

# **CLOUD COMPUTING**

## **Cloud Applications**

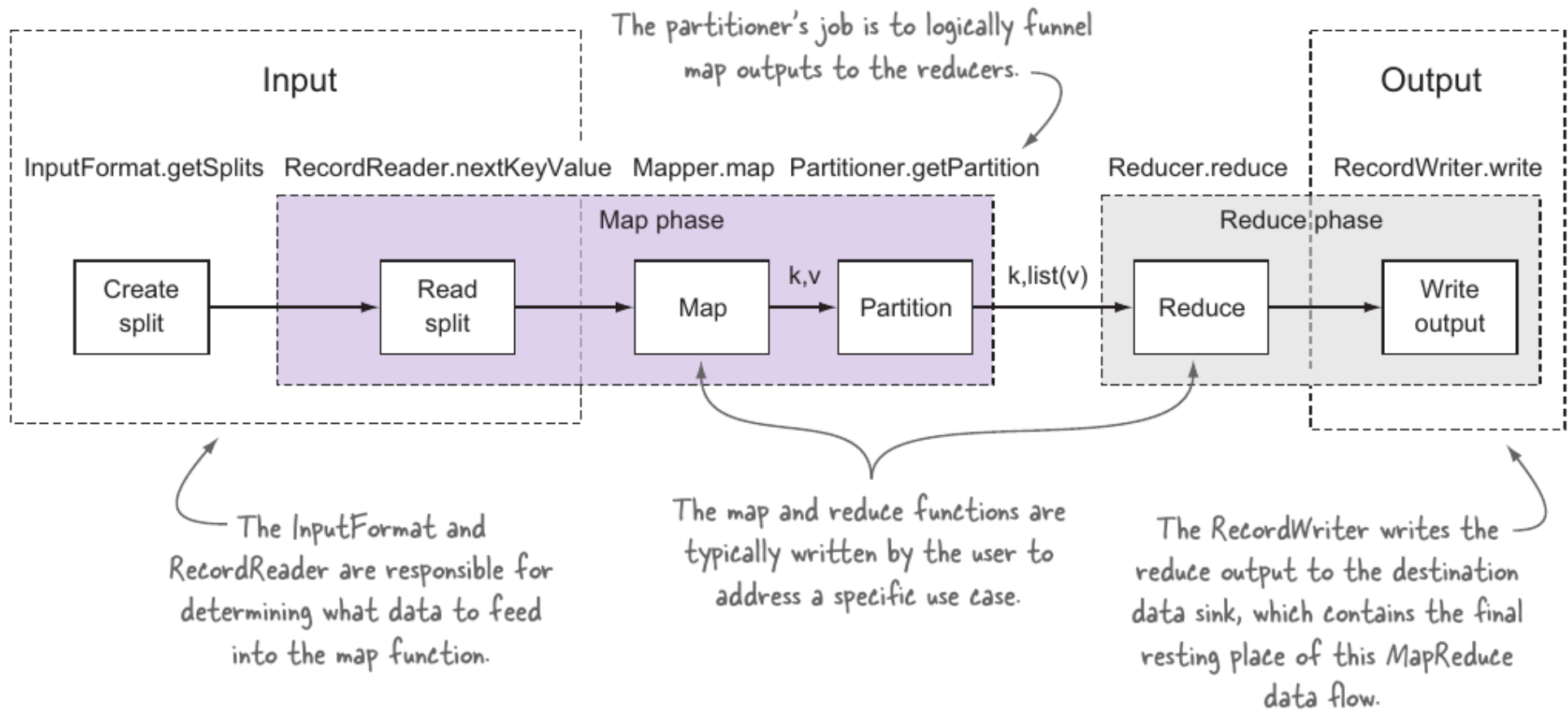
Zeinab Zali

Isfahan University Of Technology



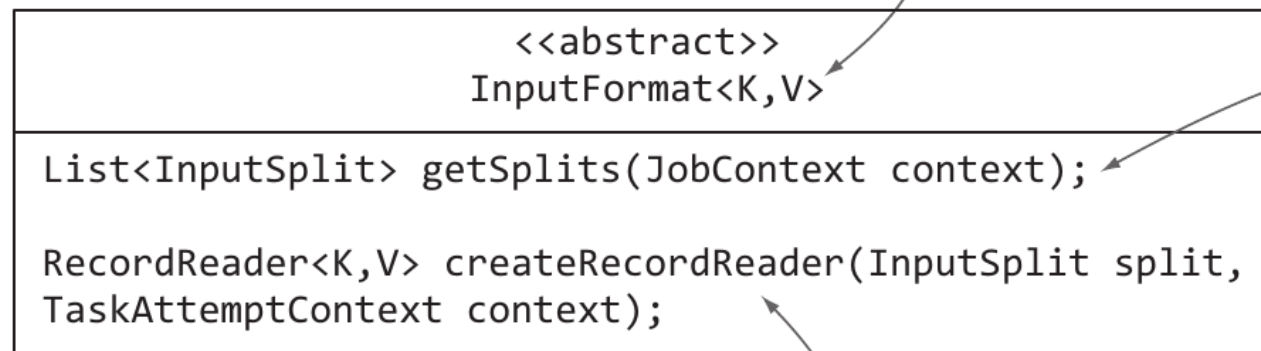
# Data Logistics

# Input/output actors in MR



# Data Input format

- **Input splits**: each split is processed by a single map task
- **Record reader**: each record is processed as a single key/value for the map function
- You can extend and implement an **InputFormat** abstract class and override its abstract methods

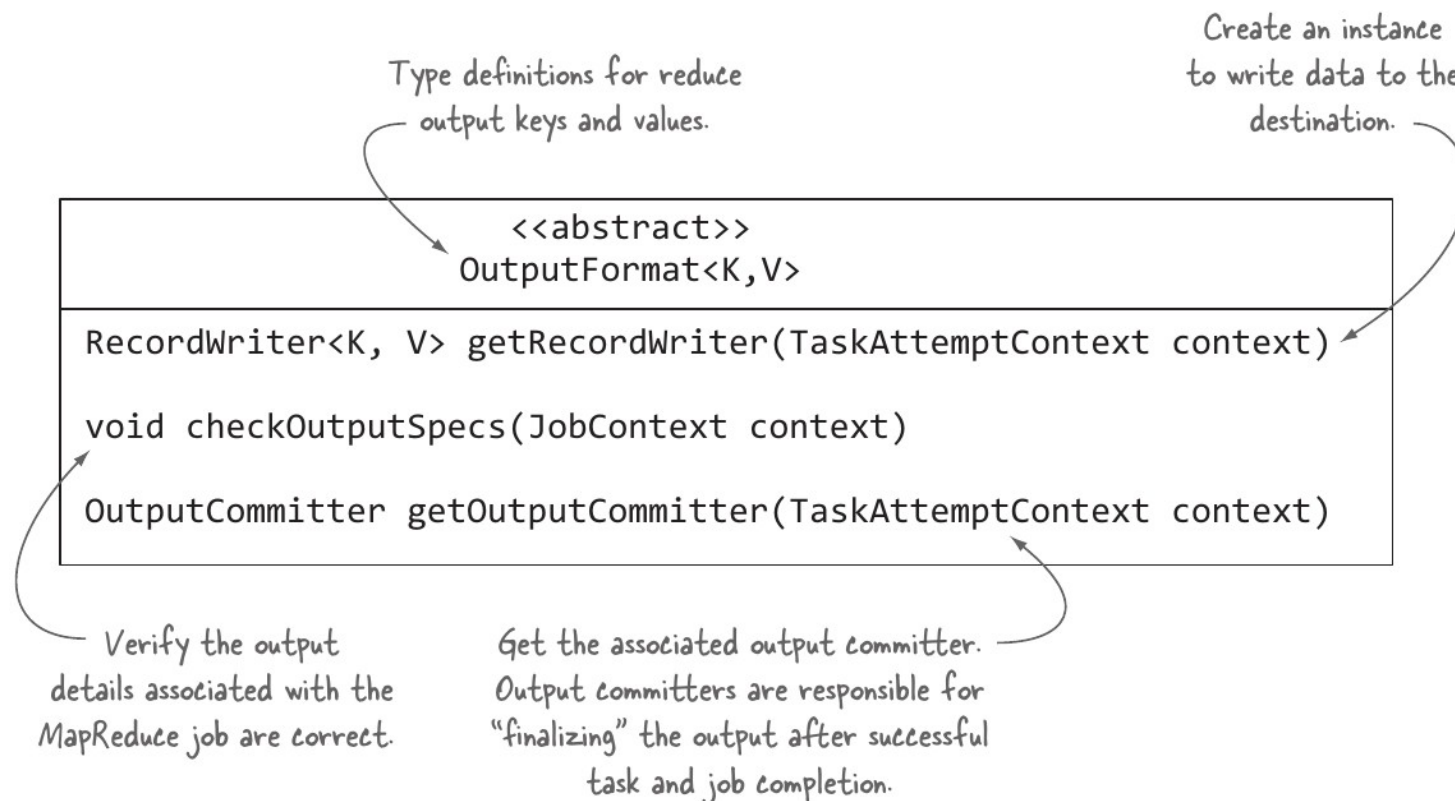


- The main Record reader class can be implemented through extending `RecordReader` abstract class

```
job.setInputFormatClass(TextInputFormat.class);
```

# Data output format

- You can extend and implement an **OutputFormat** abstract class and override its abstract methods



# XML and Json Input format

- MapReduce comes with an InputFormat that works with text, but it doesn't inherently support XML or Json
  - You should write your own inputformat and outputformat
  - **Mahout**, a machine learning system, also provides an XML InputFormat
  - **The Elephant Bird LzoJsonInputFormat** is used as a basis to create an input format class to work with JSON elements.



# Other Hadoop related Projects

# Ceph



- Ceph is a free-software storage platform, implements **object storage** on a single distributed computer cluster, and provides interfaces for object-, block- and file-level storage.
- It is more general than HDFS
  - Smaller files
- In contrast to other file systems, Ceph uses a function to calculate the block locations



# HBase



- Hbase is an open-source **non-relational distributed database** modeled after Google's Bigtable and written in Java
- Tables in HBase can serve as the input and output for MapReduce jobs run in Hadoop

# Hive



- Hive is an open-source system for data-warehousing Started at [Facebook](#)
- Hive supports queries expressed in an [SQL-like](#) declarative language called [HiveQL \(HQL\)](#)
  - These queries are then compiled into MapReduce jobs executed on Hadoop
  - **No UPDATE or DELETE** support, focuses primarily on the query part of SQL
  - Hive supports data organized as tables, partitions, and buckets.
  - Tables are inherited from relational databases and can be stored internally or externally in HDFS, NFS, or local directory

# Hive Example

- Relational join on two tables:
  - Table of word counts from Shakespeare collection
  - Table of word counts from Homer

```
SELECT s.word, s.freq, k.freq FROM shakespeare s
JOIN homer k ON (s.word = k.word) WHERE s.freq >= 1 AND k.freq >= 1
ORDER BY s.freq DESC LIMIT 10;
```

|     |       |       |
|-----|-------|-------|
| the | 25848 | 62394 |
| I   | 23031 | 8854  |
| and | 19671 | 38985 |
| to  | 18038 | 13526 |
| of  | 16700 | 34654 |
| a   | 14170 | 8057  |
| you | 12702 | 2720  |
| my  | 11297 | 4135  |
| in  | 10797 | 12445 |
| is  | 8882  | 6884  |

# Tez (I)



- Tez generalizes the MapReduce paradigm to a more powerful framework based on expressing computations as a **dataflow graph**
- Tez is an extensible framework for building high performance **batch** and **interactive data processing** applications, coordinated by YARN in Apache Hadoop.
  - MapReduce inherently supports only batch processing
  - MapReduce does not inherently support workflows as dataflow graphs

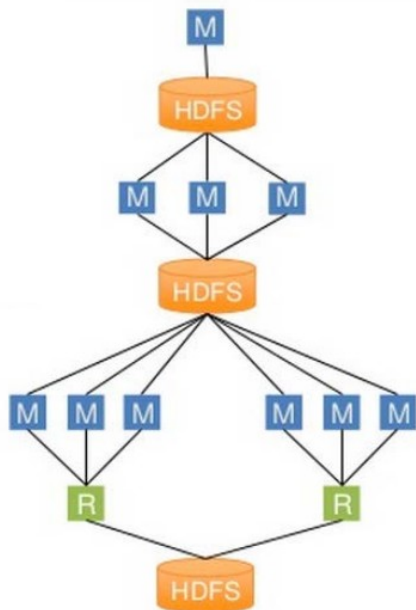
# Tez (II)



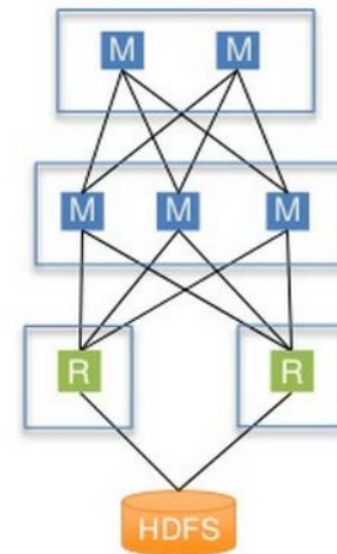
- Tez, provides higher performance than MapReduce.
- Hive embeds Tez so that it can translate complex SQL statements into highly optimized, purpose-built data processing graphs

# Dataflow with Tez

- Tez defines a **simple Java API** to express a **Directed Acyclic Graph (DAG)** of data processing
  - Express computation as a dataflow graph
  - Extensively uses **caching**



MapReduce vs Tez



# Interactive processing

- Interactive processing enable us planning reconfiguration of dataflow or execution configuration at runtime
- LLAP is low-latency analytical processing (LLAP)
  - LLAP: low Latency Analytical Processing
  - LLAP: Long Live Analytical Processing
  - LLAP is possible through interactive processing

# Pig (I)



- Pig also supports workflows, join operations for combined processing of several datasets, filtering, aggregation, and the high-level operations
- Pig compiles programs written in **Pig Latin** into a set of Hadoop jobs and coordinates their execution
- Pig interaction modes:
  - Interactive: using a shell for Pig commands
  - Batch: a user submits a script containing a series of Pig commands
  - Embedded: commands are submitted via method invocation from a Java program

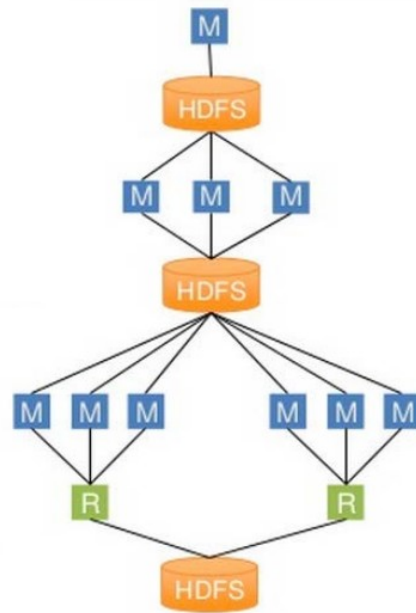


# PIG processing stages

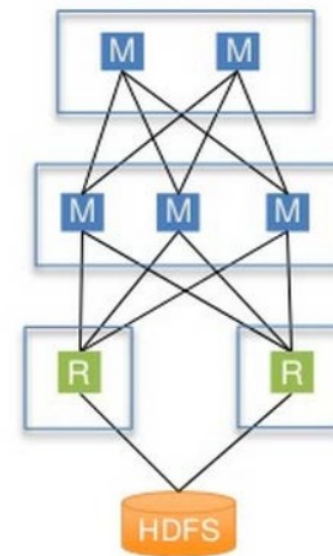
1. Parsing: the parser produces a DAG called a Logical Plan.
2. Logical optimization of the DAG and creation of a Physical Plan describing data distribution.
3. Compilation of the optimized Physical Plan into a set of MapReduce jobs (Maps and reduces)
4. The DAG is topologically sorted and jobs are submitted to Hadoop for execution

# Tez vs PIG

- Hive + Pig creates extra map/reduce jobs when implemented with Hadoop compared with using Tez on Yarn



Hive + Pig



Tez

# Impala



- Impala is a low-latency, massively parallel query engine, modeled after Google's Dremel paper describing a scalable and interactive query system
  - Impala allows you to query data in **HDFS** or **HBase** with a SQL syntax

# Impala vs Hive

- Impala is designed from the ground up as a massively parallel query engine and doesn't need to translate SQL into another processing framework
  - Hive relies on MapReduce (or more recently Tez) to execute
- Impala is faster than Hive
- Impala isn't fault-tolerant.
- Impala Only works with flat data
  - It doesn't support complex types such as maps, arrays, and structs



**CLOUDERA**

- **Hortonworks** was a data software company that developed and supported open-source software (primarily around Apache Hadoop) designed to manage Big Data and associated processing
  - From yahoo
- **Cloudera** is a software company that provides a software platform for data engineering, data warehousing, machine learning and analytics that runs in the cloud
  - It started as a hybrid open-source Apache Hadoop distribution, CDH (Cloudera Distribution Including Apache Hadoop), that targeted enterprise-class deployments of that technology