Executing MapReduce Using Pig

Overview

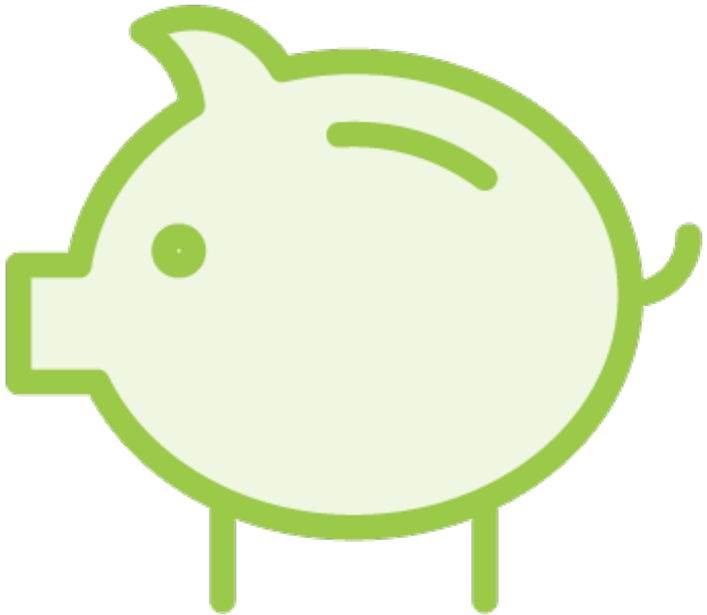
Perform multiple operations using a single foreach iterator, the nested foreach

Understand the MapReduce parallel programming paradigm

Implement the word count MapReduce program in Pig

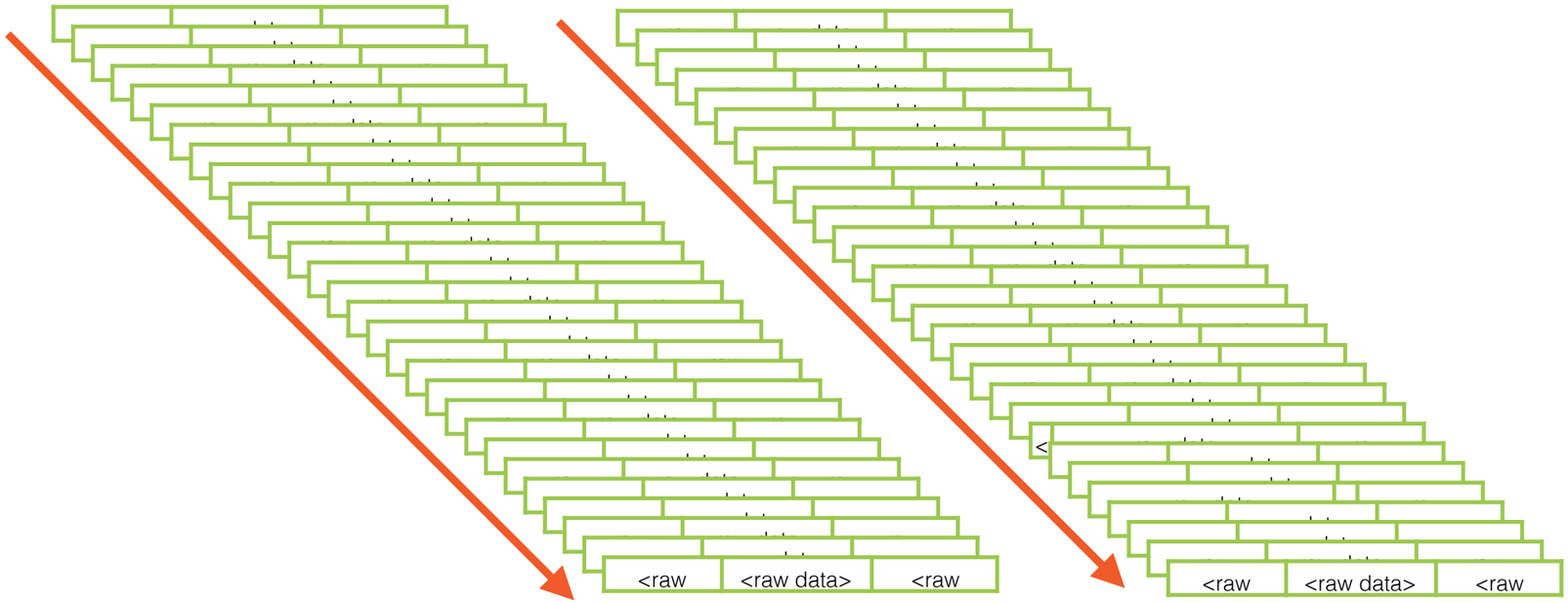
The Nested Foreach Command

Pig Works on Huge Datasets



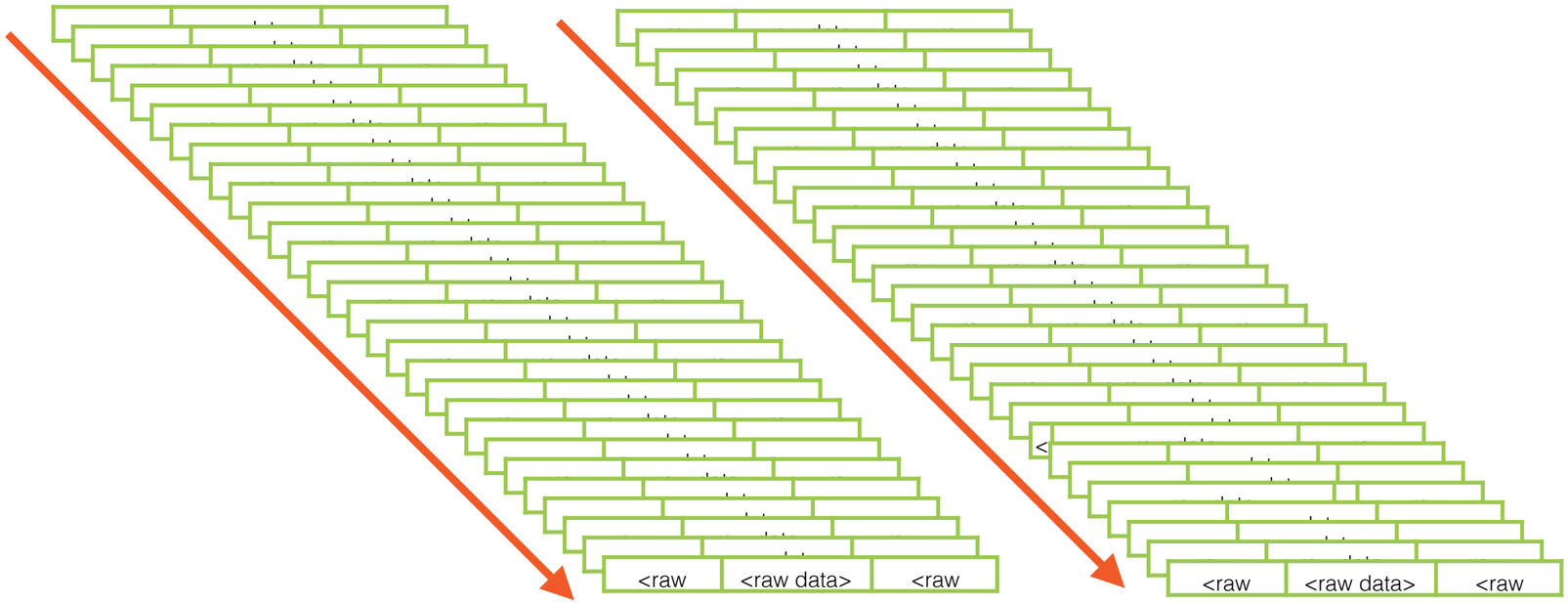
Relations may hold millions of records

Pig Works on Huge Datasets



The foreach keyword iterates through every record

Pig Works on Huge Datasets



Multiple iterations will degrade performance

Calculating Average

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o2	shoes	1	69
o3	book	2	22
o4	phone	1	149
o5	belt	2	19

Calculate the average revenue per order

Calculating Average

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o2	shoes	1	69
o3	book	2	22
o4	phone	1	149
o5	belt	2	19

SUM(amount)

Calculating Average

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o2	shoes	1	69
o3	book	2	22
o4	phone	1	149
o5	belt	2	19

SUM(amount) / COUNT(ID)

Calculating Average

ID	Product_ID	Quantity	Amount
o1	phone	1	199
o2	shoes	1	69
o3	book	2	22
o4	phone	1	149
o5	belt	2	19

Performing both SUM() and COUNT() operations in one pass is very efficient

Nested Foreach

Combine multiple operations over the records of a dataset

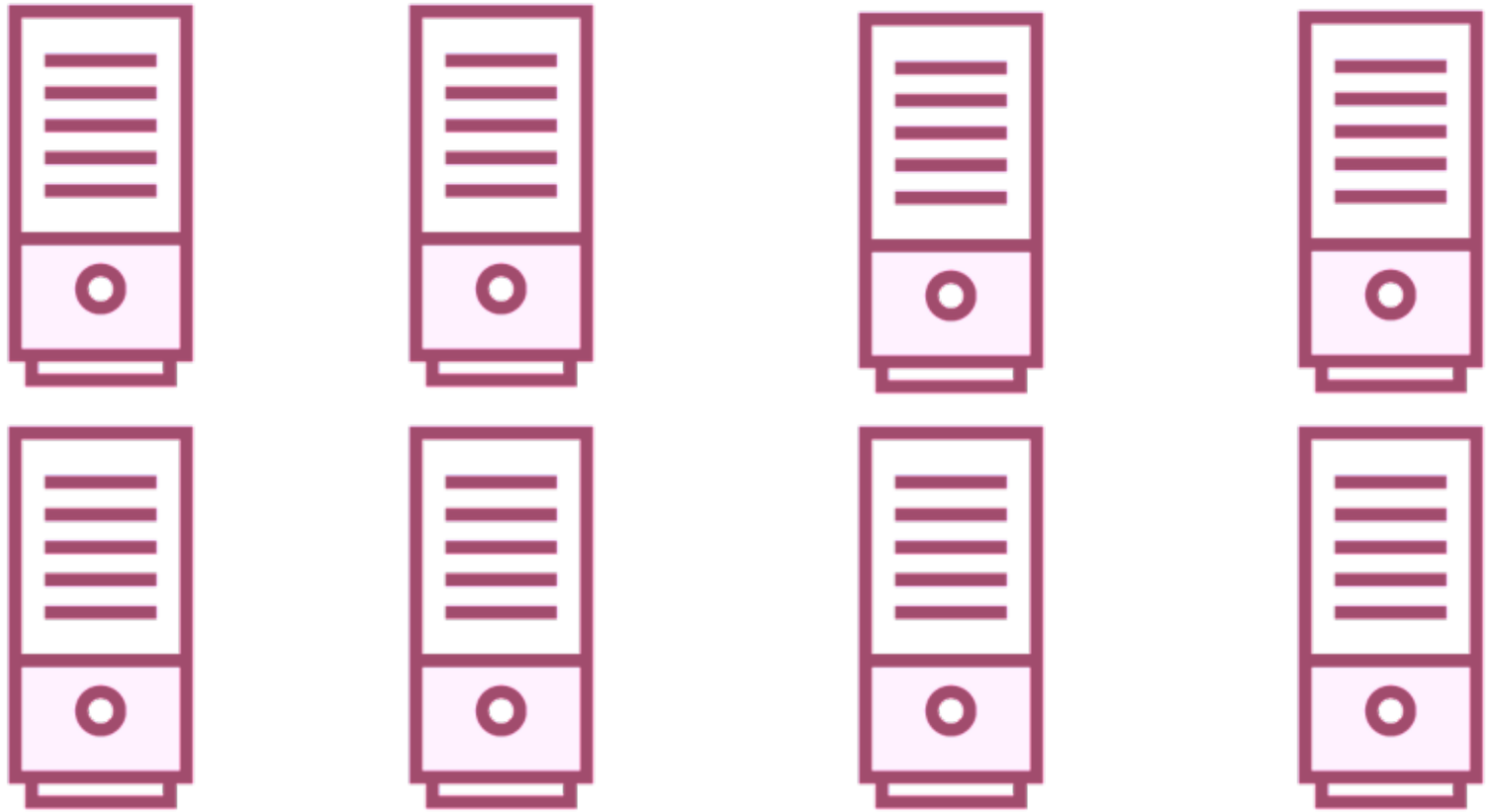
Demo

Use the nested foreach on NYC collision data to determine:

- The total number of collisions per borough
- The top 2 reasons for collisions for each borough

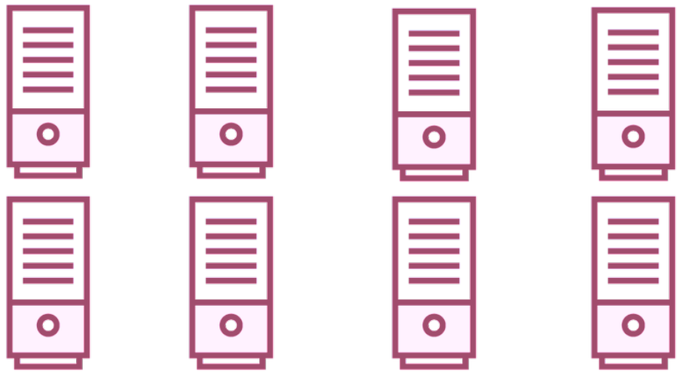
An Overview of the MapReduce Programming Model

MapReduce



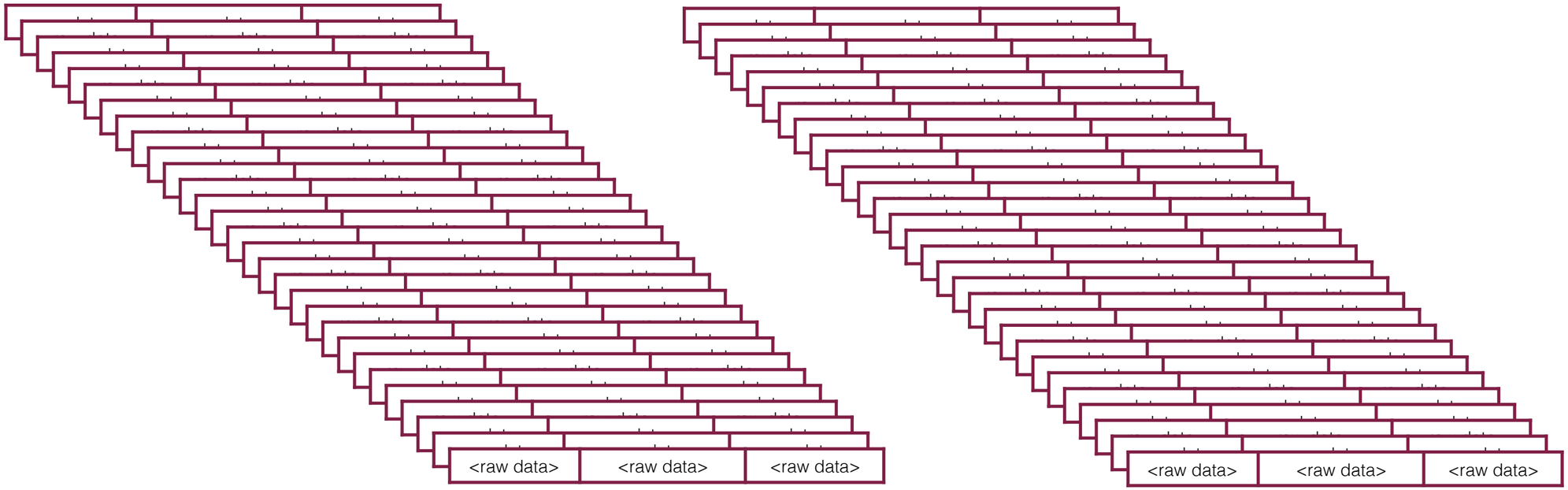
A programming paradigm which runs on a distributed system

MapReduce



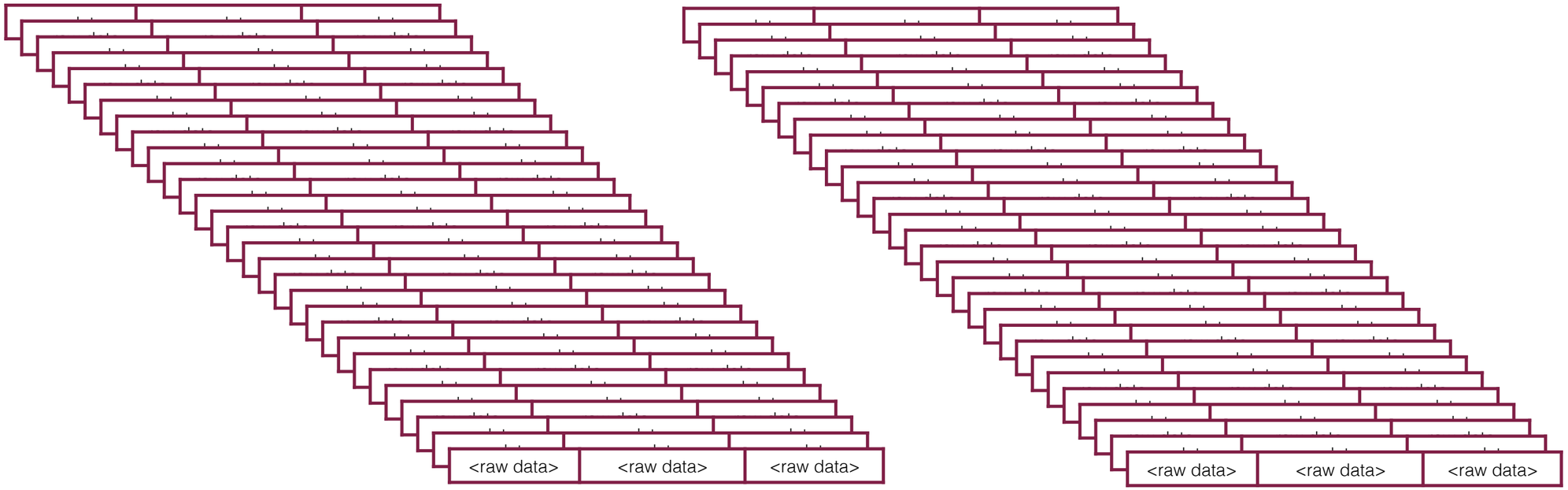
Takes advantage of the inherent parallelism in data processing

MapReduce



Modern systems generate millions of records of raw data

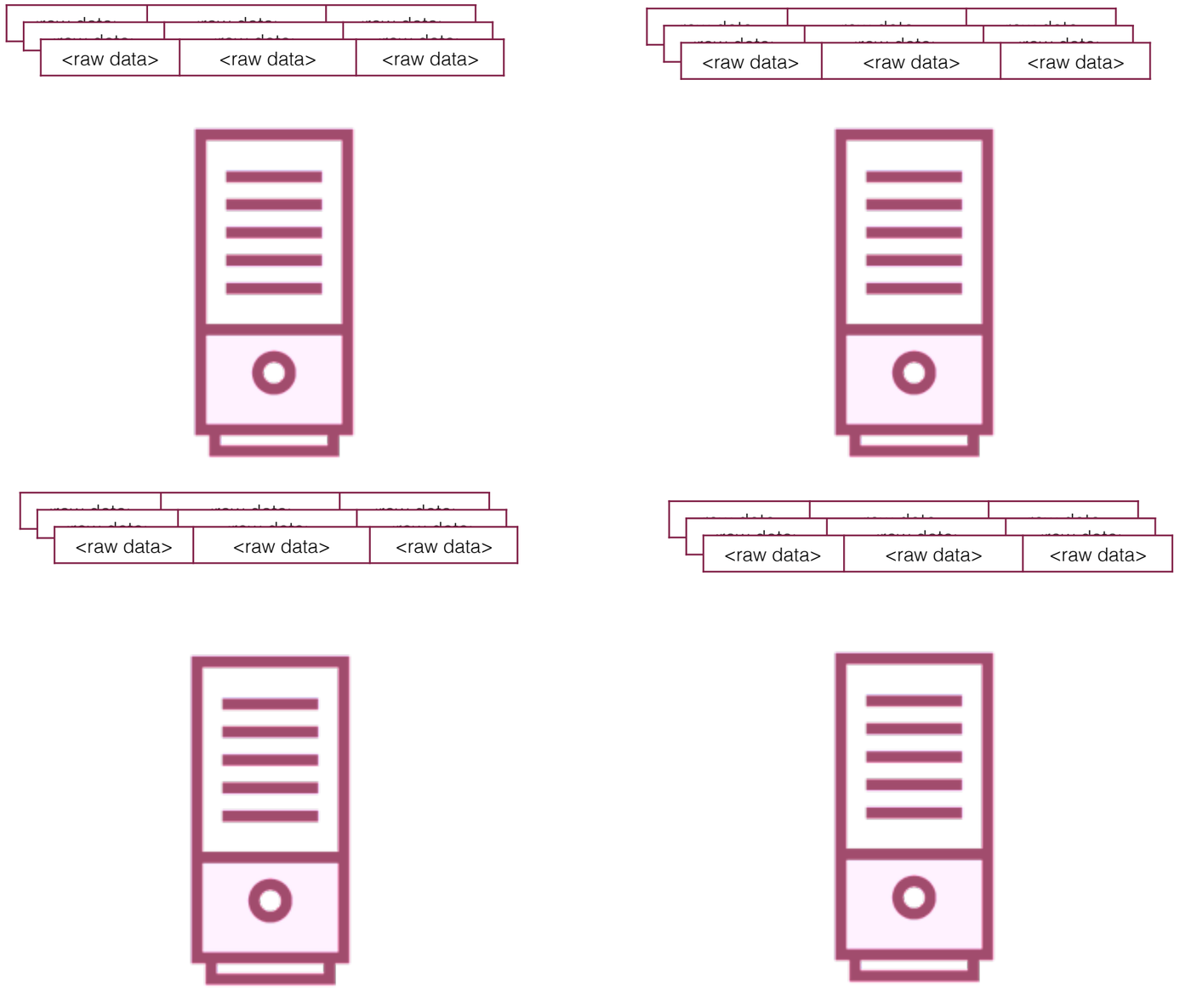
MapReduce



A task of this scale is processed in two stages

`map` `reduce`

map


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

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>

map

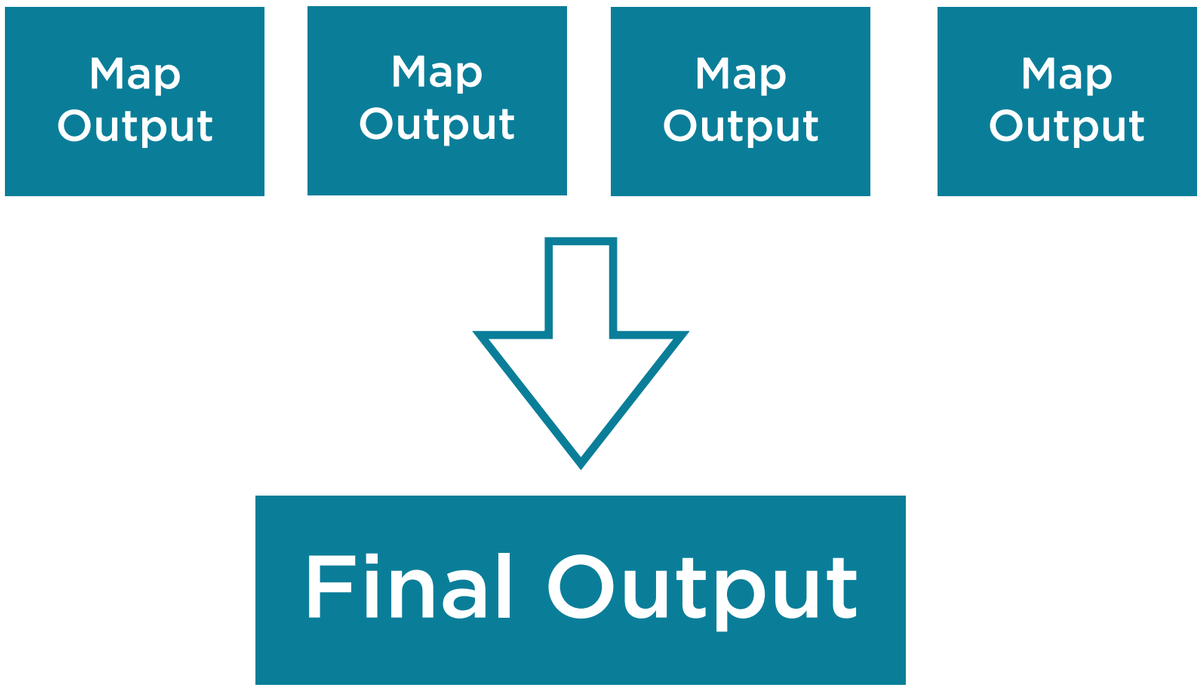
An operation performed in parallel, on small portions of the dataset

map

reduce

An operation to combine the results of the map step

reduce



map A step that can be performed in parallel

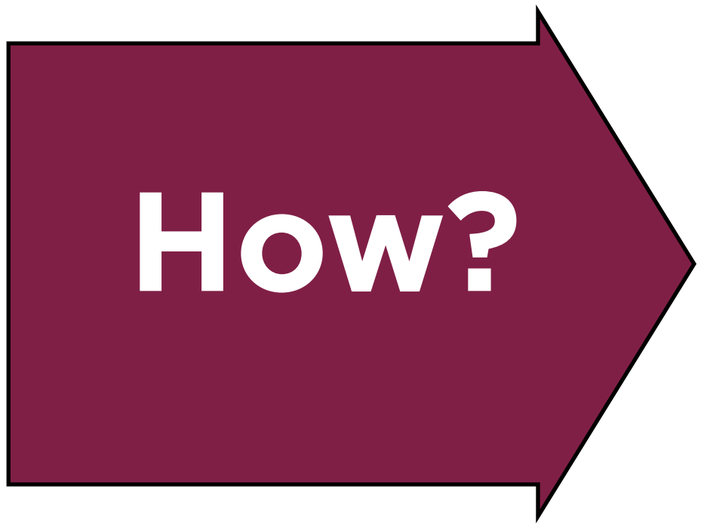
reduce A step to combine the intermediate results

The Anatomy of a MapReduce Program

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
...	..

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

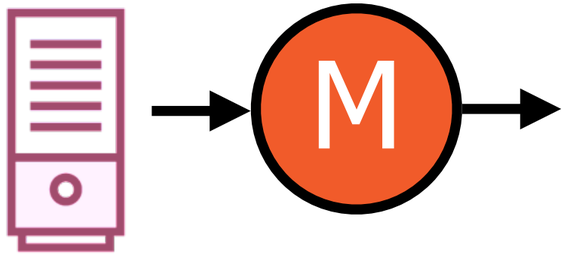
How I wonder what you are

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

MapReduce Flow

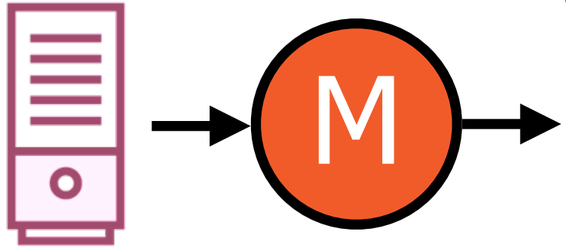
Twinkle twinkle little star

How I wonder what you are



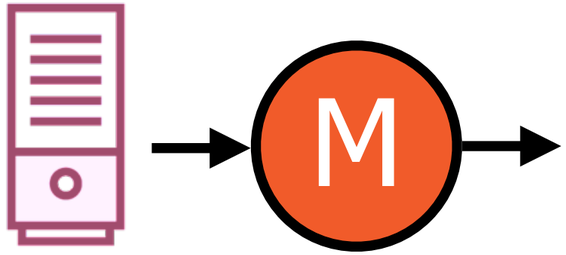
Up above the world so high

Like a diamond in the sky



Twinkle twinkle little star

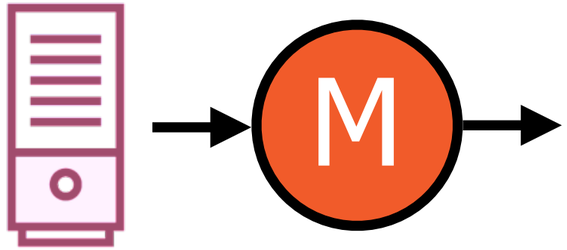
How I wonder what you are



Each mapper works in parallel

Map Flow

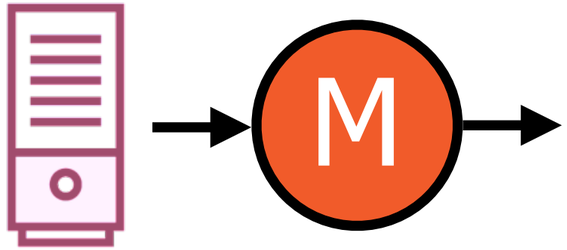
Twinkle twinkle little star
How I wonder what you are



Within each mapper, the rows are processed serially

Map Flow

Twinkle twinkle little star
How I wonder what you are

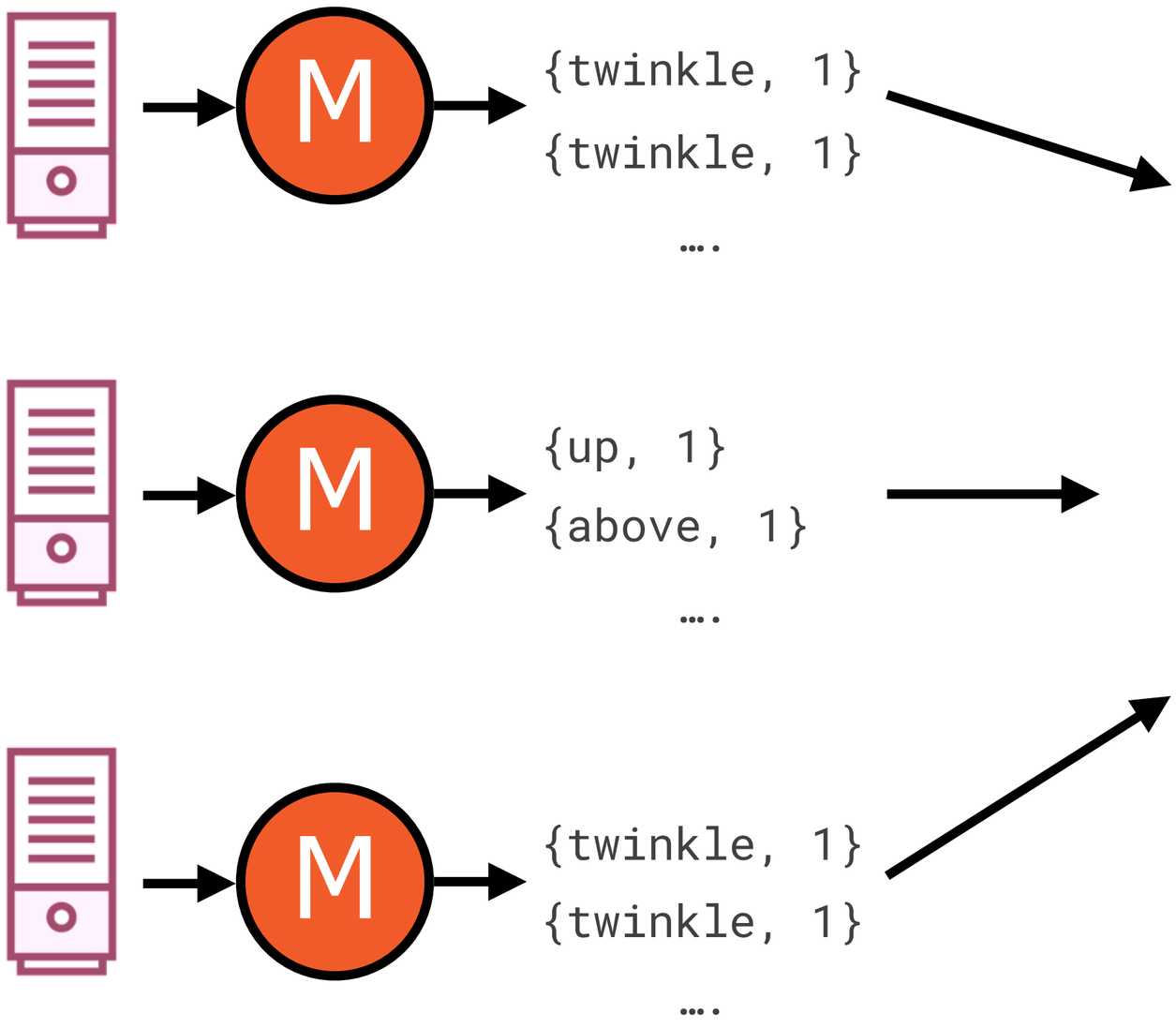


Word	# Count

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

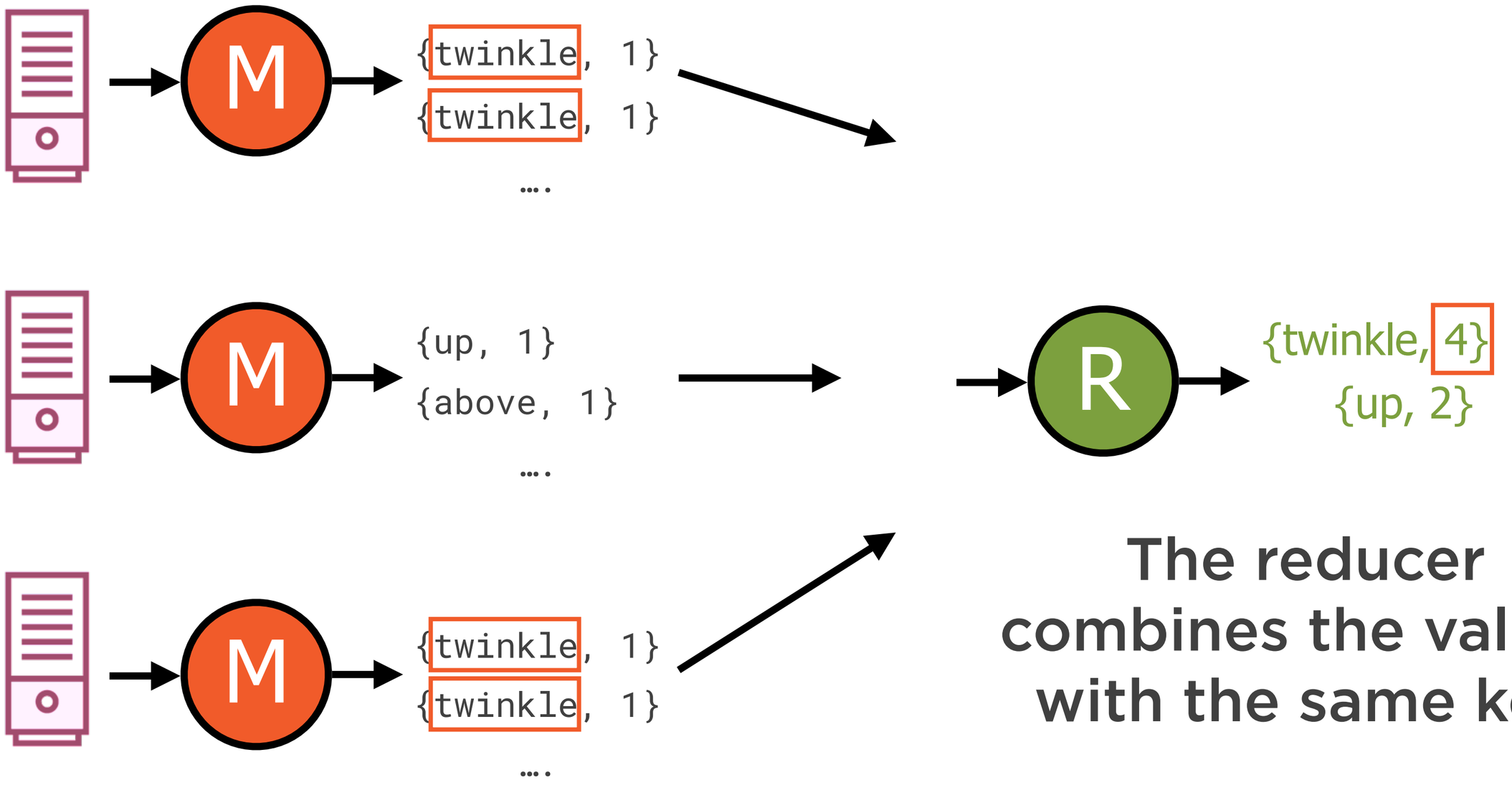
Each row emits {key, value} pairs

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

MapReduce can be implemented very simply in Pig using built-in commands

Demo

Express the word count operation using Pig Latin commands

Summary

Used the nested foreach for more powerful and efficient operations on relations

Understood the MapReduce parallel programming paradigm and learnt how the express MapReduce programs in Pig