Max Min Average Temperature by States - MapReduce Project/CS236

Andres Calderon - SID:861243796

December 8, 2015

1 Max Min Average Temperature by States

1.1 Username and node number

My username is acald013 and I used the z7 node.

1.2 Overall description

My main idea was to combine the required data in one file where station id is the key. Then a reduce job put the station id together and we can emulate an inner join. After that, different map and reduce jobs organize the data and compute aggregates (average, max and min). Finally, a mapper will format and sort the final result.

1.3 Description of mapreduce jobs

1.3.1 First job

The first job will read the files for the stations and recordings and apply StationMapper and DataMapper respectively. The mappers extract just the required data if they pass some conditions. For example, in the case of the recordings it just reads records where country is 'US' and state is not empty. The output of the mappers will be a $< key, value >$ pair where the key is the station id and the value could be:

1. a month and temperature record for that station (it comes from the recording files and is marked with a 'D'), or
2. the state where the station is located (it comes from the location file and is marked with a 'S').

An example of the output of the mappers can be seen in figure 1. Then, JoinReducer is called in this job to read the mappers' output and combine the data by station id. For each station we will have a set of samples (month and temperature) and the state where the sample belongs to. The output of the job will be a new $< key, value >$ file where the key will be the combination of state and month and the value will be its temperature. Figure 2 illustrates a possible output.

1.3.2 Second job

The second job use a simple mapper (`FileMapper`) to read the last output and `AverageReducer` to compute the average aggregation. It take advantage that the reducer collects all the values with same State-Month combination and compute the average for those values. The reducer will map the output using the state as key and a concatenation of month and average temperature as value. Figure 3 shows an example of the partial result.

```
...
008209  D~03,66.3
008209  D~03,61.1
008209  D~03,62.4
008209  D~03,68.5
...
724839  D~09,69.0
724839  D~09,65.5
724839  D~09,68.1
...
008209  S~FL
724839  S~CA
...
```

Figure 1: Output of `Data` and `Station` `Mappers`.

```
...
CA-09  69.0
CA-09  65.5
CA-09  68.1
...
FL-03  66.3
FL-03  61.1
FL-03  62.4
FL-03  68.5
...
```

Figure 2: Output of `JoinReducer`.

```
...
CA  01,49.167404
CA  02,51.227448
CA  03,53.89828
CA  04,57.585754
CA  05,64.10878
CA  06,68.97722
CA  07,73.66835
CA  08,72.76164
CA  09,68.64867
CA  10,62.290707
CA  11,56.745644
CA  12,48.054047
...
FL  01,61.281094
FL  02,61.340065
FL  03,66.74558
FL  04,71.28077
FL  05,76.623634
FL  06,80.9165
FL  07,81.821976
FL  08,82.65315
FL  09,80.26676
FL  10,74.7974
FL  11,65.25046
FL  12,65.15977
...
```

Figure 3: Output of `AverageReducer`.

```
AK  07  54.51  01  7.74  46.77
AL  08  81.29  01  47.55  33.74
AR  08  80.99  01  40.72  40.27
AZ  07  85.82  12  45.03  40.79
CA  07  73.67  12  48.05  25.61
CO  07  68.82  01  23.21  45.61
CT  07  73.01  02  31.15  41.86
DC  07  77.44  01  32.23  45.21
DE  07  76.53  02  36.42  40.11
FL  08  82.65  01  61.28  21.37
GA  08  80.87  01  48.30  32.57
HI  08  77.89  02  70.12  7.77
IA  07  74.53  01  23.05  51.48
ID  07  72.39  01  24.22  48.17
IL  08  74.71  02  28.57  46.14
IN  08  74.05  02  28.26  45.79
KS  07  79.25  12  32.60  46.64
KY  08  78.09  02  36.45  41.64
...
```

Figure 4: Output of `MaxMinReducer`.

1.3.3 Third job

The third use again `FileMapper` to read the last output. The reduce job (`MaxMinReducer`) will collect the month and its average for each state. Then, it will select the maximum and minimum value and compute the difference. For each case, it will extract the associated months and put them in the output. The job will map the output using the state as key and a concatenation of the maximum temperature, the month for the maximum temperature, the minimum temperature, the month for the minimum temperature and the difference as value. Figure 4 shows an partial example.

1.3.4 Fourth job

The final job read the last output using `SortMapper`. This map uses a custom implementation (`SortableKey`) of the `WritableComparable` interface. This implementation allow to map an output by State and Difference (of temperature). This class implements the methods `toString()`, to print just the State in the key, and `compareTo()`, to force the reducer to sort the key by the difference. As the intention of the job is just to sort the records it does not call a particular reducer, so the default reducer will pass the same records from the mappers but in ascending order. Figure 5 shows the final result.

The four jobs take around 01m21s to complete[1].

1.4 How to approach the join

Section 1.3.1 and figures 1 and 2 explain my approximation to deal with the requested join.

1.5 Possible extra-credit

I would like to put into consideration the implementation of a custom WritableComparable explained in section 1.3.4 as extra-credit.

1.6 Appendix

The bash script in figure 6 has some instructions to compile and run the jobs. It is submitted together with DataReducer.java and DataReducer.jar files which are the source code and executable of my implementation.

[1]See details at `http://www.cs.ucr.edu/~acald013/MP_Output.txt`.

```
VI       July  83.28    February  77.55   5.72
PR     August  82.80     January  76.69   6.11
HI     August  77.89    February  70.12   7.77
FL     August  82.65     January  61.28  21.37
CA       July  73.67    December  48.05  25.61
LA     August  82.63     January  52.68  29.95
OR       July  67.38    December  36.71  30.67
WA       July  66.59    December  35.59  31.01
GA     August  80.87     January  48.30  32.57
TX     August  82.87     January  49.45  33.42
MS     August  81.87     January  48.15  33.72
AL     August  81.29     January  47.55  33.74
SC     August  81.09     January  46.85  34.24
NC     August  78.99     January  44.42  34.57
VA     August  77.10    February  39.30  37.81
NM       July  75.07     January  35.49  39.58
TN     August  80.18     January  40.27  39.92
WV     August  72.57    February  32.63  39.94
DE       July  76.53    February  36.42  40.11
AR     August  80.99     January  40.72  40.27
MD       July  76.68    February  36.27  40.41
AZ       July  85.82    December  45.03  40.79
RI       July  73.09    February  32.28  40.81
KY     August  78.09    February  36.45  41.64
CT       July  73.01    February  31.15  41.86
NJ       July  75.47    February  33.40  42.08
MA       July  71.52    February  29.33  42.19
OK     August  82.27     January  40.03  42.24
PA       July  72.68    February  29.20  43.48
NY       July  71.47    February  27.15  44.32
MO     August  78.36     January  33.28  45.08
ME       July  66.20     January  21.07  45.13
OH     August  73.08    February  27.95  45.13
DC       July  77.44     January  32.23  45.21
NH       July  67.87    February  22.33  45.54
CO       July  68.82     January  23.21  45.61
IN     August  74.05    February  28.26  45.79
IL     August  74.71    February  28.57  46.14
KS       July  79.25    December  32.60  46.64
MI       July  68.59    February  21.92  46.67
AK       July  54.51     January   7.74  46.77
VT       July  68.29     January  21.12  47.17
ID       July  72.39     January  24.22  48.17
NV       July  83.59    December  34.20  49.39
WY       July  71.54    December  20.91  50.64
NE       July  76.36    December  25.54  50.82
MT       July  73.11    December  22.09  51.01
IA       July  74.53     January  23.05  51.48
WI       July  70.08    February  18.43  51.65
UT       July  79.70     January  25.44  54.26
SD       July  75.44    February  20.28  55.16
MN       July  71.11    February  13.71  57.39
ND       July  72.39    February  11.81  60.58
```

Figure 5: Output of `SortMapper`.

```bash
#!/bin/bash
#Usage: ./compile.sh DataReducer /path/to/locations/ /path/to/recordings/ /path/to/output/
#For example:
#    ./compile.sh DataReducer /user/acald013/ /user/acald013/ /user/acald013/output/
HADOOP_CLASSPATH=$(hadoop classpath)
hdfs dfs -rm -R ${4}
javac -classpath $HADOOP_CLASSPATH ${1}.java
jar -cvf ${1}.jar ${1}*
hadoop jar ${1}.jar ${1} ${2} ${3} ${4}
hdfs dfs -cat ${4}final/part-r-00000
```

Figure 6: Bash script for compilation and execution.