JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE	Jaipur Engineering College and Research Centre, Shri Ram ki Nangal, via Sitapura RIICO Jaipur- 302 022.	Academic year- 2020-2021

Vision & Mission of the Department

Vision of the Department

To become renowned Centre of excellence in computer science and engineering and make competent engineers & professionals with high ethical values prepared for lifelong learning.

Mission of the Department

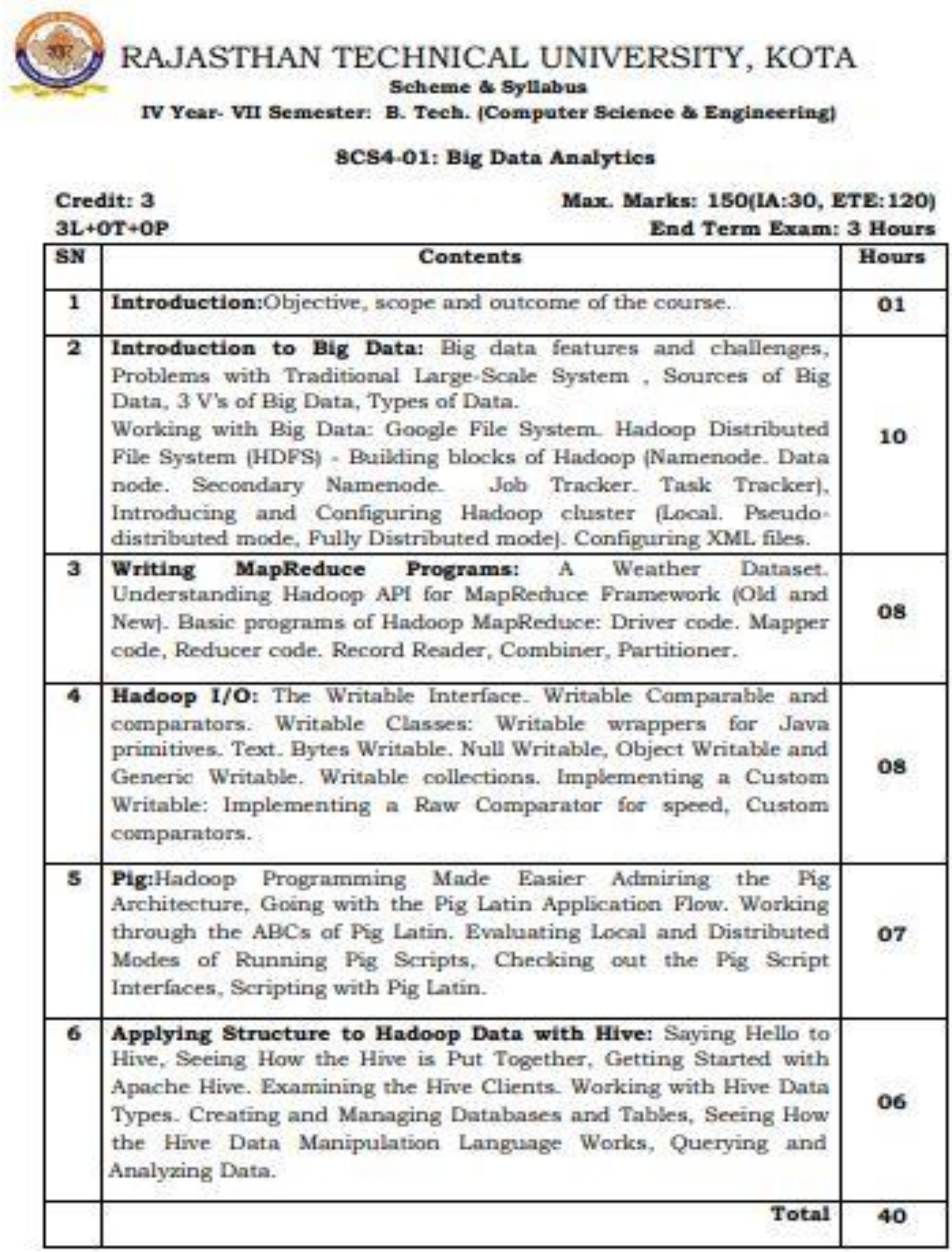
M1: To impart outcome based education for emerging technologies in the field of computer science and engineering.

M2: To provide opportunities for interaction between academia and industry.

M3: To provide platform for lifelong learning by accepting the change in technologies

M4: To develop aptitude of fulfilling social responsibilities.

SYLLABUS



RAJASTHAN TECHNICAL UNIVERSITY, KOTA

Scheme & Syllabus

IV Year- VII Semester: B. Tech. (Computer Science & Engineering)

8CS4-01: Big Data Analytics

Credit: 3 **Max. Marks: 150(IA:30, ETE:120)**

3L+0T+0P **End Term Exam: 3 Hours**

SN	Contents	Hours
1	**Introduction:**Objective, scope and outcome of the course.	01
2	**Introduction to Big Data:** Big data features and challenges, Problems with Traditional Large-Scale System , Sources of Big Data, 3 V's of Big Data, Types of Data. Working with Big Data: Google File System. Hadoop Distributed File System (HDFS) - Building blocks of Hadoop (Namenode. Data node. Secondary Namenode. Job Tracker. Task Tracker), Introducing and Configuring Hadoop cluster (Local. Pseudo-distributed mode, Fully Distributed mode). Configuring XML files.	10
3	**Writing MapReduce Programs:** A Weather Dataset. Understanding Hadoop API for MapReduce Framework (Old and New). Basic programs of Hadoop MapReduce: Driver code. Mapper code, Reducer code. Record Reader, Combiner, Partitioner.	08
4	**Hadoop I/O:** The Writable Interface. Writable Comparable and comparators. Writable Classes: Writable wrappers for Java primitives. Text. Bytes Writable. Null Writable, Object Writable and Generic Writable. Writable collections. Implementing a Custom Writable: Implementing a Raw Comparator for speed, Custom comparators.	08
5	**Pig:**Hadoop Programming Made Easier Admiring the Pig Architecture, Going with the Pig Latin Application Flow. Working through the ABCs of Pig Latin. Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin.	07
6	**Applying Structure to Hadoop Data with Hive:** Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive. Examining the Hive Clients. Working with Hive Data Types. Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analyzing Data.	06
	Total	40

PROGRAM OUTCOMES

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend

and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Jaipur Engineering College and Research Centre

Department of Computer Science & Engineering

Subject – Big Data Analytics **Subject code – 8CS4 - 01**

Semester - VIII **[L/T/P - 3/0/0]**

Course Outcome

CO1. To understand the features, file system and challenges of big data.

CO2. To learn and analyze big data analytics tools like Map Reduce, Hadoop.

CO3. To apply and evaluate Hadoop programming with respect to PIG architecture.

CO4. To create and analyze database with Hive and related tools.

CO- PO Mapping

H=3, M=2, L=1

Semester	Subject	Code	L/T/P	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
VIII	**Big Data Analytics**	8CS4 - 01	L	CO1	3	2	2	2	1	1	1	1	1	1	2	3
			L	CO2	3	3	3	2	2	1	1	1	1	1	2	3
			L	CO3	3	3	3	2	2	1	1	1	1	2	2	3
			L	CO4	3	3	3	2	2	2	2	2	2	2	2	3

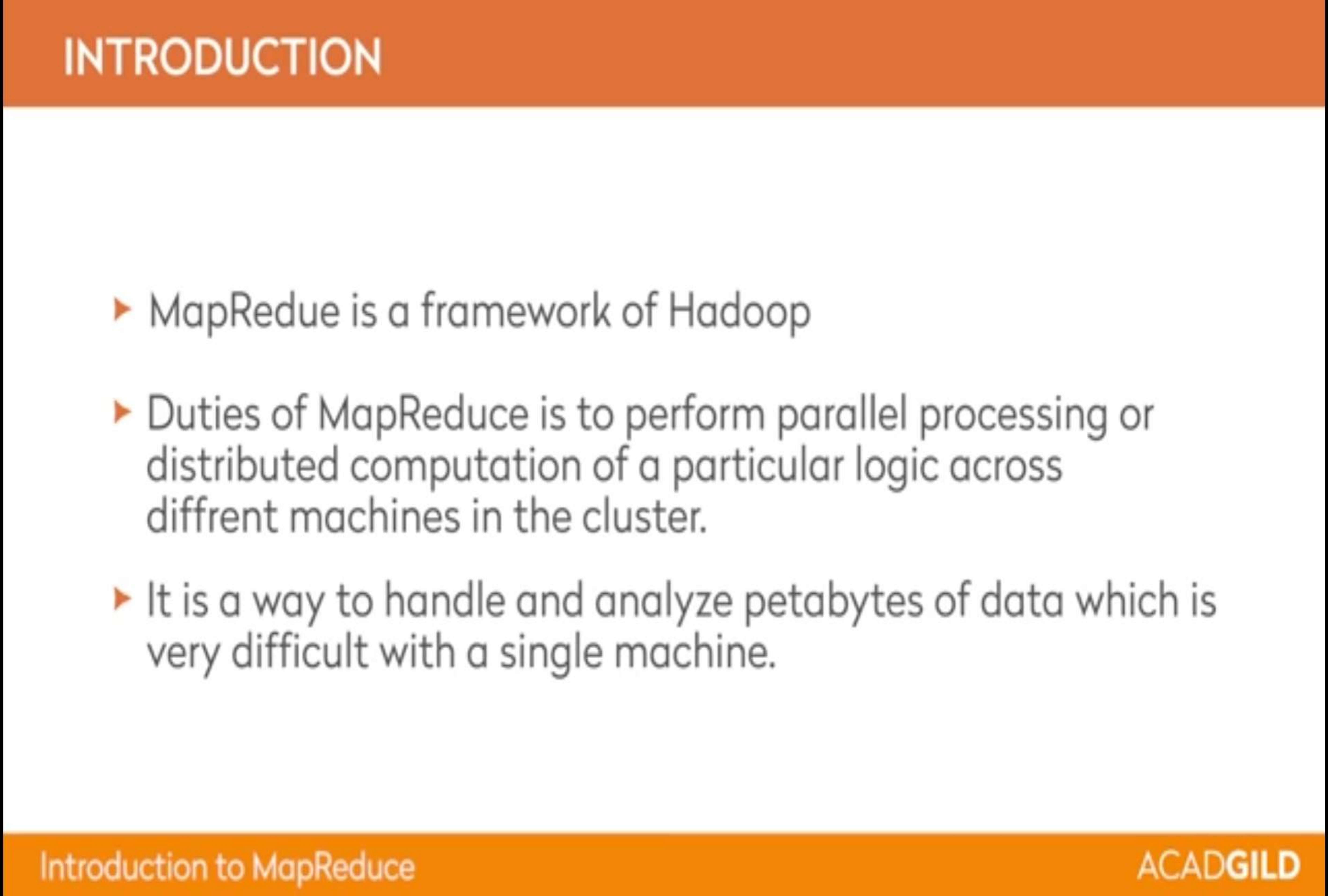
PROGRAM EDUCATIONAL OBJECTIVES:

1. To provide students with the fundamentals of Engineering Sciences with more emphasis in **Computer Science &Engineering** by way of analyzing and exploiting engineering challenges.
2. To train students with good scientific and engineering knowledge so as to comprehend, analyze, design, and create novel products and solutions for the real life problems.
3. To inculcate professional and ethical attitude, effective communication skills, teamwork skills, multidisciplinary approach, entrepreneurial thinking and an ability to relate engineering issues with social issues.
4. To provide students with an academic environment aware of excellence, leadership, written ethical codes and guidelines, and the self motivated life-long learning needed for a successful professional career.
5. To prepare students to excel in Industry and Higher education by Educating Students along with High moral values and Knowledge

JAIPUR ENGINEERING COLLEGE AND RESEARCH CENTRE

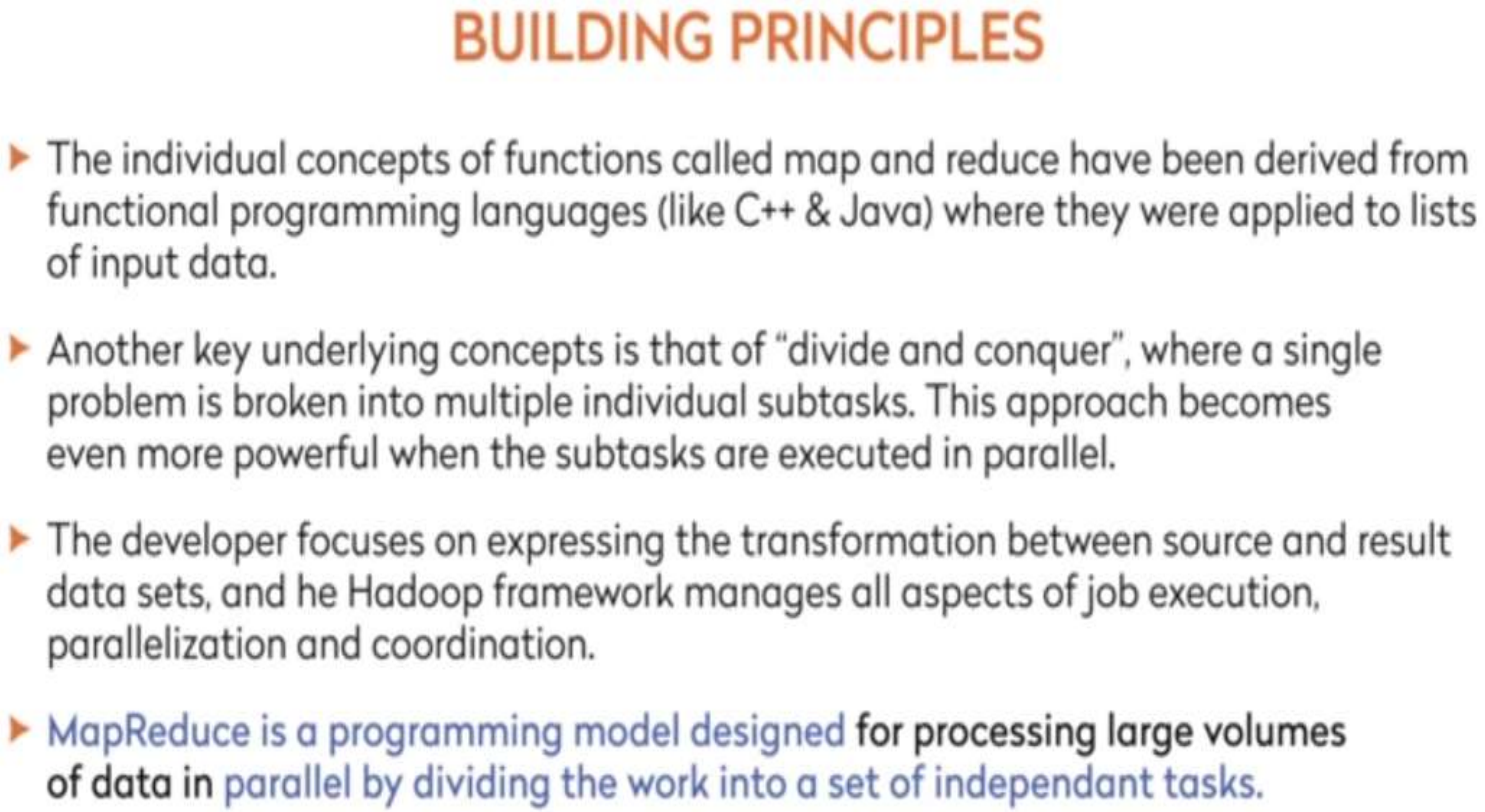
Year & Sem – IV year & VIII Sem
Subject – Big Data Analytics
Unit – II

0101
1010
%
BIG
DATA
INTRODUCTION TO MAPREDUCE



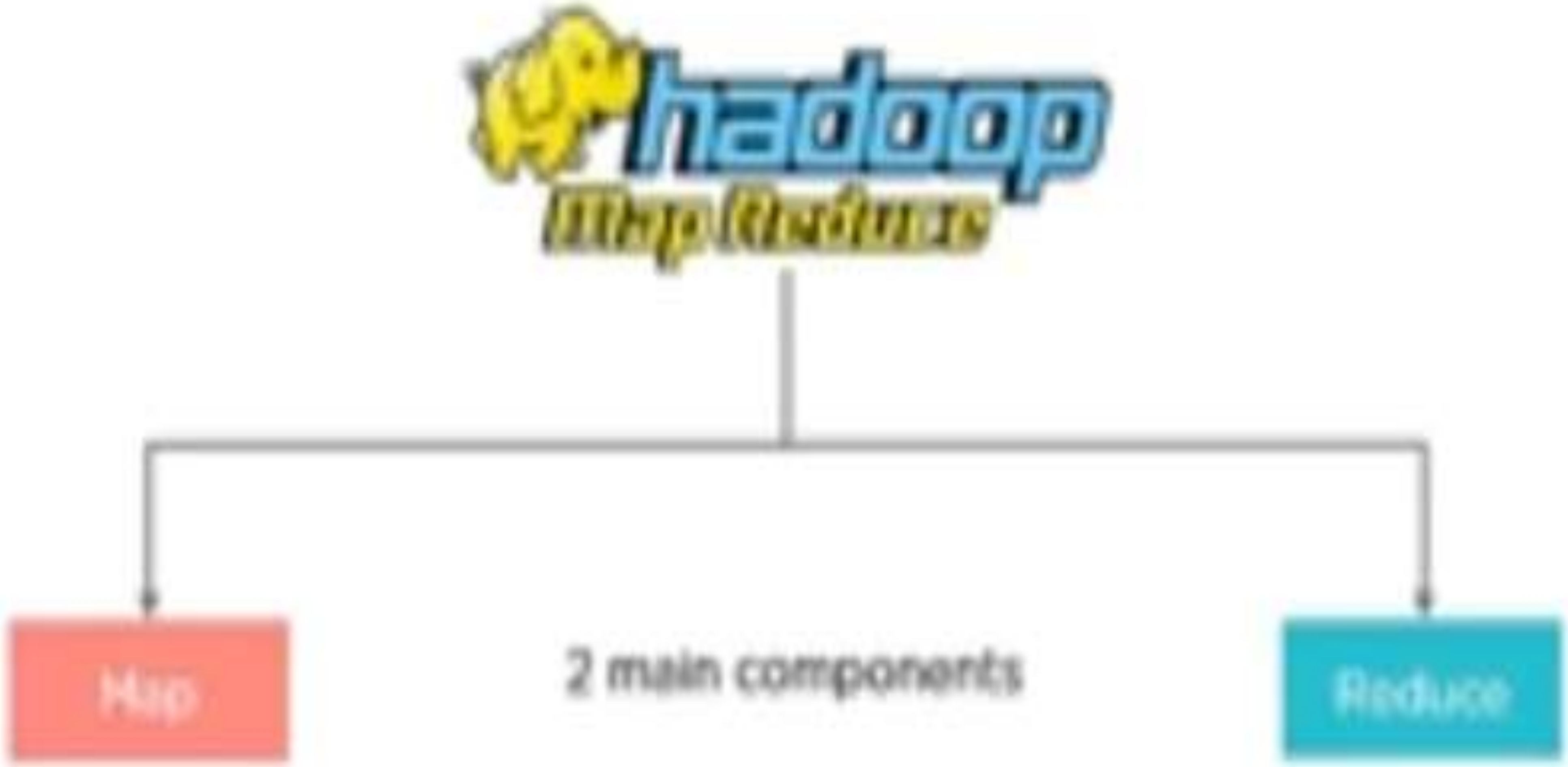
INTRODUCTION

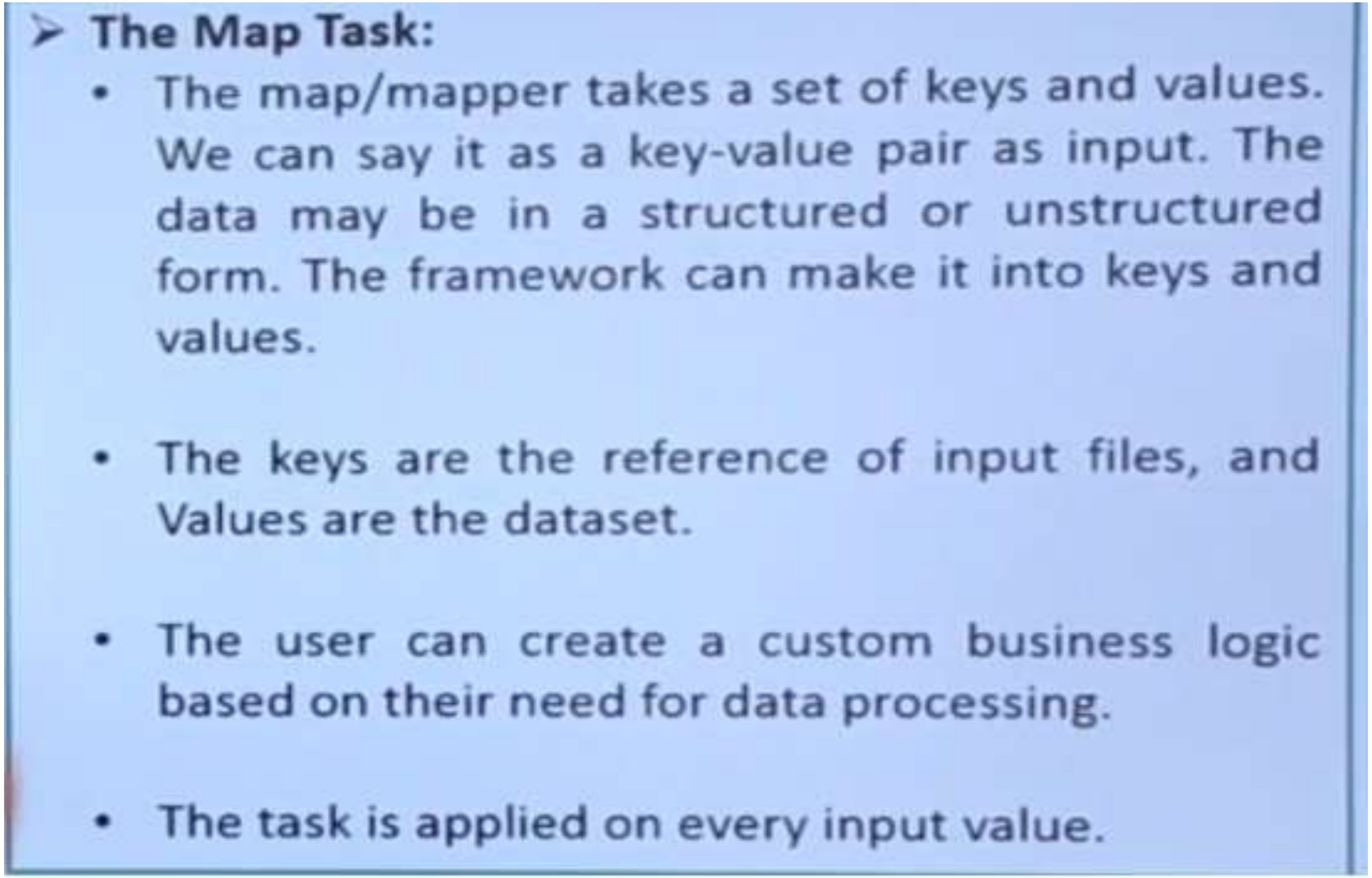
- MapRedue is a framework of Hadoop
- Duties of MapReduce is to perform parallel processing or distributed computation of a particular logic across diffrent machines in the cluster.
- It is a way to handle and analyze petabytes of data which is very difficult with a single machine.



BUILDING PRINCIPLES

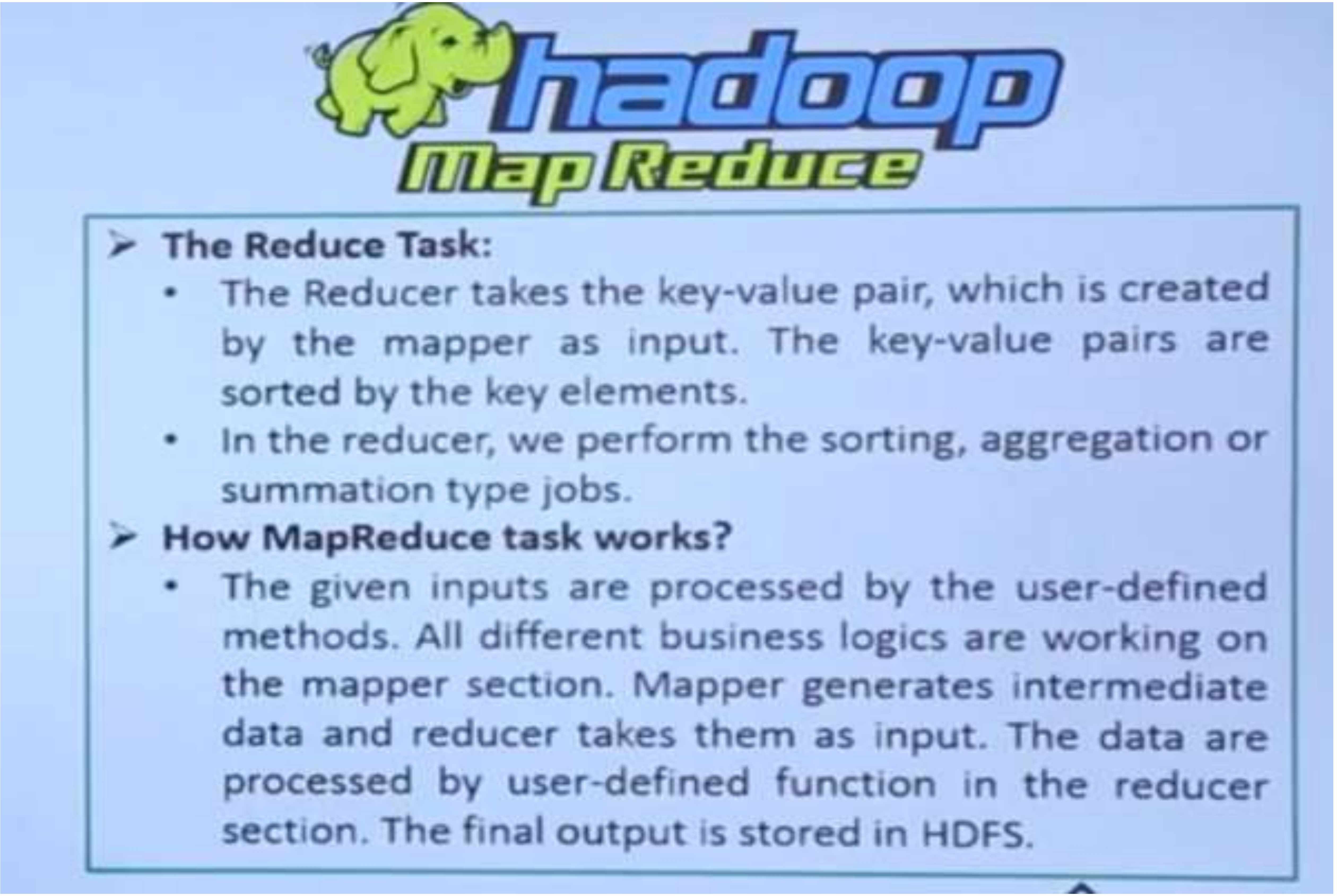
- The individual concepts of functions called map and reduce have been derived from functional programming languages (like C++ & Java) where they were applied to lists of input data.
- Another key underlying concepts is that of "divide and conquer", where a single problem is broken into multiple individual subtasks. This approach becomes even more powerful when the subtasks are executed in parallel.
- The developer focuses on expressing the transformation between source and result data sets, and he Hadoop framework manages all aspects of job execution, parallelization and coordination.
- MapReduce is a programming model designed for processing large volumes of data in parallel by dividing the work into a set of independant tasks.


hadoop
Map Reduce
Map
2 main components
Reduce



➢ **The Map Task:**

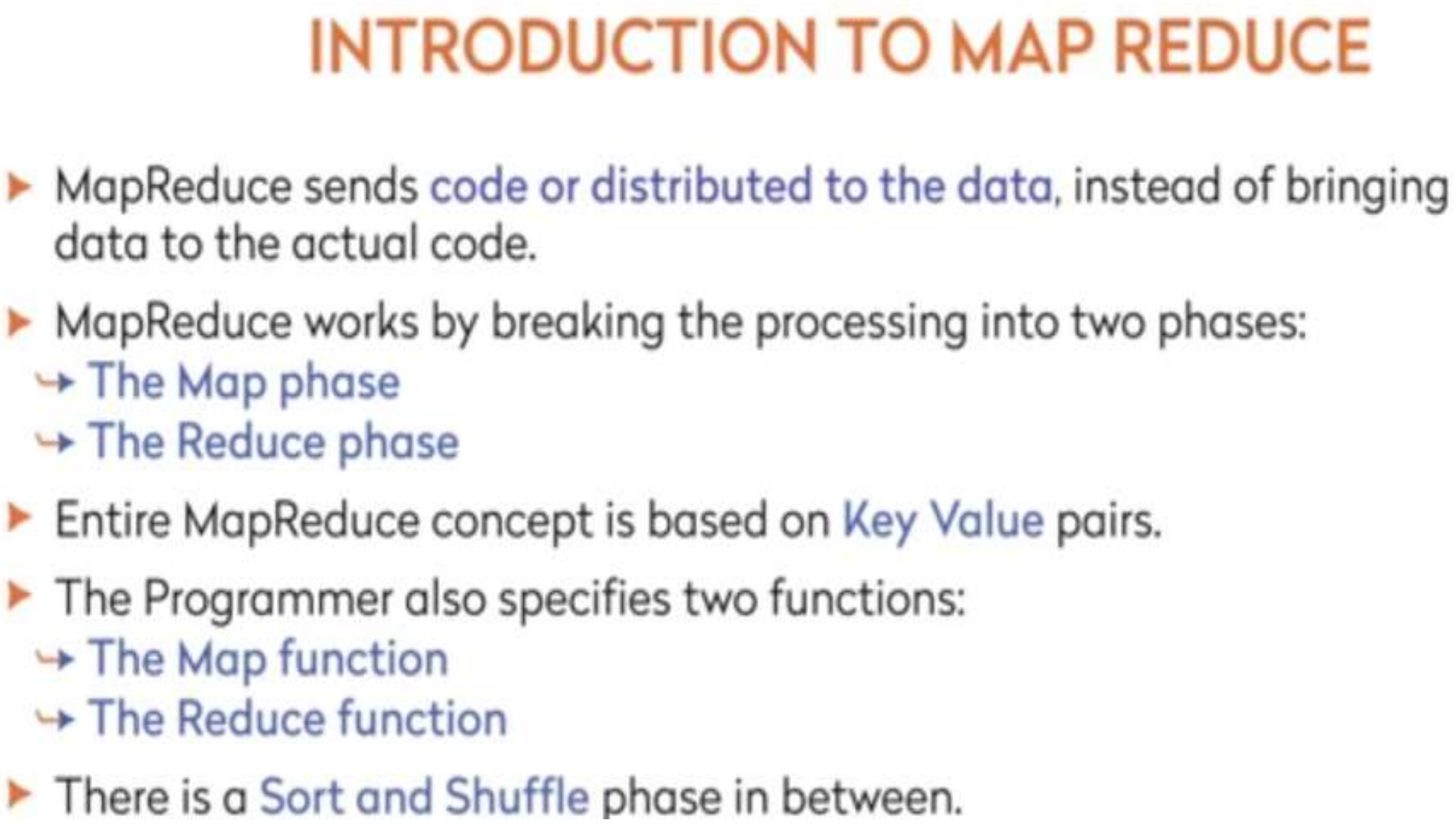
- The map/mapper takes a set of keys and values. We can say it as a key-value pair as input. The data may be in a structured or unstructured form. The framework can make it into keys and values.
- The keys are the reference of input files, and Values are the dataset.
- The user can create a custom business logic based on their need for data processing.
- The task is applied on every input value.



hadoop

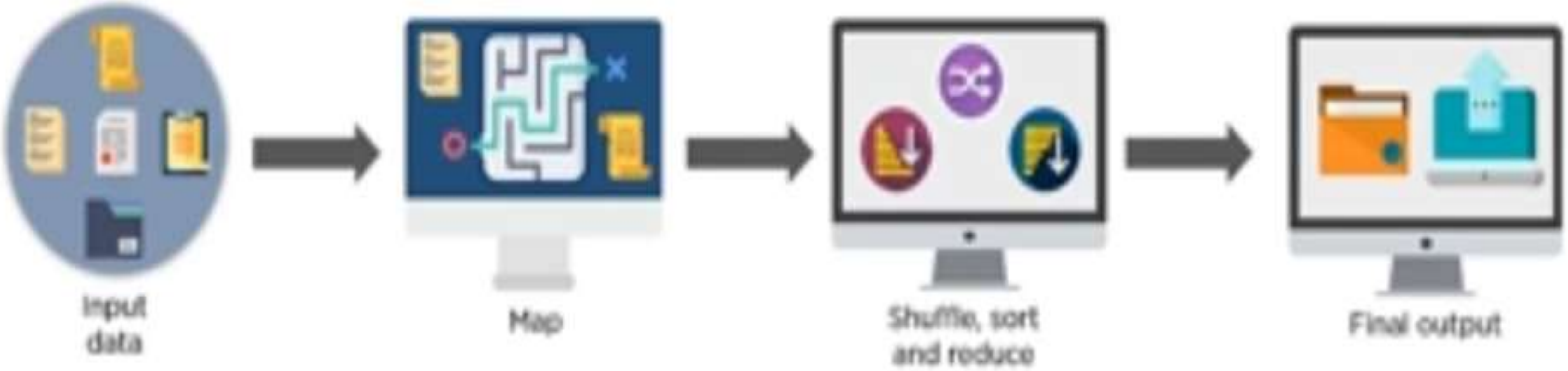
Map Reduce

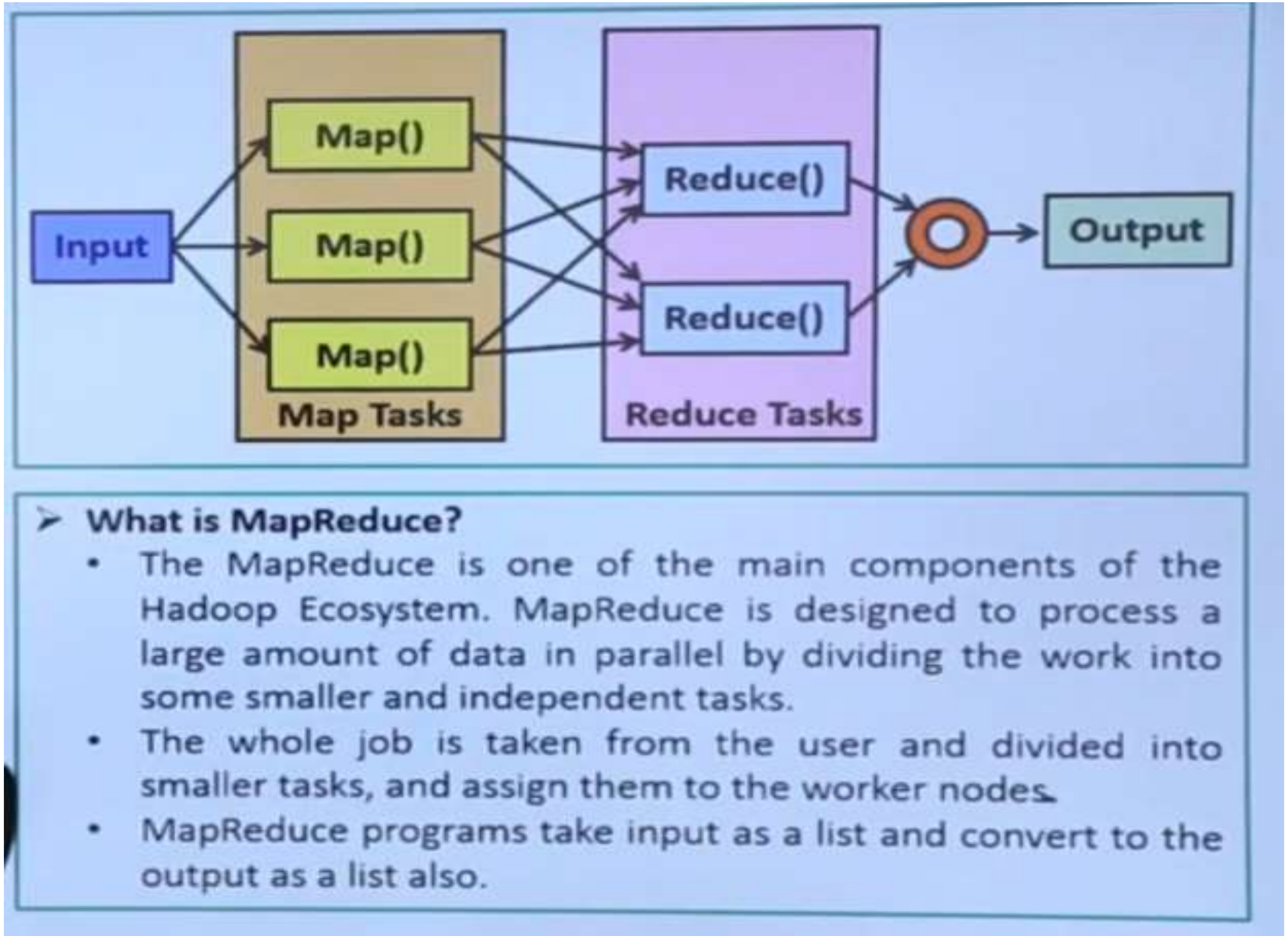
- **The Reduce Task:**
  - The Reducer takes the key-value pair, which is created by the mapper as input. The key-value pairs are sorted by the key elements.
  - In the reducer, we perform the sorting, aggregation or summation type jobs.
- **How MapReduce task works?**
  - The given inputs are processed by the user-defined methods. All different business logics are working on the mapper section. Mapper generates intermediate data and reducer takes them as input. The data are processed by user-defined function in the reducer section. The final output is stored in HDFS.



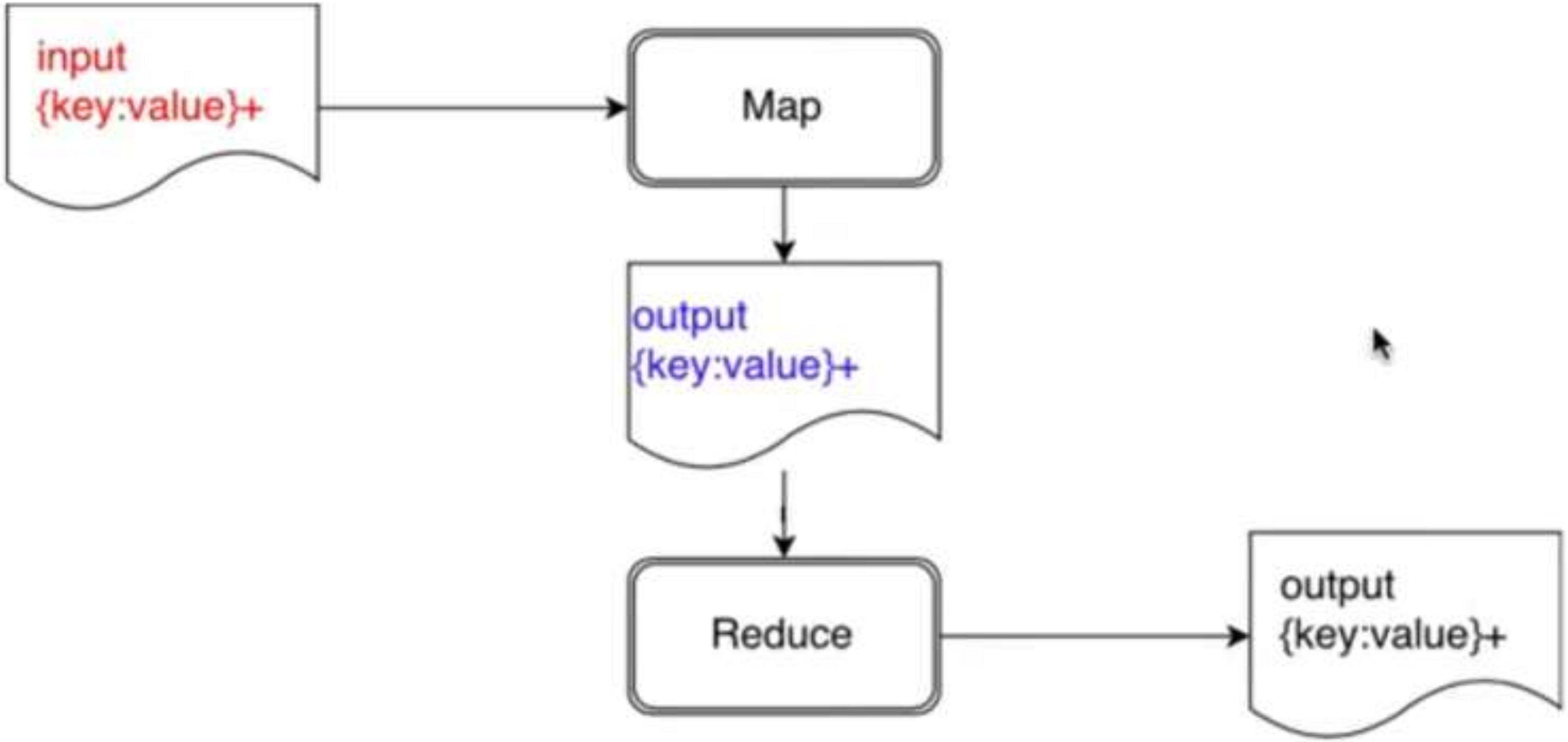
INTRODUCTION TO MAP REDUCE

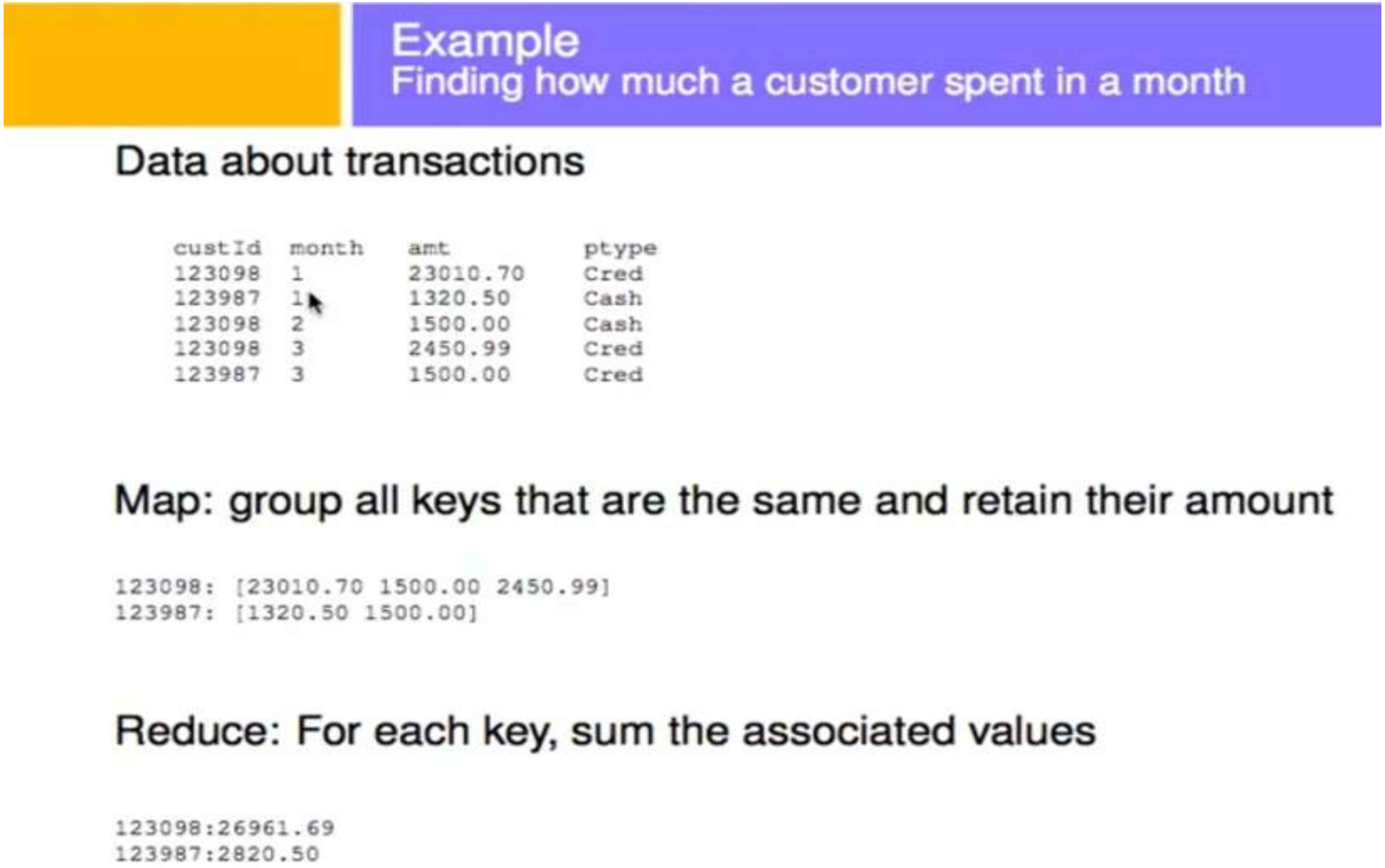
- MapReduce sends code or distributed to the data, instead of bringing data to the actual code.
- MapReduce works by breaking the processing into two phases:
  - The Map phase
  - The Reduce phase
- Entire MapReduce concept is based on Key Value pairs.
- The Programmer also specifies two functions:
  - The Map function
  - The Reduce function
- There is a Sort and Shuffle phase in between.


Input data
Map
Shuffle, sort and reduce
Final output



- **What is MapReduce?**
  - The MapReduce is one of the main components of the Hadoop Ecosystem. MapReduce is designed to process a large amount of data in parallel by dividing the work into some smaller and independent tasks.
  - The whole job is taken from the user and divided into smaller tasks, and assign them to the worker nodes.
  - MapReduce programs take input as a list and convert to the output as a list also.


input
{key:value}+
Map
output
{key:value}+
Reduce
output
{key:value}+



Example
Finding how much a customer spent in a month

Data about transactions

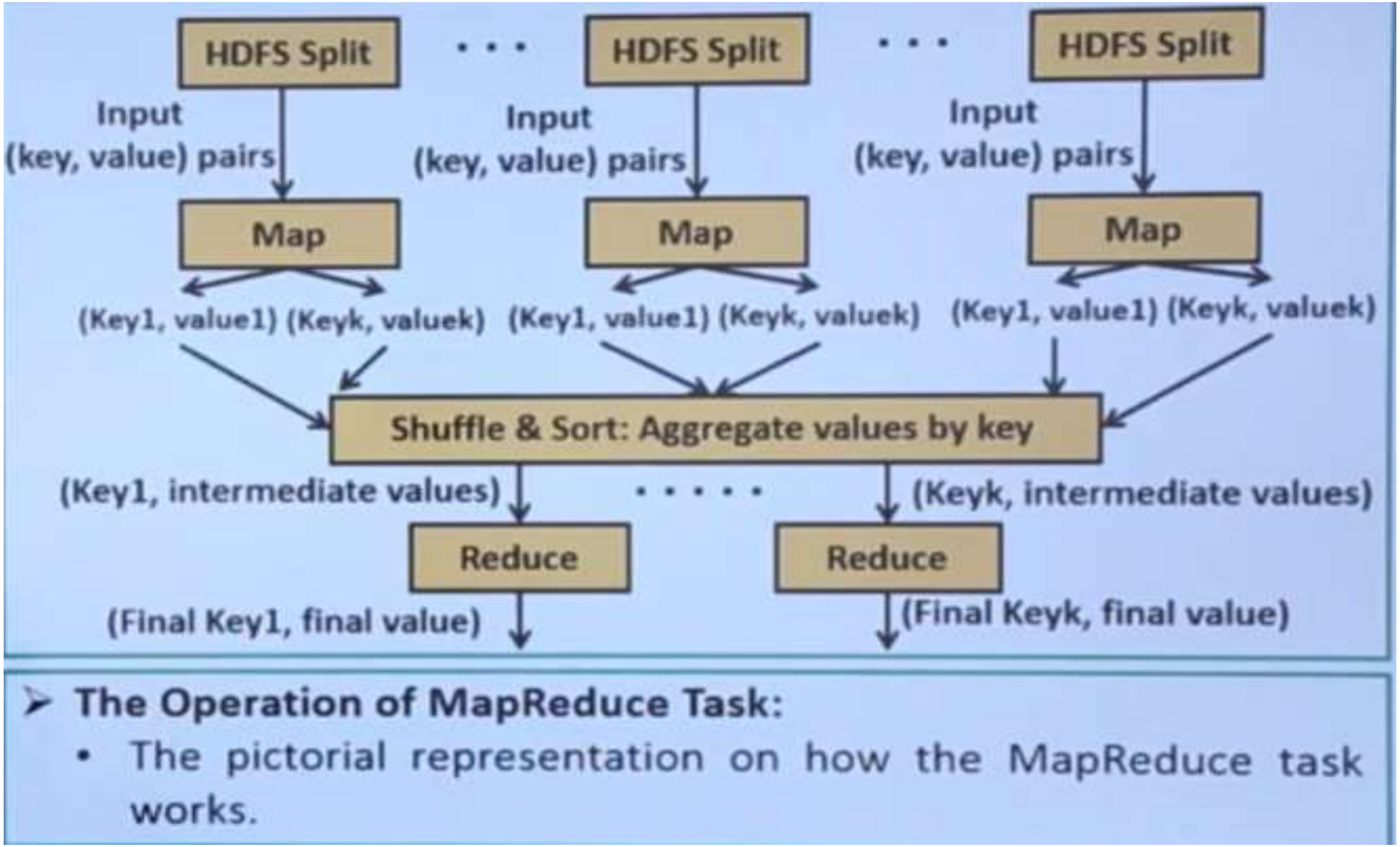
custId	month	amt	ptype
123098	1	23010.70	Cred
123987	1	1320.50	Cash
123098	2	1500.00	Cash
123098	3	2450.99	Cred
123987	3	1500.00	Cred

Map: group all keys that are the same and retain their amount

```
123098: [23010.70 1500.00 2450.99]
123987: [1320.50 1500.00]
```

Reduce: For each key, sum the associated values

```
123098:26961.69
123987:2820.50
```



HDFS Split . . . HDFS Split . . . HDFS Split

Input (key, value) pairs

Input (key, value) pairs

Input (key, value) pairs

Map

Map

Map

(Key1, value1) (Keyk, valuek)

(Key1, value1) (Keyk, valuek)

(Key1, value1) (Keyk, valuek)

Shuffle & Sort: Aggregate values by key

(Key1, intermediate values) . . . . . (Keyk, intermediate values)

Reduce

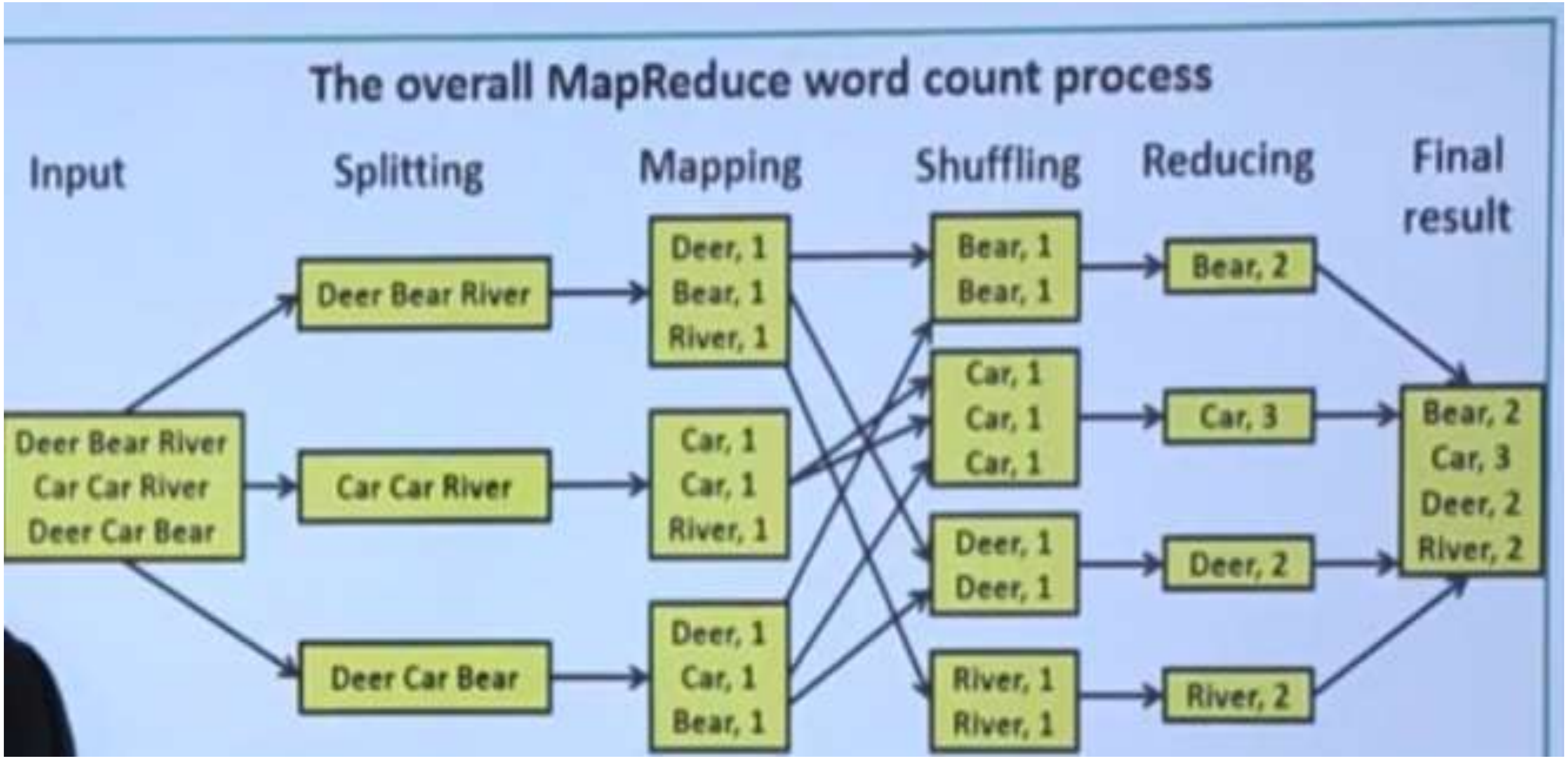
Reduce

(Final Key1, final value)

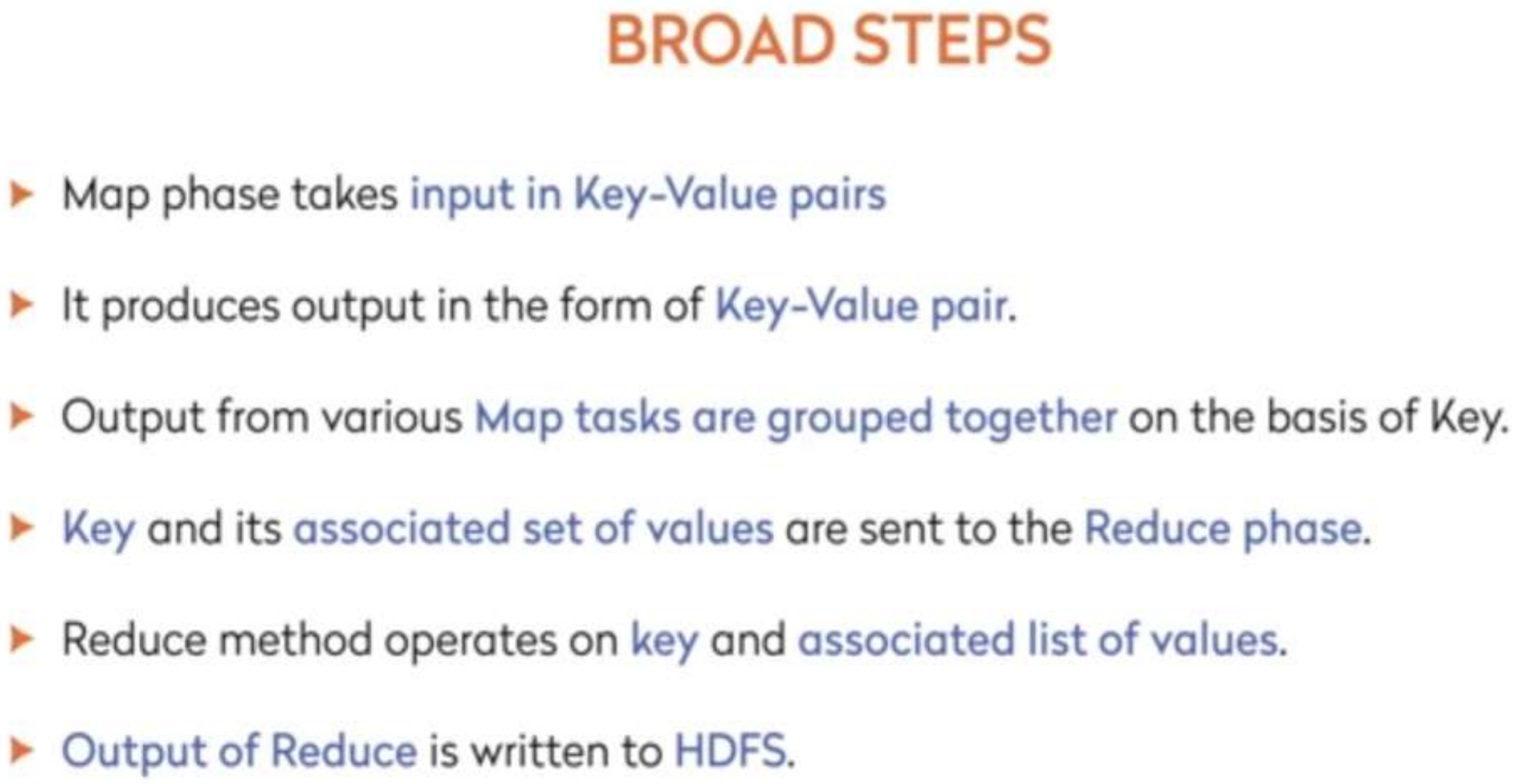
(Final Keyk, final value)

➢ The Operation of MapReduce Task:

- The pictorial representation on how the MapReduce task works.



The overall MapReduce word count process



BROAD STEPS

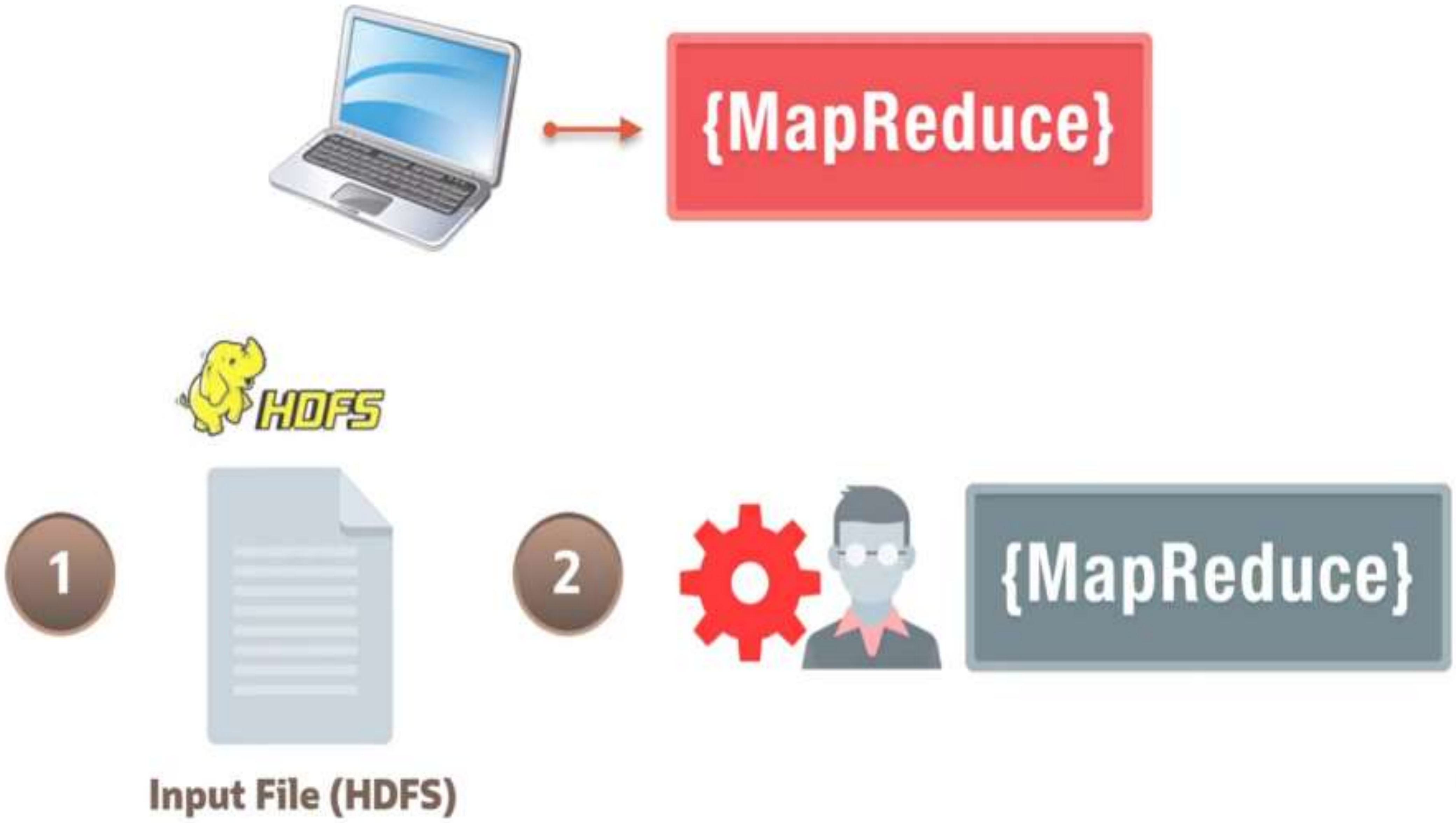
- Map phase takes input in Key-Value pairs
- It produces output in the form of Key-Value pair.
- Output from various Map tasks are grouped together on the basis of Key.
- Key and its associated set of values are sent to the Reduce phase.
- Reduce method operates on key and associated list of values.
- Output of Reduce is written to HDFS.

MapReduce Framework

- Integrates with HDFS and provides the same benefits for parallel data processing
- Parallelizes and distributes computations to where the data is stored (data locality)
- The framework:
  - Schedules and monitors tasks, and re-executes failed tasks
  - Hides complex distributed computing tasks from the developer
  - Enables developers to focus on writing the Map and Reduce functions

HDFS
hadoop
Map Reduce

MapReduce Jobs



{MapReduce}

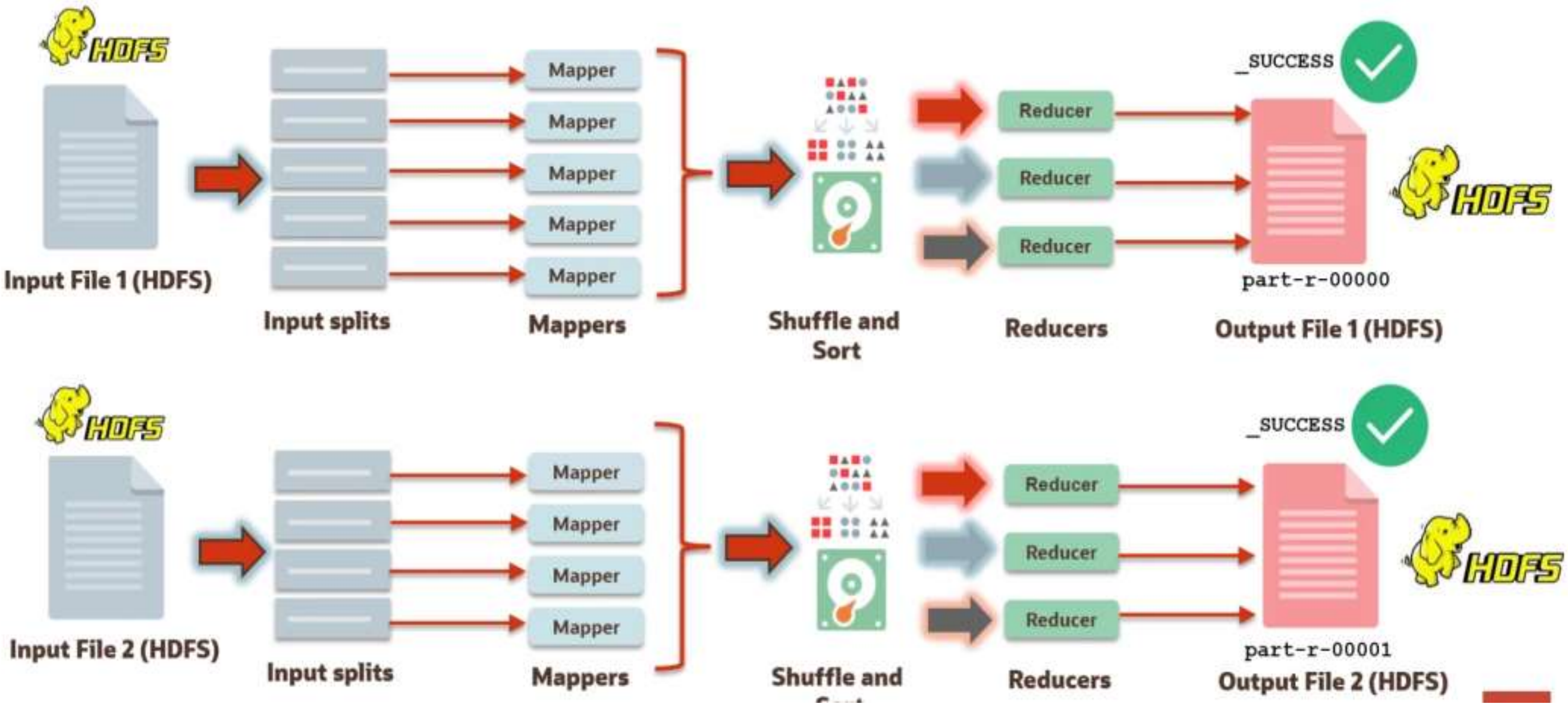
1

2

{MapReduce}

Input File (HDFS)

MapReduce Jobs



Data Locality

Data Locality in Hadoop means moving computation close to data rather than moving data towards computation. Hadoop stores data in HDFS, which splits files into blocks and distribute among various data nodes.

When a mapReduce job is submitted, it is divided into map jobs and reduce jobs.

A Map job is assigned to a datanode according to the availability of the data, ie it assigns the task to a datanode which is closer to or stores the data on its local disk.

Data locality refers the process of placing computation near to data , which helps in high throughput and faster execution of data.

1. Data Local

If a map task is executing on a node which has the input block to be processed, its called data local.

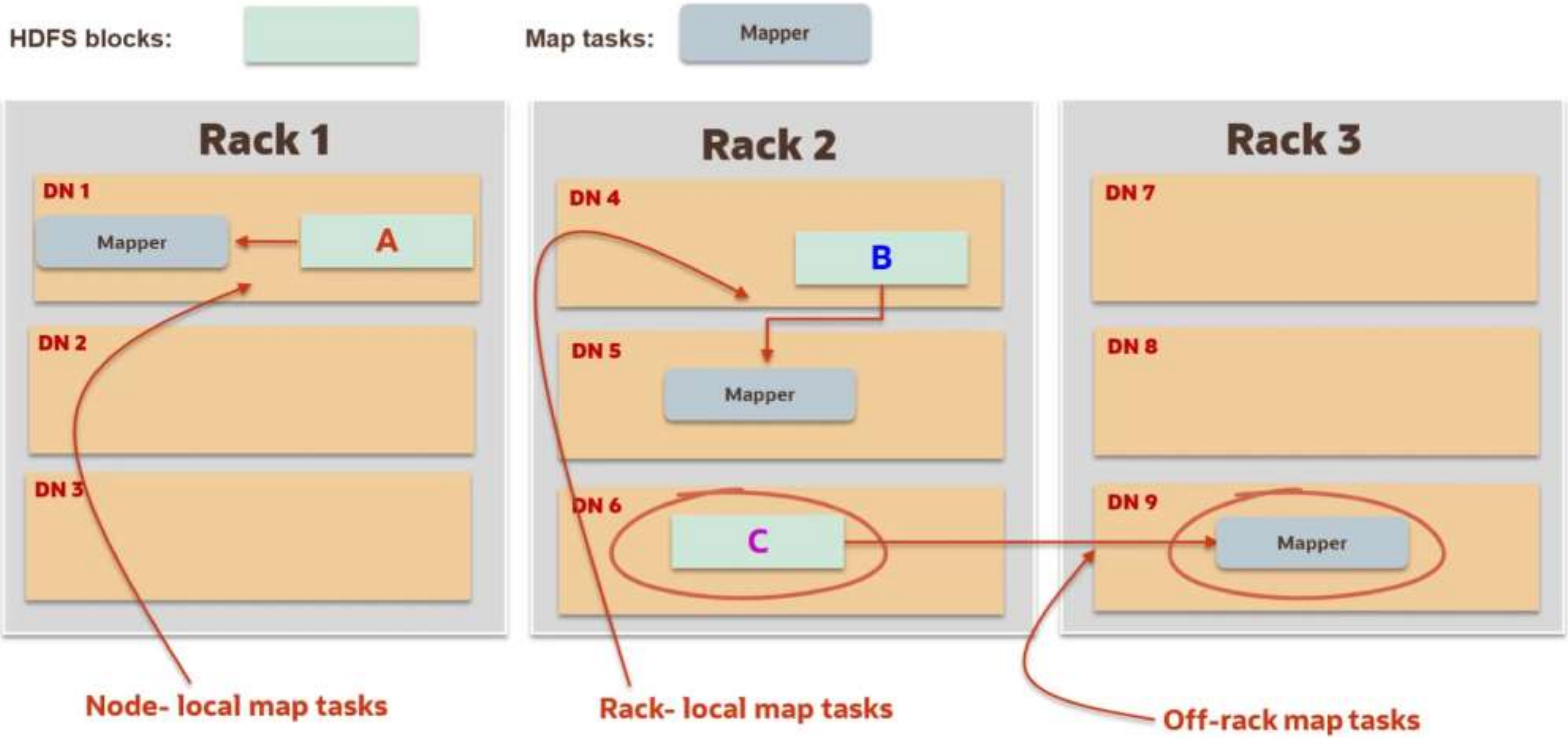
2. Intra- Rack

Its always not possible to run map task on the same node where data is located due to network constraints. In that case, mapper runs on another machine, but on the same rack. So the data need to be moved between the nodes for execution.

3. Inter-Rack

In certain cases Intra- Rack local is also not possible. In such cases, the mapper will execute from a different rack.In order to execute the mapper, the data need to be copied from the node which stores the data to the node which is executing the mapper between the racks.

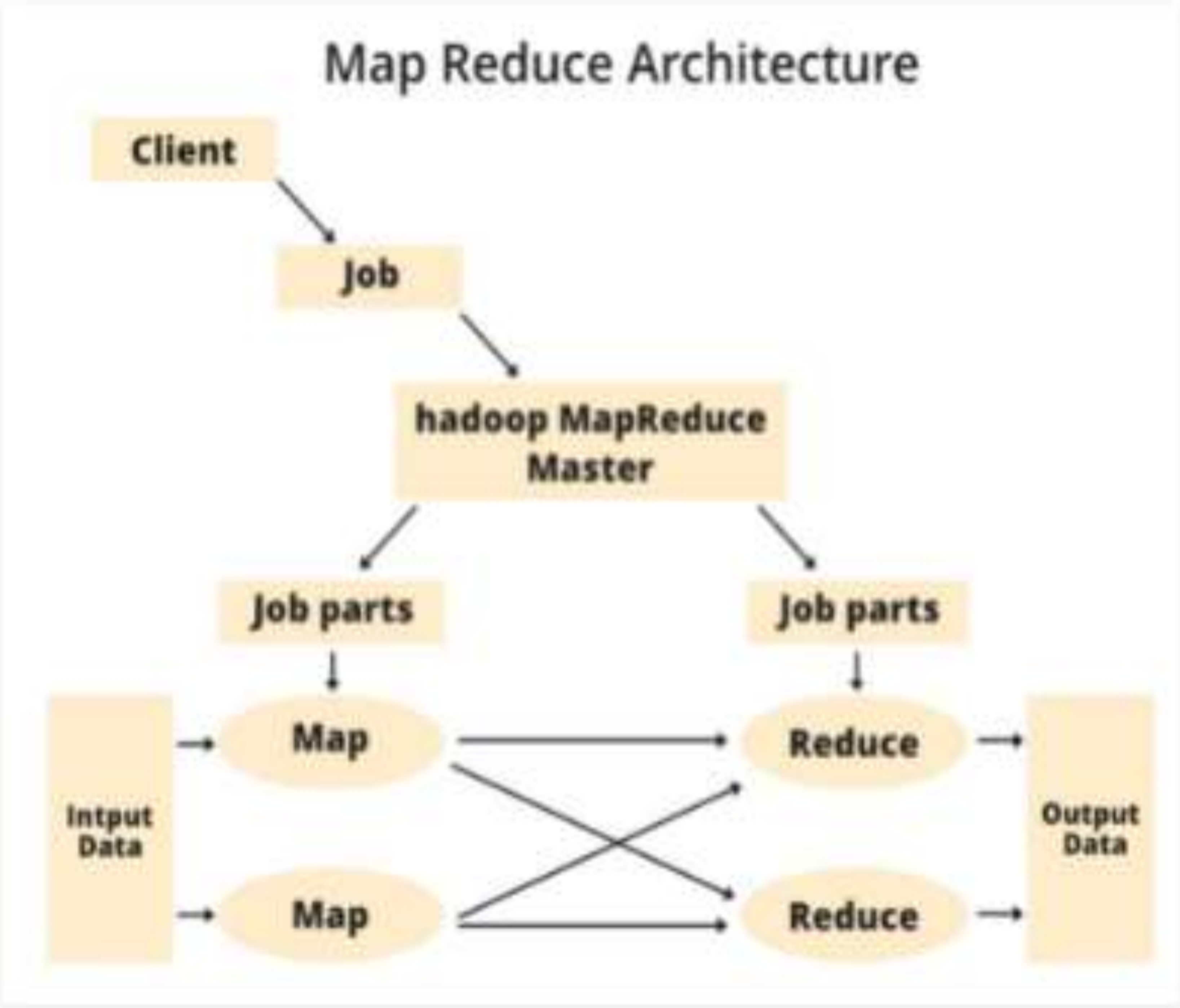
Data Locality Optimization in Hadoop



MAPREDUCE ARCHITECTURE

MapReduce and HDFS are the two major components of Hadoop which makes it so powerful and efficient to use. MapReduce is a programming model used for efficient processing in parallel over large data-sets in a distributed manner. The data is first split and then combined to produce the final result. The libraries for MapReduce is written in so many programming languages with various different-different optimizations. The purpose of MapReduce in Hadoop is to Map each of the jobs and then it will reduce it to equivalent tasks for providing less overhead over the cluster network and to reduce the processing power.

MapReduce Architecture:



Components of MapReduce Architecture:

Client: The MapReduce client is the one who brings the Job to the MapReduce for processing. There can be multiple clients available that continuously send jobs for processing to the Hadoop MapReduce Manager.

Job: The MapReduce Job is the actual work that the client wanted to do which is comprised of so many smaller tasks that the client wants to process or execute.

Hadoop MapReduce Master: It divides the particular job into subsequent job-parts.

Job-Parts: The task or sub-jobs that are obtained after dividing the main job. The result of all the job-parts combined to produce the final output.

Input Data: The data set that is fed to the MapReduce for processing.

Output Data: The final result is obtained after the processing.

How Job tracker and the task tracker deal with MapReduce:

Job Tracker: The work of Job tracker is to manage all the resources and all the jobs across the cluster and also to schedule each map on the Task Tracker running on the same data node since there can be hundreds of data nodes available in the cluster.

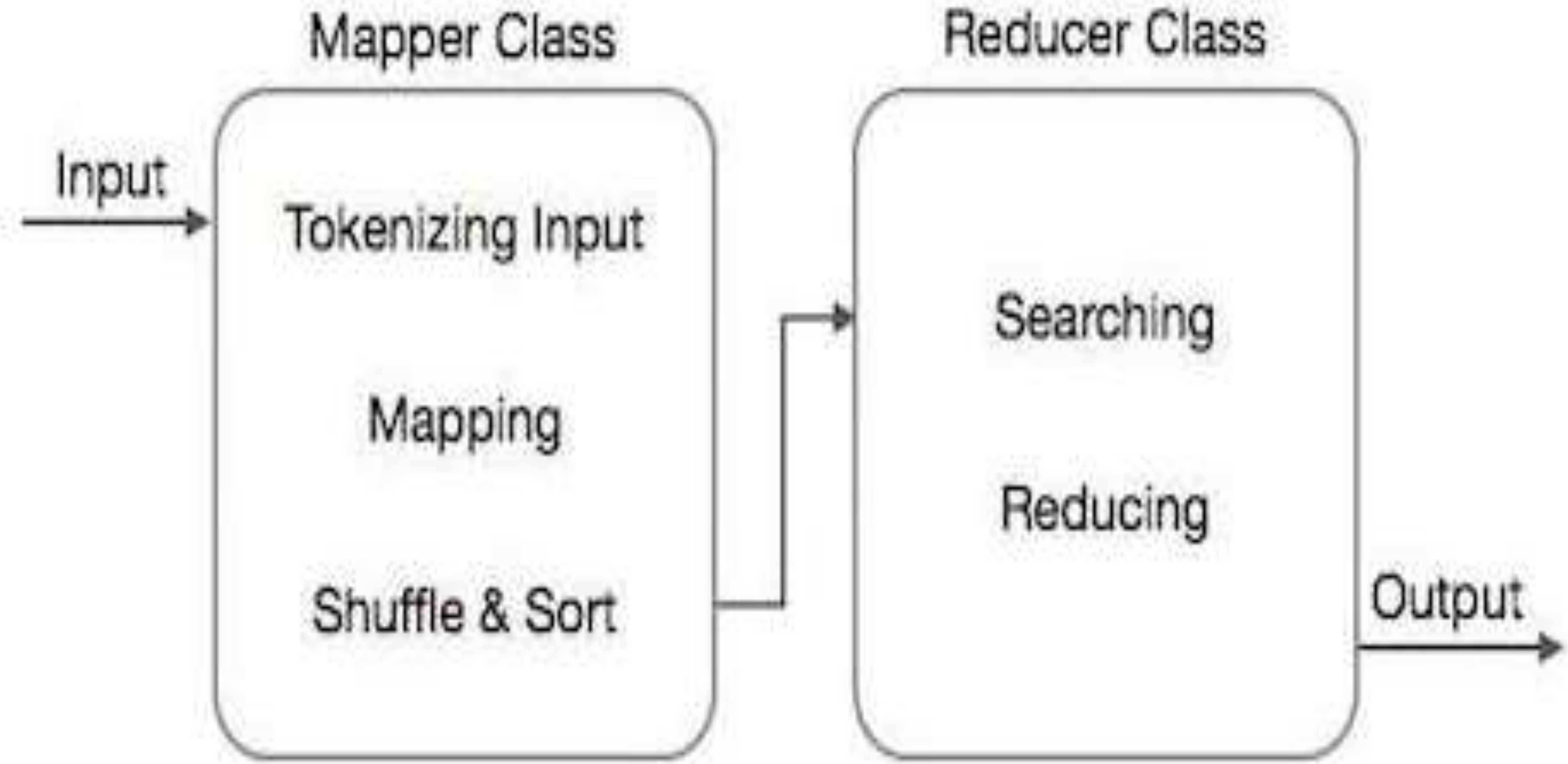
Task Tracker: The Task Tracker can be considered as the actual slaves that are working on the instruction given by the Job Tracker. This Task Tracker is deployed on each of the nodes available in the cluster that executes the Map and Reduce task as instructed by Job Tracker.

There is also one important component of MapReduce Architecture known as Job History Server.

The Job History Server is a daemon process that saves and stores historical information about the task or application, like the logs which are generated during or after the job execution are stored on Job History Server.

Mapreduce Algorithm

- The MapReduce algorithm contains two important tasks, namely Map and Reduce.
- The map task is done by means of Mapper Class
- The reduce task is done by means of Reducer Class.
- Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.


Mapper Class
Reducer Class
Input
Tokenizing Input
Mapping
Shuffle & Sort
Searching
Reducing
Output

Hadoop MapReduce Applications

- **1 — Social Networks**
- We are social networking users like Facebook, Twitter, and LinkedIn to connect with our friends and community.
- Many of the features, such as who visited your LinkedIn profile, who read your post on Facebook or Twitter, can be evaluated using the MapReduce, programming model.
- **2 — Entertainment**
- Netflix uses Hadoop and MapReduce to solve problems such as discovering the most popular movies, based on what you watched, what do you like? Providing suggestions to registered users taken into account their interests.
- MapReduce can determine how users are watching movies, analyzing their logs and clicks.

3 — Electronic Commerce

Many e-commerce providers, such as the Amazon, Walmart, and eBay, use the MapReduce programming model to identify favorite products based on users' interests or buying behavior.

It includes creating product recommendation mechanisms for e-commerce catalogs, analyzing site records, purchase history, user interaction logs, and so on.

4 — Fraud Detection

Hadoop and MapReduce are used in the financial industries, including companies such as banks, insurance providers, payment locations for fraud detection, trend identification or business metrics through transaction analysis.

Banks analyze the data of the credit card and the related expenses, for categorization of these expenses and make recommendations for different offers, analyzing anonymous purchasing behavior.

5 — Search and Advertisement Mechanisms

We can utilize it to analyze and understand search behavior, trends, and missing results for specific keywords.

Google and Yahoo use MapReduce to understand users' behavior, such as popular searches over a period of an event such as presidential elections.

Google AdWords uses MapReduce to understand the impressions of ads served, click-through rates, and engagement behavior of users.

6 — Data Warehouse

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