MapReduce Word Count

Nguyen Viet Khoa

December 5, 2025

1 Why the Chosen MapReduce Implementation

I chose a pure Python simulation of MapReduce using built-in functions like `map`, `reduce`, and `defaultdict` for grouping. This avoids dependencies on external frameworks like Hadoop (which requires setup) or libraries like mrjob (not available). It's educational, lightweight, and runs locally without a cluster, making it ideal for demonstration while illustrating core concepts: map, shuffle/sort, reduce.

2 How Mapper and Reducer Work

The Mapper takes text chunks, extracts words using regex, and emits (word, 1) pairs. The shuffle/sort phase groups and sorts these by key. The Reducer sums counts for each word.

3 Who Does What

- **Mapper**: Processes input chunks, tokenizes into words, and outputs key-value pairs (word, 1) for each occurrence.
- **Shuffle/Sort**: Groups all pairs by key (word) and sorts the keys, preparing for reduction (handled by `defaultdict` and `sorted`).
- **Reducer**: Aggregates values for each key by summing the list of 1s, producing the final count.
- **Main Function**: Splits input into chunks, orchestrates map/shuffle/reduce phases, and collects results into a dictionary.

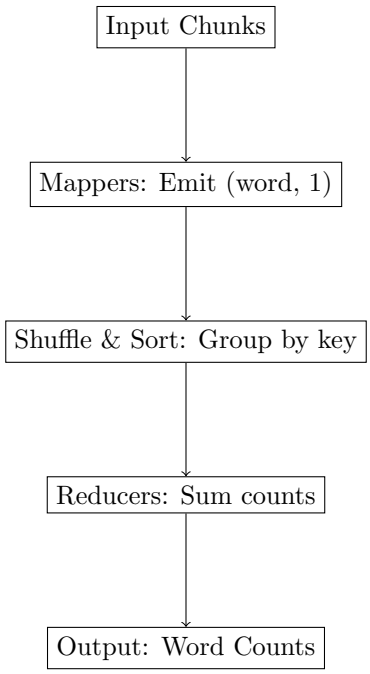

Figure 1: Mapper and Reducer Workflow