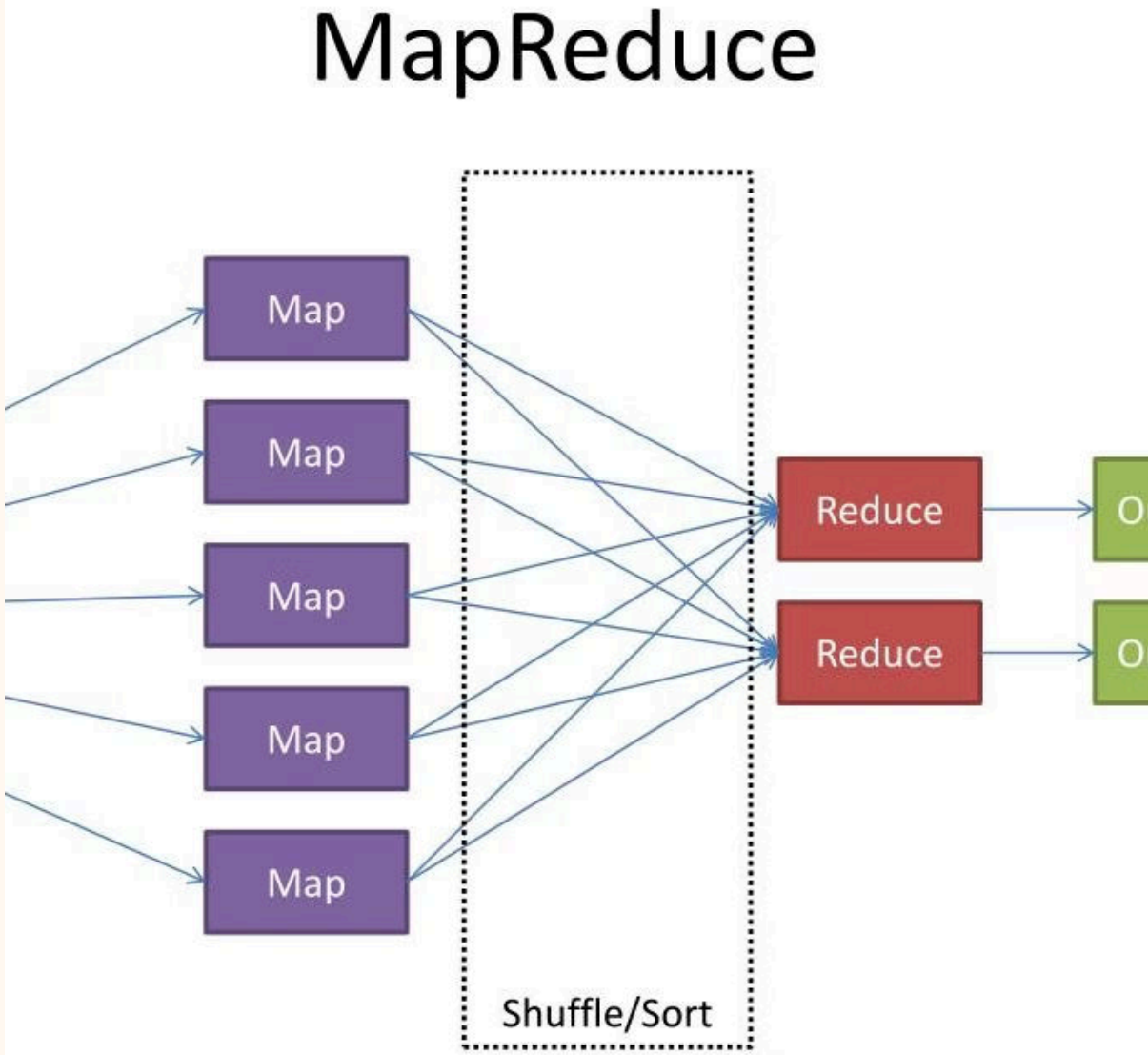
MapReduce Word Count and Most Frequent Words in Java

MapReduce in Java provides a powerful framework for processing large datasets in a distributed manner. This example demonstrates how to perform word count and find the most frequent words in a text file using MapReduce programming. The process involves mappers, reducers, and a driver to execute the map-reduce job efficiently and effectively.

G **by GUNTI VEDASRI**



Word Count Program

1

Mapper

The mapper class reads the input text and processes it to emit individual words with a count of 1, serving as the initial step in the word count program.

2

Reducer

The reducer class collects and aggregates the mapped word counts, effectively summing up the occurrences of each word to generate the final word count output.

3

Driver

The driver class sets up the MapReduce job configuration, including specifying the input and output paths and ensuring the execution of mappers and reducers.

Most Frequent Words Program

Mapper

The most frequent words mapper is analogous to the word count mapper, emitting each word and a count of 1 to continue the map-reduce process.

Reducer

Utilizing a PriorityQueue, the most frequent words reducer keeps track of the top N words by count, enabling the extraction of the most frequent words from the word count output.

Driver

The driver class sets N as a configuration parameter, allowing the distributed demonstration of the top N most frequent words from the word count output.

MapReduce Program Execution

MapReduce Program Component	Description
Mapper	Reads and processes input data, emitting intermediate key-value pairs for the reducer.
Reducer	Aggregates, filters, and summarizes the intermediate data, producing the final output.
Driver	Configures and manages the overall execution of the map-reduce job, including input/output paths and execution settings.

Java Language Integration

Java-Centric

Java serves as the foundational language for developing MapReduce programs, leveraging its robustness and wide adoption in the software industry.

Big Data Processing

Java's integration with Hadoop enables efficient and scalable processing of big data, making it an indispensable tool for data-intensive applications.

Distributed Computing

Through Java, MapReduce programs can seamlessly harness distributed computing resources, enabling parallel execution and high-performance processing.