The University of Texas at Dallas Spring 2022
Assignment 1
Name : Puru Jaiswal
NetID(s): PXJ200018

CS6350.001: Big Data Management and Analytics

Question 1

What is the difference between Hadoop and traditional database management systems?

- **Data** : RDBMS stores only structured data while hadoop can store both structured and unstructured data.
- **Processing** : Structured processing is done as the data is structured, while in hadoop both structured and unstructured data.
- RDBMS is best suited for OLTP environments, while hadoop is suitable for big data.
- In RDBMS as the volume increases querying the system becoms slow while in hadoop it queries on big data efficiently.
- RDBMS stores limited amount of data, while hadoop stores huge volume of data.
- Since RDBMS use ACID properties, they have higher integrity as compared to hadoop.
- RDBMS can be scaled vertically, while hadoop can be scaled horizontally.
- Hadoop is less expensive as it is open source in comparsion with traditional database management systems.

Question 2

Shuffling and sorting phase requires two types of sorting. Why and in which case those sorting are used ?

In Hadoop architecture, the data processing takes in the form $< key, value >$ pairs. Once the mapped has emitted a given key $< key, value >$ pair, the partitioner shuffles sort and decides on which reducer the pair needs to go. Then the reducer aggregates the $< key, value >$ pair which it receives and throw as final output.
The sorting step can be done in two ways, which are primary and secondary sorting.

Primary sorting is the default sorting which happens within each partitioner, this is based on the keys and done so that each reducer receives the value with the same key.
Now consider the scenario where we need to output the words for a particular line. In this the mapper emits $< word, line_number >$ pairs, so if it goes with primary sorting then at reducer side the $< word, line_number >$ pair will be for the same word it will tell all the line numbers whereas our original question was to give all the words which are present on a particular line number.
So here comes in the secondary sorting, so here if we sort the pairs based on the values then that would suffice our requirement. But to achieve this we need to do value to key conversion, which is done by forming a composite key and let the framework do the sorting.

Question 3

During fault tolerance, in-progress jobs in the reducer side need to be rerun whereas completed ones do not. But in the mapper side, regardless of whether the jobs are in-progress or completed they need to be rerun. Why?

The reducer doesn't really know whether it has recieved all the data from the mapper, so mapper needs to rerun.

Reducers has started reading data early before mapper completed and read some partial data. Mapper could produce more data if not failed.

Mapper has produced partial result files, then failed and new attempt has started.

Typically mappers and reducers are single-threaded and deterministic, this allows restarts and speculative execution.

Now with the case of reducers, if the jobs are in progress and the reducer has failed, so in this case only the jobs which were in progress needs to be rerun and not the jobs which have been completed, as we don't really care about the completed jobs at the reducer side.

Question 4

Suppose you have a file that stores all sales related information of a store. It contains the columns product name, price, payment mode, city, and country of client. The goal is to find out number of products sold in each country. Write a map/reduce pseudocode that computes the goal.

Assumption : The Input recieved to the mapped is in the below format :
ProductName < $TabSpace$ > Price, Payment Mode, City, Country

Map(DocId, InputString, Text, Integer) :

- InputArr := InputString.Split("tab space");
- PName := InputArr[0];
- PDesc := InputArr[1];
- PdescArr := PDesc.Split(",");
- Country := PDescArr[3];
- Emit(Country, 1);

Reduce(Text, IntIterable< $Array$ >, Text, Text)

- Sum := 0
- For V in IntItearable$Array$:
- Sum += V
- Write(Text, Sum)

Question 5

Write the pseudocode for a MapReduce program to perform matrix multiplication.

Map Side Algorithm

- **for** each element m_{ij} in M do

- produce (key, value) pairs ((i,j),(M, j, m_{ij})) for k = 1,2,3 upto the number of columns of N.
- **for** each element n_{jk} in N do
- produce (key, value) pairs ((i, j),(N, j, n_{jk})) for k = 1,2,3 upto the number of rows of M.
- **return** Set of (key, value) pairs where each key, (i,k) has a list with values (M, j, m_{ij}) and (N, j, n_{jk}) for all possible values of j.

Reduce Side Algorithm

- **for** each key(i, k) do
- sort values begin with M by j in $list_M$
- sort values begin with N by j in $list_N$
- multiply m_{ij} and n_{jk} for jth value of each list.
- sum m_{ij}*n_{jk}
- **return** (i,k), $\sum_{j=1} m_{ij}$*n_{jk}