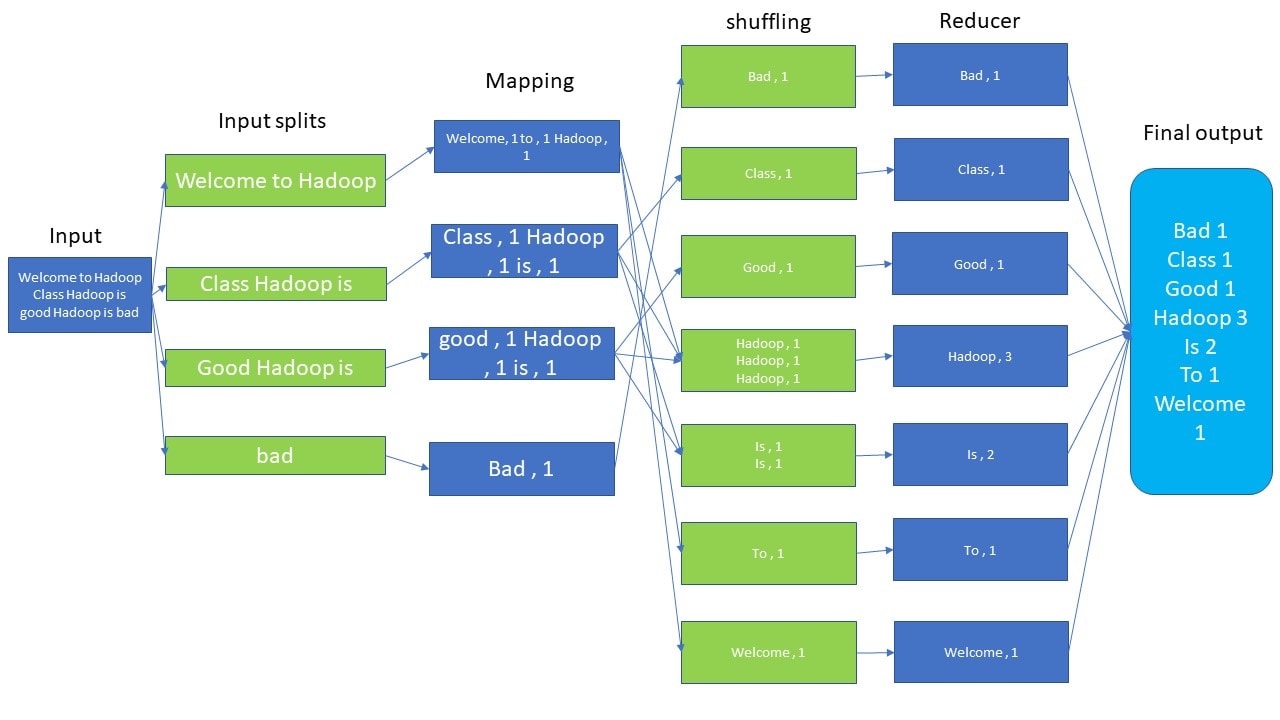
**Introduction To MapReduce**

MapReduce is a Hadoop structure utilized for composing applications that can process large amounts of data on clusters. It can likewise be known as a programming model in which we can handle huge datasets across PC clusters. This application permits information to be put away in a distributed form. It works on huge volumes of data and enormous scope of computing.

MapReduce consists of two phases:  
Map and Reduce Map generally deals with the splitting and mapping of data while reducing tasks shuffle and reducing the data.  
Hadoop is fully capable of running MapReduce programs that are written in various languages: python, java, and C++. This is very useful for performing large-scale data analysis using multiple machines in the cluster.  


**Application Of MapReduce**

**Entertainment:** To discover the most popular movies, based on what you like and what you watched in this case Hadoop MapReduce help you out. It mainly focuses on their logs and clicks.

**E-commerce:** Numerous E-commerce suppliers, like Amazon, Walmart, and eBay, utilize the MapReduce programming model to distinguish most loved items dependent on clients’ inclinations or purchasing behavior.

It incorporates making item proposal Mechanisms for E-commerce inventories, examining website records, buy history, user interaction logs, etc.

**Data Warehouse:** We can utilize MapReduce to analyze large data volumes in data warehouses while implementing specific business logic for data insights.

**Fraud Detection:** Hadoop and MapReduce are utilized in monetary enterprises, including organizations like banks, insurance providers, installment areas for misrepresentation recognition, pattern distinguishing proof, or business metrics through transaction analysis.

**How does MapReduce Works?**

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

* The Map task takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key-value pairs).
* The Reduce task takes the output from the Map as an input and combines those data tuples (key-value pairs) into a smaller set of tuples.

The reduced task is always performed after the map job.

**Input Phase −** Here we have a Record Reader that translates each record in an input file and sends the parsed data to the mapper in the form of key-value pairs.

**Map −** Map is a user-defined function, which takes a series of key-value pairs and processes each one of them to generate zero or more key-value pairs.

**Intermediate Keys −** The key-value pairs generated by the mapper are known as intermediate keys.

**Combiner −** A combiner is a type of local Reducer that groups similar data from the map phase into identifiable sets. It takes the intermediate keys from the mapper as input and applies a user-defined code to aggregate the values in a small scope of one mapper. It is not a part of the main MapReduce algorithm; it is optional.

**Shuffle and Sort −** The Reducer task starts with the Shuffle and Sort step. It downloads the grouped key-value pairs onto the local machine, where the Reducer is running. The individual key-value pairs are sorted by key into a larger data list. The data list groups the equivalent keys together so that their values can be iterated easily in the Reducer task.

**Reducer −** The Reducer takes the grouped key-value paired data as input and runs a Reducer function on each one of them. Here, the data can be aggregated, filtered, and combined in a number of ways, and it requires a wide range of processing. Once the execution is over, it gives zero or more key-value pairs to the final step.

**Output Phase −** In the output phase, we have an output formatter that translates the final key-value pairs from the Reducer function and writes them onto a file using a record writer.

**Advantage of MapReduce**

**Fault tolerance:** It can handle failures without downtime.  
Speed: It splits, shuffles, and reduces the unstructured data in a short time.  
**Cost-effective:** Hadoop MapReduce has a scale-out feature that enables users to process or store the data in a cost-effective manner.  
**Scalability:** It provides a highly scalable framework. MapReduce allows users to run applications from many nodes.  
Parallel Processing: Here multiple job-parts of the same dataset can be processed in a parallel manner. This can reduce the task that can be taken to complete a task.

**Limitations Of MapReduce**

* MapReduce cannot cache the intermediate data in memory for a further requirement which diminishes the performance of Hadoop.
* It is only suitable for Batch Processing of a Huge amounts of Data.

**Frequently Asked Questions**

**Q.1 What are the main parameters which help users to execute a MapReduce job?**

some of the main parameters are :

* Job’s input locations in the distributed file system.
* Job’s output location in the distributed file system.
* The input format of data.
* The output format of data.’

**Q.2 What is the reducer do in MapReduce?**

Reducer in Hadoop MapReduce reduces a set of intermediate values which share a key to a smaller set of values.