CSE 5331: Project 3 – Map/Reduce Implementation

Implemented by: Sanika Vijay Sudhalkar

UTA ID: 1001148173

1. Approach:

For this project, the students were required to analyze weather data provided, and compute average values of temperature, dew point and wind speed for each month, for the provided data. Each of the values were to be divided into four parts, based on time of the day.

The approach for this implementation was as follows:

Mapper 1 – The first mapper is responsible for identifying the necessary fields, i.e. station ID, hourmoda_hr (the date and hour the temperature was recorded at), Temperature, Dew Point and Wind speed. The first mapper also analyses the hour at which the temperature was taken, and assigns sections based on the following calculations:

- Section 1 – 5am (including) to 11am (i.e. hours from 5 to 10 were considered)
- Section 2 – 11am (including) to 5pm
- Section 3 – 5pm (including) to 11pm
- Section 4 – 11pm (including) to 5am

Mapper 1 Output Key – Station ID:Month:Section

Mapper 1 Output Value – Temperature, dew point and wind speed

Reducer 1 – This reducer was used to iterate over the set of values corresponding to the keys given by Mapper 1, and compute average values of temperature, dew point and wind speed. Separate average values were computed for each section of the day.

Reducer 1 Output Key – StationId:Month:Section

Reducer 1 Output Value – averageTemp|averageDewPoint|averageWindSpeed

Mapper 2 – This mapper was used to split the input data (StationId:Month:Section averageTemp|averageDewPoint|averageWindSpeed), to obtain the get the final key, and add the section attribute to the value. The section and the average values are separated by the delimiter ":", so that when the reducer splits these values, all the average values stay in the same string.

Mapper 2 Output Key – StationId:Month

Mapper 2 Output Value – Section:averageTemp|averageDewPoint|averageWindSpeed

Reducer 2 – This reducer iterates over all the sets of data, identifies the section associated with each data value, and decides the order of the data based on the section value.

Reducer 2 Output Key – "Station ID: <value of station ID> Month: <value of Month>

Reducer 2 Output Value – s1avgTemp|s1avgDP|s1avgWS| s2avgTemp|s2avgDP|s2avgWS| s3avgTemp|s3avgDP|s3avgWS| s4avgTemp|s4avgDP|s4avgWS

2. Performance Measures:

The performance of this application was measured using the data collected at the end of both the map and reduce jobs.

The Hardware configurations used were: Intel i5 quad core processor, 8gb ram, 1tb hdd

The execution details for both jobs are as follows:

For Job 1:

Total time spent by all maps in occupied slots (ms)=474489
Total time spent by all reduces in occupied slots (ms)=57646
GC time elapsed (ms)=10259
CPU time spent (ms)=204930

For Job 2:

Total time spent by all maps in occupied slots (ms)=3372
Total time spent by all reduces in occupied slots (ms)=3273
GC time elapsed (ms)=62
CPU time spent (ms)=3520

3. File Descriptions:

The JAVA file created for this project is called WeatherAnalysis.java

This file contains two inner Mapper classes (WeatherAnalysisMapper1 and WeatherAnalysisMapper2), and two inner Reducer classes (WeatherAnalysisReducer1 and WeatherAnalysisReducer2). The functionality of all these four classes has been described in section 1.

A JAR file containing the .java file as well as the .class files for all the classes has also been provided.

This file takes three command line arguments, as follows:

Argument 1 – HDFS input folder

Argument 2 – HDFS output folder for Reducer1

Argument 3 – HDFS output folder for Reducer2

4. Logical Errors and How they were handled:

- The errors majorly encountered for this project were pertaining to installation and setup of Hadoop with Ubuntu. The Hadoop Datanode and Namenode were not starting initially, due to some permission issues. That problem was fixed.
- The second problem was that once the Datanode and Namenode started, the files could not be uploaded to HDFS, due to shortage of storage. The entire Ubuntu installation needed to be done again, to allocate more hard disk space.
- While writing the actual JAVA implementation, the "Key" argument of WeatherAnalysisMapper2 was used mistakenly. Changing the type of "Key" to Text gave the ClassCastException. I later realized that the data input to the Mapper actually contains entire lines, as per the input file. The "Key" value does not need to be taken into account.
- There were some issues in correctly splitting values from the given data. These errors were debugged and fixed.

5. Output File Sample:

Reducer 1 output:

Here, keys correspond to StationId:month:day and values correspond to avgTemp|avgDewPoint|avgWindSpeed

690190:01:1	49.8413|26.6933|8.5262
690190:01:2	48.6108|26.6933|8.5262
690190:01:3	49.5121|26.6933|8.5262
690190:01:4	47.854|26.6933|8.5262

Reducer 2 output:

Here, the values correspond to:
s1avgTemp|s1avgDP|s1avgWS|s2avgTemp|s2avgDP|s2avgWS|s3avgTemp|s3avgDP|s3avgWS|s4avgTemp|s4avgDP|s4avgWS

StationID: 690190 Month: 01	49.8413|26.6933|8.5262|48.6108|26.6933|8.5262|49.5121|26.6933|8.5262|47.854|26.6933|8.5262
StationID: 690190 Month: 02	51.4204|28.6819|9.6795|51.5365|28.6819|9.6795|51.775|28.6819|9.6795|52.06|28.6819|9.6795
StationID: 690190 Month: 03	61.0637|39.4985|10.7381|61.0008|39.4985|10.7381|61.4367|39.4985|10.7381|60.9788|39.4985|10.7381