

Chapter 2: Introduction to Hadoop & Hadoop Architecture (part 2)

By Ms. Tina D'abreo

Content

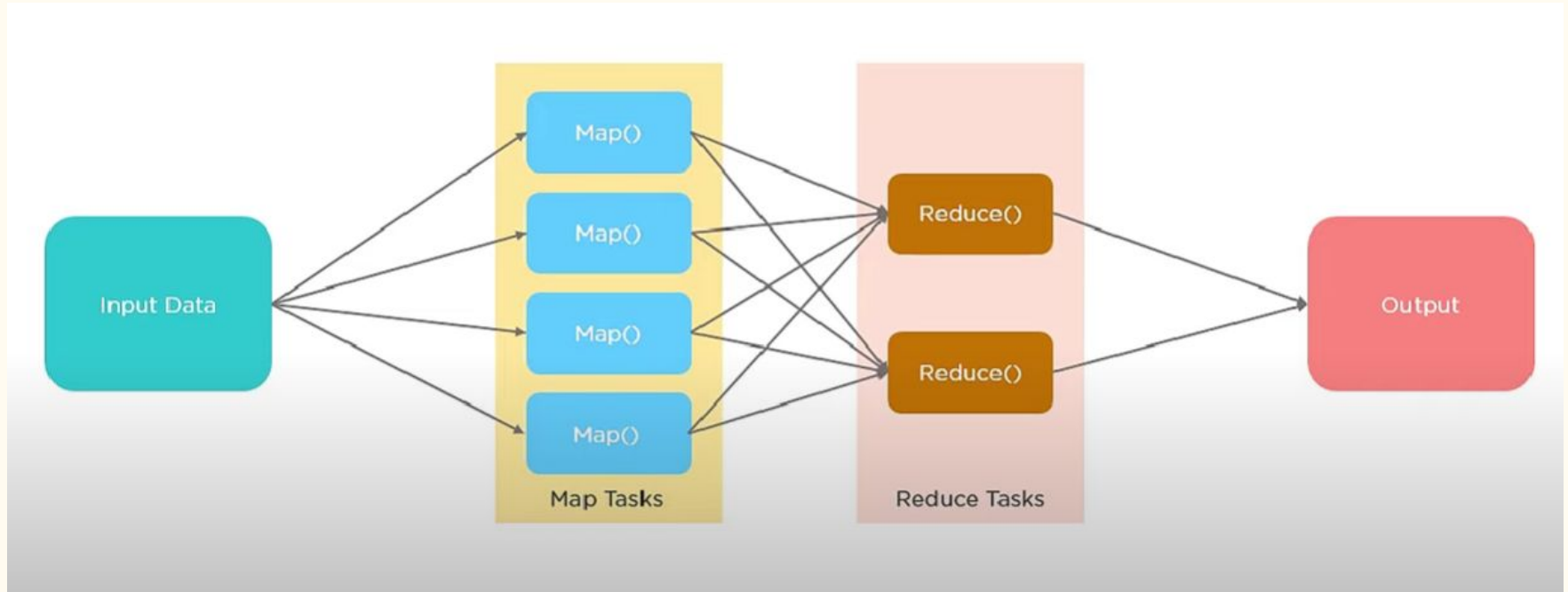
- Big Data – Apache Hadoop & Hadoop EcoSystem
- **Moving Data in and out of Hadoop**
 - **Understanding inputs and outputs of MapReduce Concept of Hadoop**
- HDFS Commands
- **MapReduce**
 - **The Map Tasks**
 - **Grouping by Key**
 - **The Reduce Tasks**
 - **Combiners**
 - **Details of MapReduce Execution**

MapReduce in Hadoop - Overview

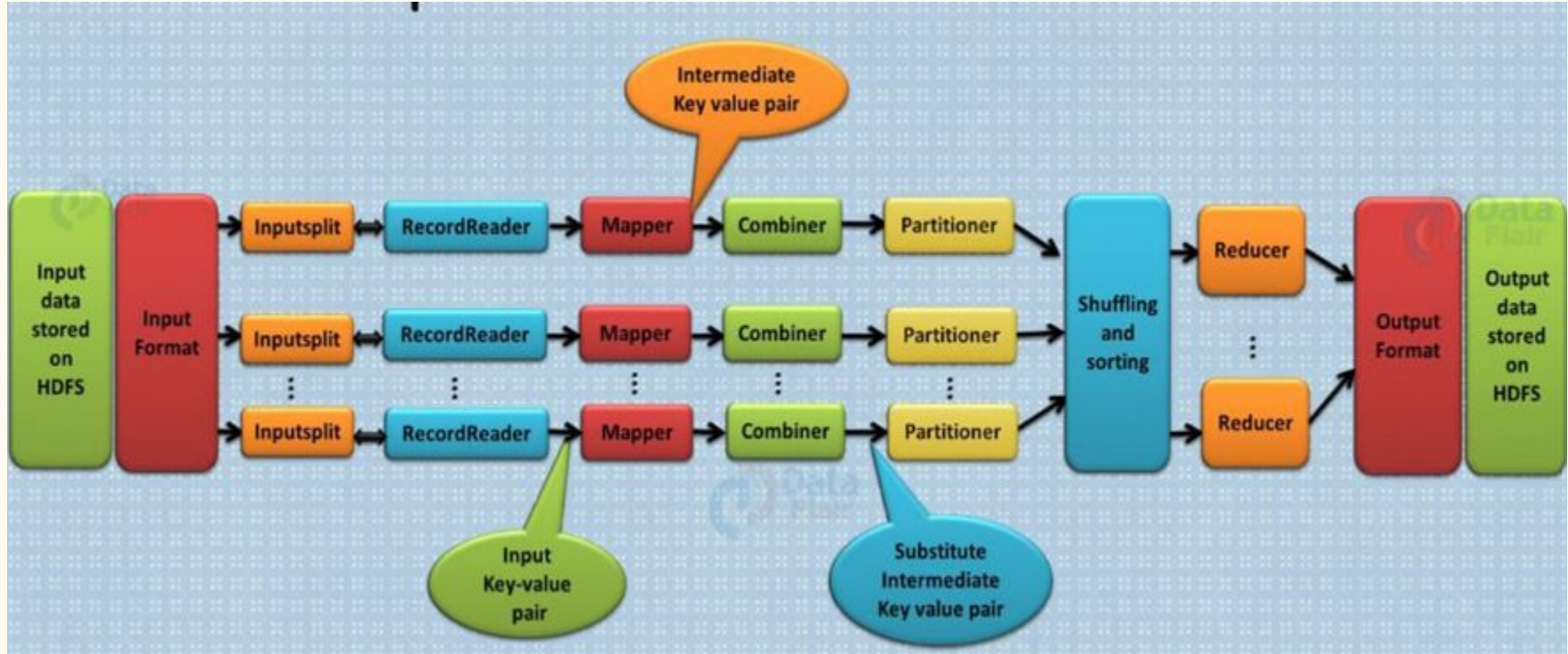
- Mapreduce is processing unit of hadoop that processes and computes vast volume of data
- Two main components in Mapreduce:
 - **Map** tasks deal with **splitting and mapping** of data
 - **Reduce** tasks **shuffle and reduce** the data.
- Flow diagram of MapReduce:



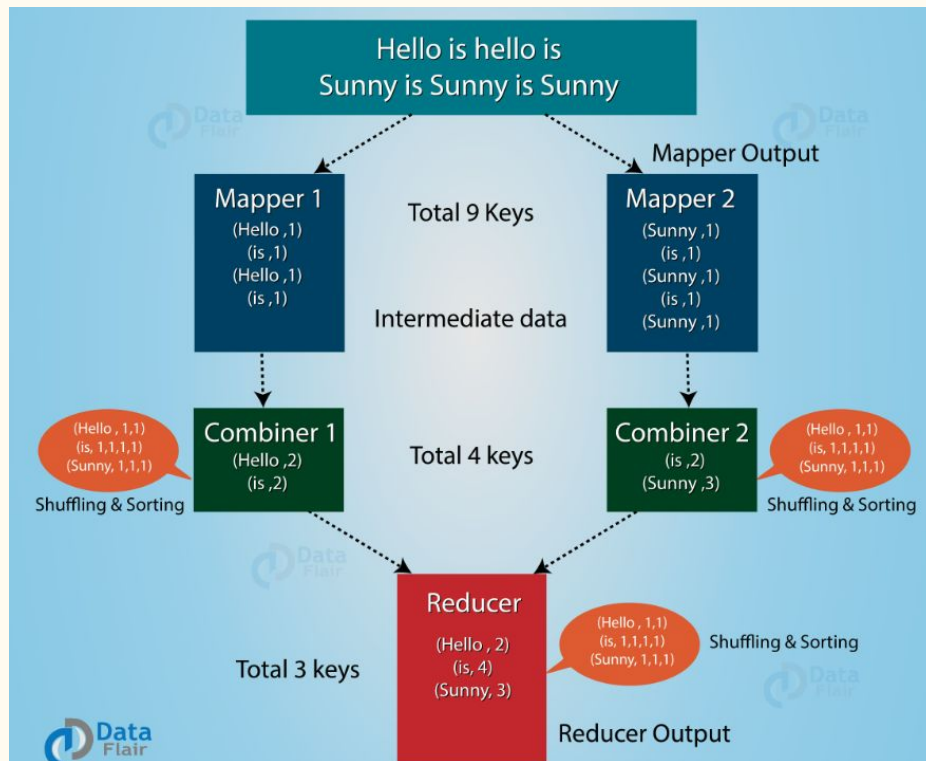
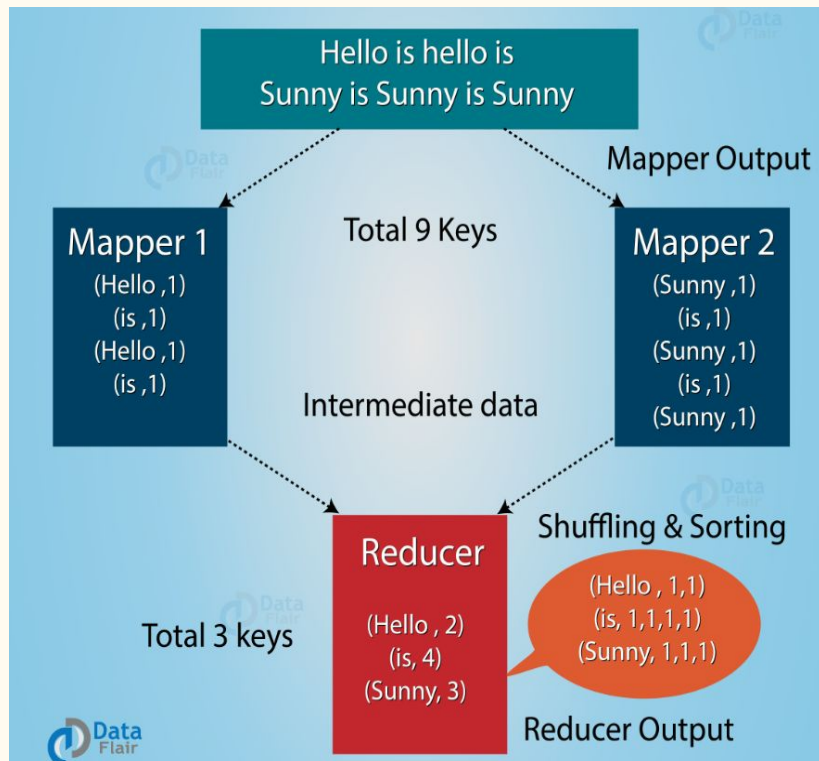
MapReduce in Hadoop - Overview



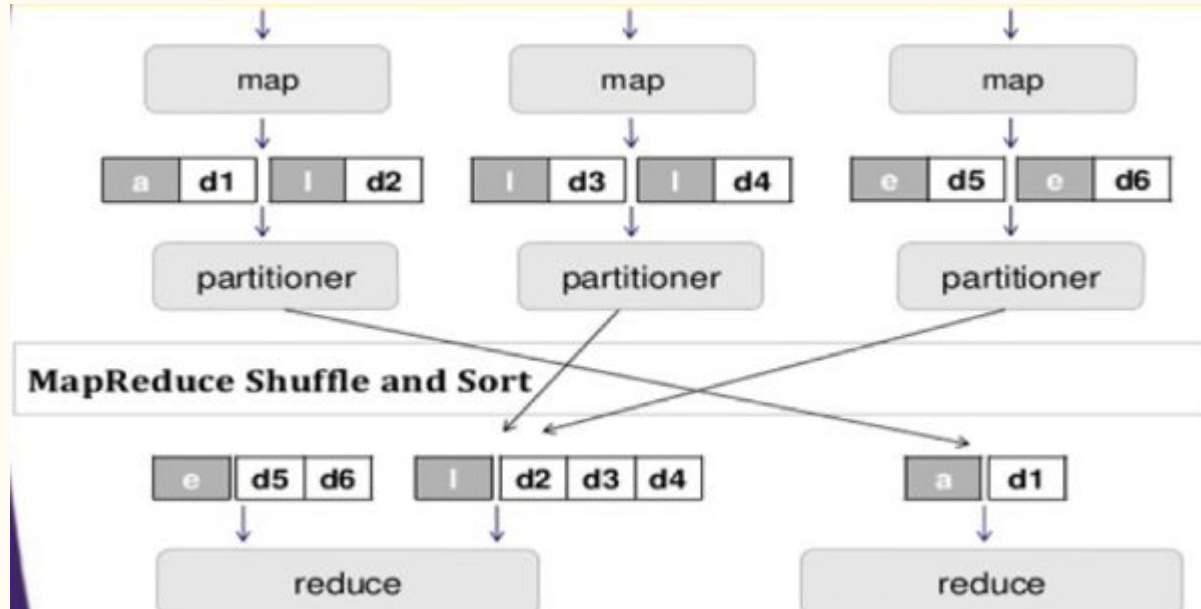
MapReduce in Hadoop - Workflow



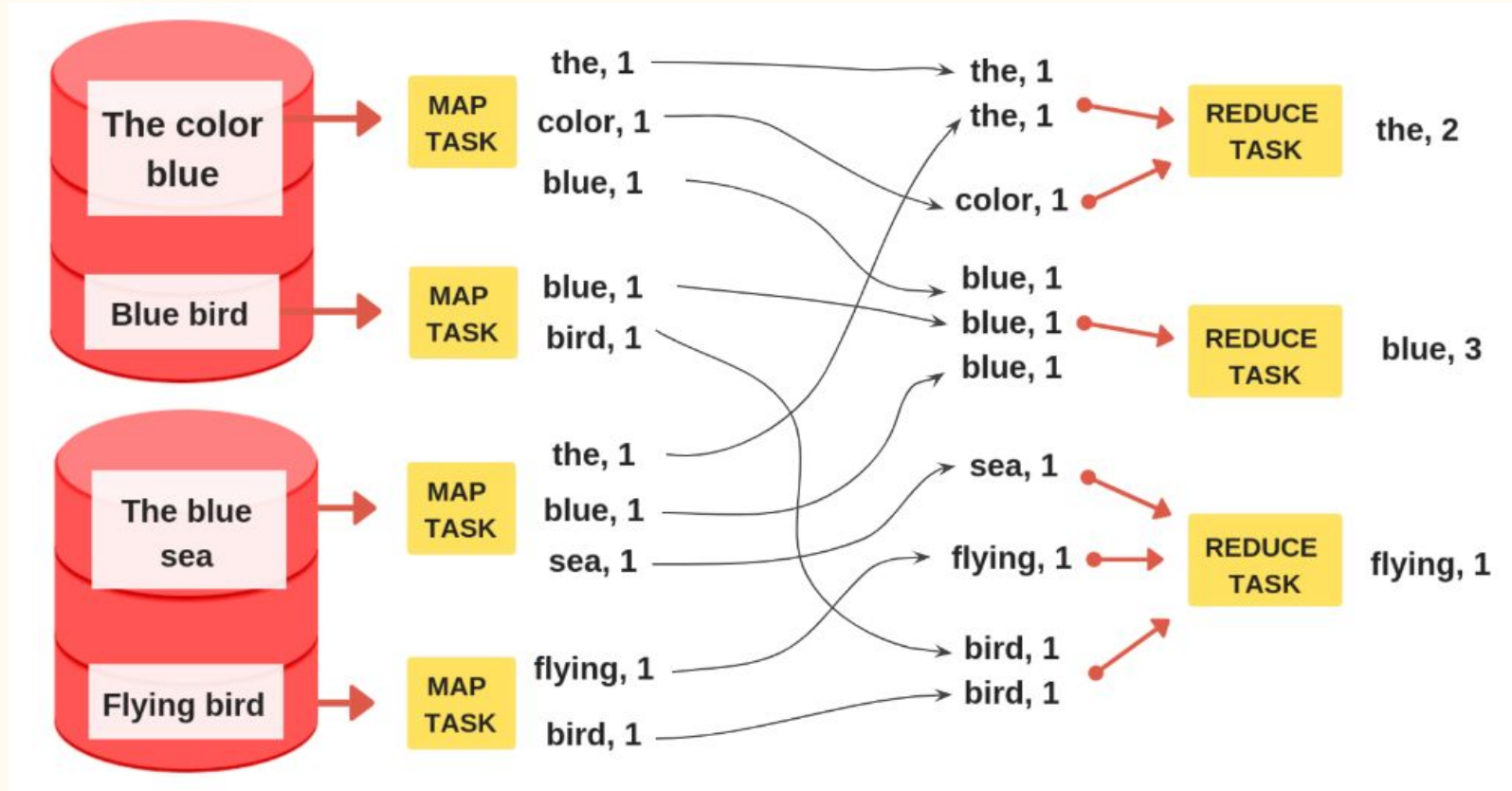
MapReduce in Hadoop - Combiner



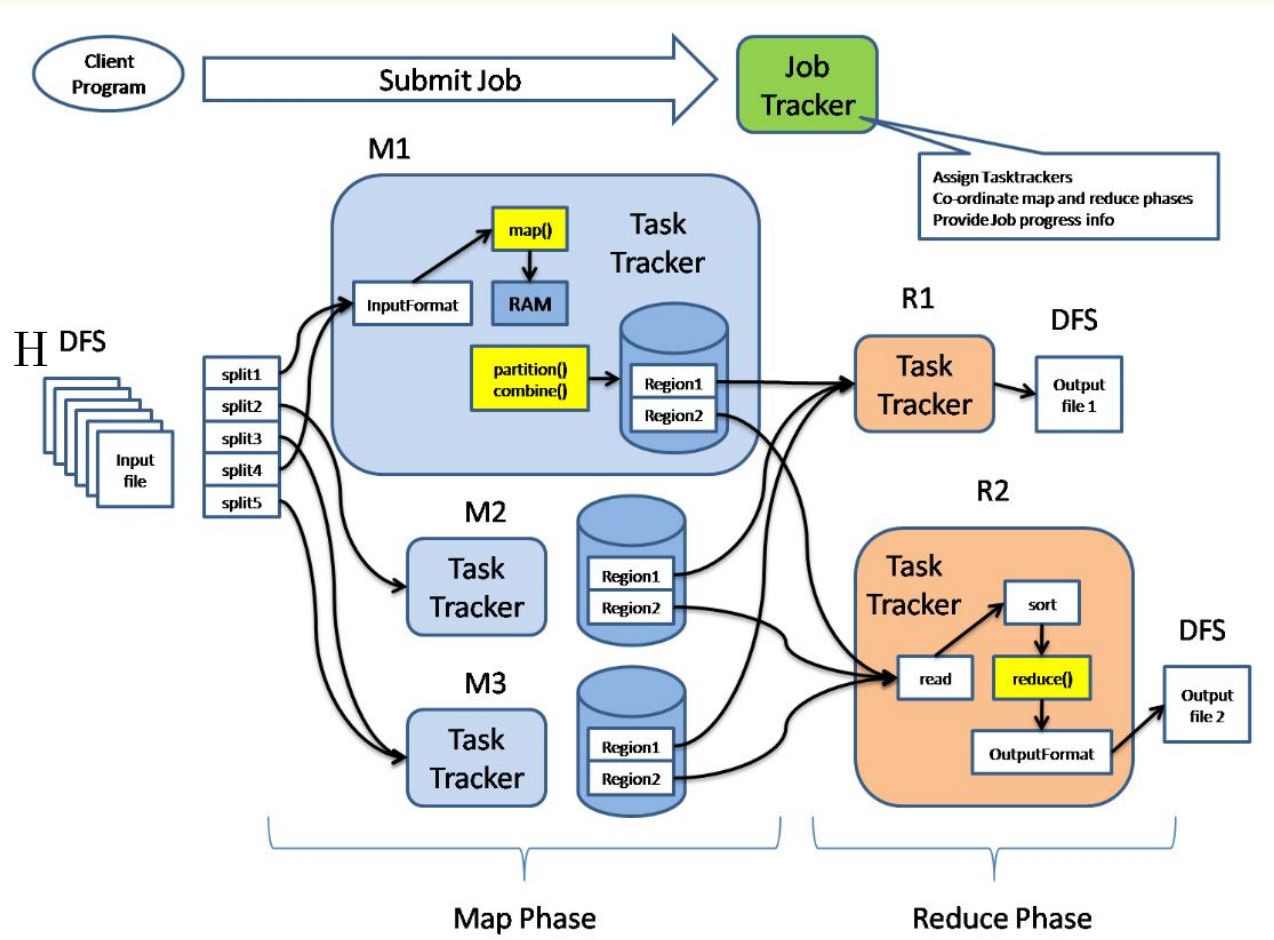
MapReduce in Hadoop - Partitioner



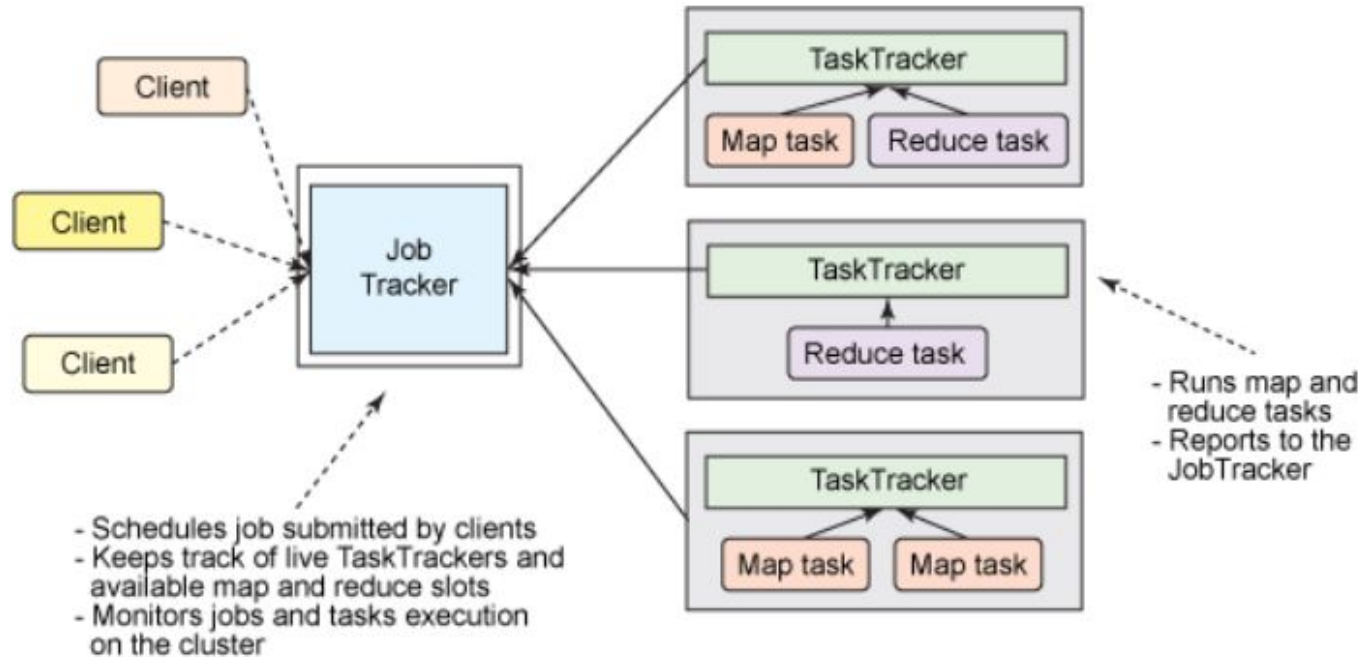
MapReduce in Hadoop - Shuffling & Sorting



MapReduce Architecture



MapReduce Job Tracker and Task Tracker



MapReduce - InputFileFormat

- FileInputFormat in Hadoop
- TextInputFormat
- KeyValueTextInputFormat
- SequenceFileInputFormat
- SequenceFileAsTextInputFormat
- SequenceFileAsBinaryInputFormat
- NLineInputFormat
- DBInputFormat

MapReduce - InputFileFormat

- *FileInputFormat in Hadoop*
 - It is the base class for all file-based InputFormats.
 - Hadoop FileInputFormat specifies input directory where data files are located.
 - When we start a Hadoop job, FileInputFormat is provided with a path containing files to read.
 - FileInputFormat will read all files and divides these files into one or more InputSplits.

MapReduce - InputFileFormat

- *TextInputFormat*
 - Default InputFormat of MapReduce.
 - TextInputFormat treats each line of each input file as a separate record and performs no parsing.
 - This is useful for unformatted data or line-based records like log files.
 - Key – It is the byte offset of the beginning of the line within the file (not whole file just one split), so it will be unique if combined with the file name.
 - Value – It is the contents of the line, excluding line terminators.

MapReduce - InputFileFormat

- ***KeyValueTextInputFormat***
 - It is similar to TextInputFormat as it also treats each line of input as a separate record.
 - While TextInputFormat treats entire line as the value,
 - The KeyValueTextInputFormat breaks the line itself into key and value by a tab character ('t').
 - Here Key is everything up to the tab character while the value is the remaining part of the line after tab character.

MapReduce - InputFileFormat

- *SequenceFileInputFormat*

- Hadoop SequenceFileInputFormat is an InputFormat which reads sequence files.
- **Sequence files are binary files that stores sequences of binary key-value pairs.**
- Sequence files block-compress and provide direct serialization and deserialization of several arbitrary data types (not just text).
- Here Key & Value both are user-defined.

MapReduce - InputFileFormat

- *SequenceFileAsBinaryInputFormat*
 - Hadoop SequenceFileAsBinaryInputFormat is a SequenceFileInputFormat using which we can extract the sequence file's keys and values as an opaque binary object.

MapReduce - InputFileFormat

- *NLineInputFormat*
 - The keys are byte offset of the line and values are contents of the line.
 - Each mapper receives a variable number of lines of input with TextInputFormat and KeyValueTextInputFormat and the number depends on the size of the split and the length of the lines.
 - And if we *want our mapper to receive a fixed number of lines of input, then we use NLineInputFormat.*
 - N is the number of lines of input that each mapper receives.
 - By default (N=1), each mapper receives exactly one line of input.
 - If N=2, then each split contains two lines. One mapper will receive the first two Key-Value pairs and another mapper will receive the second two key-value pairs.

MapReduce - Hadoop RecordWriter

- Reducer takes as input a set of an intermediate key-value pair produced by the mapper
- runs a reducer function on them to generate output that is again zero or more key-value pairs.
- *RecordWriter* writes these output key-value pairs from the Reducer phase to output files.

MapReduce - OutputFormat

- TextOutputFormat
- SequenceFileOutputFormat
- SequenceFileAsBinaryOutputFormat
- MapFileOutputFormat
- MultipleOutputs
- LazyOutputFormat
- DBOutputFormat

MapReduce - OutputFormat

- *TextOutputFormat*
 - The default OutputFormat is TextOutputFormat.
 - It writes (key, value) pairs on individual lines of text files.
 - Its keys and values can be of any type.
 - The reason behind is that TextOutputFormat turns them to string by calling toString() on them.
 - It separates key-value pair by a tab character

MapReduce - OutputFormat

- *SequenceFileOutputFormat*

- It is an Output Format which writes sequences files for its output
- It is intermediate format use between MapReduce jobs, which rapidly serialize arbitrary data types to the file;
- The corresponding SequenceFileInputFormat will deserialize the file into the same types and presents the data to the next mapper in the same manner as it was emitted by the previous reducer

MapReduce - OutputFormat

- ***SequenceFileAsBinaryOutputFormat***
 - It is another form of SequenceFileInputFormat which writes keys and values to sequence file in binary format.
- ***MapFileOutputFormat***
 - It is another form of FileOutputFormat in Hadoop Output Format, which is used to write output as map files.
 - The key in a MapFile must be added in order, so we need to ensure that reducer emits keys in sorted order.

MapReduce - OutputFormat

- *MultipleOutputs*
 - It allows writing data to files whose names are derived from the output keys and values, or in fact from an arbitrary string.
- *LazyOutputFormat*
 - Sometimes FileOutputFormat will create output files, even if they are empty.
 - LazyOutputFormat is a wrapper OutputFormat which ensures that the output file will be created only when **the record is emitted for a given partition.**

MapReduce - OutputFormat

- ***DBOutputFormat***

- DBOutputFormat in Hadoop is an Output Format for writing to relational databases and HBase.
- It sends the reduce output to a SQL table.
- It accepts key-value pairs, where the key has a type extending DBWritable.
- Returned RecordWriter writes only the key to the database with a batch SQL query.

MapReduce - Matrix Multiplication

Algorithm 1: The Map Function

```
1 for each element  $m_{ij}$  of  $M$  do
2   produce  $(key, value)$  pairs as  $((i, k), (M, j, m_{ij}))$ , for  $k = 1, 2, 3, \dots$  up
   to the number of columns of  $N$ 
3 for each element  $n_{jk}$  of  $N$  do
4   produce  $(key, value)$  pairs as  $((i, k), (N, j, n_{jk}))$ , for  $i = 1, 2, 3, \dots$  up
   to the number of rows of  $M$ 
5 return Set of  $(key, value)$  pairs that each key,  $(i, k)$ , has a list with
   values  $(M, j, m_{ij})$  and  $(N, j, n_{jk})$  for all possible values of  $j$ 
```

Algorithm 2: The Reduce Function

```
1 for each key  $(i, k)$  do
2   sort values begin with  $M$  by  $j$  in  $list_M$ 
3   sort values begin with  $N$  by  $j$  in  $list_N$ 
4   multiply  $m_{ij}$  and  $n_{jk}$  for  $j_{th}$  value of each list
5   sum up  $m_{ij} * n_{jk}$ 
6 return  $(i, k), \sum_{j=1} m_{ij} * n_{jk}$ 
```

END (Part 2)

—