Ankit Sharma
Homework 2
CS6240 Section-01

Map-Reduce Source Code

No Combiner Pseudo Code

Map(Record): For all records which have either “TMIN” or “TMAX” record type do the following:

- Emit (Station ID,(Record Type and Record Value))

Reducer(Station ID, [Value1, Value2, …]): For all the values belonging to this Station ID do the following:

- Sum up all the “TMIN” values and count and all “TMAX” values and count and find the Mean Minimum Temperature and Mean Maximum Temperature.
- Emit (Station ID, MeanMinimum, MeanMaximum)

Combiner Pseudo Code

Map(Record): For all records which have either “TMIN” or “TMAX” record type do the following:

- If the record is of type “TMIN” then create a string containing TMIN, its value and its count as 1 and TMAX, its value as 0 and its count as 0. If record is of type “TMAX” then vice-versa.
- Emit (Station ID, (“TMIN”, TMIN Value, TMIN Count, “TMAX”, TMAX Value, TMAX Count))

Combiner(Station ID, [Value1, Value2, …]): For all the values belonging to this Station ID do the following:

- Sum up all the “TMIN” values and count and all “TMAX” values and count
- Emit (Station ID, (“TMIN”, TMIN Value, TMIN Count, “TMAX”, TMAX Value, TMAX Count))

Reducer(Station ID, [Value1, Value2, …]): For all the values belonging to this Station ID do the following:

- Sum up all the “TMIN” values and count and all “TMAX” values and count and find the Mean Minimum Temperature and Mean Maximum Temperature.
- Emit (Station ID, MeanMinimum, MeanMaximum)

Ankit Sharma
Homework 2
CS6240 Section-01

In Mapper Combiner Pseudo Code

Map(Record): For all records which have either "TMIN" or "TMAX" record type do the following:

- For all the station ID's in this mapper, calculate the total sum and count of records
- For all the stations ID's Emit (Station ID, ("TMIN", TMIN Value, TMIN Count, "TMAX", TMAX Value, TMAX Count))

Reducer(Station ID, [Value1, Value2, …]): For all the values belonging to this Station ID do the following:

- Sum up all the "TMIN" values and count and all "TMAX" values and count and find the Mean Minimum Temperature and Mean Maximum Temperature.
- Emit (Station ID, MeanMinimum, MeanMaximum)

Secondary Sort Pseudo Code

Map(Record): For all records which have either "TMIN" or "TMAX" record type do the following:

- For all the (station ID, year) intermediate keys in this mapper, calculate the total sum and count of records on per year basis
- For all the (station ID, year) intermediate keys Emit ((Station ID, Year), ("TMIN", TMIN Value, TMIN Count, "TMAX", TMAX Value, TMAX Count))

getPartition((station ID, year), Value): Create partitions based only on Station ID.

- return myPartition(Station ID)

keyComparator((station ID, year)): Sort in increasing order of Station ID. If Station ID is same, sort in increasing order of year. [Note: I have only implemented keyComparator but not used it in Secondary Sort].

groupingComparator((Station ID, year)): Sort in increasing order of Station ID only. Ignore the year part. So if two records have the same station ID they will be identical.

Reducer((Station ID, year), [Value1, Value2, …]): For all the values belonging to this Station ID do the following:

- The reducer will receive the keys sorted in increasing order of station ID. The value for each key won't be sorted in any order.
- Sum up all the "TMIN" values and count and all "TMAX" values and count for each year and find the Mean Minimum Temperature and Mean Maximum Temperature on the basis of per year.
- Emit (Station ID, [(year0, MeanMinimum0, MeanMaximum0), (year1, MeanMinimum1, MeanMaximum1), …..])

Performance Comparison

Running Time

	Run 1 (in seconds)		
	Map Tasks	Reduce Tasks	Total
No Combiner	796.1	152.8	948.9
Combiner	817.9	125.9	943.8
In Map Combiner	858.7	100.3	959
Sec. Sorting	189.1	77.6	266.7

	Run 2 (in seconds)		
	Map Tasks	Reduce Tasks	Total
No Combiner	834.0	157.8	991.8
Combiner	857.9	135.3	993.2
In Map Combiner	821.2	121.8	943
Sec. Sorting	179.3	76.6	255.9

- Yes, the Combiner was called in the Combiner program. The syslogs from the Combiner program print out the amount of input and output records processed by the combiner and it was more than 0 which means that the combiner was called. The map output records are equal to the combiner input records which points to the fact that combiner was called once.

- The number of records to be processed by the reducer went down from 8798241 in No Combiner to 223782 in Combiner program. Also, the reducer running time is less in Combiner as compared to No Combiner due to less data present.

- Yes, the local aggregation was effective in In-Mapper Combiner as compared to No Combiner since we needed to process less data at the Reducer which resulted in decrease of processing time at the Reducer task as well.

Ankit Sharma
Homework 2
CS6240 Section-01

- In-Mapper Combiner generally should take more time at the Mapper since it has to do more work at the Map. However, we are never sure if and when a Combiner would be called. The programmer cannot control this. As such, In-Mapper combiner would be better since it guarantees that it will decrease the data to be transferred from Mapper to Reducer whereas a Combiner cannot guarantee that.

- The sequential run of Map-Reduce program finishes in around 15 seconds. The values are the same as for the Map-Reduce programs. This is not a clear representation though. If the file were in terabytes of size, then the Map-Reduce program would have been faster.