Map Reduce Project - BAN632

1) Python files (mapper and reducer) ← in project submission

2) The commands for executing the Python application in Hadoop are as follows:

put data into correct directory

hdfs dfs -copyFromLocal CourseProjectData/ hdfs://msba-hadoop-name:9000/user/student29/

run mapper

spark-submit NcdcRecordMapper.py CourseProjectData/*.gz

run reducer

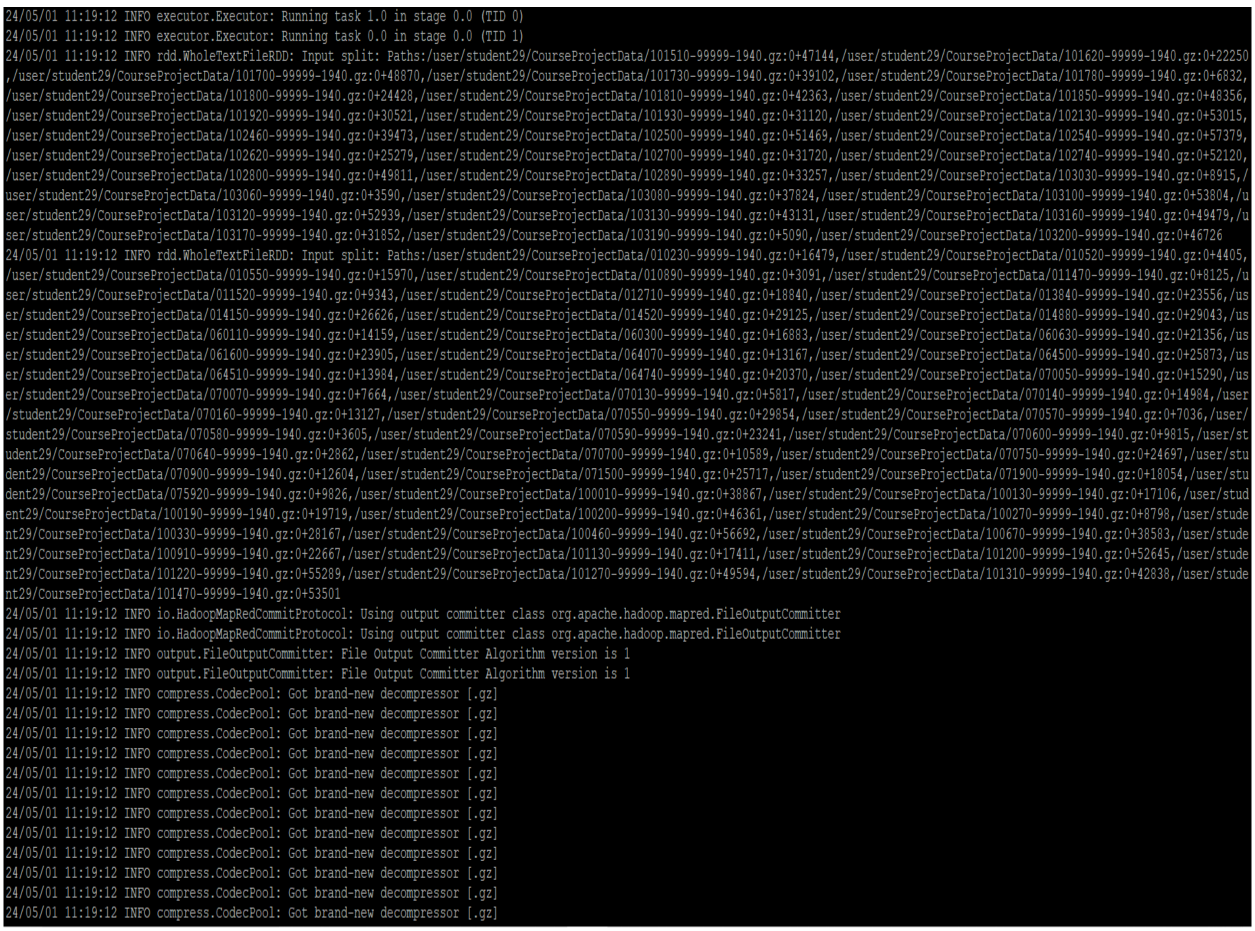
spark-submit --master local[*] NcdcRecordReducer.py

3) The text file including Year and Temperature data created by you ← in project submission

4) The screenshot of the text file being created

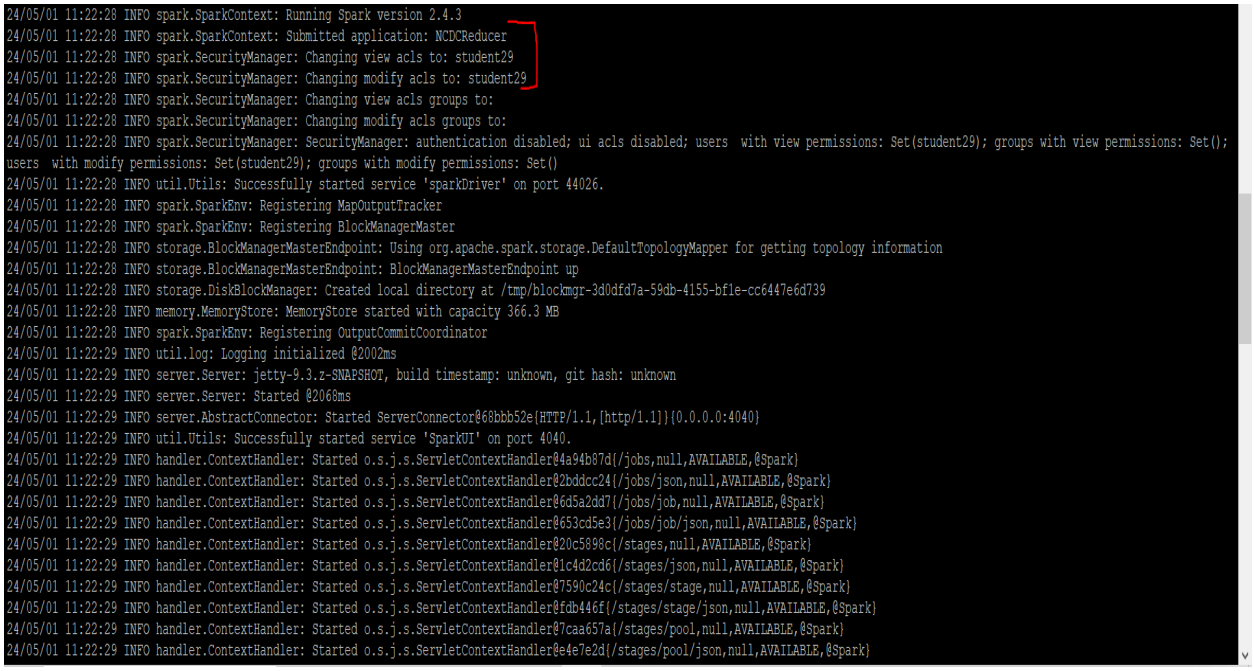
The file is created when running NcdcRecordReducer.py.

First Screenshot: mapper



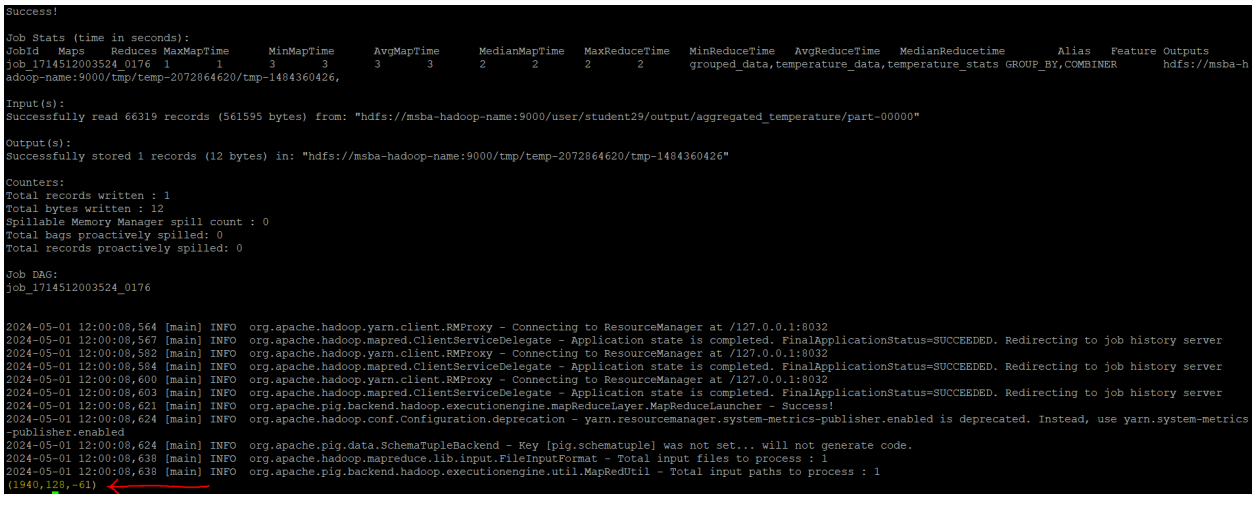
```
24/05/01 11:19:12 INFO executor.Executor: Running task 1.0 in stage 0.0 (TID 0)
24/05/01 11:19:12 INFO executor.Executor: Running task 0.0 in stage 0.0 (TID 1)
24/05/01 11:19:12 INFO rdd.WholeTextFileRDD: Input split: Paths:/user/student29/CourseProjectData/101510-99999-1940.gz:0+47144,/user/student29/CourseProjectData/101620-99999-1940.gz:0+22250
,/user/student29/CourseProjectData/101700-99999-1940.gz:0+48870,/user/student29/CourseProjectData/101730-99999-1940.gz:0+39102,/user/student29/CourseProjectData/101780-99999-1940.gz:0+6832,
/user/student29/CourseProjectData/101800-99999-1940.gz:0+24428,/user/student29/CourseProjectData/101810-99999-1940.gz:0+42363,/user/student29/CourseProjectData/101850-99999-1940.gz:0+48356,
/user/student29/CourseProjectData/101920-99999-1940.gz:0+30521,/user/student29/CourseProjectData/101930-99999-1940.gz:0+31120,/user/student29/CourseProjectData/102130-99999-1940.gz:0+53015,
/user/student29/CourseProjectData/102460-99999-1940.gz:0+39473,/user/student29/CourseProjectData/102500-99999-1940.gz:0+51469,/user/student29/CourseProjectData/102540-99999-1940.gz:0+57379,
/user/student29/CourseProjectData/102620-99999-1940.gz:0+25279,/user/student29/CourseProjectData/102700-99999-1940.gz:0+31720,/user/student29/CourseProjectData/102740-99999-1940.gz:0+52120,
/user/student29/CourseProjectData/102800-99999-1940.gz:0+49811,/user/student29/CourseProjectData/102890-99999-1940.gz:0+33257,/user/student29/CourseProjectData/103030-99999-1940.gz:0+8915,/
user/student29/CourseProjectData/103060-99999-1940.gz:0+3590,/user/student29/CourseProjectData/103080-99999-1940.gz:0+37824,/user/student29/CourseProjectData/103100-99999-1940.gz:0+53804,/u
ser/student29/CourseProjectData/103120-99999-1940.gz:0+52939,/user/student29/CourseProjectData/103130-99999-1940.gz:0+43131,/user/student29/CourseProjectData/103160-99999-1940.gz:0+49479,/u
ser/student29/CourseProjectData/103170-99999-1940.gz:0+31852,/user/student29/CourseProjectData/103190-99999-1940.gz:0+5090,/user/student29/CourseProjectData/103200-99999-1940.gz:0+46726
24/05/01 11:19:12 INFO rdd.WholeTextFileRDD: Input split: Paths:/user/student29/CourseProjectData/010230-99999-1940.gz:0+16479,/user/student29/CourseProjectData/010520-99999-1940.gz:0+4405,
/user/student29/CourseProjectData/010550-99999-1940.gz:0+15970,/user/student29/CourseProjectData/010890-99999-1940.gz:0+3091,/user/student29/CourseProjectData/011470-99999-1940.gz:0+8125,/u
ser/student29/CourseProjectData/011520-99999-1940.gz:0+9343,/user/student29/CourseProjectData/012710-99999-1940.gz:0+18840,/user/student29/CourseProjectData/013840-99999-1940.gz:0+23556,/us
er/student29/CourseProjectData/014150-99999-1940.gz:0+26626,/user/student29/CourseProjectData/014520-99999-1940.gz:0+29125,/user/student29/CourseProjectData/014880-99999-1940.gz:0+29043,/us
er/student29/CourseProjectData/060110-99999-1940.gz:0+14159,/user/student29/CourseProjectData/060300-99999-1940.gz:0+16883,/user/student29/CourseProjectData/060630-99999-1940.gz:0+21356,/us
er/student29/CourseProjectData/061600-99999-1940.gz:0+23905,/user/student29/CourseProjectData/064070-99999-1940.gz:0+13167,/user/student29/CourseProjectData/064500-99999-1940.gz:0+25873,/us
er/student29/CourseProjectData/064510-99999-1940.gz:0+13984,/user/student29/CourseProjectData/064740-99999-1940.gz:0+20370,/user/student29/CourseProjectData/070050-99999-1940.gz:0+15290,/us
er/student29/CourseProjectData/070070-99999-1940.gz:0+7664,/user/student29/CourseProjectData/070130-99999-1940.gz:0+5817,/user/student29/CourseProjectData/070140-99999-1940.gz:0+14984,/user
/student29/CourseProjectData/070160-99999-1940.gz:0+13127,/user/student29/CourseProjectData/070550-99999-1940.gz:0+29854,/user/student29/CourseProjectData/070570-99999-1940.gz:0+7036,/user/
student29/CourseProjectData/070580-99999-1940.gz:0+3605,/user/student29/CourseProjectData/070590-99999-1940.gz:0+23241,/user/student29/CourseProjectData/070600-99999-1940.gz:0+9815,/user/st
udent29/CourseProjectData/070640-99999-1940.gz:0+2862,/user/student29/CourseProjectData/070700-99999-1940.gz:0+10589,/user/student29/CourseProjectData/070750-99999-1940.gz:0+24697,/user/stu
dent29/CourseProjectData/070900-99999-1940.gz:0+12604,/user/student29/CourseProjectData/071500-99999-1940.gz:0+25717,/user/student29/CourseProjectData/071900-99999-1940.gz:0+18054,/user/stu
dent29/CourseProjectData/075920-99999-1940.gz:0+9826,/user/student29/CourseProjectData/100010-99999-1940.gz:0+38867,/user/student29/CourseProjectData/100130-99999-1940.gz:0+17106,/user/stud
ent29/CourseProjectData/100190-99999-1940.gz:0+19719,/user/student29/CourseProjectData/100200-99999-1940.gz:0+46361,/user/student29/CourseProjectData/100270-99999-1940.gz:0+8798,/user/stude
nt29/CourseProjectData/100330-99999-1940.gz:0+28167,/user/student29/CourseProjectData/100460-99999-1940.gz:0+56692,/user/student29/CourseProjectData/100670-99999-1940.gz:0+38583,/user/stude
nt29/CourseProjectData/100910-99999-1940.gz:0+22667,/user/student29/CourseProjectData/101130-99999-1940.gz:0+17411,/user/student29/CourseProjectData/101200-99999-1940.gz:0+52645,/user/stude
nt29/CourseProjectData/101220-99999-1940.gz:0+55289,/user/student29/CourseProjectData/101270-99999-1940.gz:0+49594,/user/student29/CourseProjectData/101310-99999-1940.gz:0+42838,/user/stude
nt29/CourseProjectData/101470-99999-1940.gz:0+53501
24/05/01 11:19:12 INFO io.HadoopMapRedCommitProtocol: Using output committer class org.apache.hadoop.mapred.FileOutputCommitter
24/05/01 11:19:12 INFO io.HadoopMapRedCommitProtocol: Using output committer class org.apache.hadoop.mapred.FileOutputCommitter
24/05/01 11:19:12 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
24/05/01 11:19:12 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 1
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
24/05/01 11:19:12 INFO compress.CodecPool: Got brand-new decompressor [.gz]
```

Second Screenshot: reducer (red markup shows reducer works), this is when the yr_temp_data.txt file is created.



5) the screenshot of the final Pig output showing the year and the highest and lowest temperatures

```
# load data from text file
temperature_data = LOAD 'hdfs://msba-hadoop-name:9000/user/student29/output/aggregated_temperature/yr_temp_data.txt' USING PigStorage(',') AS (year:int, temperature:int);
# group data by year
grouped_data = GROUP temperature_data BY year;
# extract highest and lowest temperatures
temperature_stats = FOREACH grouped_data { max_temp = MAX(temperature_data.temperature); min_temp = MIN(temperature_data.temperature); GENERATE group AS year, max_temp AS max_temperature, min_temp AS min_temperature; }
# MIN MAX TEMP OUTPUT AT THE BOTTOM OF SCREENSHOT
STORE temperature_stats INTO 'output/temperature_stats' USING PigStorage(',');
DUMP temperature_stats;
```



6) the screenshot of the final Hive output showing the average year and temperature.

```
# typical hive setup
ls -l | grep meta
mv metastore_db metastore_db.old
schematool -dbType derby -initSchema
hive
set hive.metastore.warehouse.dir;
# create table in a manner that I can load the text file into it
CREATE TABLE IF NOT EXISTS temperature_data (year INT, temperature INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n' STORED AS TEXTFILE;
# load text file into table
LOAD DATA INPATH 'hdfs://msba-hadoop-name:9000/user/student29/output/aggregated_temperature/yr_temp_data.txt' INTO TABLE temperature_data;
# query for average temperature
SELECT year, AVG(temperature) AS avg_temperature FROM temperature_data WHERE year IS NOT NULL AND temperature IS NOT NULL GROUP BY year;
```