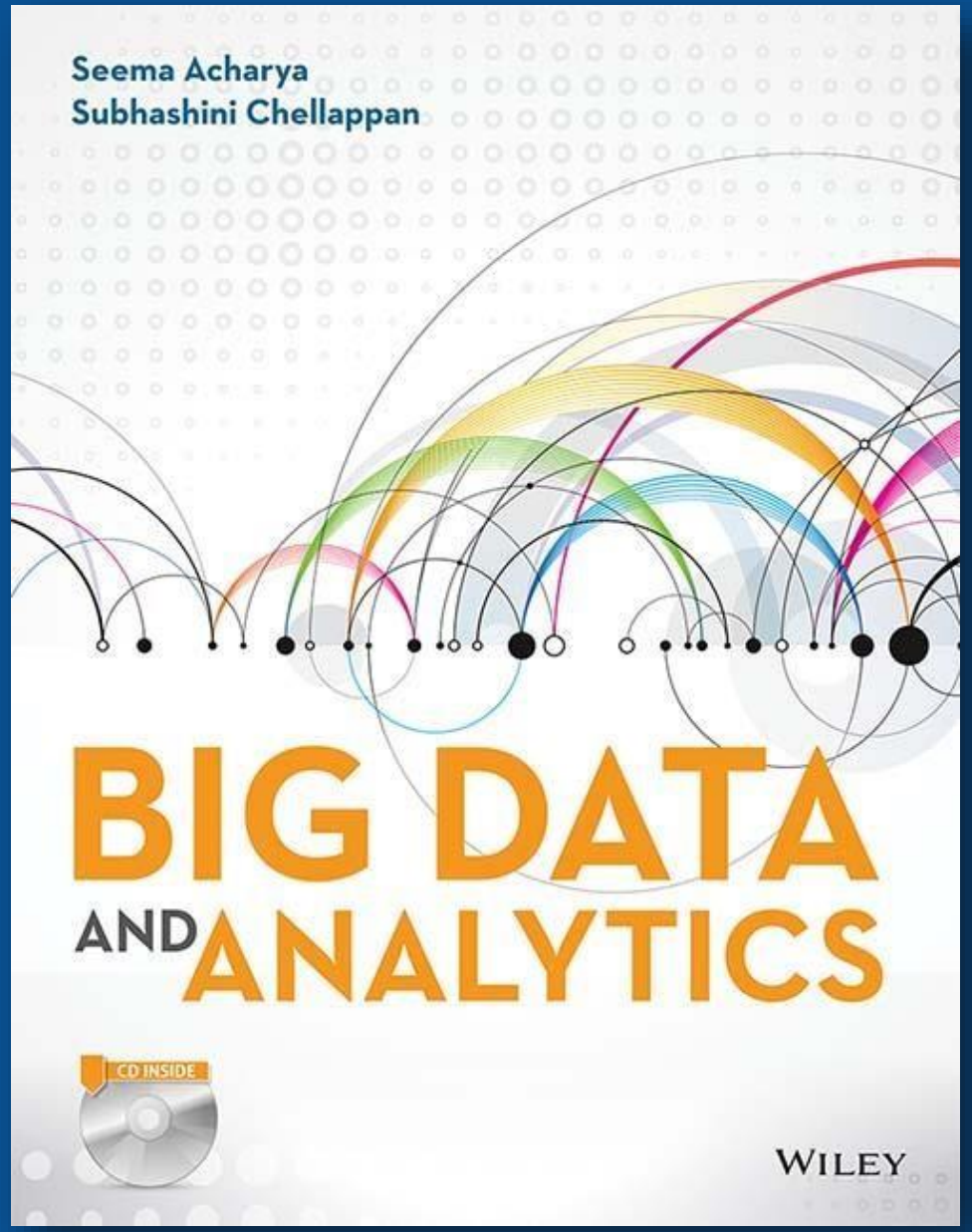


Big Data And Analytics

Seema Acharya
Subhashini Chellappan

Chapter 8

Introduction to MapReduce Programming

Learning Objectives and Learning Outcomes

Learning Objectives	Learning Outcomes
Introduction to MapReduce Programming 1. To study the optimization techniques of MapReduce Programming. 2. To learn the combiner, partitioner technique. 3. To study sorting, searching and compression.	a) To comprehend the reasons behind the popularity of MapReduce Programming. b) To be able to perform combiner and partitioner program. c) To comprehend searching, sorting and compression.

Session Plan

Lecture time	90 minutes
Q/A	5 minutes

Agenda

- Introduction
- Mapper
  - RecordReader
  - Map
  - Combiner
  - Partitioner
- Reducer
  - Shuffle
  - Sort
  - Reduce
  - Output Format
- Combiner
- Partitioner
- Searching
- Sorting
- Compression

Introduction

Introduction

In MapReduce Programming, Jobs (Applications) are split into a set of map tasks and reduce tasks. Then these tasks are executed in a distributed fashion on Hadoop cluster.

Each task processes small subset of data that has been assigned to it. This way, Hadoop distributes the load across the cluster.

MapReduce job takes a set of files that is stored in HDFS (Hadoop Distributed File System) as input.

Map task takes care of loading, parsing, transforming and filtering.

Reduce task is responsible for grouping and aggregating data that is produced by map tasks to generate final output

Mapper

Mapper

A mapper maps the input key–value pairs into a set of intermediate key–value pairs. Maps are individual tasks that have the responsibility of transforming input records into intermediate key–value pairs.

Mapper Consists of following phases:

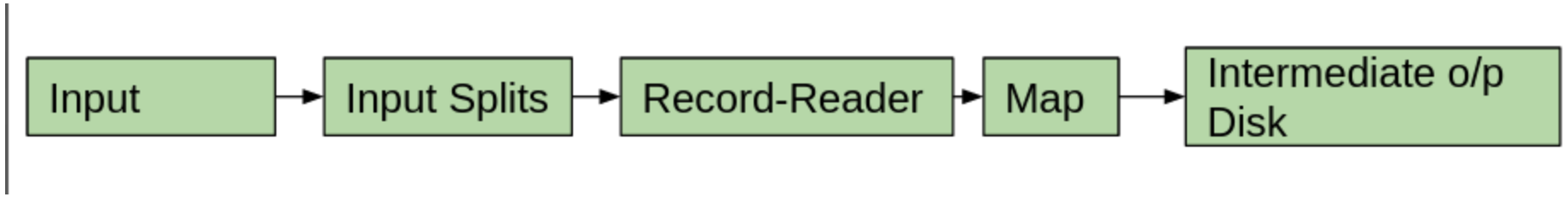
- **RecordReader**
- **Map**
- **Combiner**
- **Partitioner**

- **RecordReader**

It takes the byte-oriented view of input, provided by the InputSplit and presents as a record-oriented view for Mapper. It uses the data within the boundaries that were created by the InputSplit and creates Key-value pair.

- **Map**

The map function works on key-value pair (from RecordReader) and generates an intermediate key-value pairs.



- **Combiner**

The combiner in MapReduce is also known as ‘Mini-reducer’. It takes the intermediate key-value pairs provided by mapper and applies user-specific aggregate function only to that mapper. It runs after the mapper and before the Reducer and its use is optional but provides high performance in terms of network band width and disk space.

- **Partitioner**

The **Partitioner** in MapReduce controls the partitioning of the key of the intermediate mapper output. It splits them into shard and sends the shard to the particular reducer as per the user specific code. Usually the key with the same values goes to the same reducer. (By hash function, key or a subset of the key is used to derive the partition). A total number of partitions depends on the number of reduce task

Reducer

Reducer

The primary chore of the Reducer is to reduce a set of intermediate values (the ones that share a common key) to a smaller set of values.
The Reducer has three primary phases:

- Shuffle and Sort

This phase takes the output of all the partitioners and downloads them into the local machine where the reducer is running. Then, these individual data pipes are sorted by keys which [produce larger data list. (grouping similar words)

- Reduce

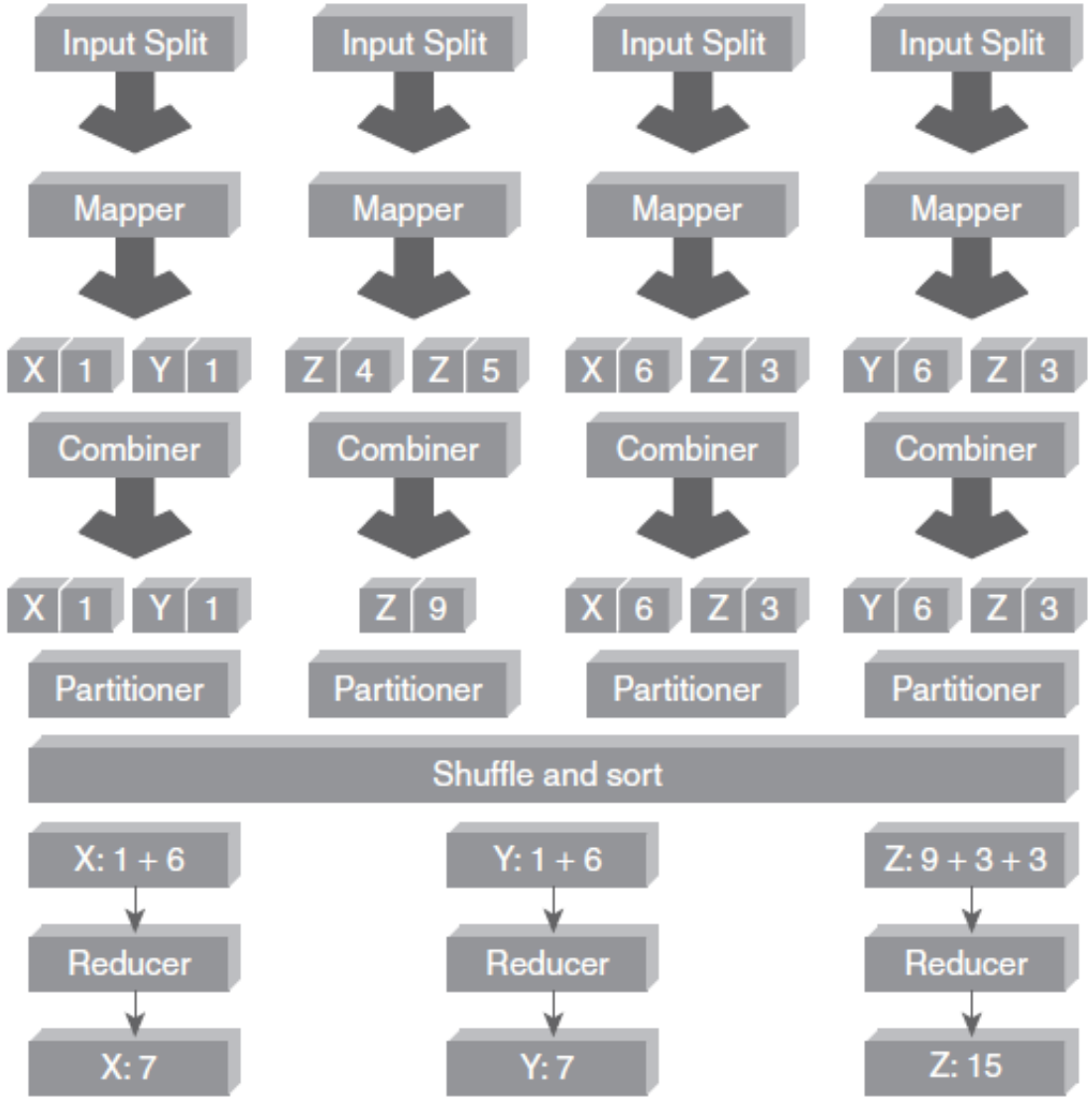
Takes the grouped data produced by Shuffle and Sort, applies reduce function and processes one group at a time. It produces various operations like aggregation, filtering, and combining data. Output of reducer is sent to the output format

- Output Format

It separates key-value pair with rtab and writes it out to the file using record writer

The chores of Mapper, Combiner, Partitioner, and Reducer

The chores of Mapper, Combiner, Partitioner, and Reducer



Combiner

Combiner

It is an optimization technique for MapReduce Job. Generally, the reducer class is set to be the combiner class. The difference between combiner class and reducer class is as follows:

- Output generated by combiner is intermediate data and it is passed to the reducer.
- Output of the reducer is passed to the output file on disk.

Partitioner

Partitioner

The partitioning phase happens after map phase and before reduce phase. Usually the number of partitions are equal to the number of reducers. The default partitioner is hash partitioner.

Searching and Sorting Demo

Compression

In MapReduce programming, you can compress the MapReduce output file. Compression provides two benefits as follows:

1. Reduces the space to store files.
2. Speeds up data transfer across the network.

You can specify compression format in the Driver Program as shown below:

```
conf.setBoolean("mapred.output.compress",true);
conf.setClass("mapred.output.compression.codec",
GzipCodec.class,CompressionCodec.class);
```

Here, codec is the implementation of a compression and decompression algorithm. GzipCodec is the compression algorithm for gzip. This compresses the output file.

Answer a few questions…

Fill in the blanks

1. Partitioner phase belongs ---------- to task.
2. Combiner is also known ----------.
3. RecordReader converts byte-oriented view into ---------- view.
4. MapReduce sorts the intermediate value based on ---------- .
5. In MapReduce Programming, reduce function is applied ---------- group at a time.

Thank You