CSOE18 – Big Data Analytics

Name:-	Kshitij Kanade
Roll No:-	107121045
Date:-	02/12/2023
Batch:-	2025
Dept:-	EEE Sec A

Assignment - 2

Hadoop MapReduce for
Climate Data Analytics

Task_3b Answers

Task #3b:-

i. **What will the key and values, output by the Map tasks be? What types will they be?**

- Key: The key emitted by the Map tasks will be the date (Text type).
- Value: The value will be the temperature difference (FloatWritable type).

ii. **Can the Mapper produce a key/value pair from a single input? How can we solve this issue?**

Yes, the Mapper can produce a key/value pair from a single input.
To address this issue, the Mapper in this case processes each input line from the CSV. It checks if the record is either TMAX or TMIN for the specified station (USW00094728). When it finds both TMAX and TMIN records for the same date, it emits the date as the key and the temperature difference as the value.

iii. **For each day and weather station, the TMAX record always precedes the TMIN in NCDC's data, and we suppose that no split occurs between a TMAX and a TMIN record. By following this approach, what work will be left to the Reduce task?**

- Since TMAX records always precede TMIN records and no split occurs between them, the Map phase filters and pairs TMAX and TMIN records for each date. Hence, the Reduce task's primary job is to gather these paired temperature data per date and perform no additional computation, simply emitting the temperature differences calculated by the Mapper.

iv. **Reducer classes extend Hadoop's Reducer class. Read Hadoop's documentation for that class, in particular, what the default behaviour of its reduce() function is. What can you deduce from this? Write the code for both solutions and plot the results.**

- The default behavior of the reduce() function in Hadoop's Reducer class is to iterate through all values associated with a particular key.

- The reduce() function gets called once for each unique key, receiving an iterable of values.
- It's expected to perform processing on these values, possibly aggregating or computing something based on the logic provided within the reduce() method.
- In this scenario, the Reducer's reduce() method will receive temperature differences associated with each date, and for each date, it will directly emit these differences without any further processing.
- This MapReduce job is optimized to pair TMAX and TMIN records, computing temperature differences in the Mapper and emitting these differences per date for the Reduce phase to write them to the output. The Reduce task doesn't need to perform complex calculations, merely collecting and emitting the calculated temperature differences for each date.

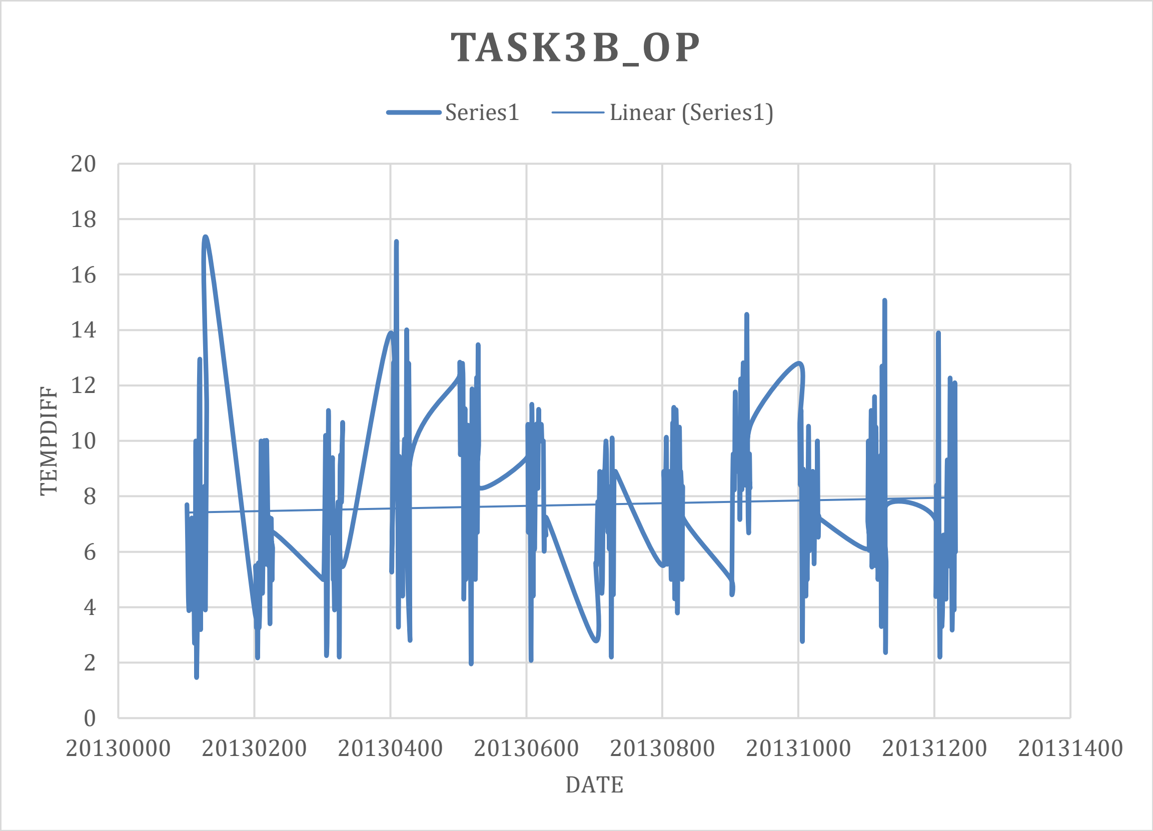
NOTE:- Code files are present in the directory along with output files (text as well as Excel) – refer to the plot in the Excel sheet too in case the one here is not visible.

Further possible modifications:-

One possible enhancement for Task_3b involves implementing a combiner to perform a partial aggregation in the Map phase. This combiner could efficiently aggregate temperature differences per date before sending them to the Reducer, further reducing data transmission and Reducer workload.

Additionally, employing data compression techniques in the output stage could optimize storage utilization. Moreover, introducing a partitioner based on dates might enhance parallelism by distributing data across reducers evenly. Implementing speculative execution for specific tasks can enhance fault tolerance and job completion time. Furthermore, employing a custom data serialization method could optimize data transfer. Lastly, utilizing in-memory caching mechanisms for frequently accessed data can boost performance in subsequent computations.

Plot:-



References:-

1. The Word Count program on MapReduce demonstrated in class.
2. Regular expressions application in Java.
3. Verbal Discussion with Classmates Rohit Meena, Harshit Malik, Atharv Bhavey, Akanksh Muvva, Vinay.
4. StackOverflow, GeeksForGeeks, etc online platforms.

Thank You!