GANDHINAGAR INSTITUTE OF TECHNOLGY

Information Technology Department

Big Data Analytics (2171607)

Understanding i/o and o/p of MapReduce - Data Serialization

Prepared By:

Patel Maulik Satishkumar (150124116006)

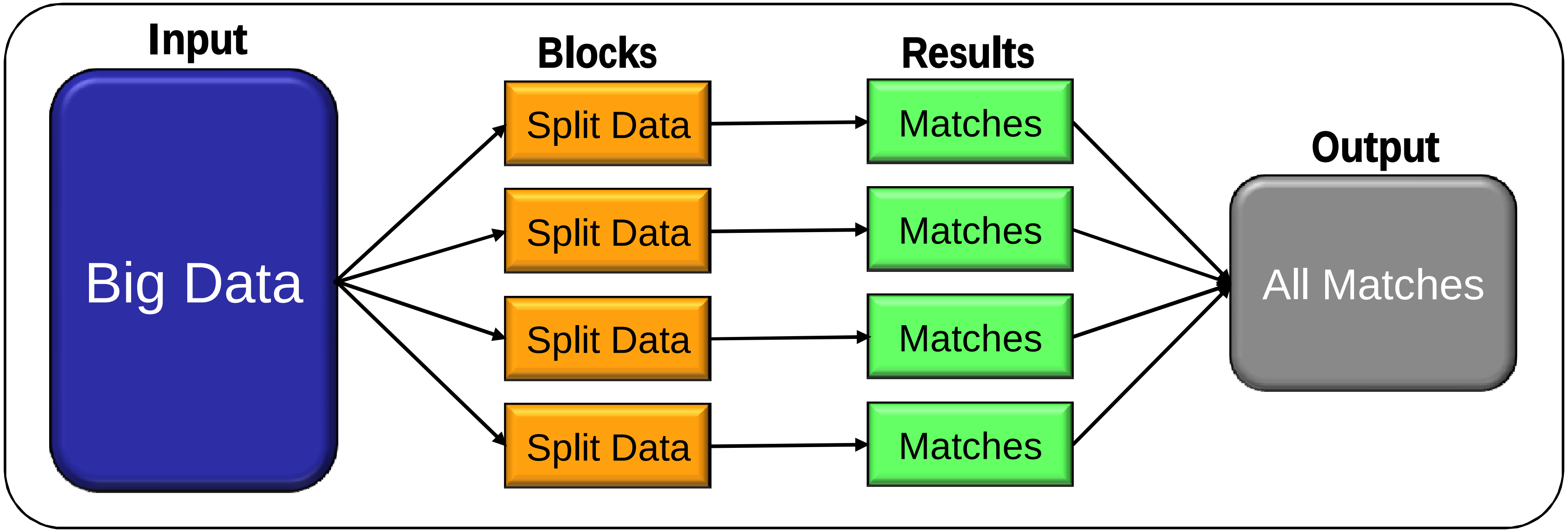
Guided By: **Prof. Prakash B. Patel**

Content

- ❑ Why MapReduce ?
- ❑ What is MapReduce ?
- ❑ Inputs and Outputs of MapReduce
- ❑ MapReduce Example
- ❑ What is Data Serialization ?
- ❑ Data Serialization in Java & Hadoop

Why **MapReduce** ?

- ❑ **Traditional Approach** of Big Data Processing



- ✓ High Processing Power is required
- ✓ Large no. of Computer Systems are required
- ✓ We have to write code for individual block.
- ✓ Splitting & Merging is difficult to manage

Why **MapReduce** ?

❑ Process lots of data.

- ✓ Google processed about **24 petabytes** of data per day in 2009.

❑ A **single machine** cannot serve all the data

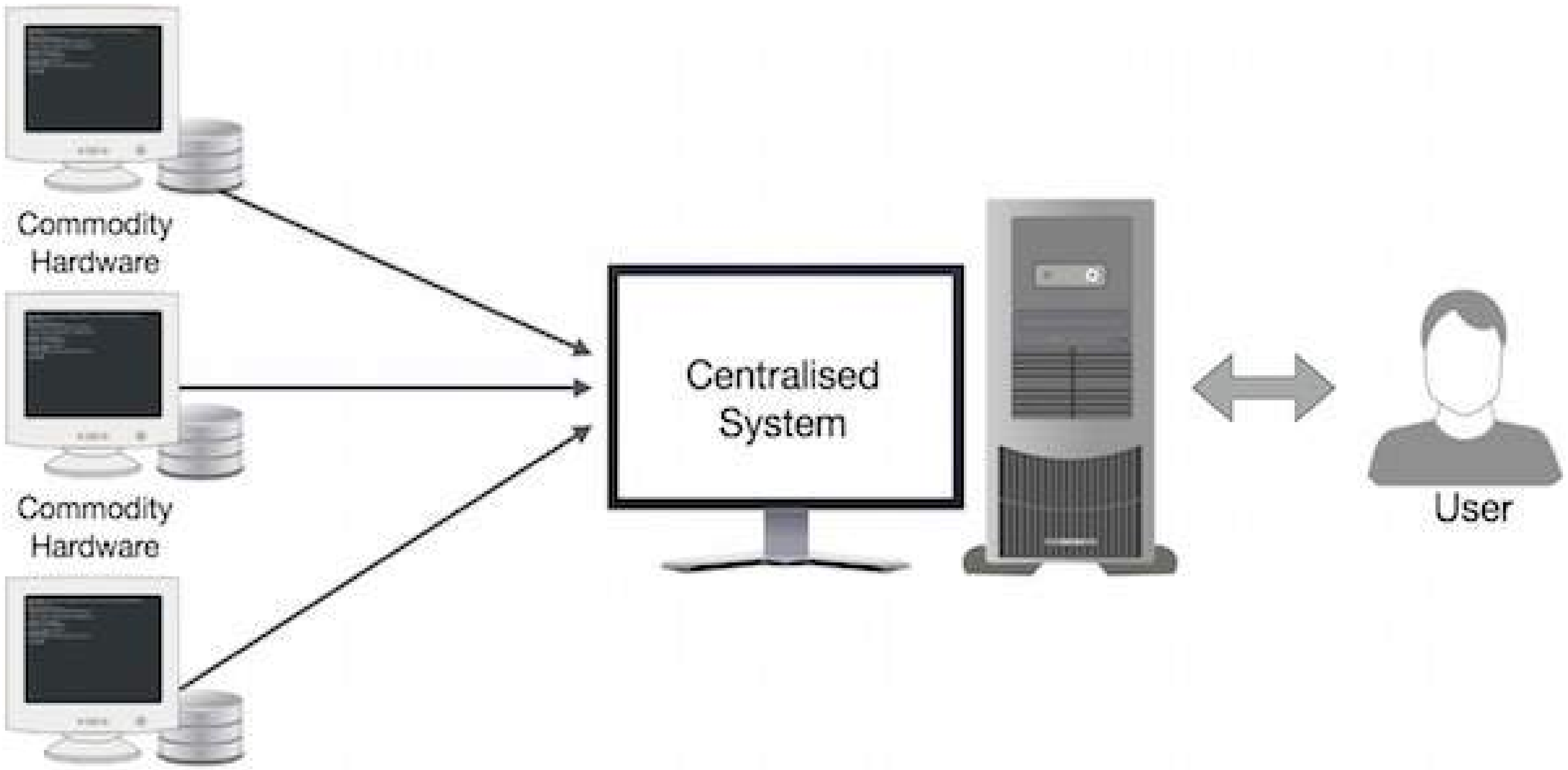
- ✓ You need a ***distributed system*** to *store and process* in parallel.

❑ Parallel programming?

- ✓ **Threading** is hard!
- ✓ How do you facilitate **communication** between nodes?
- ✓ How do you scale to **more machines**?
- ✓ How do you handle **machine failures**?

What is **MapReduce** ?

- MapReduce divides a task into small parts and assigns them to many computers. Later, the results are collected at one place and integrated to form the result dataset.



What is **MapReduce** ?

❑ Hadoop MapReduce is a software framework for easily writing applications which process huge amounts of data in-parallel on large clusters of commodity hardware in a reliable and fault-tolerant manner.

MapReduce = **Map** + **Reduce**

(Mapper) (Reducer)

- ✓ Mappers read in data from the file system, and output (typically) modified data.
- ✓ Reducers collect all of the mappers output on the keys, and output (typically) reduced data.
- ✓ The outputted data is written to disk.
- ✓ All data is in terms of key value pairs.

What is **MapReduce** ?

- ❑ The MapReduce algorithm consists two important tasks, namely **Map** and **Reduce**.

1) Map Task : It takes set of data and convert it into another set of data, where individual elements are broken down into tuples (key-value pairs).

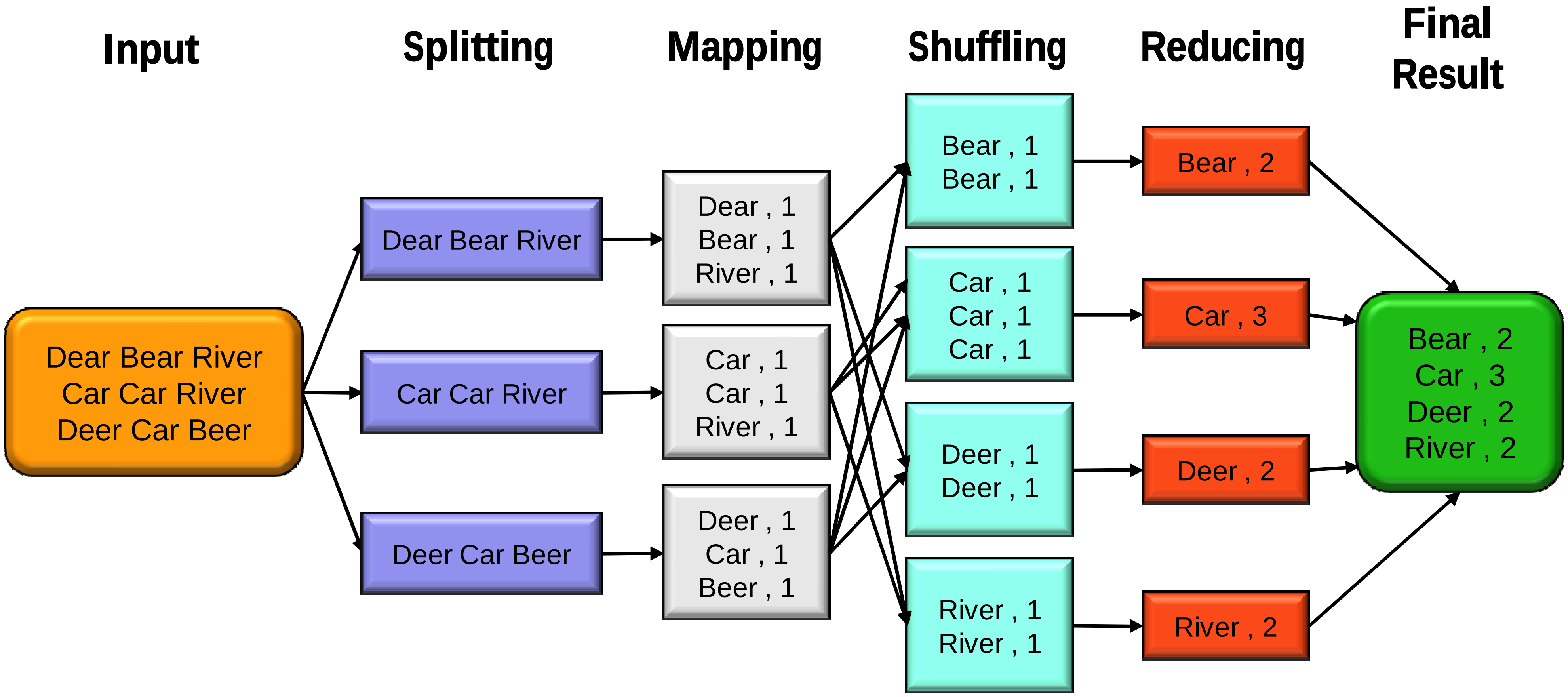
2) Reduce Task : It takes the outputs from the Map as an input and combines those data tuples (key-value pairs) into a smaller set of tuples. The reduce task is always performed after the map job.

Inputs & Outputs of MapReduce

- The MapReduce framework operates on < ***key, value*** > pairs , that is , the framework views the input as a set of < key, value > pairs and produces a set of < key, value > pairs as the output.

Task	Input	Output
Map	< k1 , v1 >	list (< k2 , v2 >)
Reduce	< k2 , list(v2) >	list (< k3 , v3 >)

MapReduce Example



The overall MapReduce for Word Count Process

What is **Data serialization** ?

- Serialization is the *process of translating* ***data structures or objects state*** *into* ***binary or textual form*** to transport the data over network or to *store on some* **persistent storage**.

- Once the data is transported over network or retrieved from the persistent storage, it needs to be deserialized again.

- Serialization is termed as marshalling and deserialization is termed as unmarshalling.

Data serialization in **Java** & **Hadoop**

- ❑ **Java** provides mechanism, called **object serialization** where an object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the types of data stored in the object.

- ❑ In **Hadoop**, the concept of serialization is used for *Interprocess communication* and *Persistence storage*.

- ❑ Persistence storage is a digital storage facility that does not lose its data with the lose of power supply.

  Ex:- *Magnetic disks* and *Hard Disk Drives*.

THANK YOU