**What is MapReduce?**

MapReduce is a programming model used for processing large datasets in a distributed and parallel manner across many computers (nodes). It was introduced by Google and is widely used in Hadoop for big data processing.

**Simple Definition:**

MapReduce splits a big task into smaller tasks (Map), processes them in parallel, then combines the results (Reduce).

**🔄 MapReduce Workflow (Step-by-Step)**

Let’s understand it using a common example: Word Count

**1. Input Data (e.g., a large text file)**

nginx

CopyEdit

Hello world

Hello Hadoop

**2. Map Phase 🗺️**

Each line is split and processed as key-value pairs:

| Input Line | Mapper Output |
| --- | --- |
| Hello world | (Hello, 1), (world, 1) |
| Hello Hadoop | (Hello, 1), (Hadoop, 1) |

Multiple Mappers work in parallel on different chunks of data.

**3. Shuffle and Sort Phase 🔀**

The output from mappers is grouped by key (word):

| **Key** | **Values** |
| --- | --- |
| Hello | [1, 1] |
| world | [1] |
| Hadoop | [1] |

This phase ensures all values for the same key go to the same reducer.

**4. Reduce Phase 🧮**

Reducers sum the values:

| **Key** | **Final Output** |
| --- | --- |
| Hello | (Hello, 2) |
| world | (world, 1) |
| Hadoop | (Hadoop, 1) |

Multiple Reducers can work in parallel on different keys.

**5. Final Output**

Stored in HDFS (Hadoop Distributed File System) as:

bash

CopyEdit

/output/part-r-00000

Hello 2

world 1

Hadoop 1

**⚙️ MapReduce Phases Summary**

| **Phase** | **What it Does** |
| --- | --- |
| Map | Processes input and emits key-value |
| Combine (optional) | Optimizes local output from mappers |
| Shuffle | Groups same keys across all mappers |
| Reduce | Aggregates values for each key |
| Output | Writes final result to HDFS |

**✅ Advantages of MapReduce**

* Scalable to petabytes of data
* Parallel processing across clusters
* **Fault-tolerant**
* Works well with HDFS