**Fundamentals of MapReduce with Example**

Hadoop has two core components – HDFS and MapReduce.

HDFS or Hadoop Distributed File System is used to store the large, voluminous data coming from different sources which are commonly known as the Big Data.

MapReduce is the mechanism to process that Big Data in HDFS.

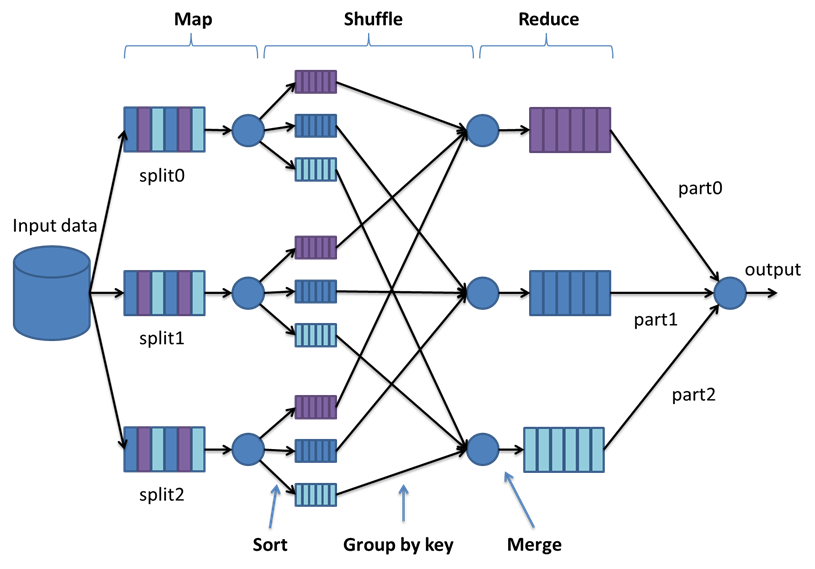
**Definition: -**

Map Reduce is the programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop Cluster, i.e. suppose you have a job to run and you write the Job using the MapReduce framework and then if there are a thousand machines available, the Job could run potentially in those thousand machines.

**Why MapReduce used –**

The Big Data is not stored traditionally in HDFS. The data gets divided into chunks of small blocks of data which gets stored in respective data nodes. No complete data’s present in one centralized location and hence a native client application cannot process the information right away. So a particular framework is needed with the capability of handling the data that stays as blocks of data into respective data nodes, and the processing can go there to process that data and bring back the result. In a nutshell, data is processed in parallel which makes processing faster.

**How Map Reduce works –**



The above diagram illustrates the working mechanism of the Map-Reduce algorithm which operates on three phases – Mapper Phase, Sort and Shuffle Phase and the Reducer Phase.

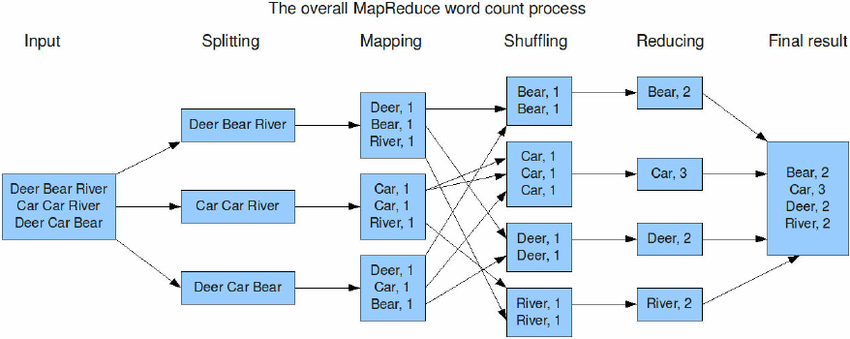
Map – In this stage, the input data is mapped into respective key-value pairs on all the mappers assigned to the data.

Shuffle and Sort – This phase acts as a bridge between the Map and the Reduce phase to decrease the computation time. The data here is shuffled and sorted simultaneously based on the keys.

Reduce – The sorted data is the input to the Reducer which aggregates the value corresponding to each key and produces the desired output.

**Example:-**

The Map-Reduce algorithm could be used in various applications such as finding the number of sales in a store based on each product category or finding the monetary value for the highest sales grouped by stores. However, here we will look into the simple word count problem to understand how the algorithm works.



In the above figure, the input data is first is split into three mappers where each unique word is a key, and the corresponding count is labelled as its value.

In the shuffling phase, data taken from each mapper is shuffled and sorted based on the keys.

In the last step, i.e., the Reduce phase, the value of each key is aggregated and produce the final output.