Applying MapReduce to Common Data Problems

THINKING MAPREDUCE

Janani Ravi
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Overview

Data flow in a MapReduce

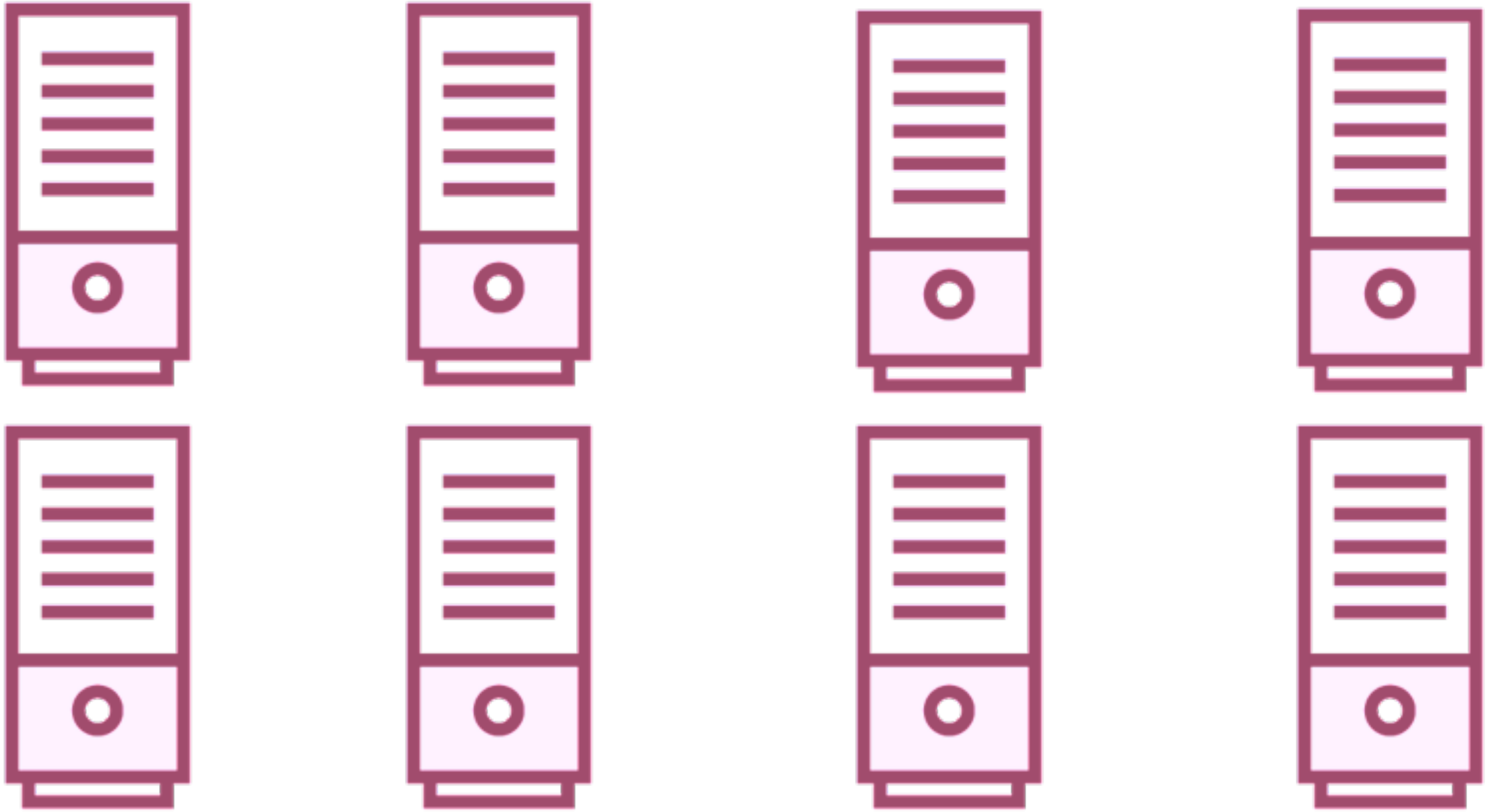
Break down tasks into Map and Reduce phases

Optimize MapReduce using a Combiner

MapReduce

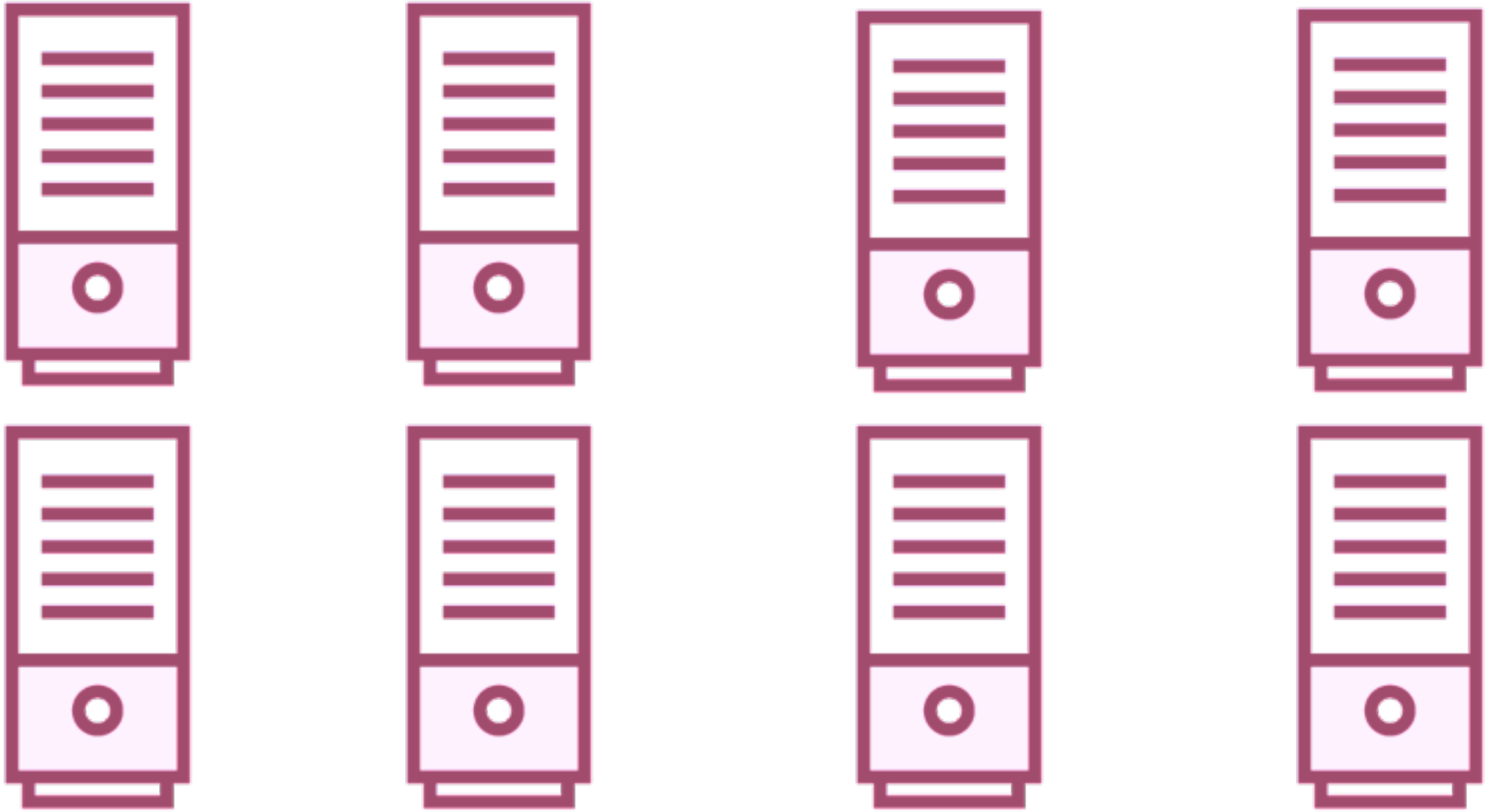
Processing huge amounts of data

MapReduce



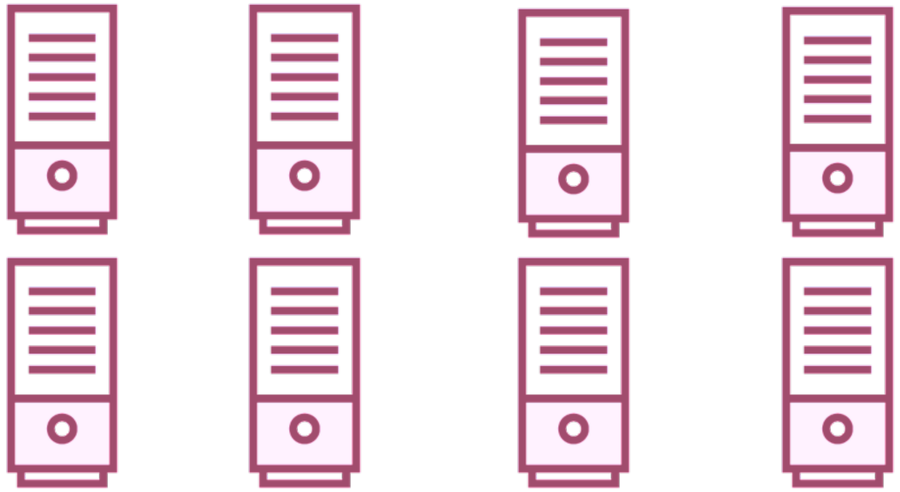
Requires running processes on many machines

MapReduce



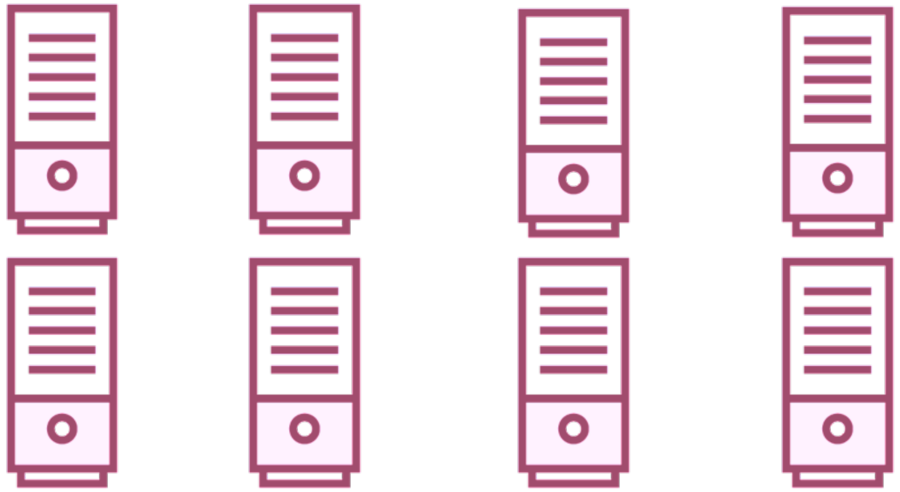
A distributed system

MapReduce



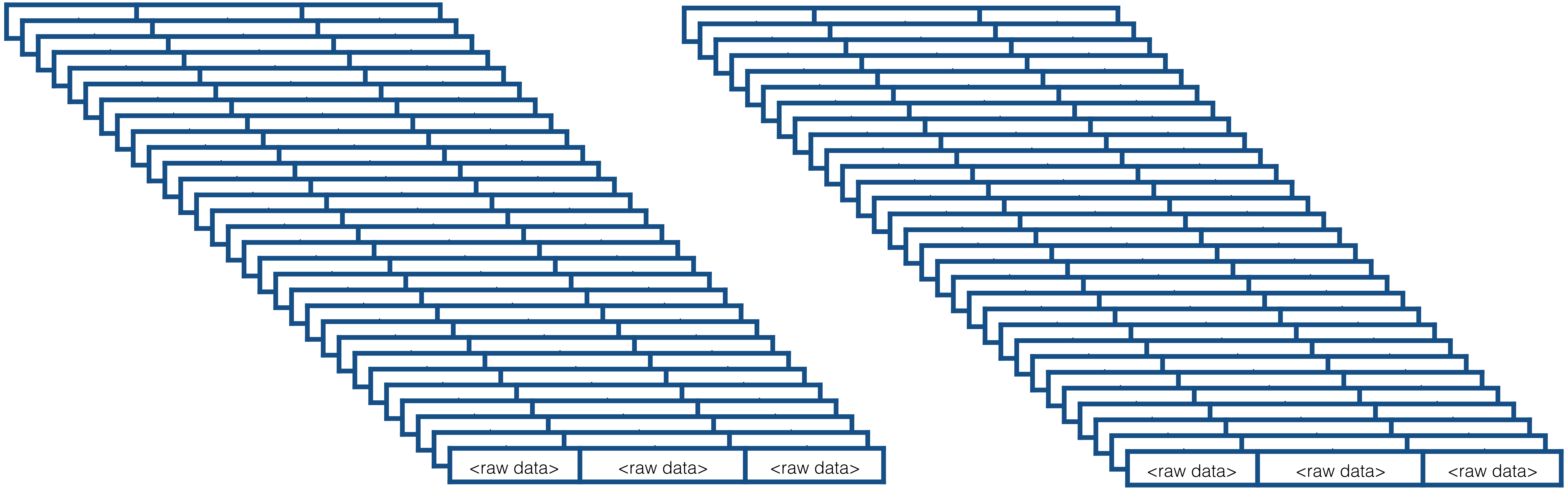
MapReduce is a programming paradigm

MapReduce



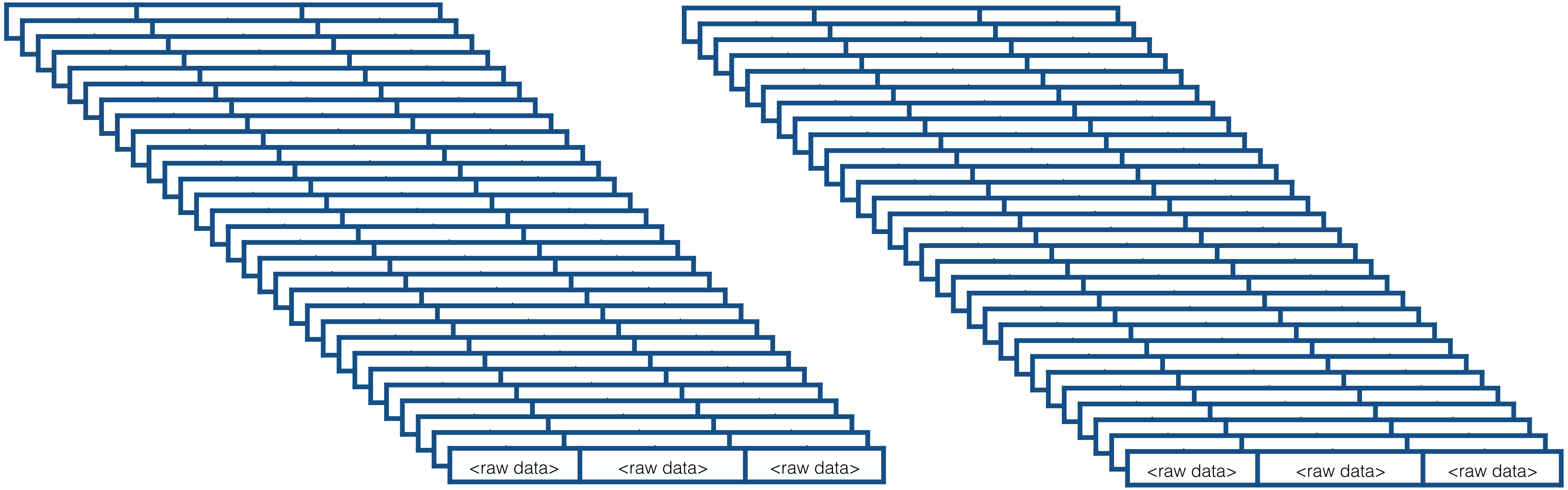
Takes advantage of the inherent parallelism in data processing

MapReduce



Modern systems generate millions of records of raw data

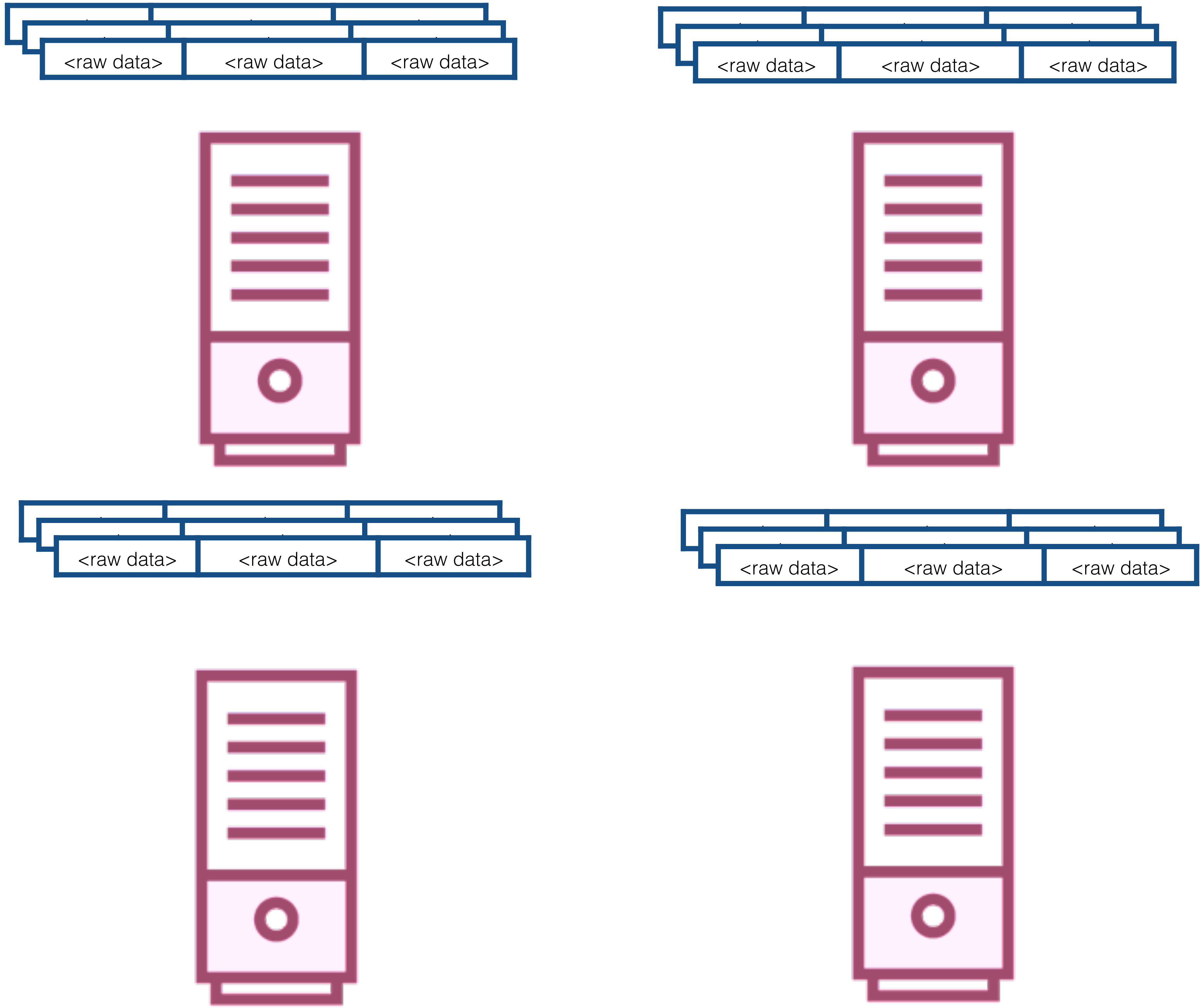
MapReduce



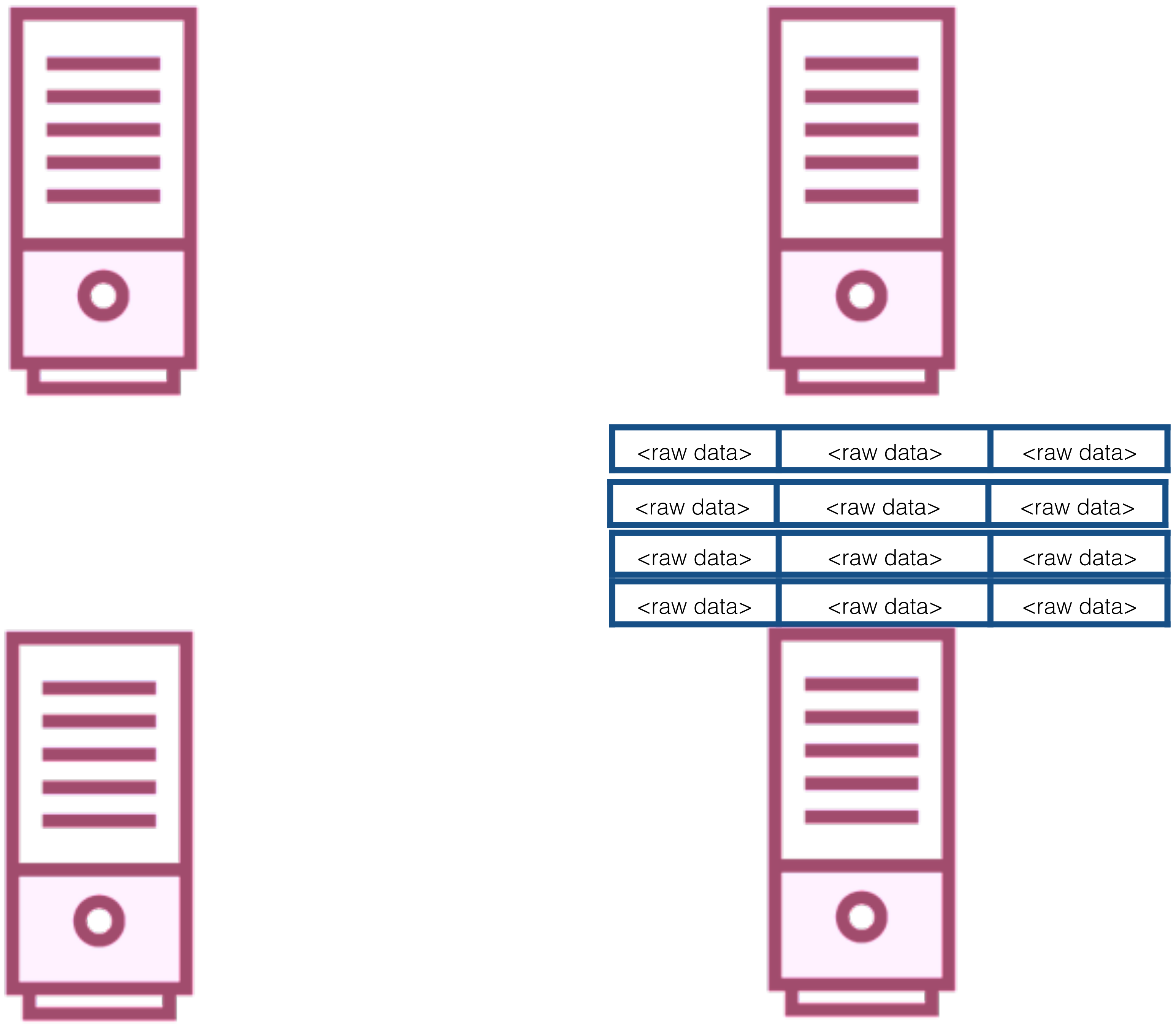
A task of this scale is processed in two stages

map reduce

map



reduce



<raw data>	<raw data>	<raw data>
<raw data>	<raw data>	<raw data>
<raw data>	<raw data>	<raw data>
<raw data>	<raw data>	<raw data>

MapReduce

`map` `reduce`

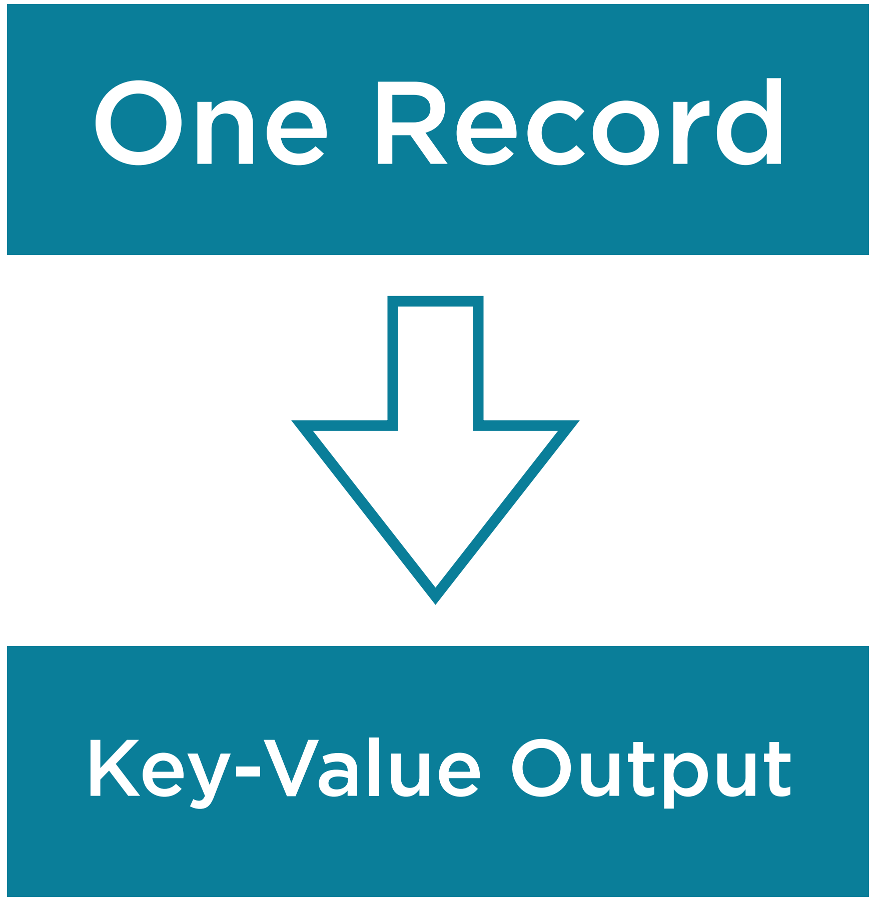
The programmer defines these 2 functions

Hadoop does the rest - behind the scenes

map

An operation performed in parallel, on small portions of the dataset

map



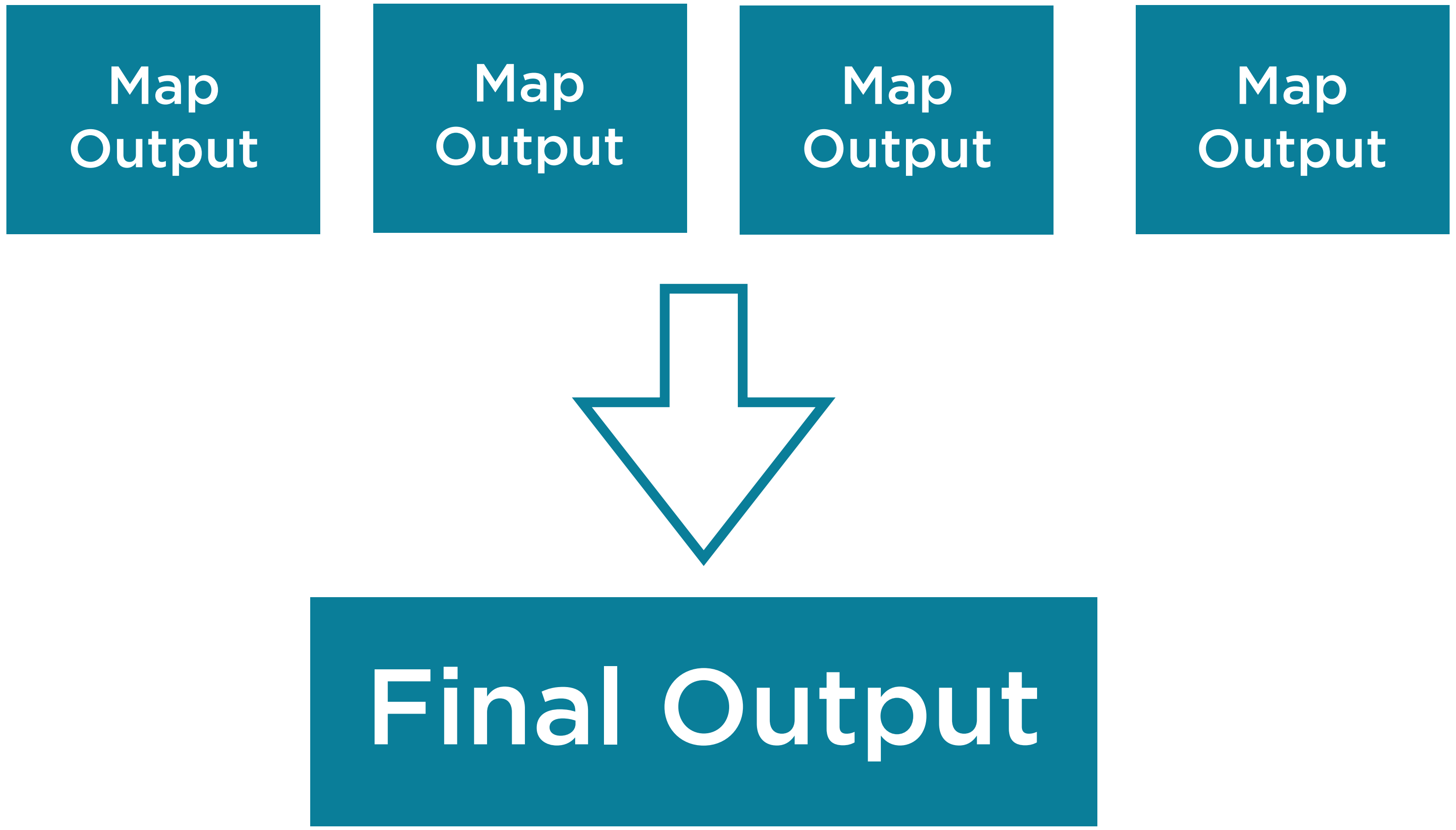
One Record

Key-Value Output

reduce

An operation to combine the results of the map step

reduce



Map Output

Map Output

Map Output

Map Output

Final Output

map A step that can be performed in parallel

reduce A step to combine the intermediate results

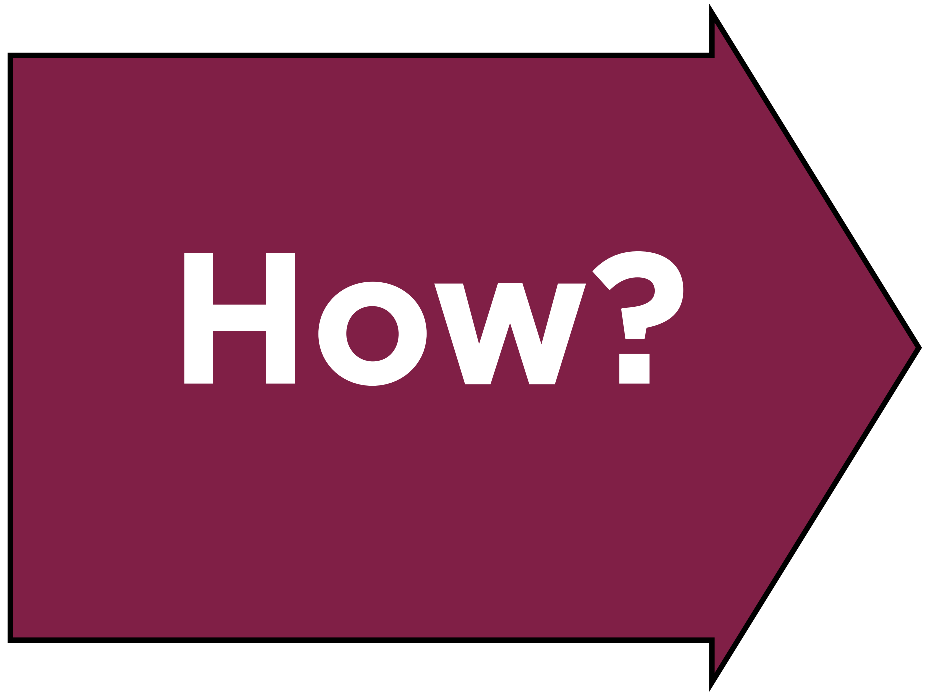
Breaking down any task into these two steps is almost an art

This course will teach you this art - with lots of opportunities to practice it

Counting Word Frequencies

Consider a large text file

Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky
Twinkle twinkle little star
How I wonder what you are
.....



Word	Frequency
above	14
are	20
how	21
star	22
twinkle	32
...	..

Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky
Twinkle twinkle little star
How I wonder what you are
.....

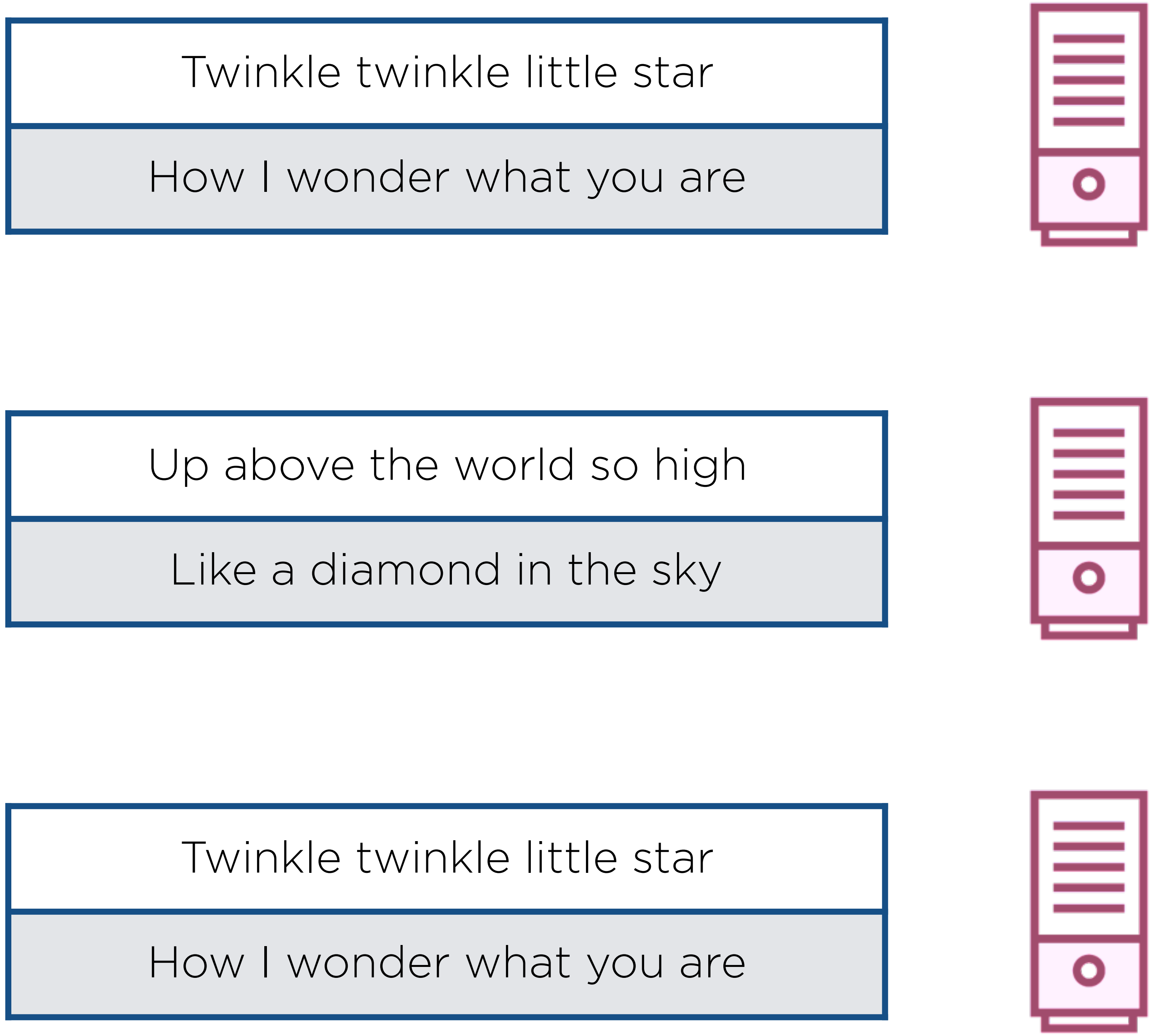
MapReduce Flow

The raw data is really large (potentially in PetaBytes)

It’s distributed across many machines in a cluster

Each machine holds a partition of data

MapReduce Flow



Twinkle twinkle little star

How I wonder what you are

Up above the world so high

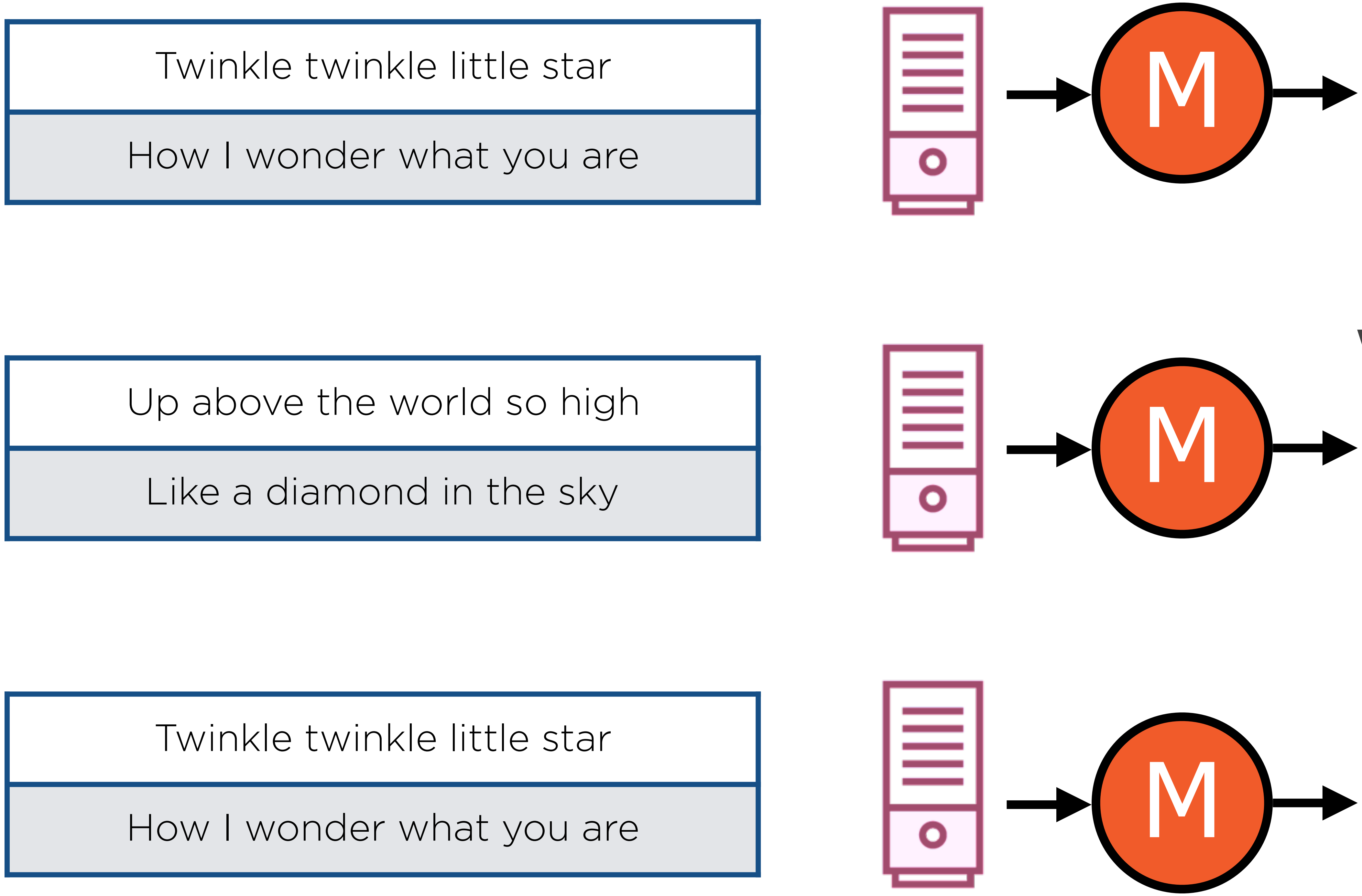
Like a diamond in the sky

Twinkle twinkle little star

How I wonder what you are

Each partition is given to a different process i.e. to mappers

MapReduce Flow



Twinkle twinkle little star

How I wonder what you are

Up above the world so high

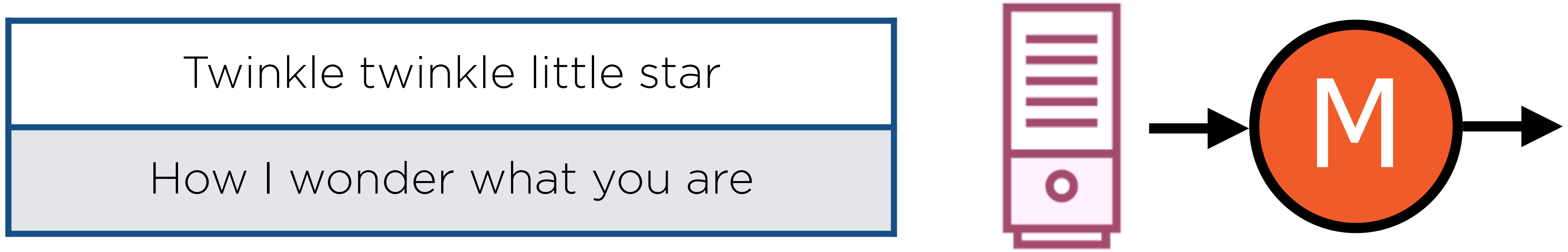
Like a diamond in the sky

Twinkle twinkle little star

How I wonder what you are

Each mapper works in parallel

Map Flow

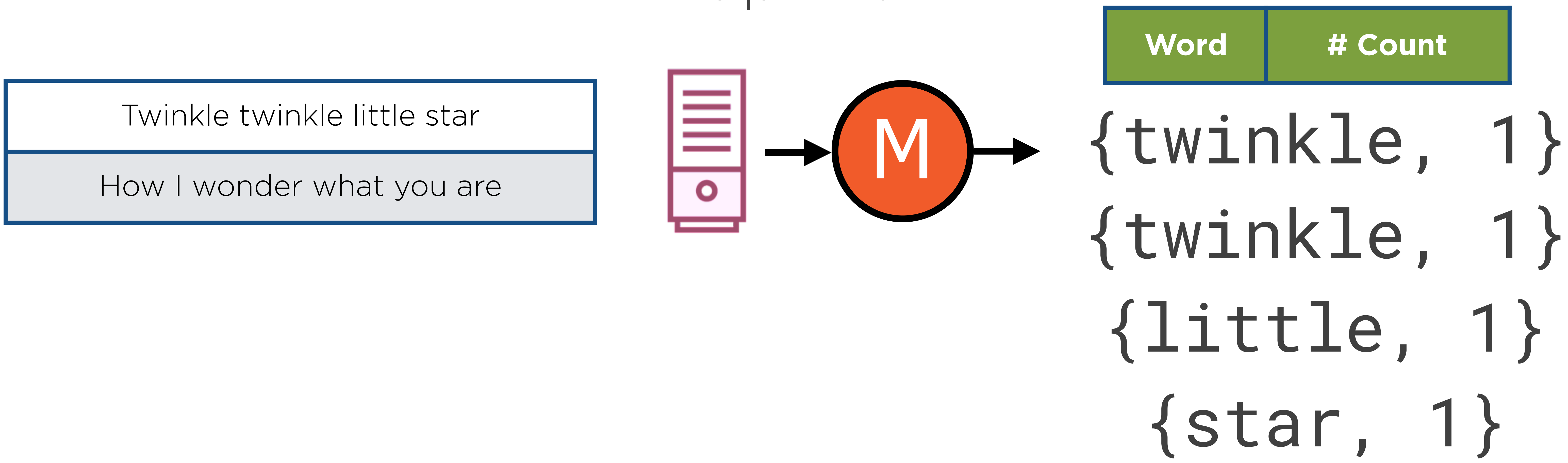


Twinkle twinkle little star

How I wonder what you are

Within each mapper, the rows are processed serially

Map Flow



Twinkle twinkle little star

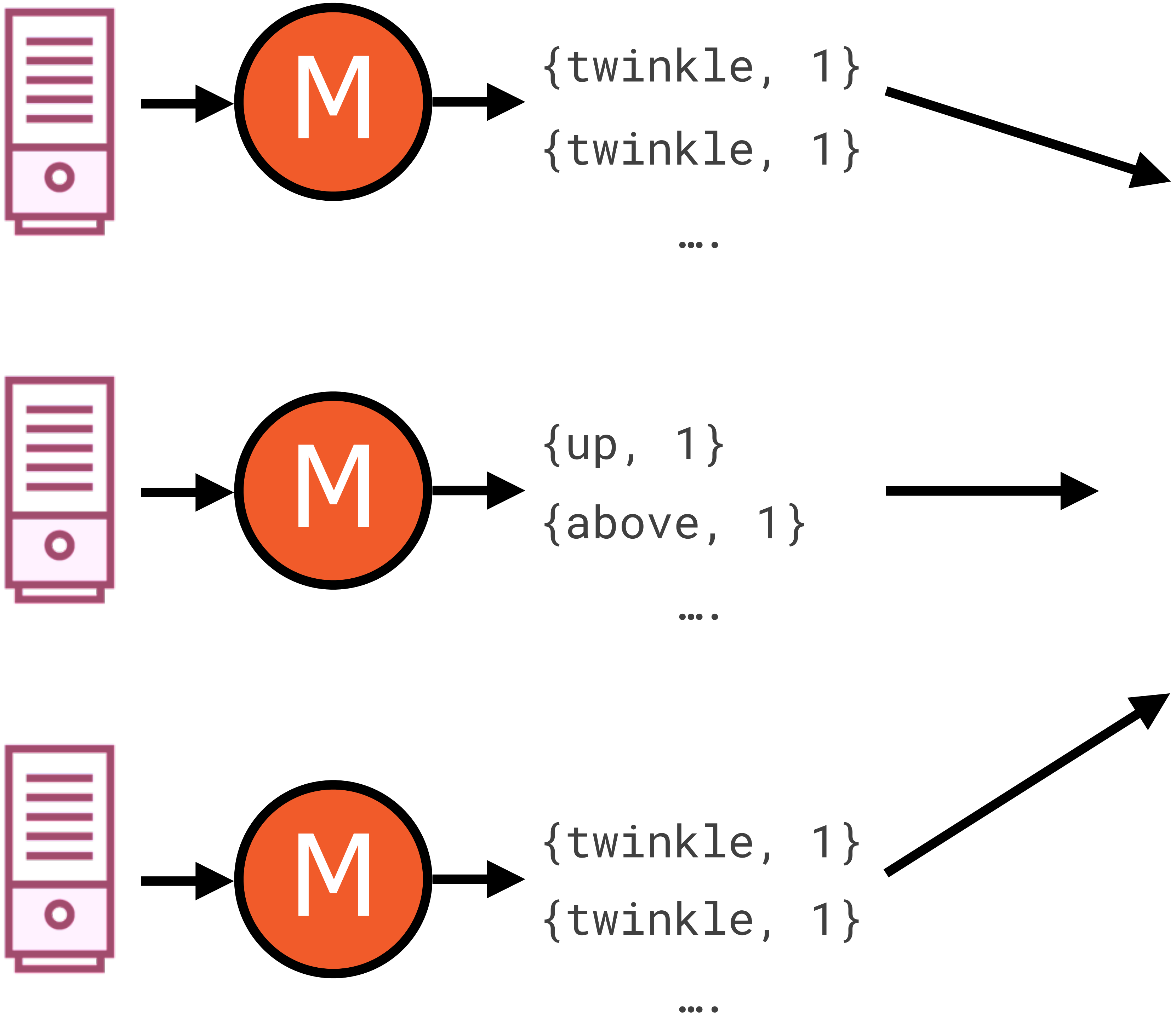
How I wonder what you are

Word	# Count

```
{twinkle, 1}
{twinkle, 1}
{little, 1}
{star, 1}
```

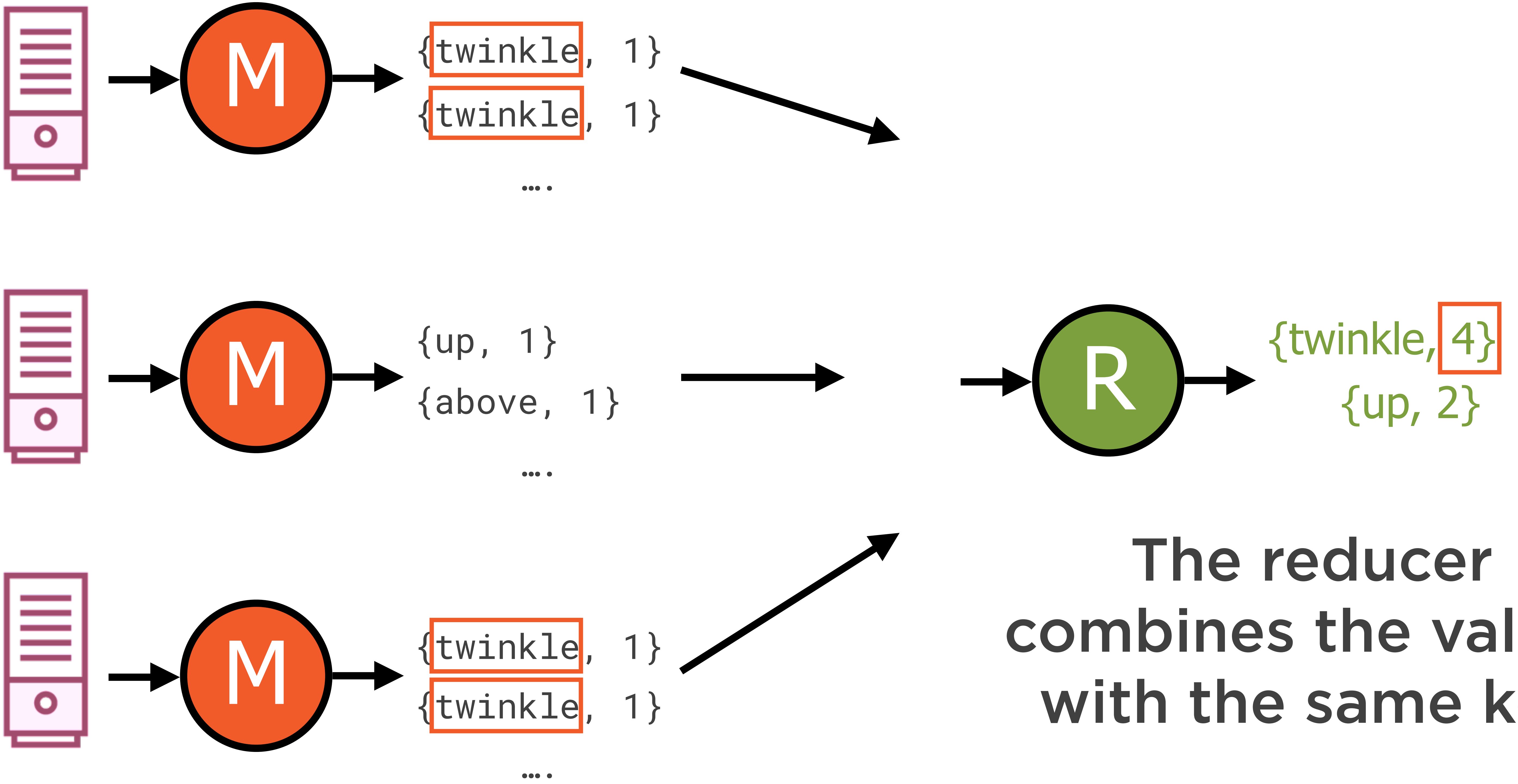
Each row emits {key, value} pairs

Reduce Flow



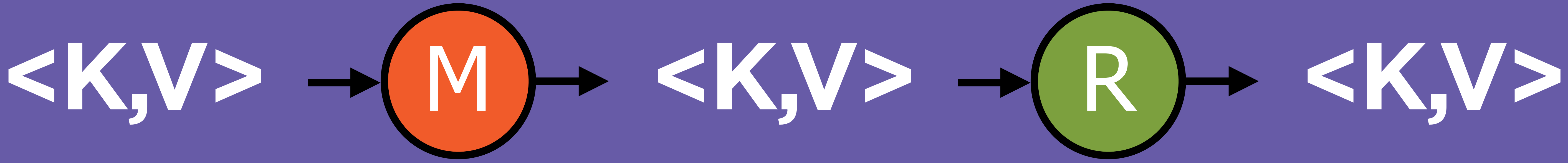
The results are passed on to another process i.e. a reducer

Reduce Flow



The reducer combines the values with the same key

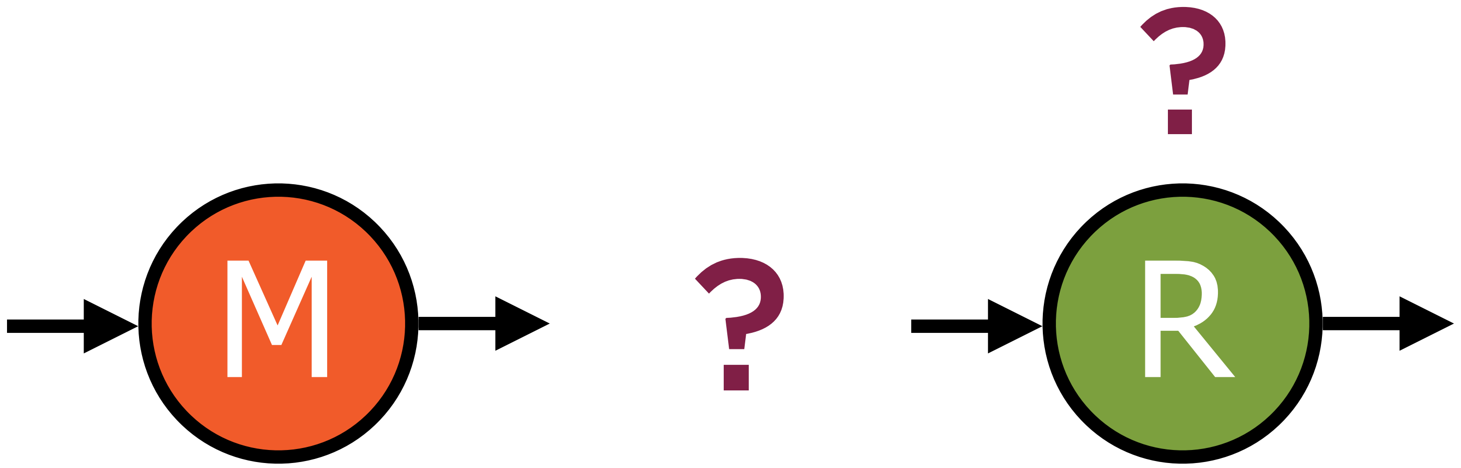
Key Insight Behind MapReduce



<K,V> → M → <K,V> → R → <K,V>

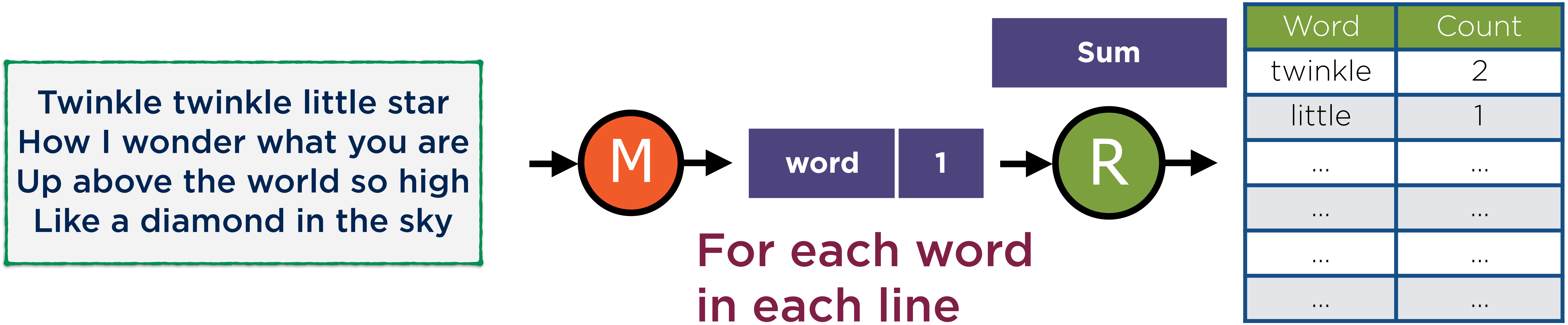
Many data processing tasks can be expressed in this form

Answer Two Questions



1. What {key, value} pairs should be emitted in the map step?
2. How should values with the same key be combined?

Counting Word Frequencies

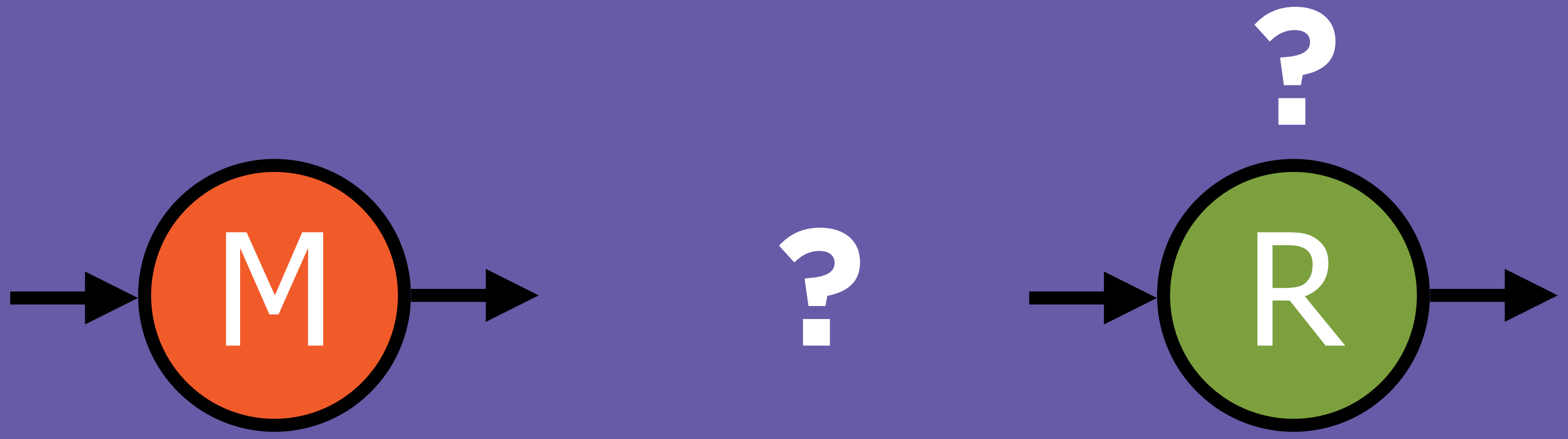


Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky

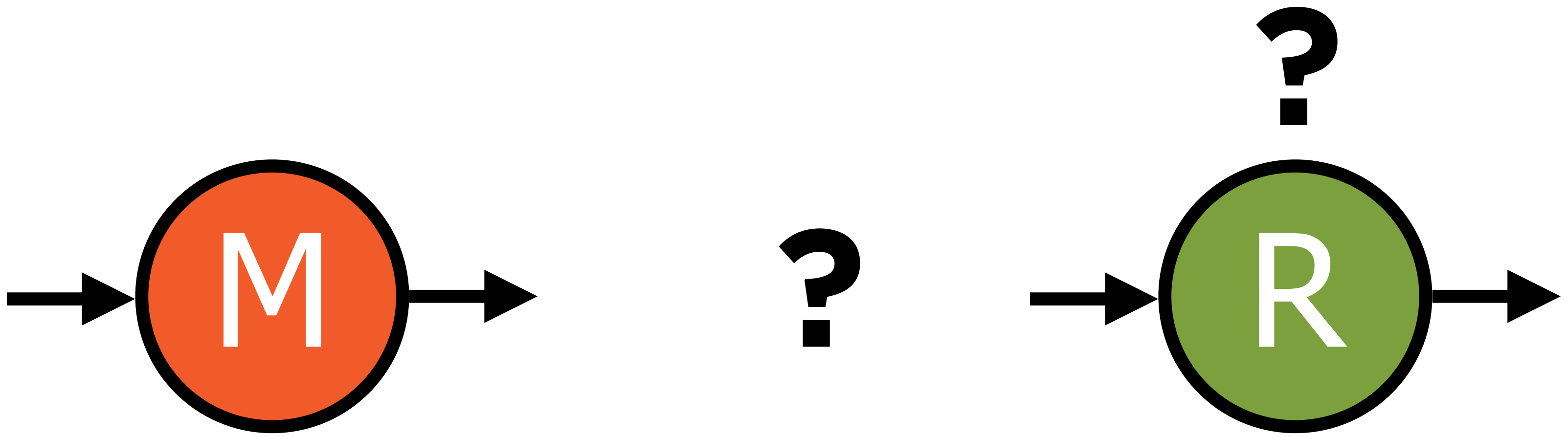
Word	Count
twinkle	2
little	1
…	…
…	…
…	…
…	…

For each word in each line

```
{twinkle, 1}
{twinkle, 1}
{little, 1}
{star, 1}
. .
…
```



Answer these to parallelize any task :)

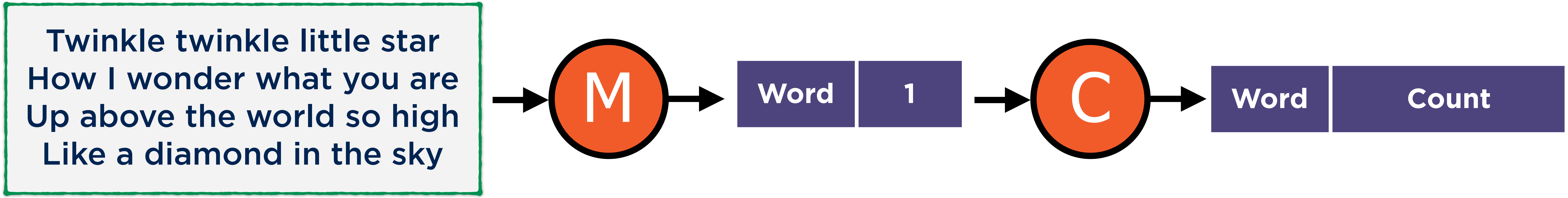


The parallelism here is in the map phase

Can we do more?

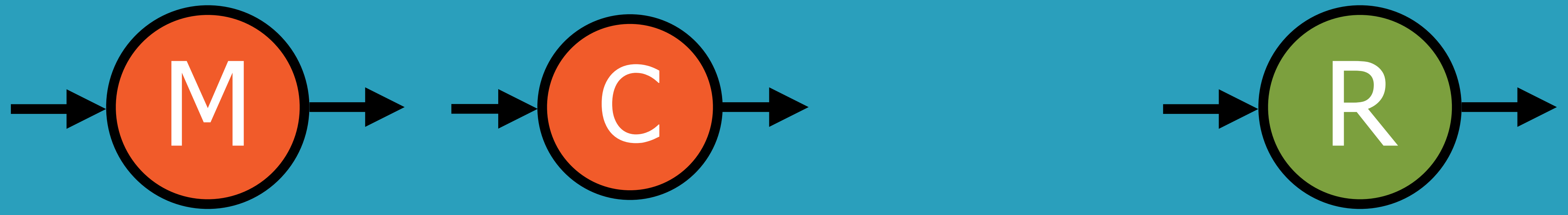
Use a combiner

Using a Combiner



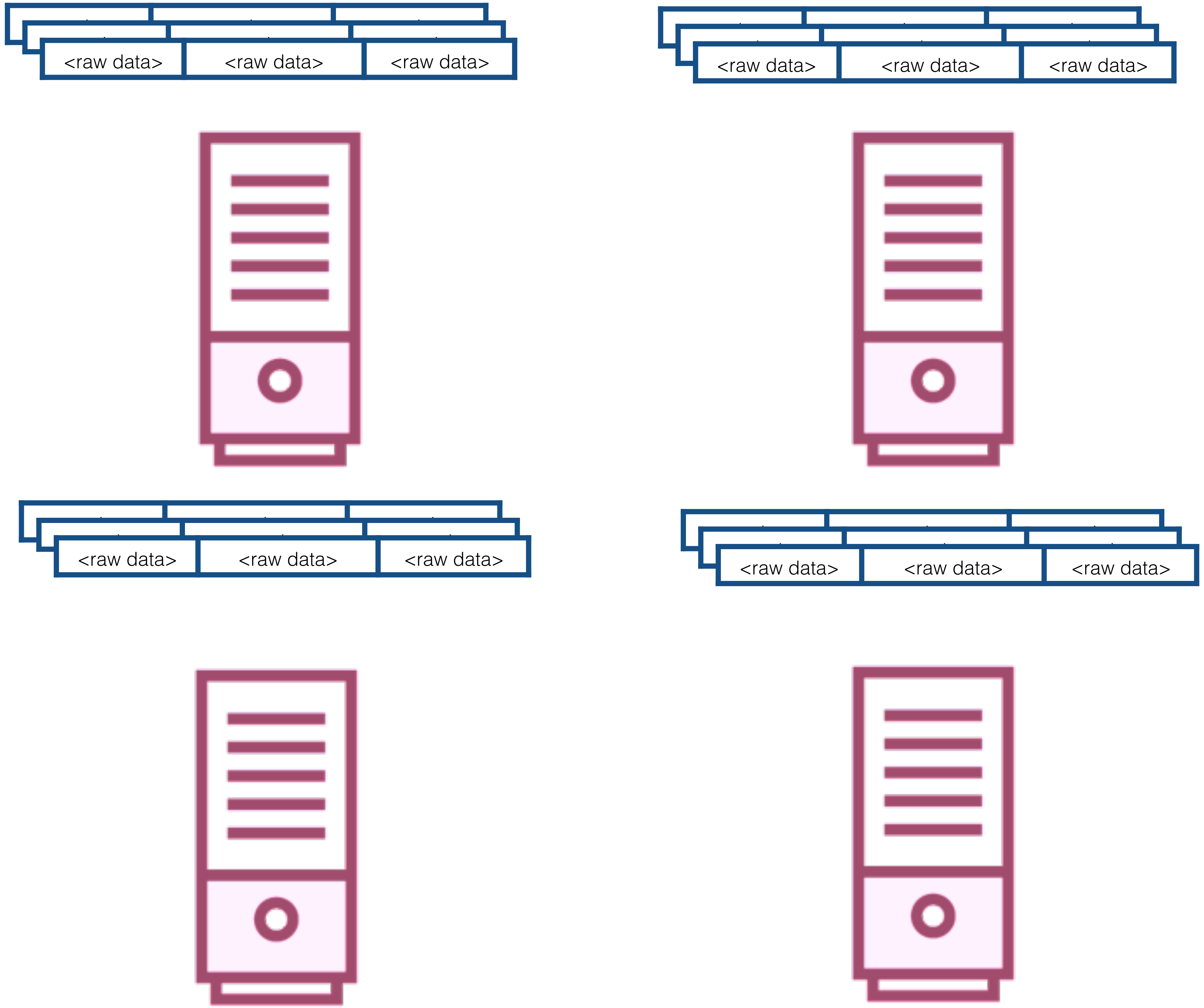
Twinkle twinkle little star
How I wonder what you are
Up above the world so high
Like a diamond in the sky

Combine values with the same key before they are copied over to the reducer

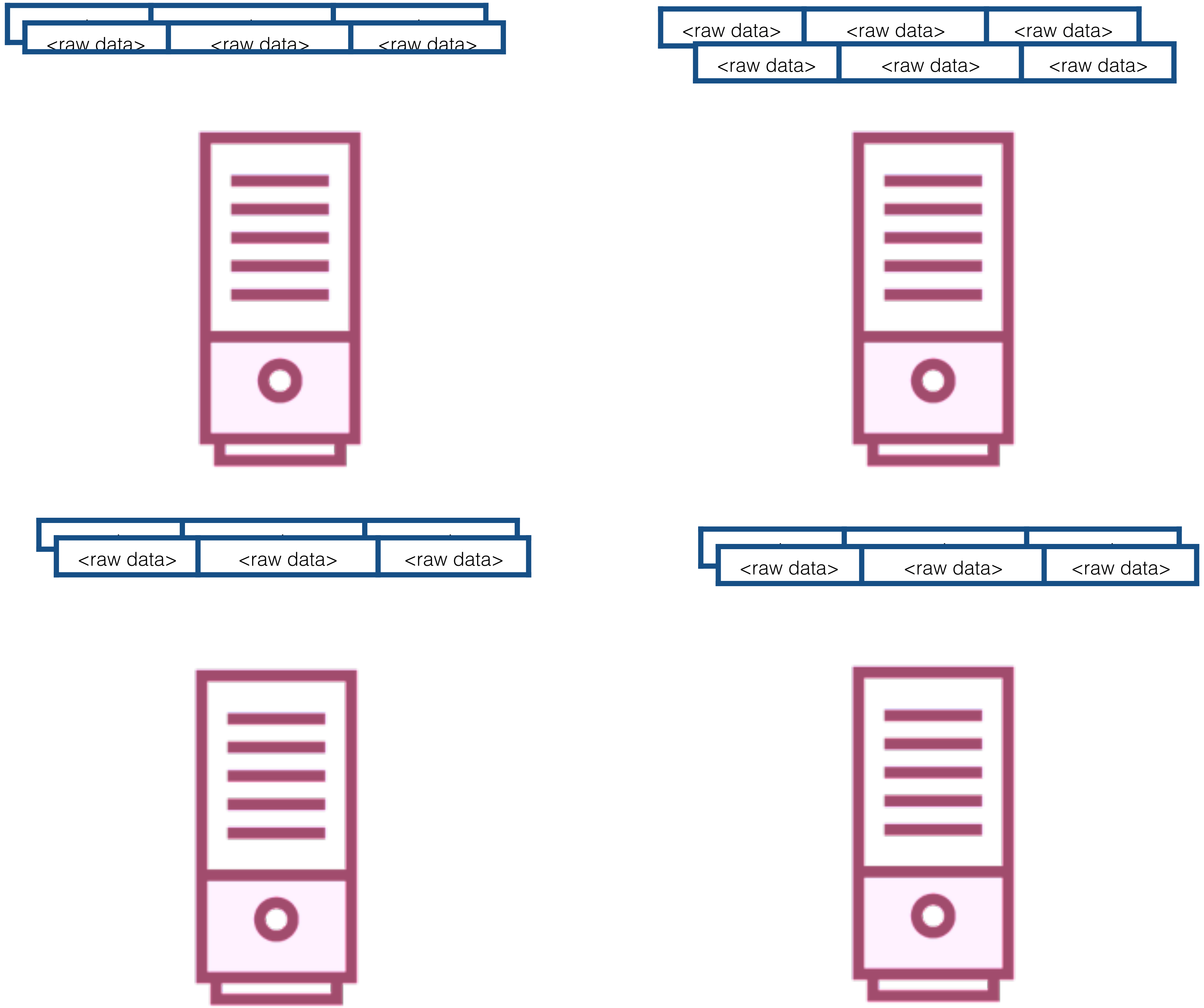


A combiner works on the mapper output before it is sent to the reducer

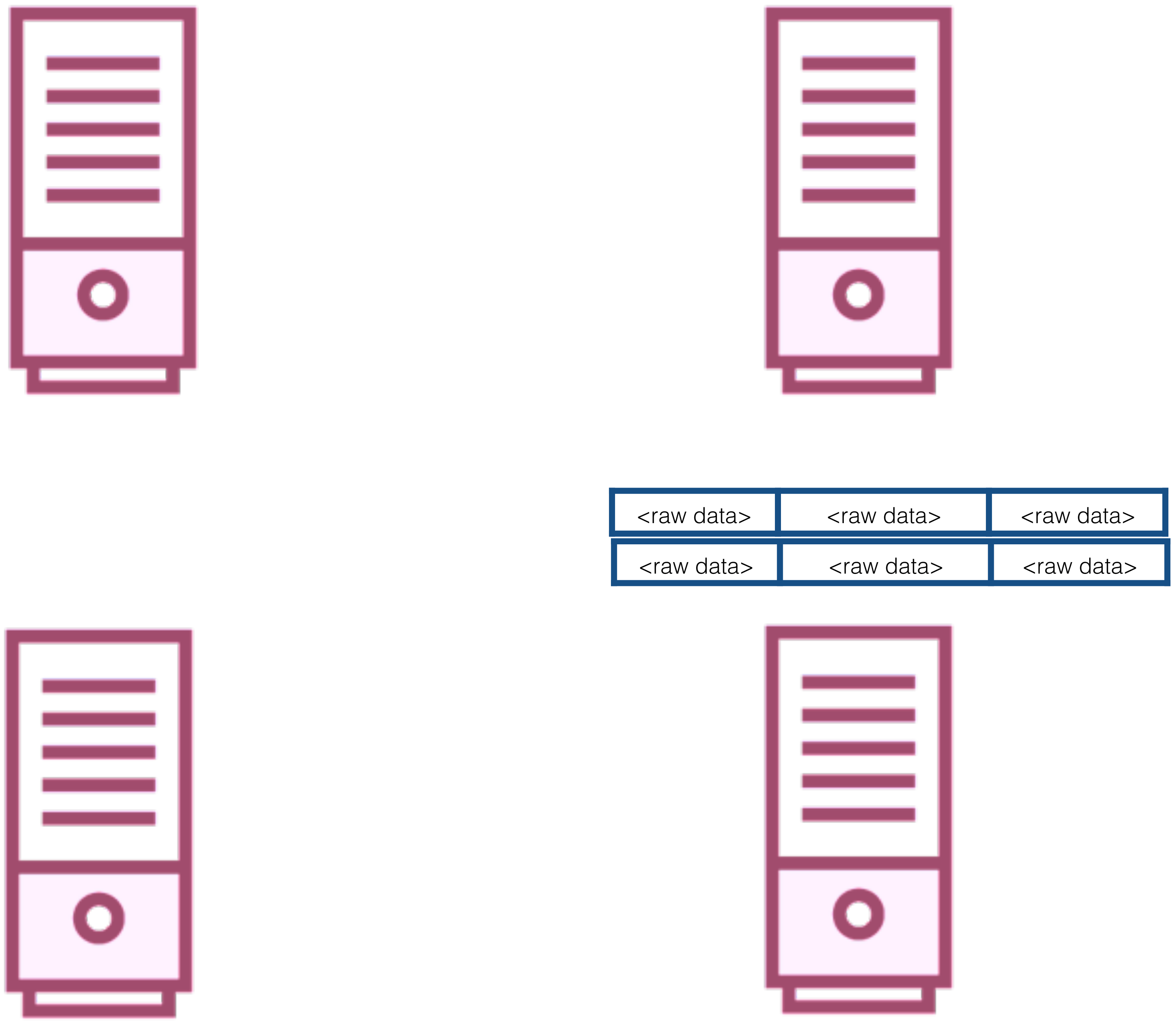
map



combine



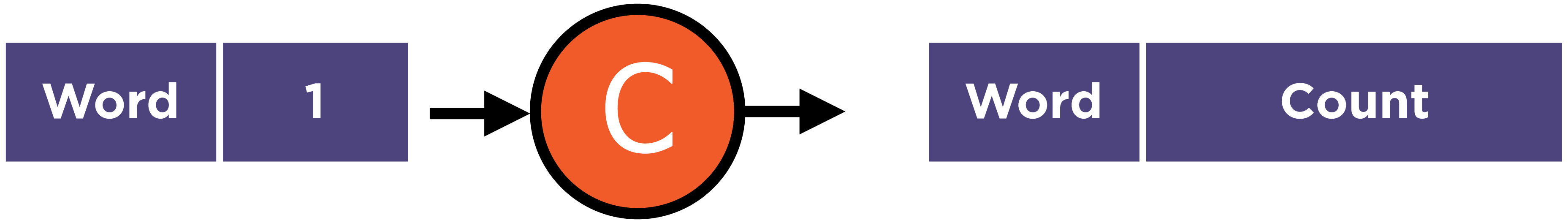
reduce



<raw data>	<raw data>	<raw data>
<raw data>	<raw data>	<raw data>

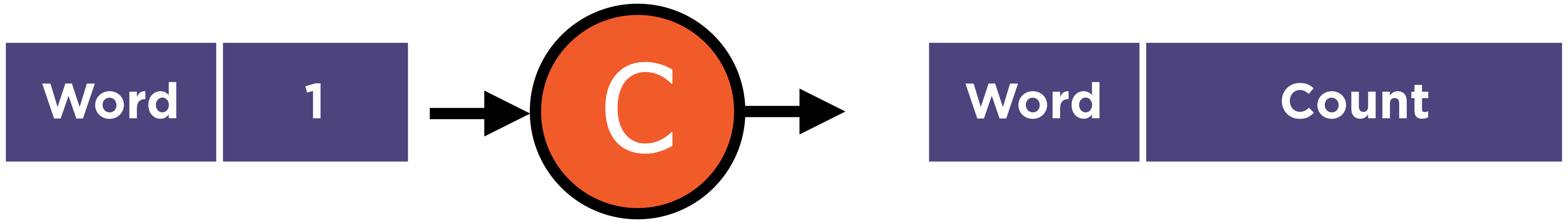
Using a Combiner

Very often, this is the same logic that happens in the Reducer!



Using a Combiner

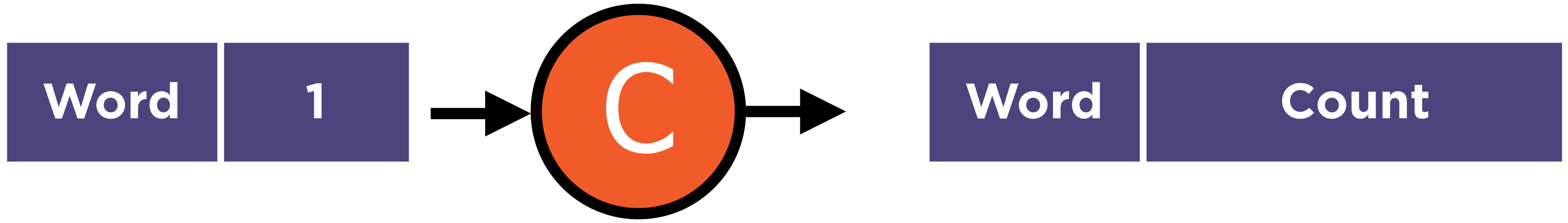
The result of the MapReduce with and without combiners should be the same



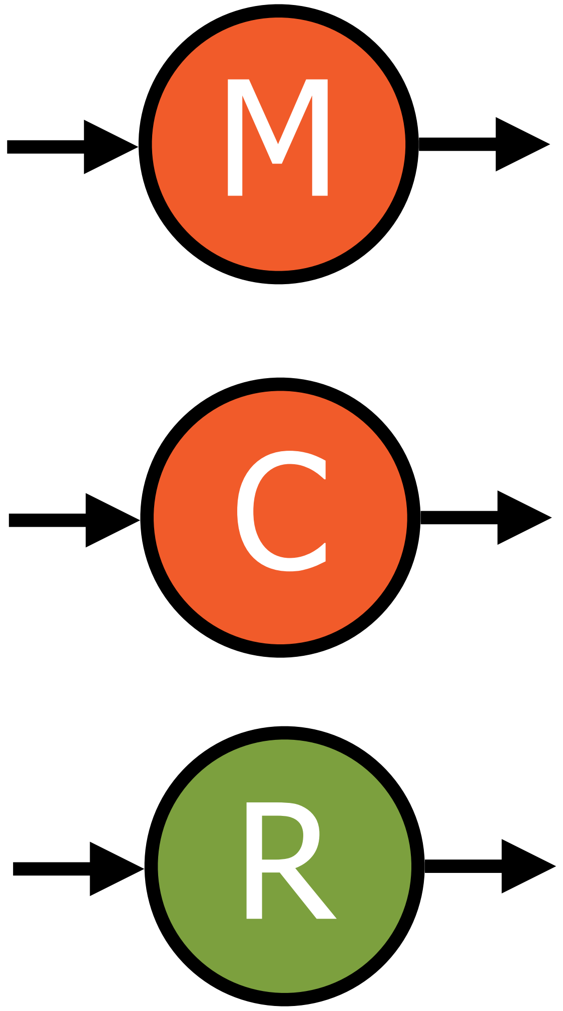
Using a Combiner

Improve parallelism by doing more in the map phase

Reduce data transfer across the cluster to the reduce nodes

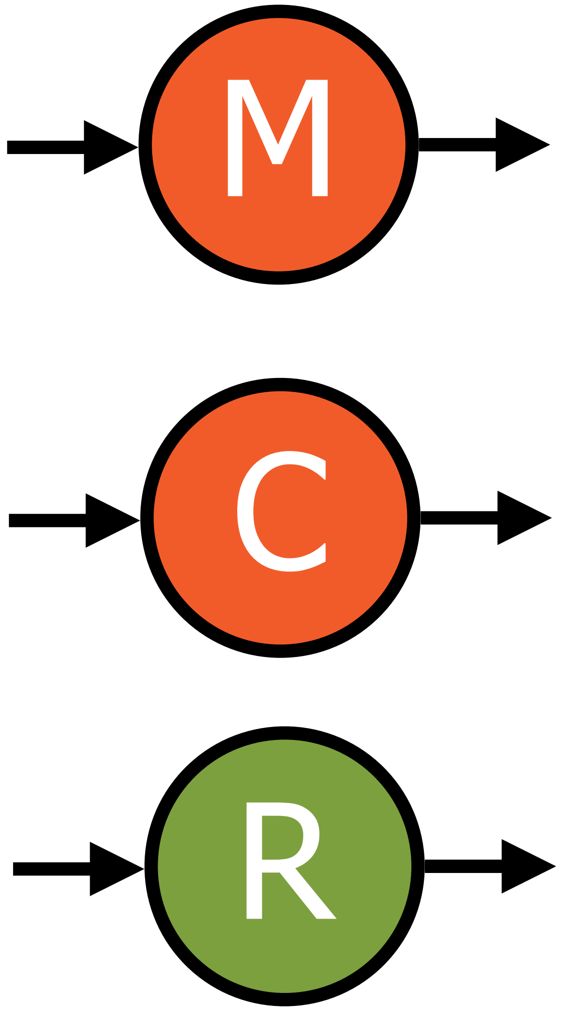


Implementing in Java



Implement a class with the logic for each step

Implementing in Java



A Mapper Class

A Reducer Class

A Reducer Class

MapReduce forces the programmer
to **Think Parallel**

Think Parallel

Filtering

Counting

Ranking

Min/Max/Avg

Whatever the task, break it down into 2 steps

- A step that can be performed in parallel
- A step to combine the intermediate results

Demo

Download Hadoop jars

Setup a MapReduce project in IntelliJ

Summary

An overview of MapReduce

Processing using Map and Reduce phases

More parallelism using a Combiner