**The Course Project**

Name: Abhishek Malla

NET-ID: VB7520

**Plagiarism and cheating will not be tolerated for this individual and independent project assignment. All submissions will undergo a thorough plagiarism check. Sharing your work with other students, copying others' work, or using AI tools such as ChatGPT is strictly prohibited. Any instances of plagiarism or cheating will result in a grade of 0 for the entire project submission.**

The course project consists of four parts. Parts 1 and 2 require the use of the project data zip file, while parts 3 and 4 require the use of a text file named “skyceiling.txt”.

**Part 1:**

**The first part is to develop a Mapper and Reducer application to calculate the *range* (the difference between max and min values) of *sky ceiling height* (meters) for *each observation month* from NCDC records (note: 99999 indicates missing value, and [01459] indicate good quality value).**

gunzip -r /home/student3/ProjectData/

****

chmod +x height\_map\_range.py

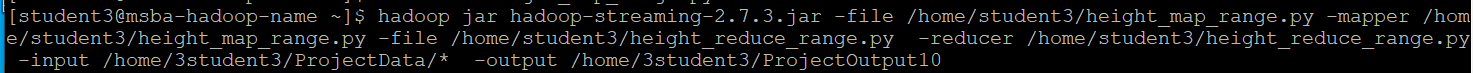
chmod +x height\_reduce\_range.py

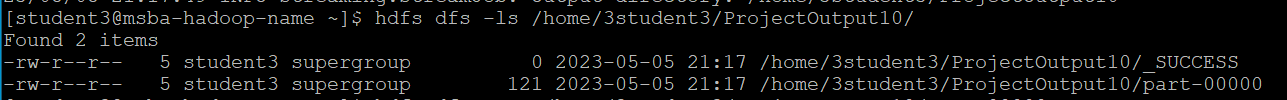
****

hdfs dfs -copyFromLocal ProjectData/\* /home/3student3

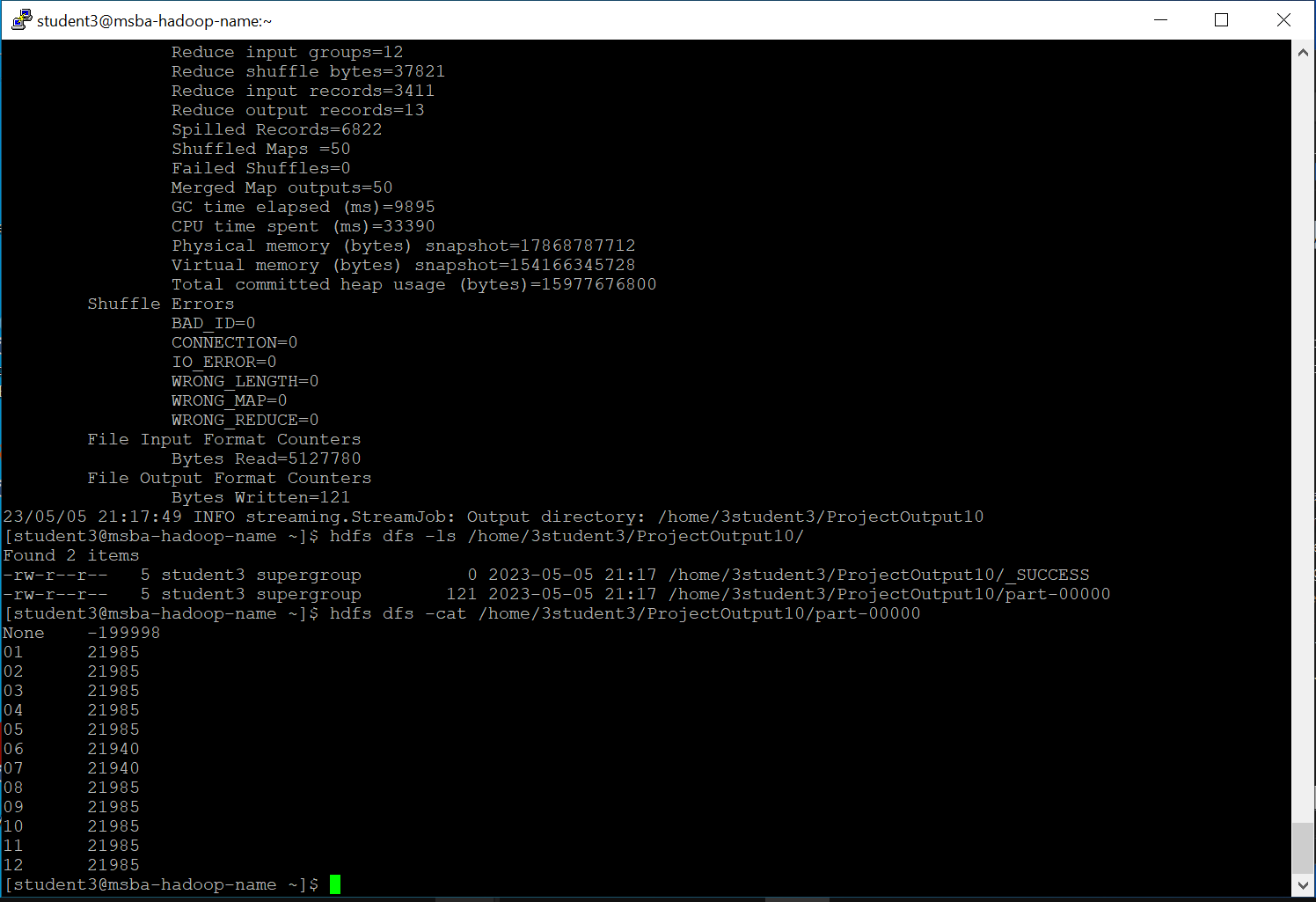


hadoop jar hadoop-streaming-2.7.3.jar -file /home/student3/height\_map\_range.py -mapper /home/student3/height\_map\_range.py -file /home/student3/height\_reduce\_range.py -reducer /home/student3/height\_reduce\_range.py -input /home/3student3/ProjectData/\* -output /home/3student3/ProjectOutput10



hdfs dfs -ls /home/3student3/ProjectOutput10/

hdfs dfs -cat /home/3student3/ProjectOutput10/part-00000



**Part 2:**

**The second part is to develop a python application that can be implemented in PySpark to calculate the *average visibility distance* (meters) for *each USAF weather station ID* from NCDC records (note: 999999 indicates missing value, and [01459] indicate good quality value).**

[hadoop@ec2-54-147-30-89.compute-1.amazonaws.com](mailto:hadoop@ec2-54-147-30-89.compute-1.amazonaws.com)

**hdfs dfs -mkdir /user/Hadoop/input/**



hdfs dfs -copyFromLocal ProjectData/\* /user/hadoop/input/



spark-submit –master local avg\_distance.py



Text

Description automatically generated

hdfs dfs -ls /user/hadoop/projectoutputspark2/

A picture containing text

Description automatically generated

hdfs dfs -cat /user/hadoop/projectoutputspark2/part-00\*

Graphical user interface, text

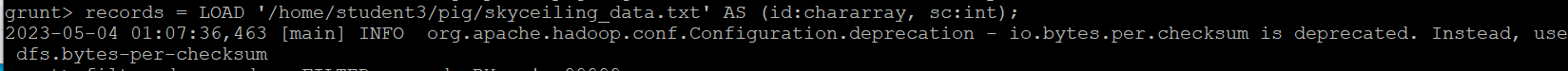
Description automatically generated

**Part 3:**

**The third part is to load the text file into Pig and get the range of sky ceiling height for each USAF weather station ID.**

pig -x local



records = LOAD '/home/student3/pig/skyceiling\_data.txt' AS (id:chararray, sc:int); 

filtered\_records = FILTER records BY sc!= 99999;

grouped\_records = GROUP filtered\_records BY id;

A picture containing text, orange, dark

Description automatically generated

range\_result = FOREACH grouped\_records {

max\_value=MAX(filtered\_records.sc);

min\_value=MIN(filtered\_records.sc);

GENERATE group AS id,(max\_value-min\_value) AS range;}

Text

Description automatically generated

DUMP range\_result;



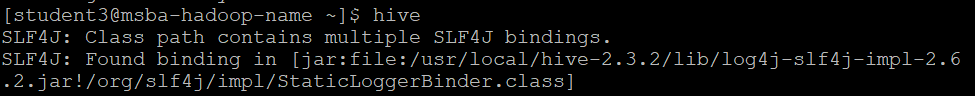
Text

Description automatically generated

**Part 4:**

**The fourth part is to load the text file into Hive and get the average sky ceiling height for each USAF weather station ID.**

hive



DROP TABLE IF EXISTS records3;

Graphical user interface, text

Description automatically generated

CREATE TABLE records3 (id STRING, sc INT)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t';

Text

Description automatically generated

LOAD DATA LOCAL INPATH '/home/student3/pig/skyceiling\_data.txt'

OVERWRITE INTO TABLE records3;Text

Description automatically generated

SELECT id, avg(sc)

FROM records3

WHERE sc!= 99999

GROUP BY id;

Text

Description automatically generated

You need to turn in:

**1)** Part 1:

a. *if you are using JAVA to develop the Mapper and Reducer applications:* the three java files (mapper, reducer and main);

b. *if you are using Hadoop streaming jar and developing two python programs (mapper python file and reducer python file)*: the two python files (mapper and reducer);

c. *if you are using mrjob library and developing one python program with two functions:* the python file (with the mapper and reducer functions);

2) Part 2: the python program you developed;

**3)** the commands from converting java files into a Jar file to running the Jar file in Hadoop, or the commands to execute the python files in Hadoop and in Spark;

**4)** the step by step commands and screenshots of solutions from all the parts;

The original dataset for this project is available on Blackboard.