



# MapReduce Architecture

---

Ram Krishna Pudasaini  
Roll: 24



# What is Mapreduce?

- Definition: MapReduce is a programming model and processing framework for distributed computing.
- Origin: Popularized by Google and implemented in open-source frameworks like Apache Hadoop.
- Purpose: Designed to handle massive datasets across clusters of computers efficiently.



# Key Components of MapReduce

Client Program

## JobTracker

Description: Master daemon responsible for job scheduling, task assignment, and monitoring.

Functionality: Accepts job submissions, coordinates task execution across nodes.

Example: In Hadoop, JobTracker manages MapReduce jobs.

## TaskTrackers

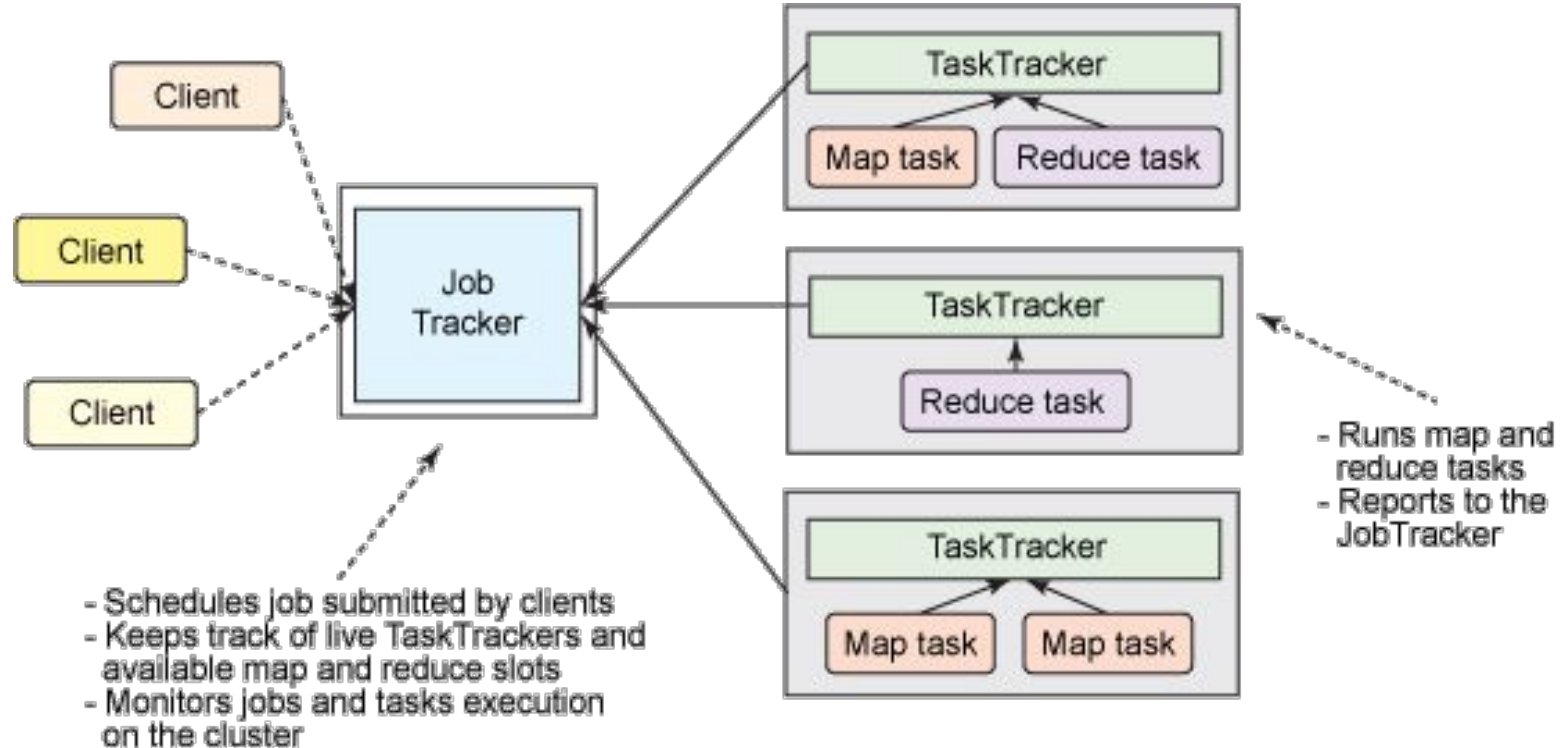
Description: Worker daemons running on individual cluster nodes.

Functionality: Execute tasks assigned by the JobTracker, send status updates.

Example: Each node may have multiple TaskTrackers, executing map and reduce tasks.



# How do Job Tracker works?





# Key Components of MapReduce

Input Split

## MapReduce Job

Description: Main processing unit consisting of map and reduce phases.

Map Phase: Applies a function to input data, generates intermediate key-value pairs.

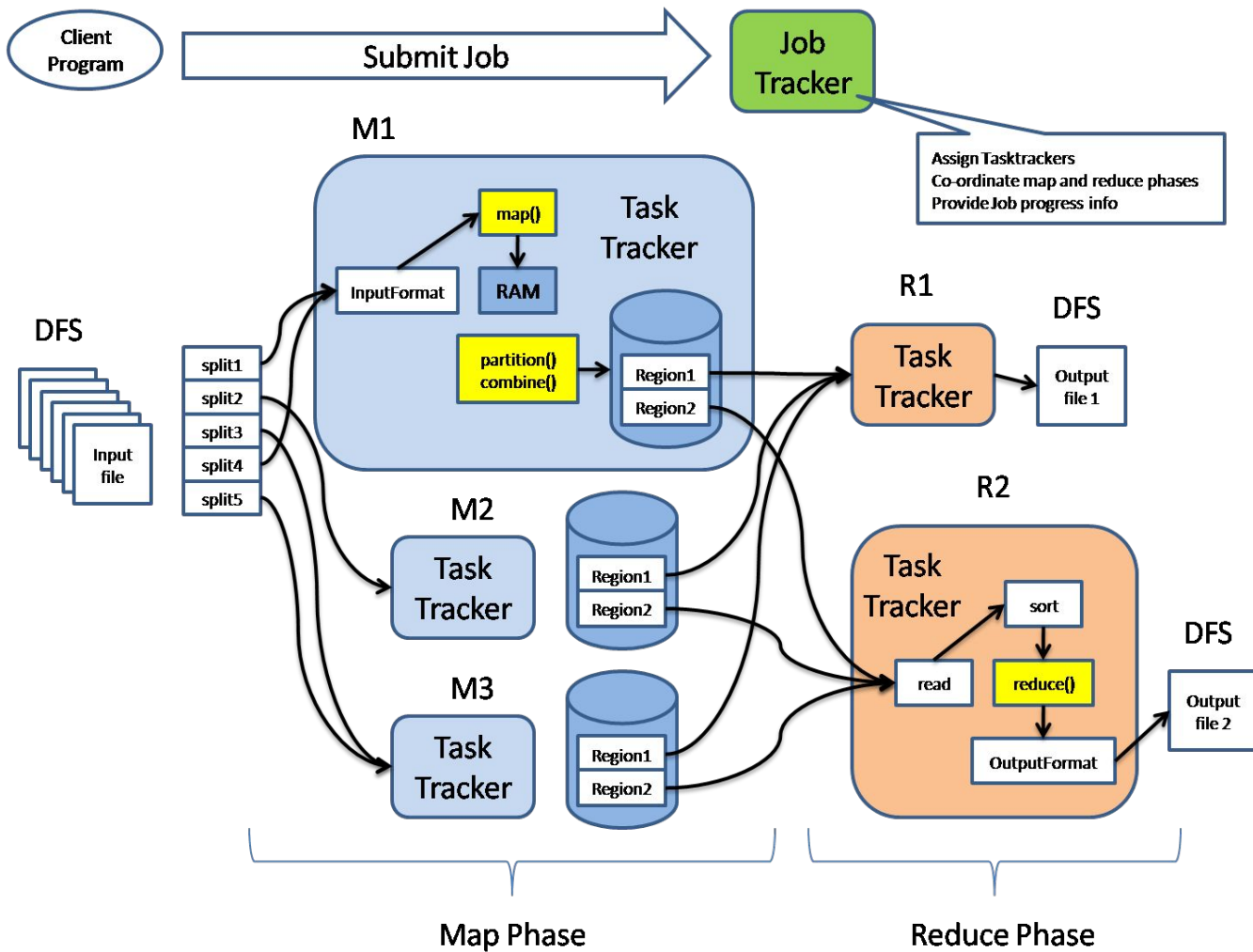
Reduce Phase: Aggregates and processes intermediate pairs, produces final output.

## OutputFormat

Description: Specifies the format of final output data produced by the MapReduce job.

Functionality: Defines how output is written and stored.

Example: Various OutputFormats available, including text, sequence, and custom formats.





# Combiner

- Purpose: The Combiner, also known as the mini-reducer, is an optional optimization technique employed during the MapReduce framework's shuffle and sort phase.
- Functionality: After the Map phase, before the data is transferred over the network to the reducers, the Combiner aggregates the intermediate key-value pairs produced by the mapper tasks locally on each node. It performs a "mini-reduction" operation on these pairs.



# Partitioner

- Purpose: The Partitioner is responsible for determining which reducer instance will receive the output of each map task based on the intermediate keys.
- Functionality: It partitions (divides) the intermediate key-value pairs produced by the mapper tasks into groups or partitions, ensuring that all key-value pairs with the same key end up in the same partition.



# Mapreduce Workflow

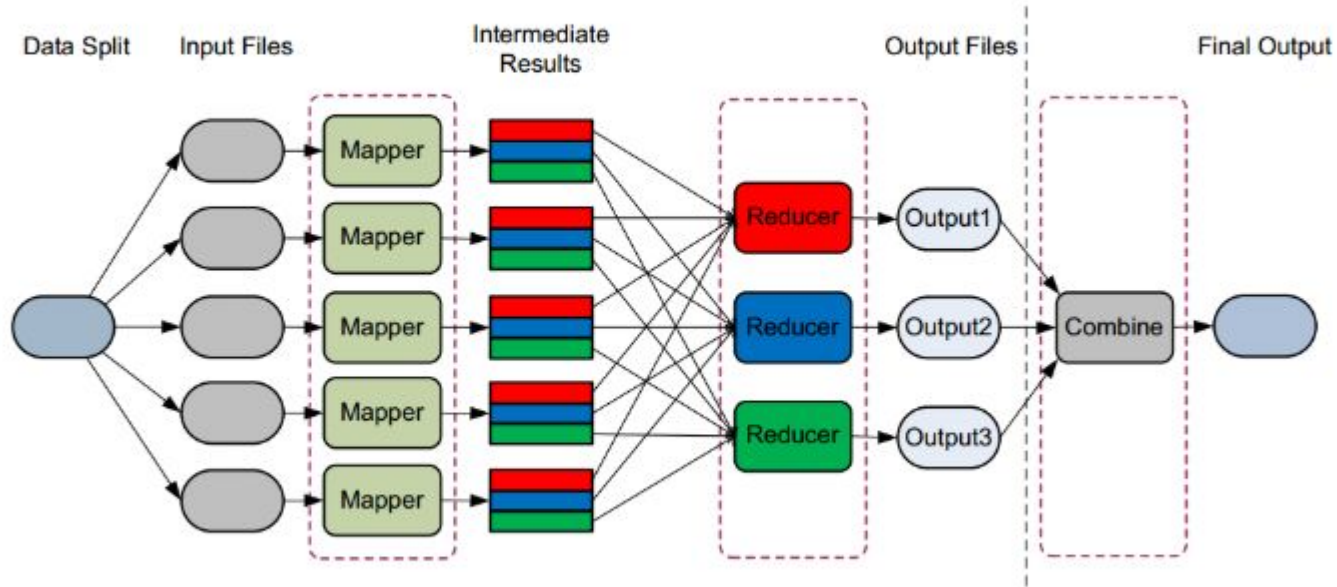
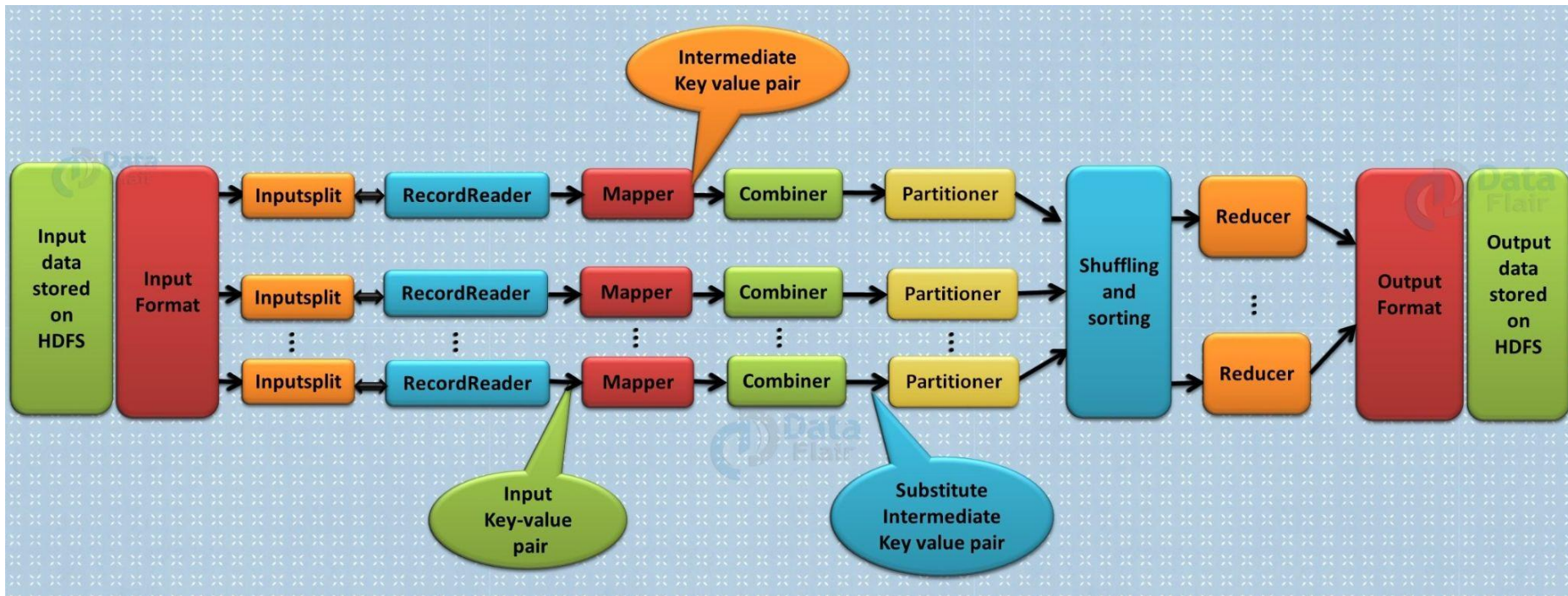


Figure 1: Implementation of MapReduce, overview of the dataflow

# Final Overview





**Thank You**