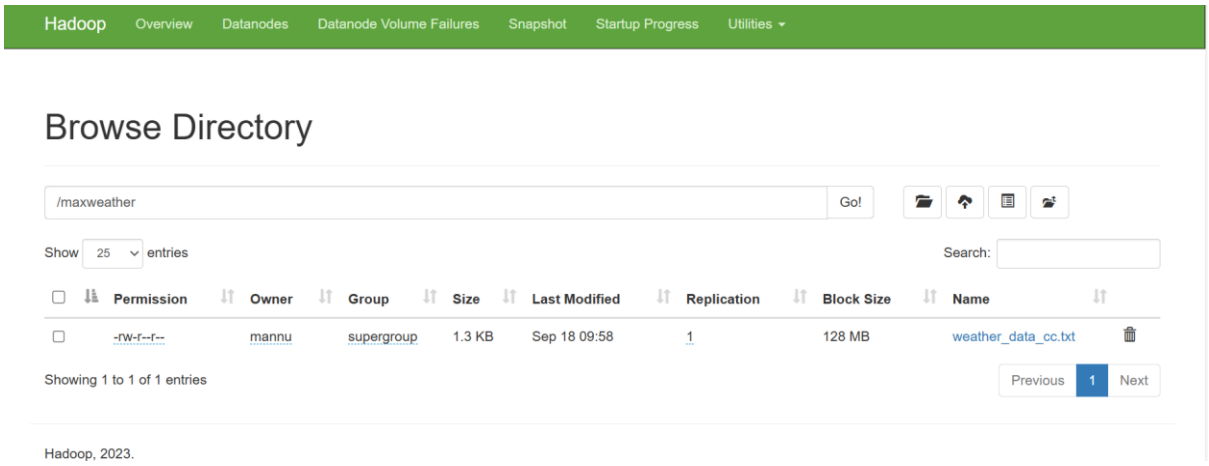



```
C:\>hdfs dfs -mkdir -p /maxweather

C:\>hdfs dfs -put C:\\Users\\mannu\\Documents\\Ex_10_cc\\weather_data_cc.txt /maxweather

C:\>hdfs dfs -ls /maxweather
Found 1 items
-rw-r--r--   1 mannu supergroup       1333 2024-09-18 09:58 /maxweather/weather_data_cc.txt

C:\>_
```




```
C:\>hdfs dfs -cat /maxweather/weather_data_cc.txt
0029029070999991902010720004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9-00331+99999098351ADDGF1029919999
99999999999999
0029029070999991955050520004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00251+99999098351ADDGF1029919999
99999999999
0029029070999992000123120004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00171+99999098351ADDGF1029919999
99999999999
0029029070999991984051520004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00381+99999098351ADDGF1029919999
99999999999
0029029070999991920062020004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9-00021+99999098351ADDGF1029919999
99999999999
0029029070999992018101520004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00411+99999098351ADDGF1029919999
99999999999
0029029070999991970072220004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00361+99999098351ADDGF1029919999
99999999999
0029029070999992003051120004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00391+99999098351ADDGF1029919999
99999999999
0029029070999991931083120004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00251+99999098351ADDGF1029919999
99999999999
0029029070999991999090912004+64333+023450FM-12+000599999V0202501N027819999999N0000001N9+00321+99999098351ADDGF1029919999
99999999999
```

```
C:\>hadoop jar %HADOOP_HOME%\share\hadoop\tools\lib\hadoop-streaming-*.jar ^
More? -input /maxweather/weather_data_cc.txt ^
More? -output /maxweather/output ^
More? -mapper "python C:\Users\mannu\Documents\Ex_10_cc\mapper.py" ^
More? -combiner "python C:\Users\mannu\Documents\Ex_10_cc\combiner.py" ^
More? -reducer "python C:\Users\mannu\Documents\Ex_10_cc\reducer.py" ^
More? -file C:\Users\mannu\Documents\Ex_10_cc\mapper.py ^
More? -file C:\Users\mannu\Documents\Ex_10_cc\combiner.py ^
More? -file C:\Users\mannu\Documents\Ex_10_cc\reducer.py
2024-09-18 10:07:51,902 WARN streaming.StreamJob: -file option is deprecated, please use generic option -files instead.
packageJobJar: [C:\Users\mannu\Documents\Ex_10_cc\mapper.py, C:\Users\mannu\Documents\Ex_10_cc\combiner.py, C:\Users\ma
nu\Documents\Ex_10_cc\reducer.py, /C:/Users/mannu/AppData/Local/Temp/hadoop-unjar1389565929964994158/] [] C:\Users\mann
\AppData\Local\Temp\streamjob8644228401151669202.jar tmpDir=null
2024-09-18 10:07:52,711 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-09-18 10:07:52,863 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
```

## File information - part-00000

Download  Head the file (first 32K)  Tail the file (last 32K)

**Block information --** Block 0

Block ID: 1073741835

Block Pool ID: BP-2025921802-192.168.182.1-1726641338029

Generation Stamp: 1011

Size: 79

Availability:

- Shreeya

### File contents

```
1902    33
1920    2
1931    25
1955    25
1970    36
1984    38
1999    32
2000    17
```

```
C:\>hdfs dfs -cat /maxweather/output/part-00000
1902    33
1920    2
1931    25
1955    25
1970    36
1984    38
1999    32
2000    17
2003    39
2018    41
```

**Result:**

Thus the implementation of max temperature mapreduce program to identify the year wise maximum temperature from sensor data has been executed successfully.