GENERAL ASSEMBLY

# Intro to the big data ecosystem

*Patrick Smith*

# LEARNING OBJECTIVES

- Recognize big data problems
- Explain how the map reduce algorithm works
- Perform a map-reduce on a single node using python

# OPENING

# What *really* is “big data?”

# What is big data?

- Big data is a term used when the data exceeds the processing capacity of typical database.

# What is big data?

- Big data is a term used when the data exceeds the processing capacity of typical database.
- We need a big data analytics when the data grows quickly and we need to uncover hidden patterns, unknown correlations, and other useful information.

# Examples of big data

- Facebook social graph
- Netflix movie preferences
- Large recommender systems
- Activity of visitors to a website
- Customer activity in a retail store (ie: Target)

# What challenges exist with big data?

- Processing time
- Cost
- Architecture maintenance and setup
- Hard to visualize

# The Three Vs:

- **Volume:** Large amounts of data

# The Three Vs:

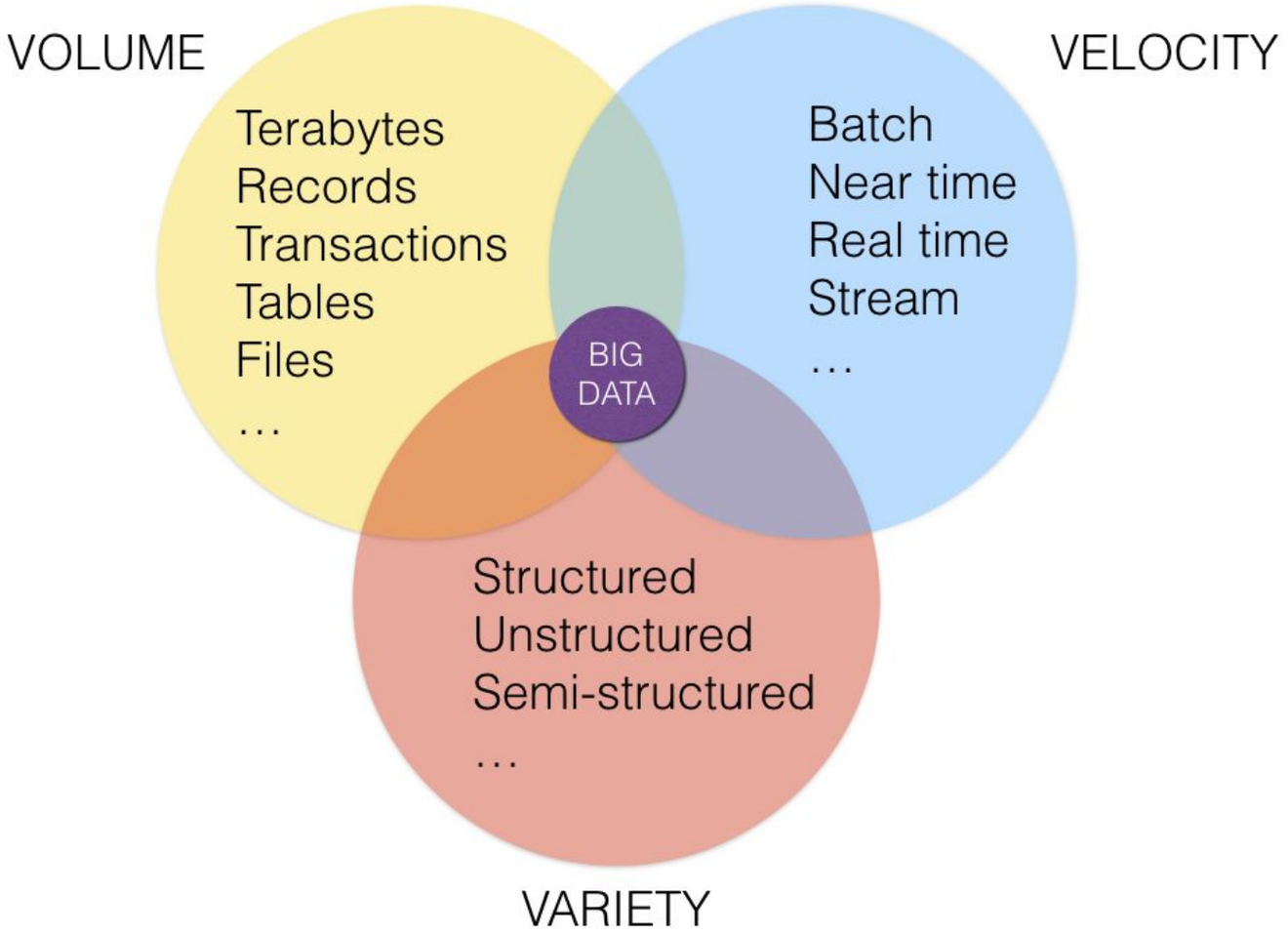
- **Volume:** Large amounts of data
- **Variety:** Different types of structured, unstructured, and multi-structured data

# The Three Vs:

- **Volume:** Large amounts of data
- **Variety:** Different types of structured, unstructured, and multi-structured data
- **Velocity:** Needs to be analyzed quickly

# Intro to big data

## The Three Vs:



# Two approaches to Big Data: High Performance Computing and Cloud.

## Intro to big data: HPC

# Supercomputers are very expensive, very powerful calculators used by researchers to solve complicated math problems.

# Can you think of advantages and disadvantages of this configuration?

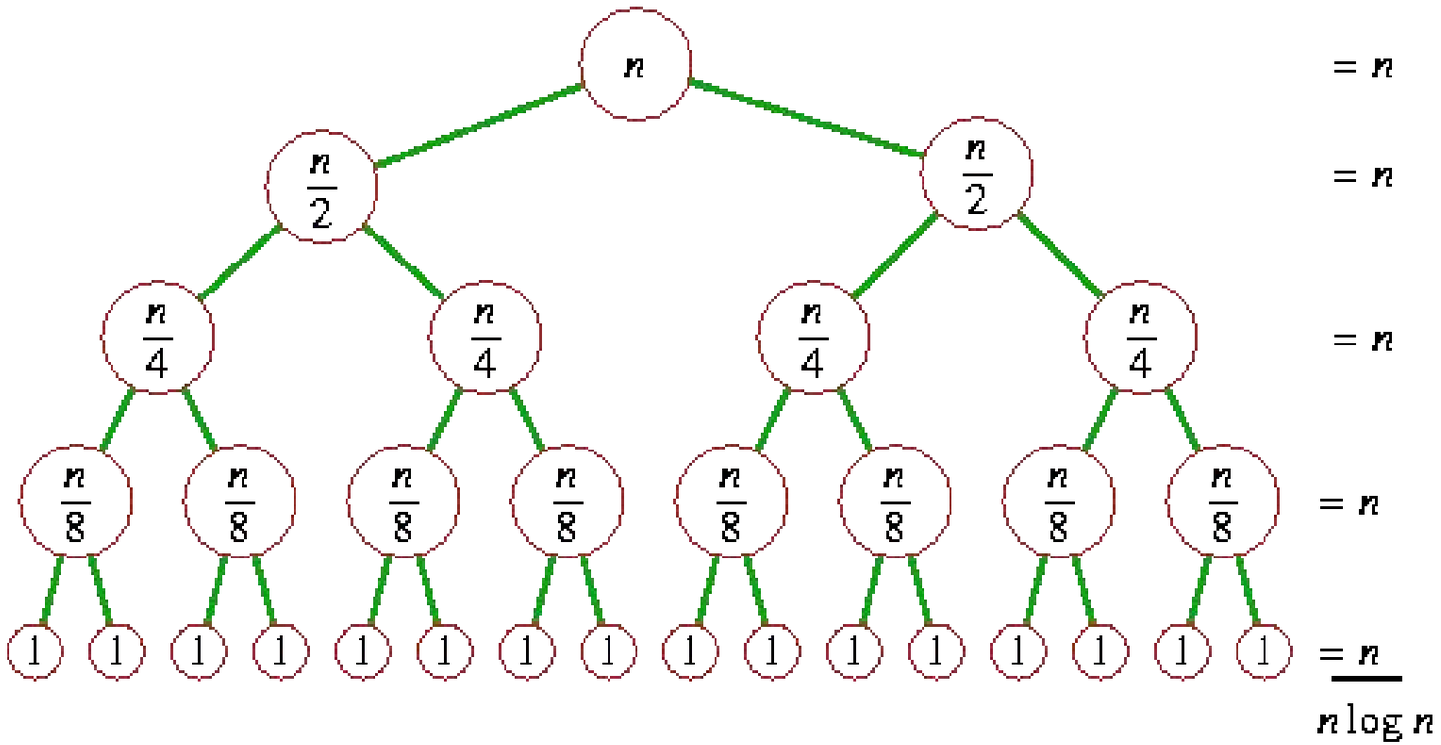
# Intro to big data: Cloud Computing

## Instead of one huge machine, what if we got a bunch of (commodity) machines?

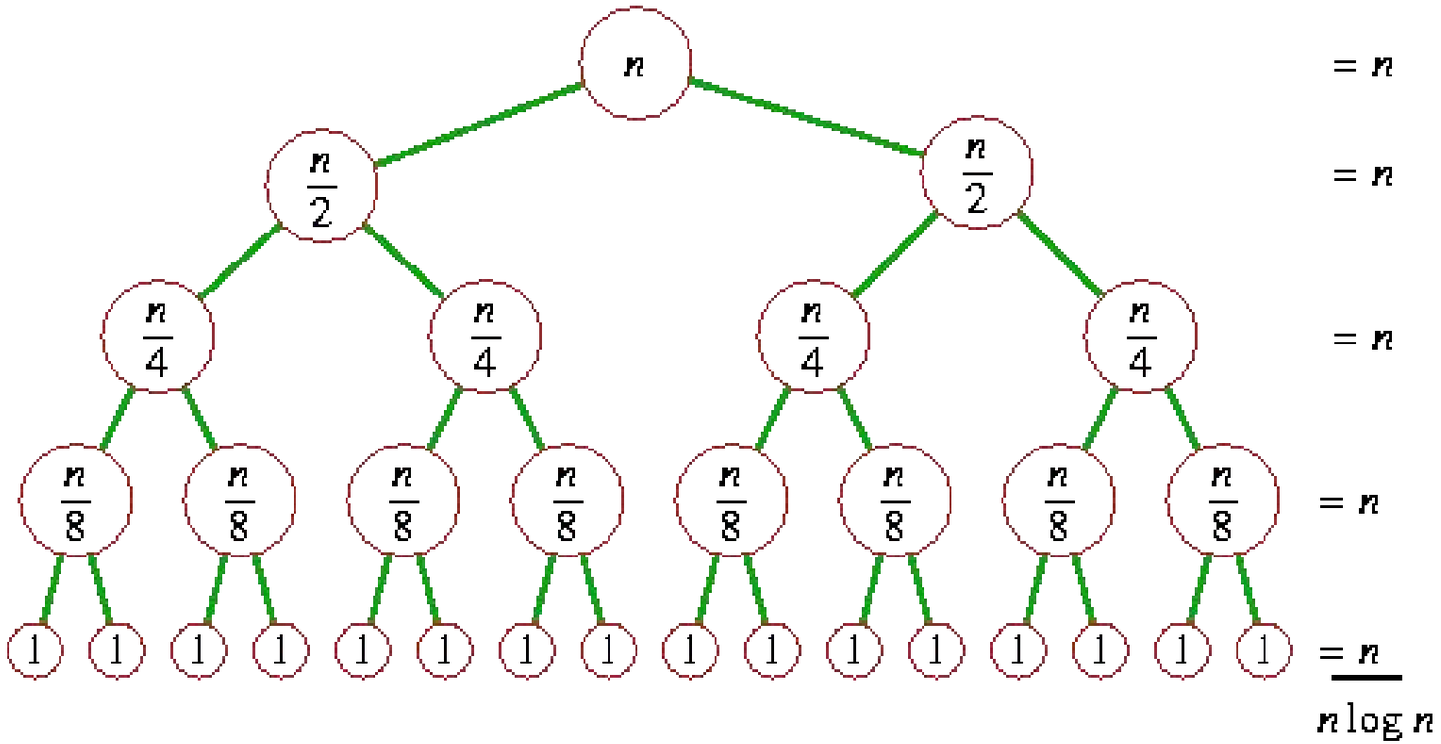
# Can you think of advantages and disadvantages of this configuration?

# How do you think many computers process data?

# Parallelism

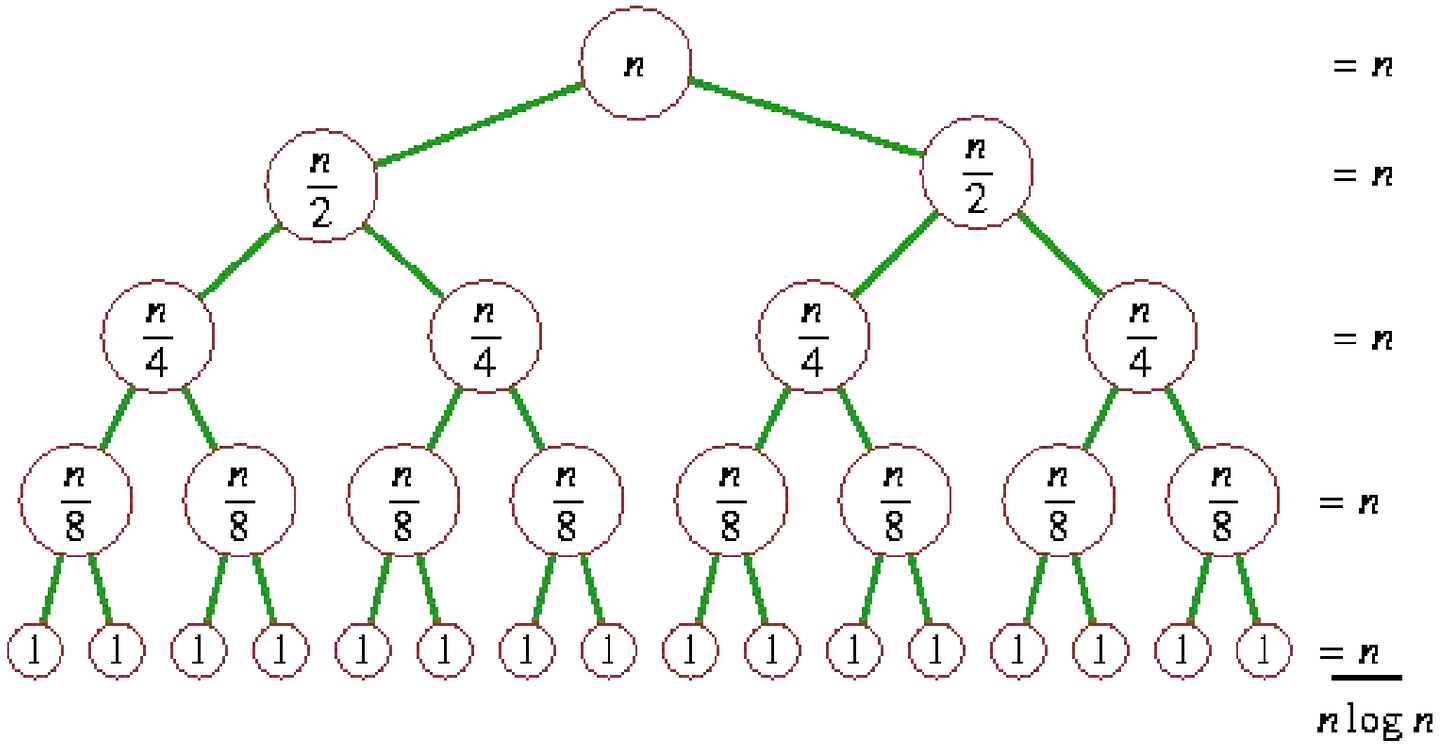


# Parallelism



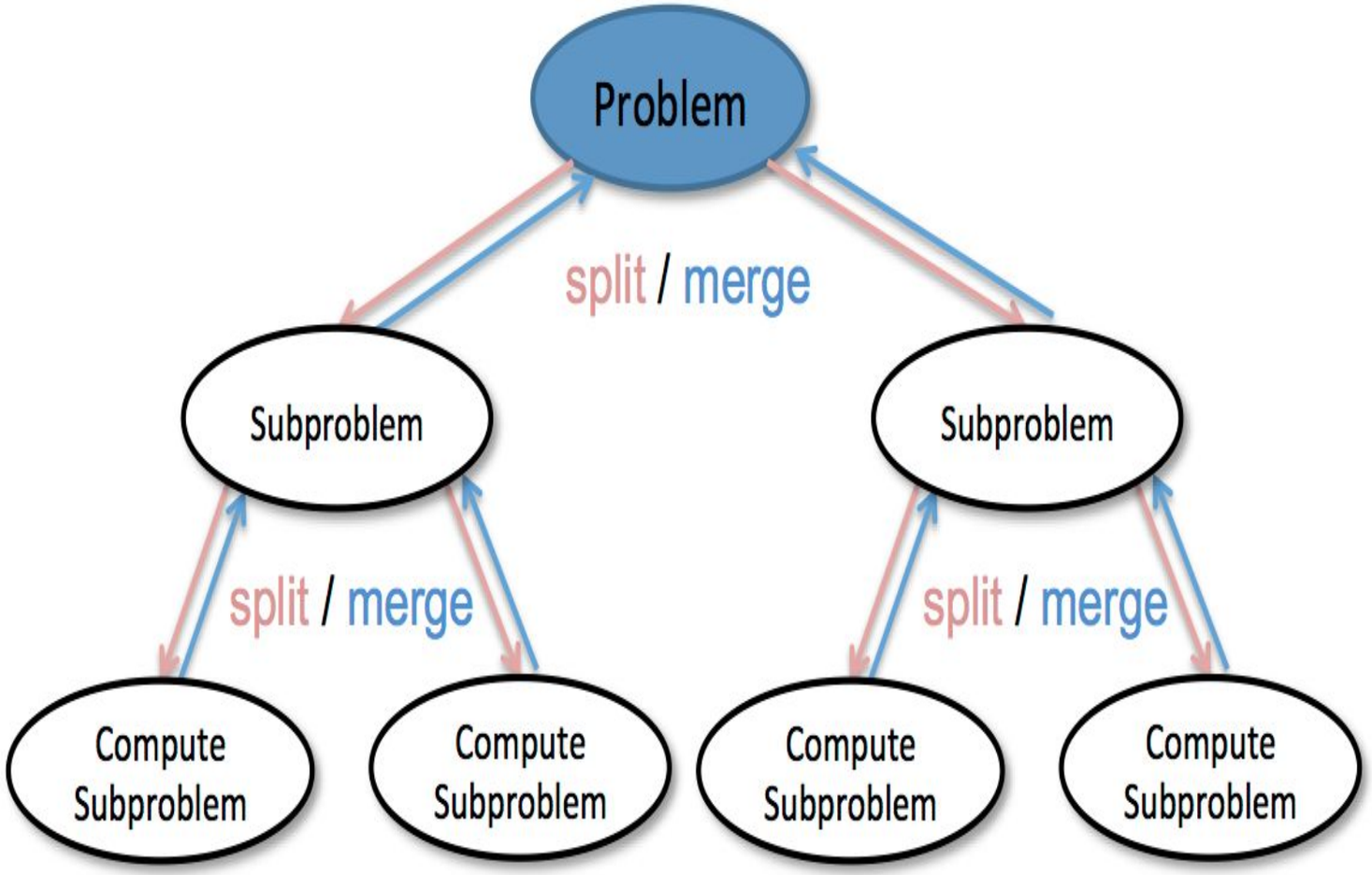

The foundation of Big Data processing, is the idea that a problem can be computed by multiple machines together. This allows many resources to be used in "parallel".

# Parallelism



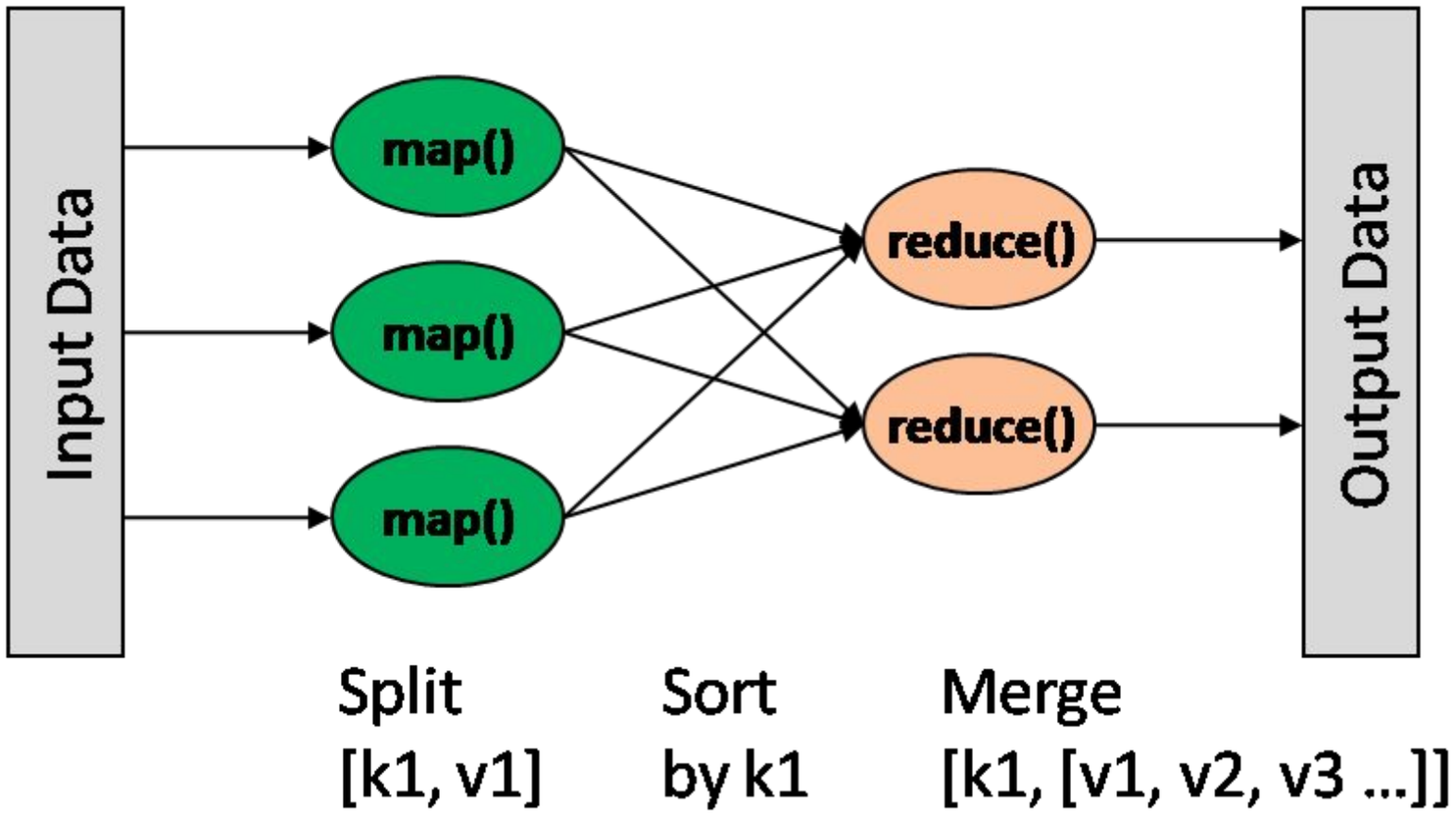

- Running multiple instances to process data
- Data can be subsetted and solved iteratively
- Sub-solutions can be solved independently

# Divide and Conquer



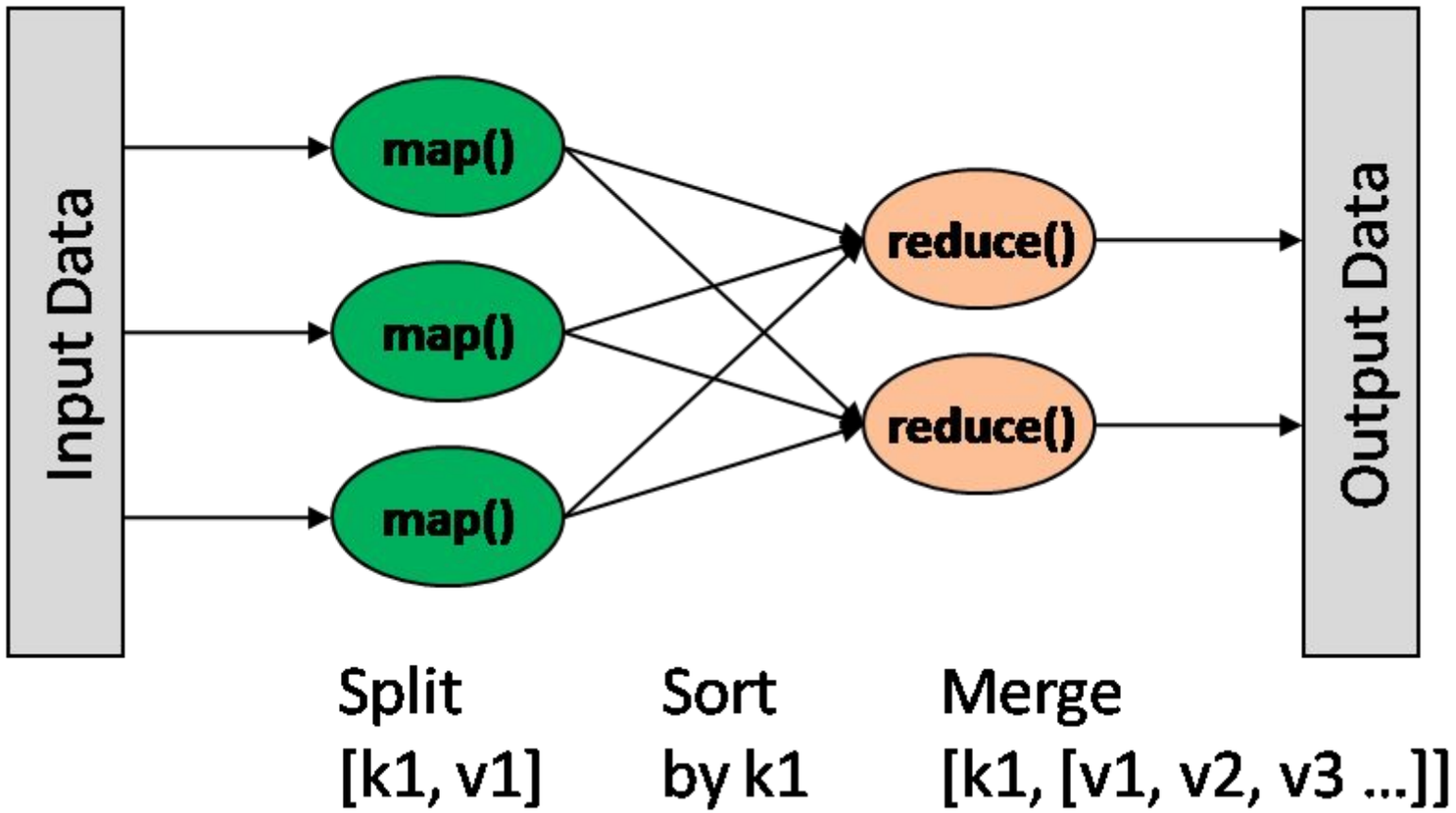

The defining characteristic of a problem that is suitable for the divide and conquer approach is that it can be broken down into independent subtasks.

# MapReduce




The term Map Reduce indicate a two-phase divide and conquer algorithm initially invented and publicized by Google in 2004.

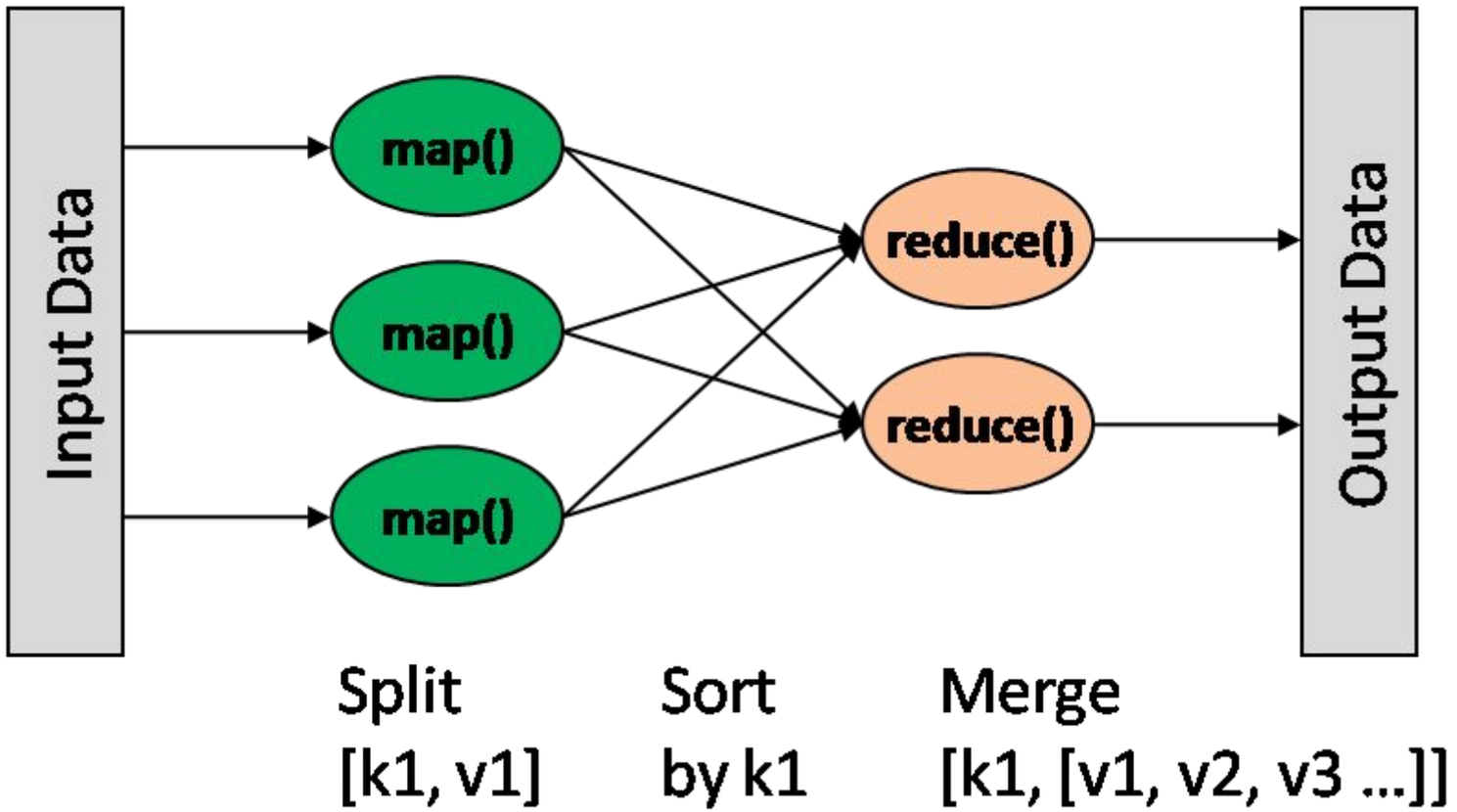
# MapReduce




It involves splitting a problem into subtasks and processing these subtasks in parallel and it consists of two phases:

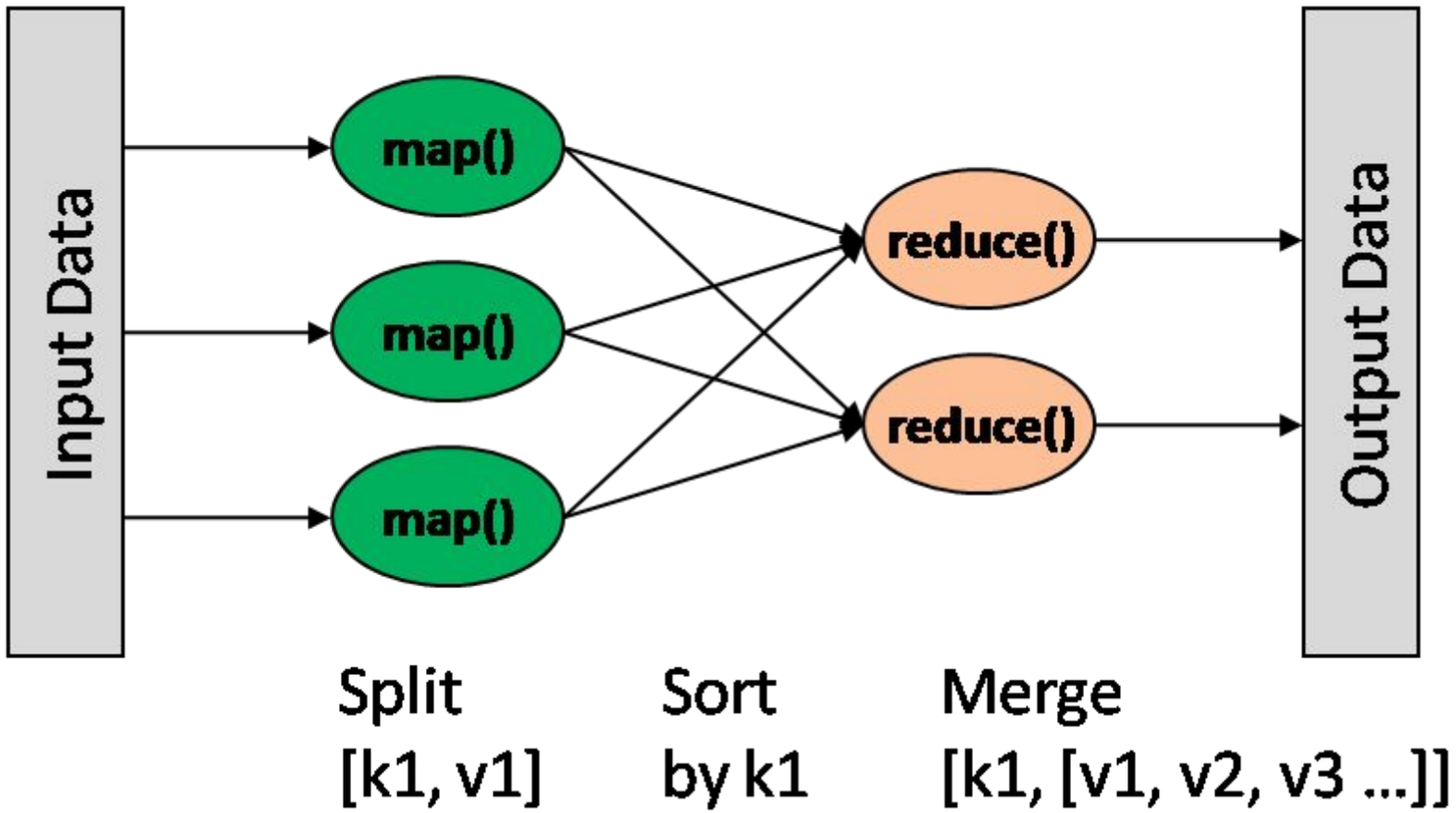
- the mapper phase
- the reducer phase

# MapReduce




In the ***mapper phase***, data is split into chunks and the same computation is performed on each chunk, while in the ***reducer phase,*** data is aggregated back to produce a final result.

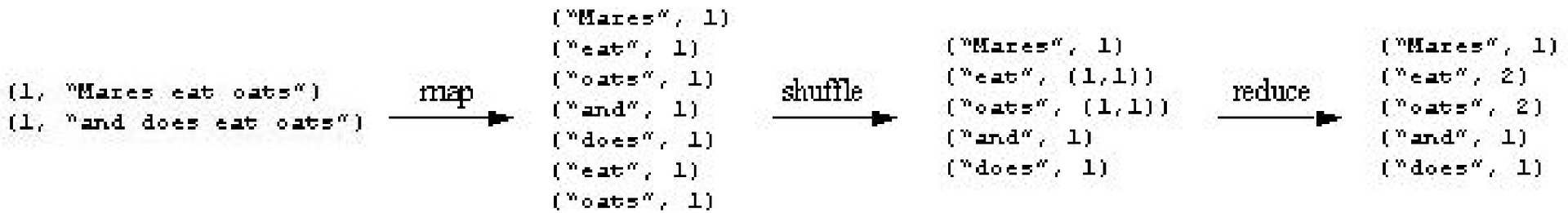
# MapReduce




Map-reduce uses a functional programming paradigm. The data processing primitives are mappers and reducers, as we've seen.

- mappers – filter & transform data
- reducers – aggregate results

# Key Value pairs

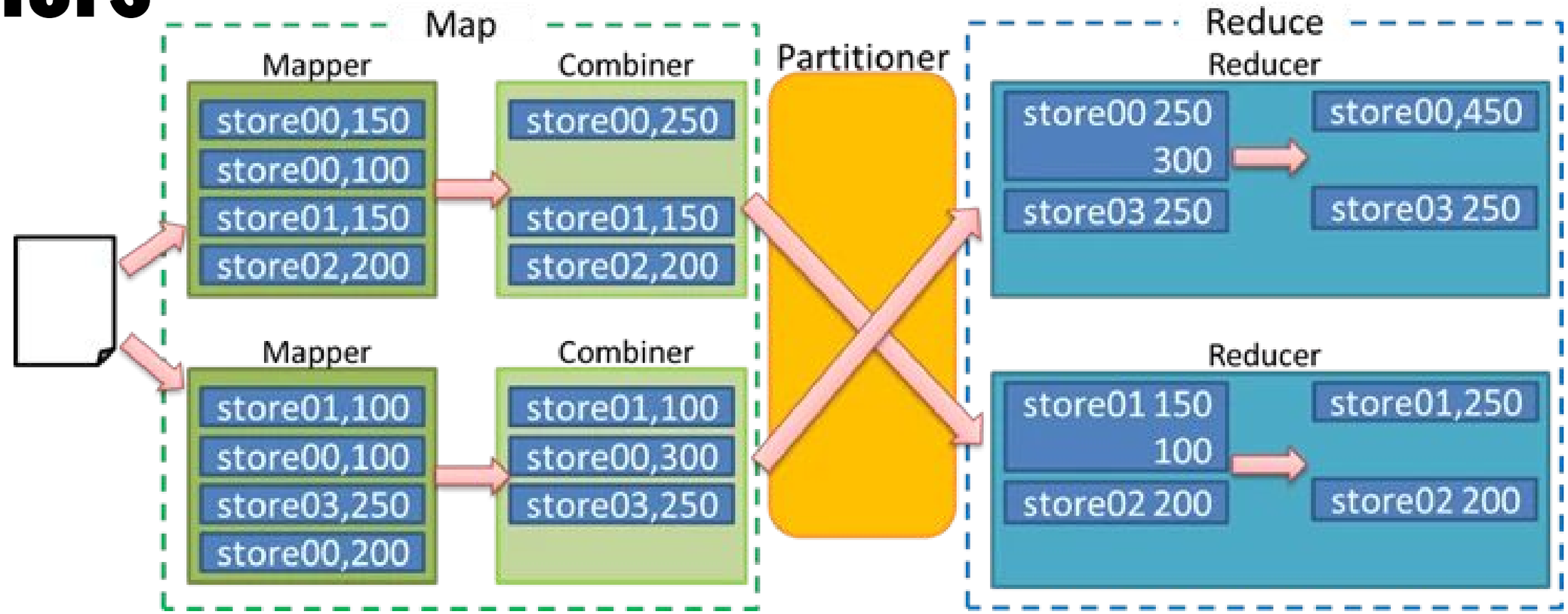



Data is passed through the various phases of a map-reduce pipeline as key-value pairs.

# What python data structures could be used to implement a key value pair?

# Exercises

# Combiners

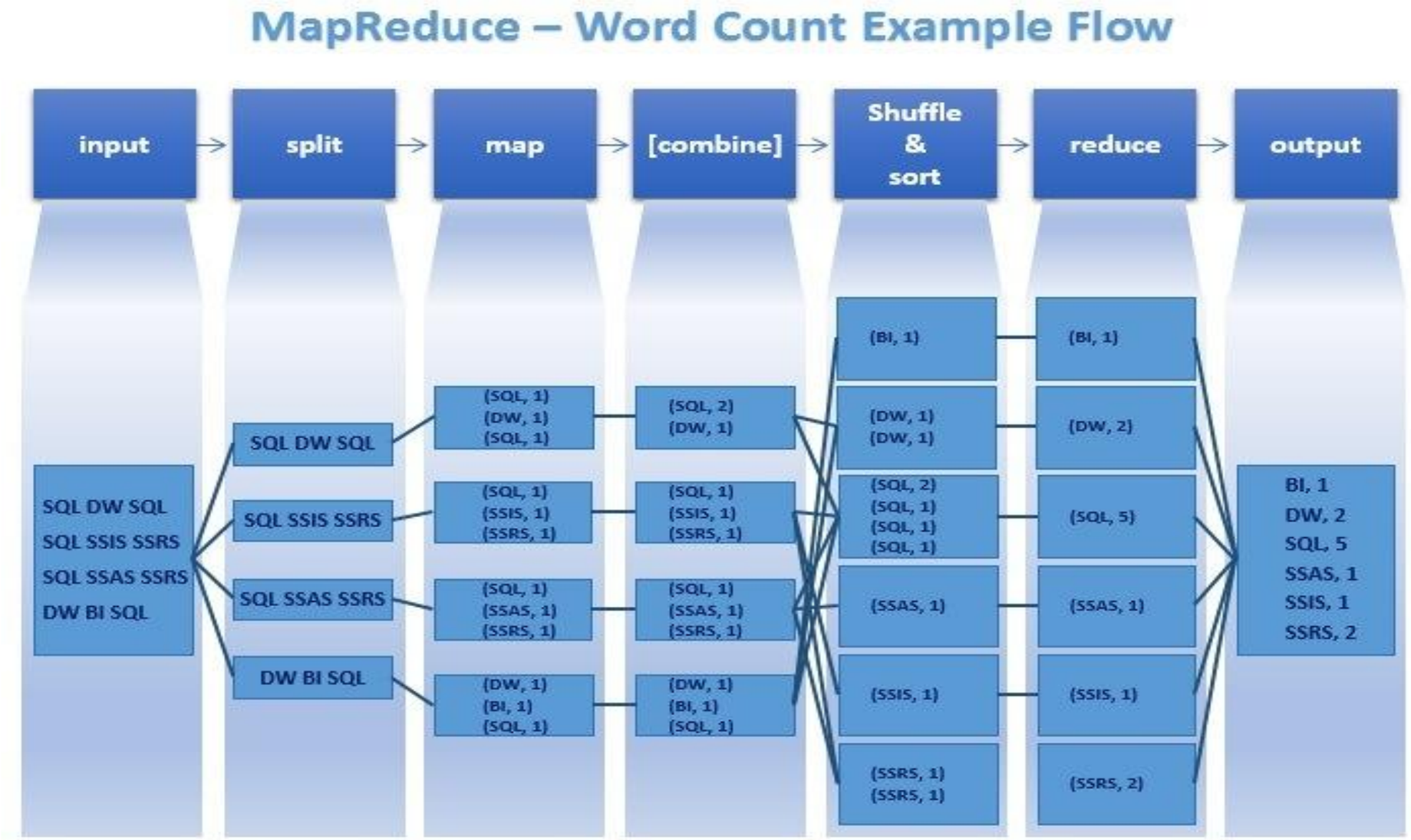



Combiners are intermediate reducers that are performed at node level in a multi node architecture.

Intro to Big Data

# MapReduce in python

# Intro to big data



# Optional Demo: Manual ARIMA

# Conclusion

# Q & A