MapReduce

BY
PROF. PRAKASH PATEL
ASST.PROF.-IT DEPT.
GIT

What is MapReduce?

- MapReduce is a framework using which we can write applications to process huge amounts of data
- MapReduce is a processing technique and a program model for distributed computing based on java.
- The MapReduce algorithm contains two important tasks, namely Map and Reduce.
- It is used in parallel, on large clusters of commodity hardware in a reliable manner.

Continue

- Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).
- *Reduce* task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples.
- MapReduce implies, the reduce task is always performed after the map job.
- The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes.

- Under the MapReduce model, the data processing primitives are called mappers and reducers.
- we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster
- This simple scalability is what has attracted many programmers to use the MapReduce model.

The Algorithm

- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.

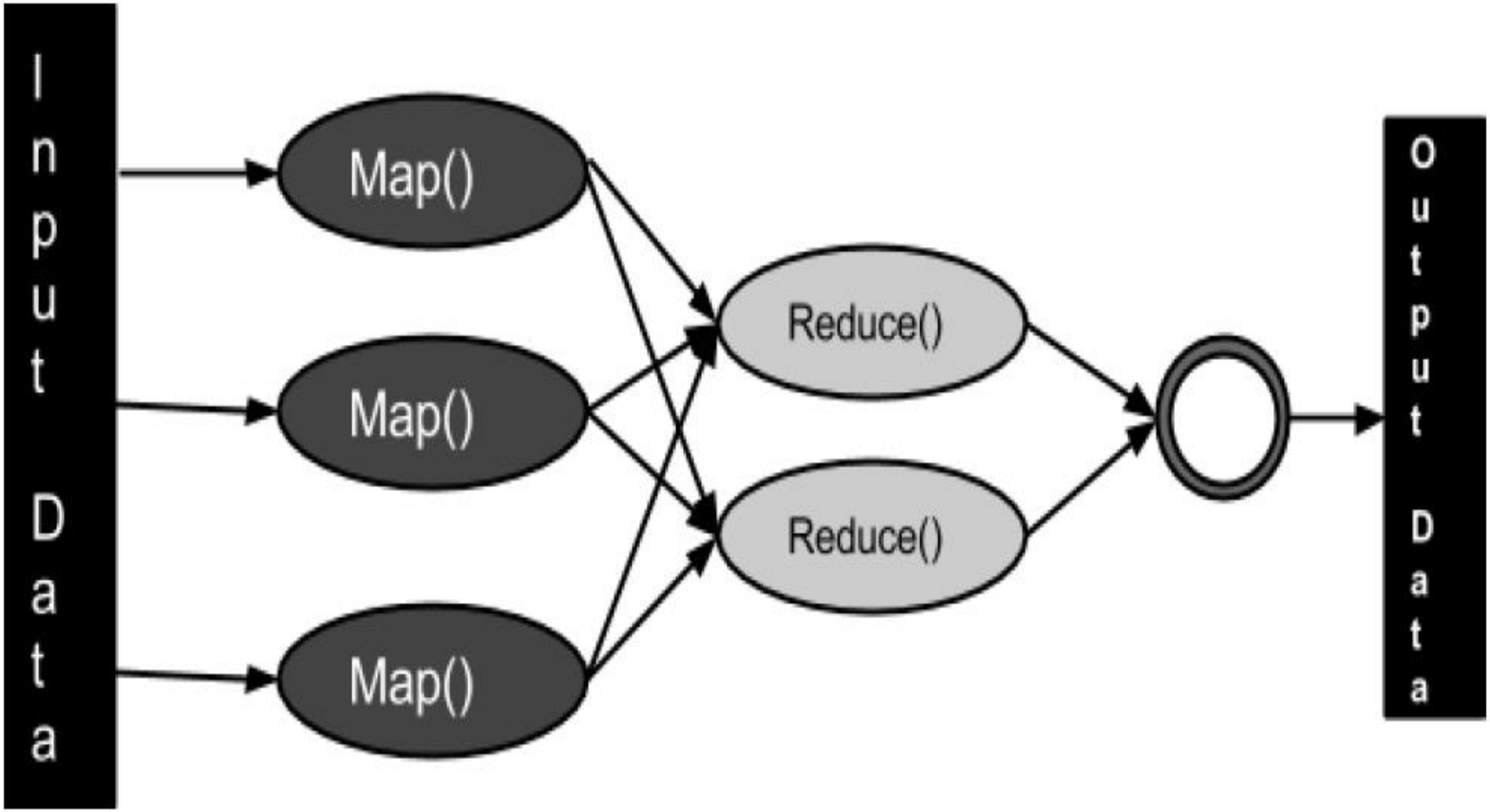
- **Map stage**: The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS).

- The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

- **Reduce stage**: This stage is the combination of the **Shuffle** stage and the **Reduce** stage.
- The Reducer's job is to process the data that comes from the mapper.
- After processing, it produces a new set of output, which will be stored in the HDFS.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

The MapReduce framework

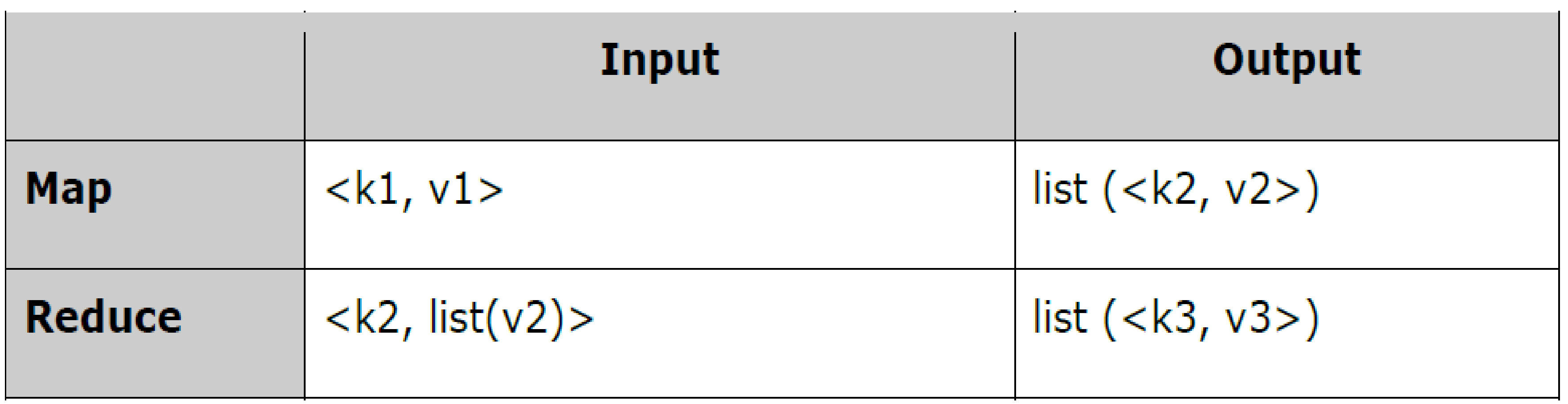


Inputs and Outputs

- The MapReduce framework operates on <key, value> pairs
- the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job
- The key and the value classes should be in serialized manner by the framework
- The key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework.

MapReduce I/O

- Input and Output types of a **MapReduce job**: (Input) <k1, v1> **->** map **->** <k2, v2>**->** reduce **->** <k3, v3> (Output).



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