**The Course Project**

The course project includes 3 parts. The first part is to develop a Mapper and Reducer application to retrieve Year and Temperature from original NCDC records (i.e., the dataset we are using for this class) and then write the Year and Temperature data into a text file. The second part is to load the text file into Pig and get the highest and lowest temperatures for each year. The third part is to load the text file into Hive and get the average temperature for each year.

You need to turn in 1) the three java files (mapper, reducer and main), 2) the commands from converting them into a Jar file to running the Jar file in Hadoop, 3) the text file including Year and Temperature data created by you, 4) the screenshot of the text file being created, 5) the screenshot of the final Pig output showing the year and the highest and lowest temperatures, and 6) the screenshot of the final Hive output showing the year and average temperature.

The original dataset for this project is available on Blackboard.

**Solution-**

**Part-1**

**Develop a Mapper and Reducer application to retrieve Year and Temperature from original NCDC records (i.e., the dataset we are using for this class) and then write the Year and Temperature data into a text file**

**CSUEB HADOOP URL- 134.154.190.204**

1. Command for creating folder on HDFS:

hdfs dfs -mkdir /home/sb\_project1

1. Command for copying a file from local disk to HDFS:

hdfs dfs -copyFromLocal ./Input\_Data2 /home/sb\_project1/

1. Command for compiling Java file:

javac -classpath /home/student23/hadoop-common-2.6.1.jar:/home/student23/hadoop-mapreduce-client-core-2.6.1.jar:/home/student23/commons-cli-2.0.jar -d . ListTemperatureWithCompression.java ListTemperatureMapper.java ListTemperatureReducer.java

1. Command for creating jar file:

jar -cvf list-temperature.jar ListTemperatureWithCompression.class ListTemperatureMapper.class ListTemperatureReducer.class

1. Command to set a hadoop class path:

export HADOOP\_CLASSPATH=/home/sb\_project1

1. Command for running the file in HDFS:

hadoop jar list-temperature.jar ListTemperatureWithCompression /home/sb\_project1/Input\_Data2/\*.gz /home/sb\_project1/sb\_project1\_output1

1. Command to list the output in HDFS **(output of map-reduce is compressed file)**:

hdfs dfs -ls /home/sb\_project1/sb\_project1\_output1/

1. Command for compiling Java file:

javac -classpath /home/student23/hadoop-common-2.6.1.jar:/home/student23/hadoop-mapreduce-client-core-2.6.1.jar:/home/student23/commons-cli-2.0.jar -d . FileDecompressor.java

1. Command for creating jar file:

jar -cvf file-decompressor.jar FileDecompressor.class

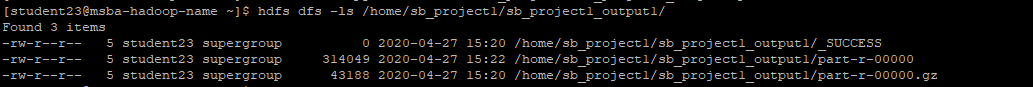
1. Command for running the file in HDFS **(Executing file decompressor program)**:

hadoop jar file-decompressor.jar FileDecompressor /home/sb\_project1/sb\_project1\_output1/part-r-00000.gz

1. Command to list the files in HDFS:

hdfs dfs -ls /home/sb\_project1/sb\_project1\_output1/

**The screenshot of a text file created including Year and Temperature data-**



1. Command to display the data from the file in HDFS:

hdfs dfs -text /home/sb\_project1/sb\_project1\_output/part-r-00000

1. Command for copying a file to local disk from HDFS:

hdfs dfs -copyToLocal /home/sb\_project1/sb\_project1\_output1/part-r-00000 /home/student23/Reducer-Output/

**Part-2**

**Load the text file into Pig and get the highest and lowest temperatures for each year**

1. Command to open the pig environment:

pig -x local

1. Command to load the input file into the pig environment:

records = LOAD '/home/student23/Reducer-Output/part-r-00000' AS (year:chararray,temperature:int);

1. Command to filter the valid records:

filtered\_records = FILTER records BY temperature != 9999;

1. Command to group the records by each year:

grouped\_records = GROUP filtered\_records BY year;

1. Command to retrieve the maximum temperature for each year:

max\_temp = FOREACH grouped\_records GENERATE group, MAX(filtered\_records.temperature);

1. Command to retrieve the minimum temperature for each year:

min\_temp = FOREACH grouped\_records GENERATE group, MIN(filtered\_records.temperature);

1. Command to join two results from step 5 and step 6 using common column ‘year’:

max\_min\_temp = JOIN max\_temp by $0, min\_temp by $0;

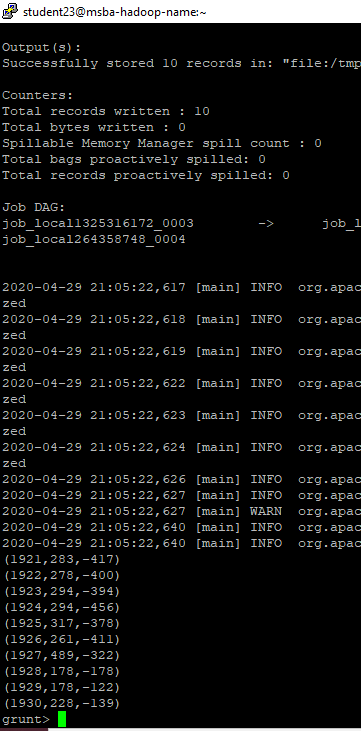
1. Command to assign required columns from step 7 in the Final\_records (year,max\_temp,min\_temp):

Final\_records = FOREACH max\_min\_temp GENERATE $0,$1,$3;

1. Command to display the result:

DUMP Final\_records;

**The screenshot of the final Pig output showing the year and the highest and lowest temperatures-**



**Part-3**

**Load the text file into Hive and get the average temperature for each year**

1. Commands to activate hive on CSUEB Hadoop:
2. ls -l | grep meta
3. mv metastore\_db metastore\_db.old
4. schematool -dbType derby -initSchema
5. hive
6. set hive.metastore.warehouse.dir;
7. Command to drop a table if already exists:

DROP TABLE IF EXISTS temp\_table;

1. Command to create a schema for the table:

CREATE TABLE temp\_table (year STRING, temperature INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘\t’;

1. Command to load the data into the created table in the step 3:

LOAD DATA LOCAL INPATH '/home/student23/Reducer-Output/part-r-00000' OVERWRITE INTO TABLE temp\_table;

1. Command to query the table to fetch/display the required data:

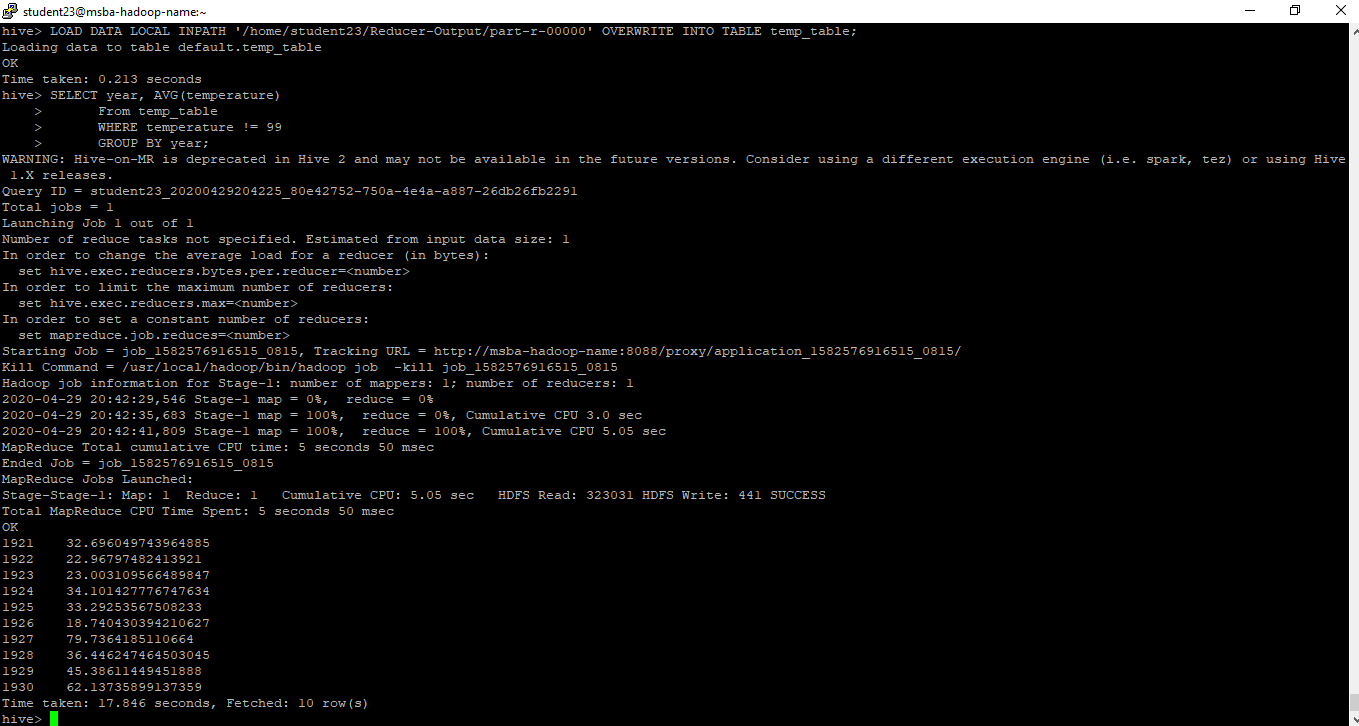
SELECT year, AVG(temperature)

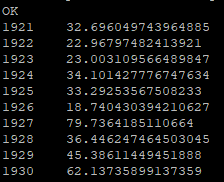
From temp\_table

WHERE temperature != 99

GROUP BY year; `

**The screenshot of the final Hive output showing the year and average temperature-**



**Zoom-in version:-0**