**1. Input Split**

What is it?

* A logical chunk of your input data that a single Mapper will process.
* Defined by the Input Format (e.g., TextInputFormat) before the Map phase starts.

Example:

If you have a 1 GB file and the split size is 128 MB:

* The file will be divided into 8 input splits.
* Hadoop will assign 8 Mapper tasks.

Purpose:

* Enables parallel processing by distributing the workload across multiple nodes.
* Increases performance and scalability.

**2. Multiple Mappers**

What are Mappers?

* Mapper is the first phase in MapReduce.
* It takes input data (from input splits), processes it, and emits intermediate key-value pairs.

Example**:**

For a Word Count job, each mapper reads lines and emits:

text

Copy code

("Hadoop", 1)

("is", 1)

("great", 1)

Purpose:

* Processes each split independently and in parallel.
* Handles data transformation or extraction tasks.

**Combiner (Mini-Reducer)**

What is it?

* An optional optimization step that runs after Mapper and before data is shuffled to Reducers.
* Performs local aggregation to minimize data transferred over the network.

Example:

If the Mapper emits:

text

Copy code

("Hadoop", 1)

("Hadoop", 1)

("Hadoop", 1)

The Combiner converts this locally to:

text

Copy code

("Hadoop", 3)

Purpose:

* Reduces network congestion and improves performance.
* Not guaranteed to run, and must be associative & commutative.

**4. Zero Reducers**

What does it mean?

* You configure the job to skip the Reduce phase:

java

Copy code

job.setNumReduceTasks(0);

Effect:

* Map output (with optional combining) is directly written to HDFS.
* You get multiple part files, one per Mapper.

Use Cases:

* Filtering, data formatting, preprocessing
* When global aggregation is not needed

Each file contains the output of one Mapper (optionally passed through the Combiner).

**🎯 Considerations:**

* If you need consolidated output, you'll need to merge manually (e.g., using hdfs dfs -getmerge).
* This design is faster and simpler for jobs that don’t require final aggregation.

**Final Overview**

| **Component** | **Description** |
| --- | --- |
| **Input Split** | Logical unit of work for Mappers (not physical block) |
| **Mapper** | Processes each split independently and outputs key-value pairs |
| **Combiner** | Optional mini-reducer for local aggregation (runs on Mapper output) |
| **Reducer** | Disabled here – no global aggregation |
| **Output** | Multiple files, each representing one Mapper’s final output |

Top of Form

Bottom of Form