8.3: Apache Hadoop

- **Instructor**: Dr. GP Saggese - gsaggese@umd.edu

- Resources
  - Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung: The Google File System, 2003
  - Jeffrey Dean and Sanjay Ghemawat: MapReduce: Simplified Data Processing on Large Clusters, 2004

Implementations of MapReduce

- **Google**
  - Not available outside Google
- **Hadoop**
  - Website
  - An open-source implementation in Java
  - Uses HDFS for stable storage
  - Hadoop Wiki
    - Introduction, Getting Started, Map/Reduce Overview
- **Amazon Elastic MapReduce (EMR)**
  - Website
  - Hadoop MapReduce running on Amazon EC2
  - Can also run Spark, HBase, Hive, ...
- **Spark**
- **Dask**

MapReduce: Hadoop

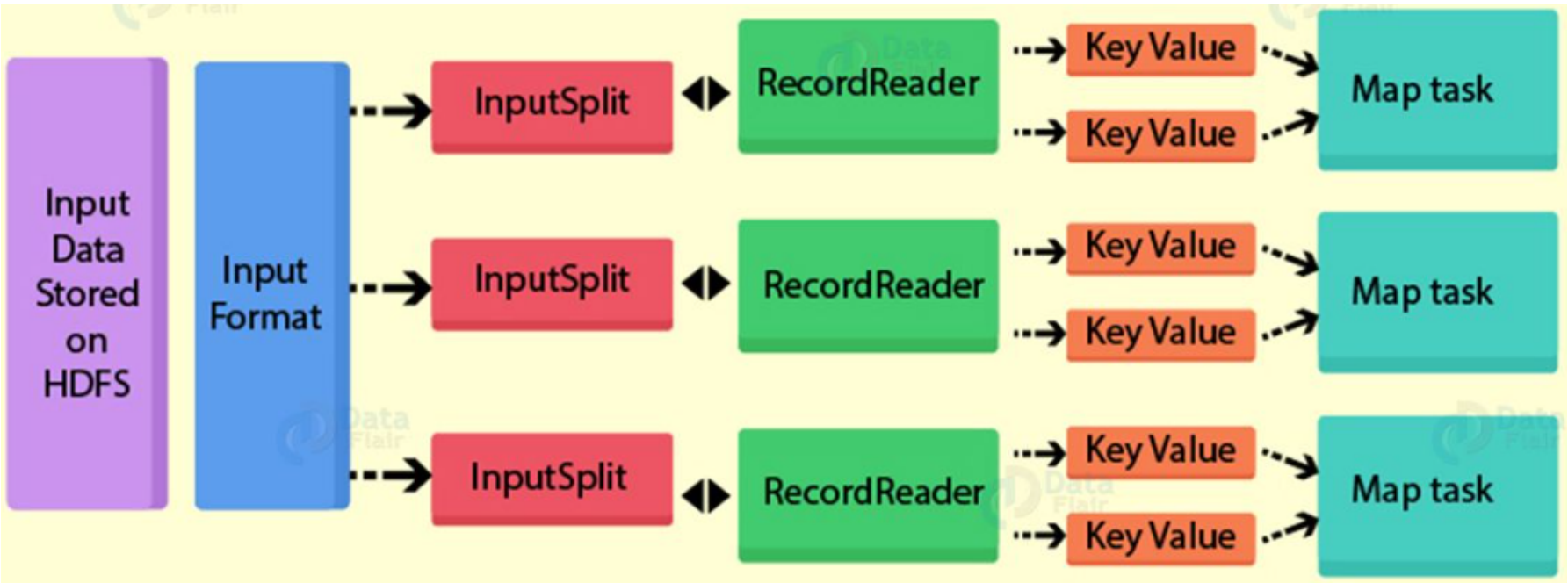
- Hadoop is an open-source implementation of MapReduce
- Functionalities
  - Partition the input data (HDFS)
  - Input adapters
    - E.g., HBase, MongoDB, Cassandra, Amazon Dynamo
  - Schedule program's execution across a set of machines
  - Handle machine failures
  - Manage inter-machine communication
  - Perform the *GroupByKey* step
  - Output adapters
    - E.g., Avro, ORC, Parquet
  - Schedule multiple *MapReduce* jobs

Data Flow

- Input, intermediate, final outputs are stored in a distributed file system (HDFS)
  - Every operation in Hadoop goes from disk to disk
- Adapters to read / partition the data in chunks
- Scheduler tries to schedule map tasks "close" to physical storage location of input data
  - Intermediate results (e.g., GroupBy) are stored on local FS of Map and Reduce workers
- Output is often input to another MapReduce task

Input Data

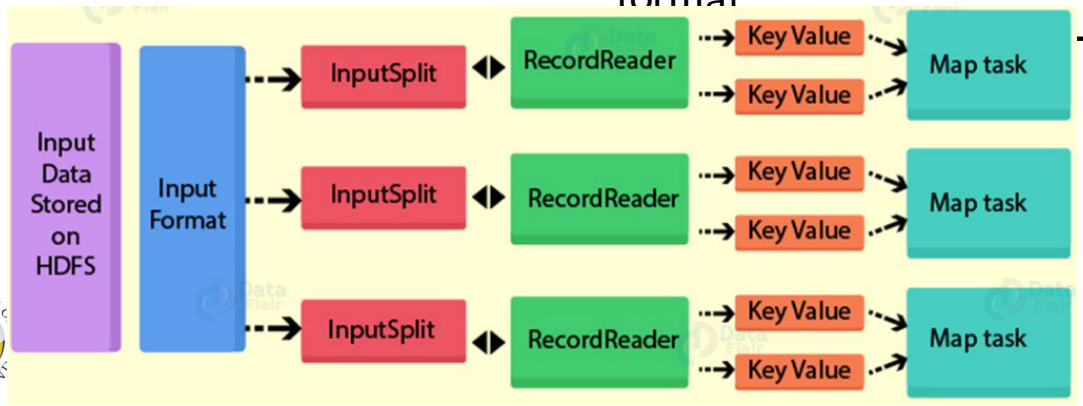
- **InputData** stores the data for a **MapTask** typically in a distributed file system (e.g., HDFS)
- The format of input data is arbitrary
  - Line-based log files
  - Binary files
  - Multi-line input records
  - Something else (E.g., an SQL database)



InputFormat

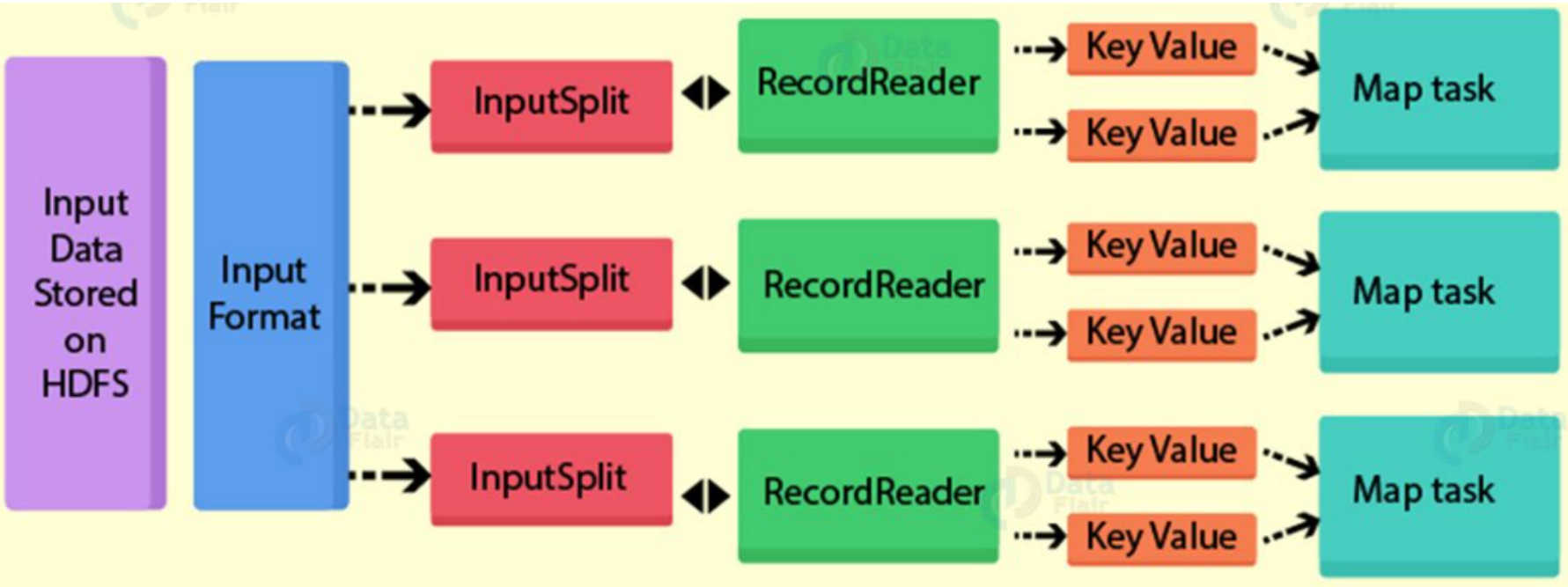
- **InputFormat** class reads and splits up the input files
- Select the files that should be used for input
- Defines the **InputSplits** that break a file
- Provides a factory for **RecordReaders** objects that read the file

InputFormat	Description	Key	Value
TextInputFormat	Default format; reads lines of text files	The byte offset of the line	The line contents
KeyValueInputFormat	Parses lines into (K, V) pairs	Everything up to the first tab character	The remainder of the line
SequenceFileInputFormat	A Hadoop-specific high-performance binary format	User-defined	User-defined



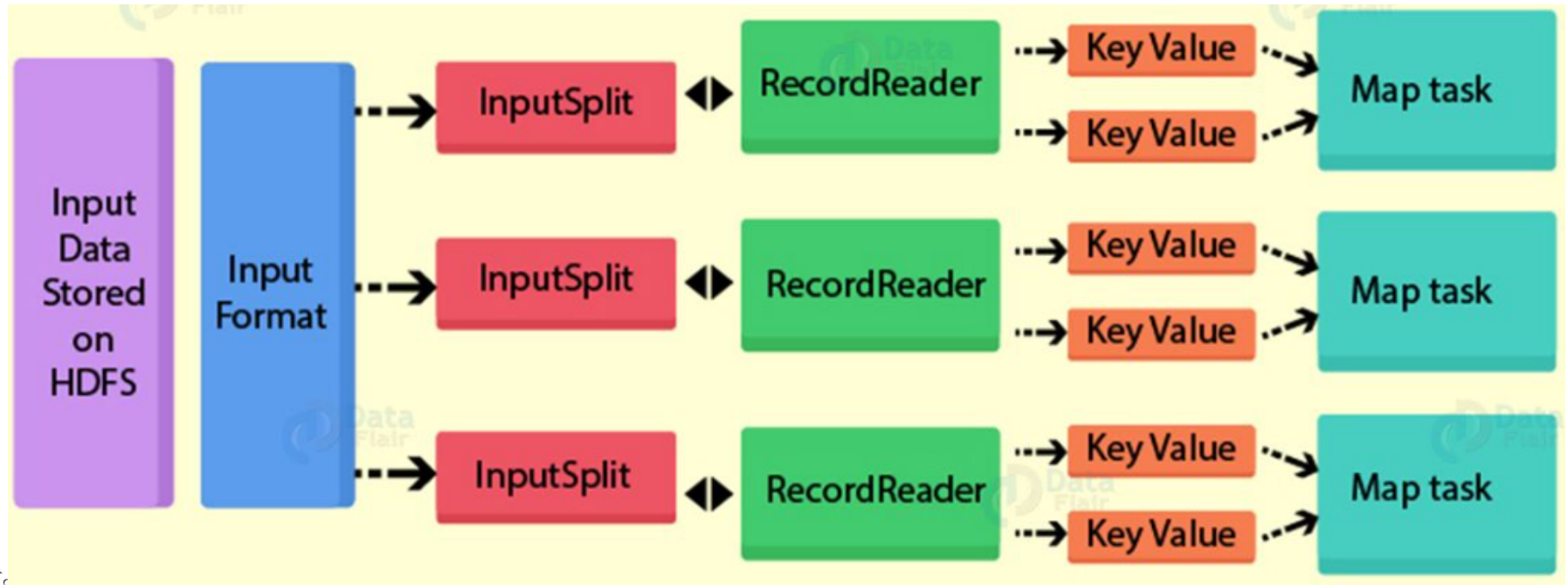
InputSplit

- **InputSplit** describes a unit of work that comprises a single **MapTask**
  - By default, the **InputFormat** breaks a file up into 64MB splits
- By dividing the file into splits
  - Each **MapTask** corresponds to a single input split
  - Several **MapTasks** to operate on a single file in parallel



RecordReader

- The **InputSplit** defines a slice of work but does not describe how to access it
- The **RecordReader** class
  - Loads data from its source and converts it into **(K, V) pairs** suitable for reading by **MapTasks**
  - Is invoked repeatedly on the input until the entire **InputSplit** is consumed
  - Each invocation leads to a call of the map function defined by the programmer



OutputFormat

- The **OutputFormat** class
  - defines the way (K,V) pairs produced by **Reducers** are written to output files
  - write to files on the local disk or in HDFS in different formats

OutputFormat	Description
TextOutputFormat	Default; writes lines in "key value" format
SequenceFileOutputFormat	Writes binary files suitable for reading into subsequent MapReduce jobs
NullOutputFormat	Generates no output files