EXPERIMENT NO:05

Aim: Implementing simple algorithms in Map Reduce: Matrix multiplication, aggregates, joins, sorting, searching etc.

System software requirements: Ubuntu, MongoDB

Theory:

MapReduce is a technique in which a huge program is subdivided into small tasks and run parallelly to make computation faster, save time, and mostly used in distributed systems. It has 2 important parts:-

- **Mapper:-**
  It takes raw input data and organizes it into key, value pairs. For example, in a dictionary "Data" and its associated meaning is "facts and statistics collected together for reference or analysis". Here, the key is data and the value associated with its facts and statistics collected together for references or analysis
- **Reduber:-**
  It is responsible for processing data in parallel and produce final output.

Algorithm 1: The map function

1. For each element m_{ij} of M do
2. product (key value) pair as $((i,k),(M,j,m_{ij}))$, for $k = 1,2,3...$ up to the number of columns of N.
3. for each element n_{jk} of N do.
4. product (key value) pair as $((i,k),(N,j,n_{jk}))$, for $i = 1, 2, 3...$ up to number of rows of M.
5. return set of (key value) pairs that each key, (i,k), has a list with values (M,j,m_{ij}) and (N,j,n_{jk}) for all possible values of j.

Algorithm 2 : The reduce function

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

* Aggregation in MongoDB:

- In MongoDB, aggregation operations process the data records/ documents and return computed results.
- It collects values from various documents and groups them together and then performs different types of operations on that grouped data like sum, average, minimum, maximum, etc to return a computed result.

* Conclusion :

Thus, we have successfully implemented the simple algorithms like matrix multiplication, aggregation and joins