**Cloud Computing for Data Analysis**

**Assignment – 1**

|  |
| --- |
| **Submitted by:** Abdullah Al Raqibul Islam  **UNCC ID:** 801151189 |

1. The DISTINCT(X) operator is used to return only distinct (unique) values for datatype (or column) X in the entire dataset .

As an example, for the following table A:

|  |  |  |
| --- | --- | --- |
| **A.ID** | **A.ZIPCODE** | **A.AGE** |
| 1 | 12345 | 30 |
| 2 | 12345 | 40 |
| 3 | 78910 | 10 |
| 4 | 78910 | 10 |
| 5 | 78910 | 20 |

DISTINCT(A.ID) = (1, 2, 3, 4, 5)

DISTINCT(A.ZIPCODE) = (12345, 78910)

DISTINCT(A.AGE) = (30, 40, 10, 20)

Implement the DISTINCT(X) operator using Map-Reduce. Provide the algorithm pseudocode. You should use only one Map-Reduce stage, i.e. the algorithm should make only one pass over the data.

|  |
| --- |
| We can use MapReduce’s ability to group keys together to fetch the DISTINCT elements. The mapper function will emit the value of X column as the key for the reducer phase. The reducer phase will just emit the key as the output. Here is the pseudocode,  mapper(key, record):  emit (value, NULL) -> where value is the column X’s value for each input  reducer(key, records):  emit (key, NULL) |

1. The SHUFFLE operator takes a dataset as input and randomly re-orders it.

*Hint: Assume that we have a function rand(m) that is capable of outputting a random integer between [1, m].*

Implement the SHUFFLE operator using Map-Reduce. Provide the algorithm pseudocode.

|  |
| --- |
| Mapper function will map the record as a value of random key so that it will go to a random reducer. The reducer will simply emit the values.  rand(m):  return a random integer between [1, m]  mapper(key, record):  emit (rand(m), record)  reducer(key, record):  for each value in record:  emit (value) |

1. What is the communication cost (in terms of total data flow on the network between mappers and reducers) for following query using Map-Reduce:

**Get DISTINCT(A.ID from A WHERE A.AGE > 30 )**

The dataset A has 1000M rows, and 400M of these rows have A.AGE <= 30. DISTINCT(A.ID) has 1M elements. A tuple emitted from any mapper is 1 KB in size.

|  |
| --- |
| We will apply the WHERE clause in the mapper function and forward the data to the reducer to perform the DISTINCT part (as done in the Task 1).  So, mapper function will transfer (1000M – 400M) = 600M tuple data to the reducer. As given, each tuple’s size is 1KB, so the total data flow on the network between mappers and reducers,  600M \* 1KB = 572.20 GB |

1. Consider the checkout counter at a large supermarket chain. For each item sold, it generates a record of the form [Product-Id, Supplier, Price]. Here, Product-Id is the unique identifier of a product, Supplier is the supplier name of the product and Price is the sales price for the item. Assume that the supermarket chain has accumulated many terabytes of data over a period of several months.

The CEO wants a list of suppliers, listing for each supplier the average sales price of items provided by the supplier. How would you organize the computation using the Map-Reduce computation model?

|  |
| --- |
| The mapper will simply emit the supplier as key and price as value for the reducer phase. Reducer will perform the average operation on the list of the values of a supplier’s price list. Here is the pseudocode,  mapper(key, record):  emit (record.suplier, record.price)  reducer(supplier, list of price):  emit (supplier, AVG(price)) |

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

*For the following questions give short explanations of your answers.*

1. **True or False:** Each mapper/reducer must generate the same number of output key/value pairs as it receives on the input.

|  |
| --- |
| **False.**  Mapper and reducer can generate any number of key/value pairs. For example, in the Task 3 we have observed that the mapper is receiving 1000M records and forwarding only 600M to the reducer. |

1. **True or False:** The output type of keys/values of mappers/reducers must be of the same type as their input.

|  |
| --- |
| **False.**  Mappers/reducers can produce any type of key/values. For example, in the above Task 4 the reducer is receiving a supplier-name and a list of prices for that supplier, but returning supplier-name and average price for that supplier (different than the input key/value pair). |

1. **True or False:** The input to reducers is grouped by key.

|  |
| --- |
| **True.**  The input for reducer is generated by the mapper, and mapper’s sole task is to forward the data to reducer grouped by key. |

1. **True or False:** It is possible to start reducers while some mappers are still running.

|  |
| --- |
| **False.**  As we know from Task 7 that, the input to reducers is grouped by key. If any mapper is still running when the reducers has already been started, then the running mapper can still produce a that already consumed by the running reducers. |