Understanding the MapReduce Programming Model

INTRODUCING MAPREDUCE

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

Summary

Understand the need for Distributed Computing

Understand the role of MapReduce in a distributed computing setup

Spot applications of MapReduce

Know the typical flow of a MapReduce task

What Do These Products Have in Common?

Facebook Social Network

Google Search

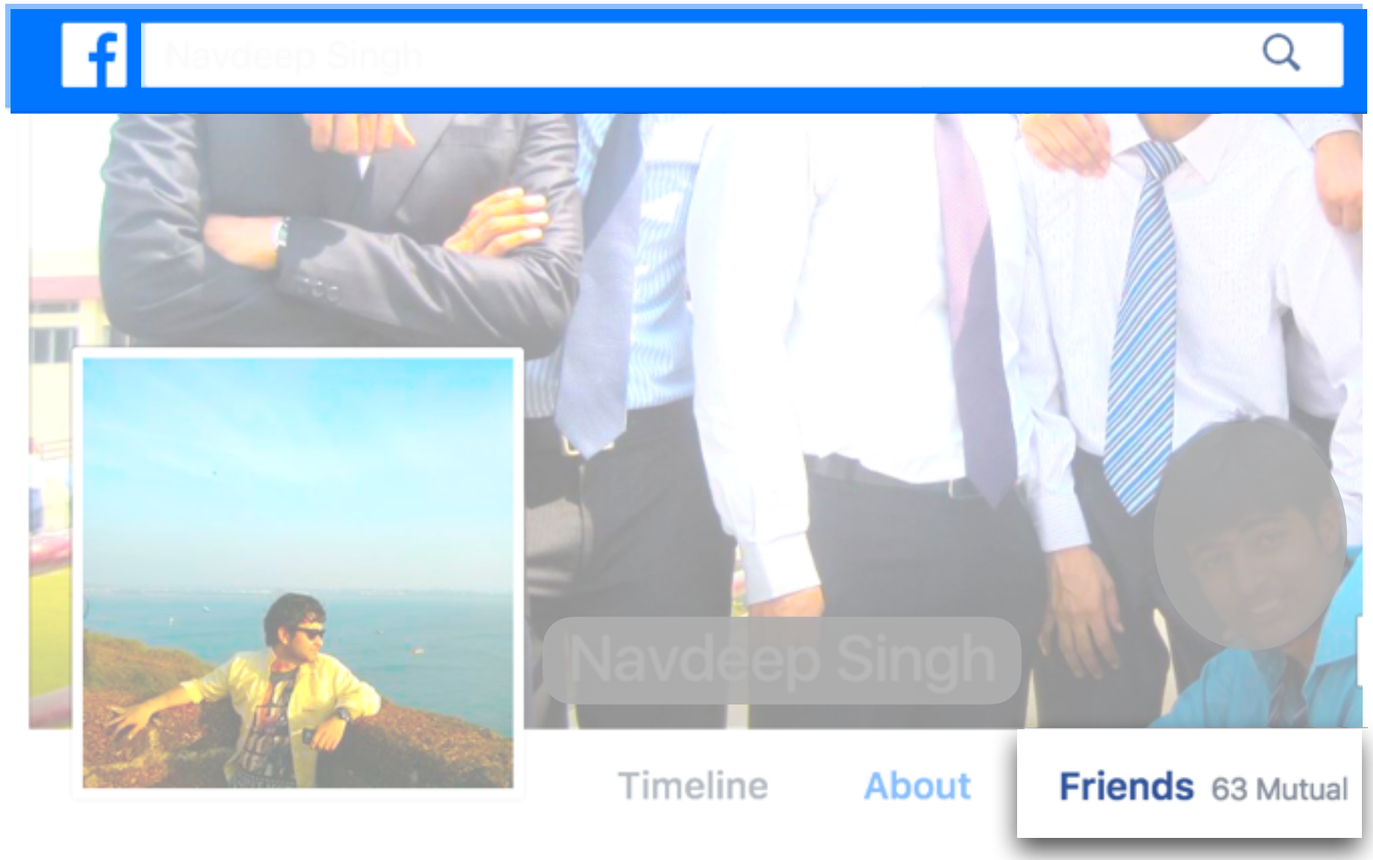
LinkedIn Member Analytics

Facebook Social Network

More than a billion users log on to Facebook every month..

Facebook Social Network

Yet, Facebook takes milliseconds to return the number of mutual friends for every pair of users



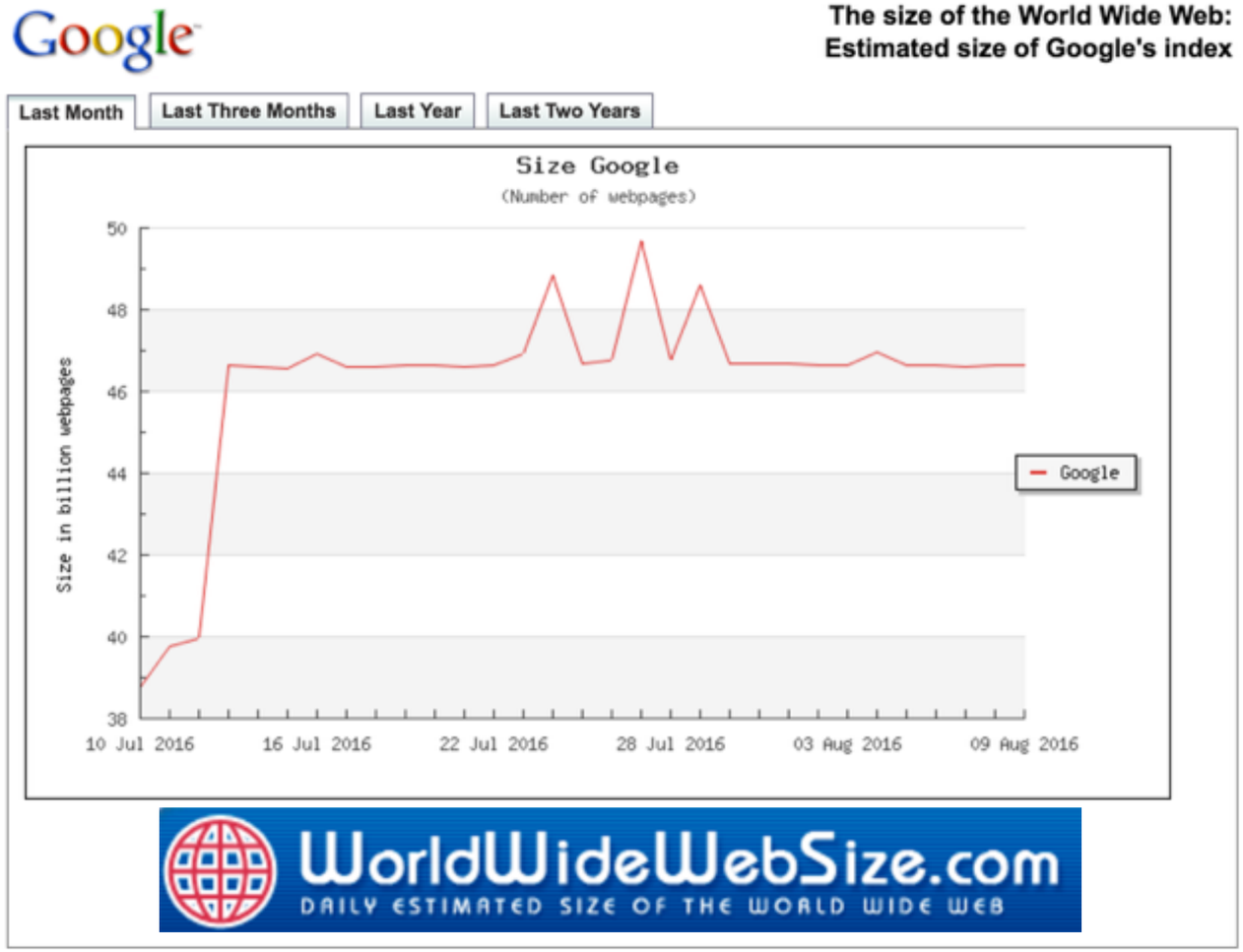
Facebook Social Network

Google Search

LinkedIn Member Analytics

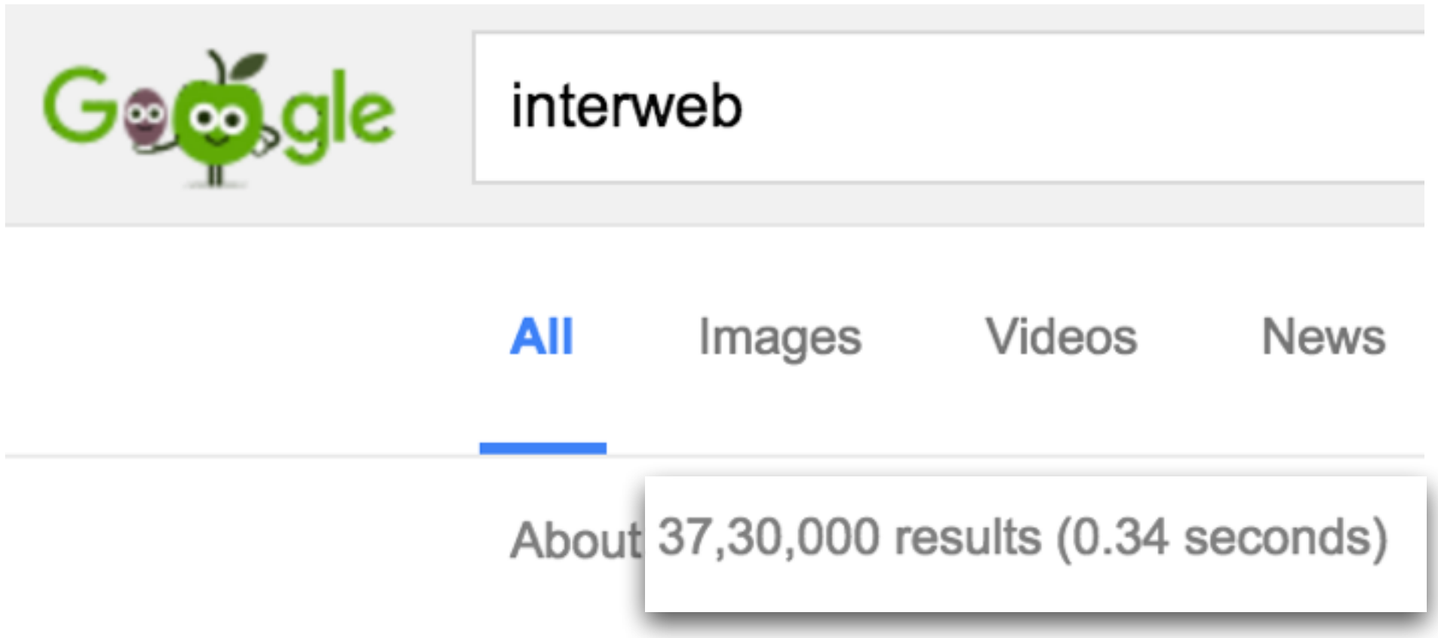
Google Search

The Internet has more than 40 billion webpages



Google Search

Yet, Google can return a search result in less than a second



Facebook Social Network

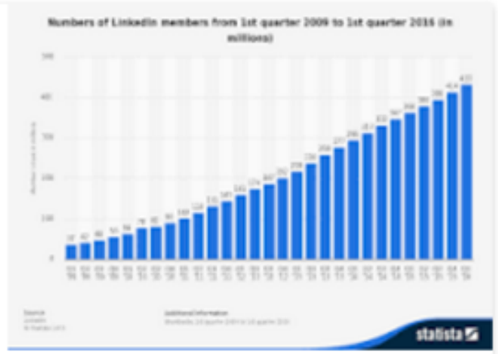
Google Search

LinkedIn Member Analytics

LinkedIn Member Analytics

LinkedIn has nearly half a billion members..

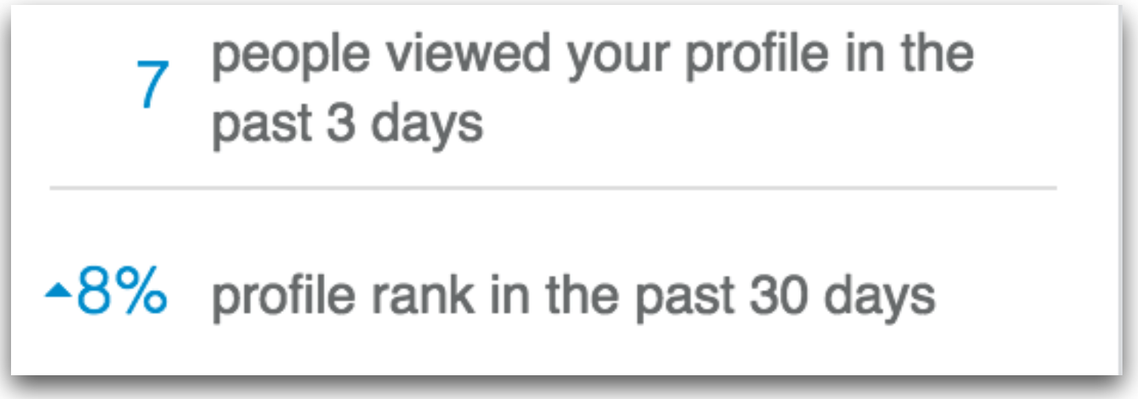
This timeline displays **member numbers** of social network **LinkedIn** from the first quarter of 2009 to the second quarter of 2016, in millions. During the most recently reported quarter, **LinkedIn** had 450 million **members**, up from 414 million **members** in the preceding quarter.



Numbers of LinkedIn members 2009-2016 - Statista
www.statista.com/statistics/274050/quarterly-**numbers**-of-**linkedin**-**members**/

LinkedIn Member Analytics

Yet, LinkedIn can display complicated metrics for each and every member on demand..



What's special about this?

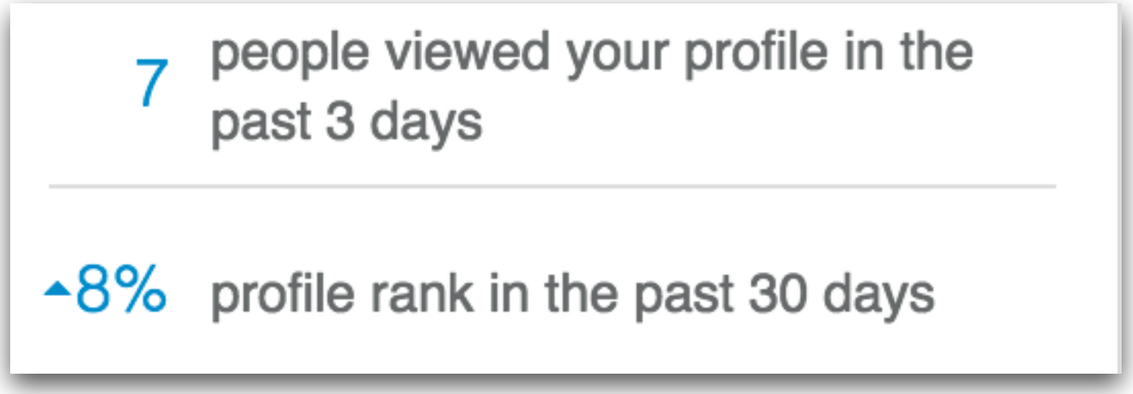
Facebook Social Network

Google Search

LinkedIn Member Analytics

Huge data set

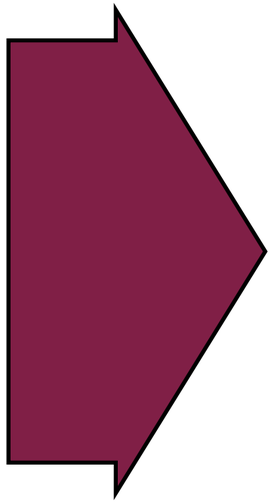
One clear insight



The power of indexes!

Billions of rows of raw data

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Swetha



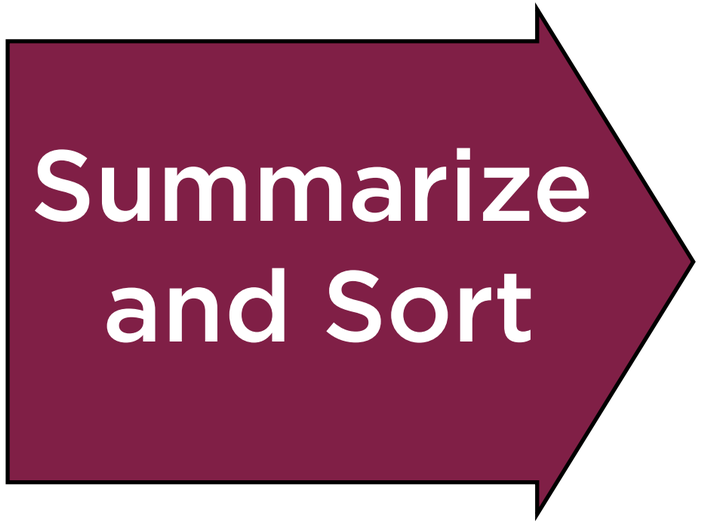
7 people viewed your profile in the past 3 days

▲8% profile rank in the past 30 days

Building an Index

Billions of rows of raw data

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Swetha



Index

Member	# Profile Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

Huge data = Massive Indexes

An Index is a Map with a really fast way to look it up

Member	# Profile Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

Massive Indexes

Building an index is

- hugely data intensive
- repetitive

Member	# Profile Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

Big data systems are all
about building these indexes

What are the requirements
of such a system?

System Requirements

System Requirements

Raw data

Store massive amounts of data

System Requirements

Raw data

Summarize and Sort

Index

Store massive amounts of data

Process it in a timely manner

System Requirements

Store

Store massive amounts of data

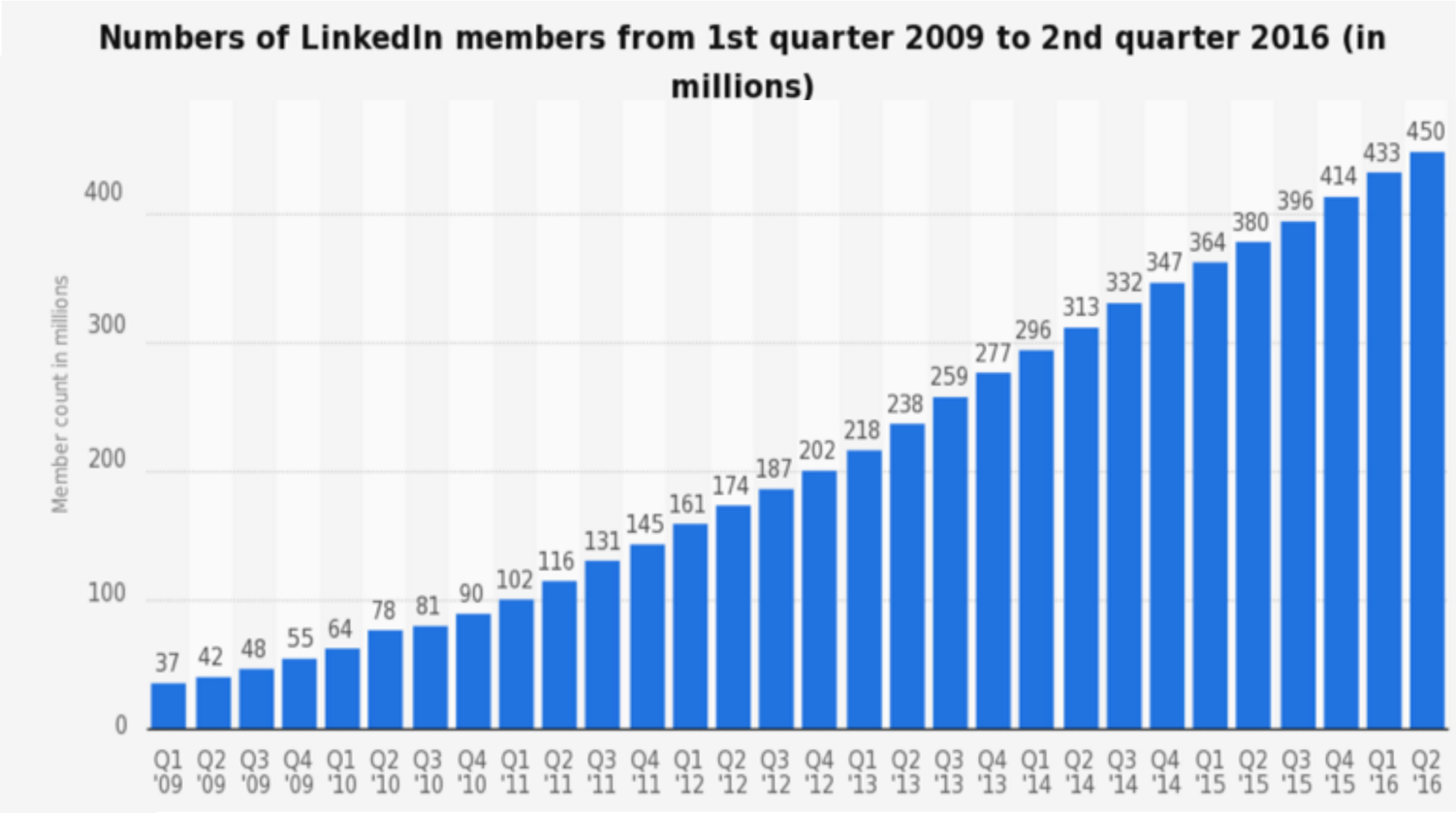
Process

Process it in a timely manner

?

Store Process ?

Growing size of data



Store Process ?

The infrastructure needs to keep up with the growing size of data

System Requirements

Store

Store massive amounts of data

Process

Process it in a timely manner

Scale

Scale easily as data grows

Store **Process** **Scale**

Distributed Computing Frameworks
like Hadoop were developed for
exactly this

Two Ways to Build a System

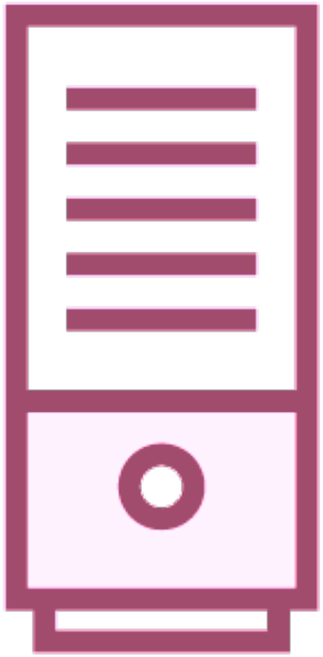
Monolithic

Distributed

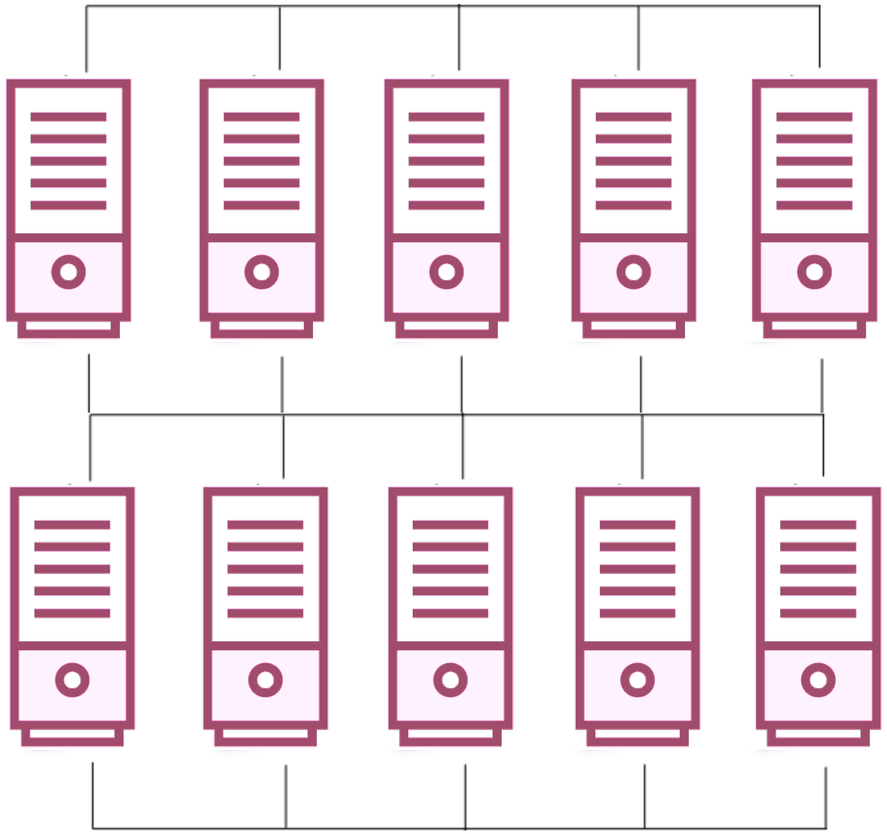
Two Ways to Build a Team

A star who dribbles and shoots

A team of good players who know how to pass



A supercomputer



A cluster of decent machines
that know how to parallelize

Two Ways to Build a System

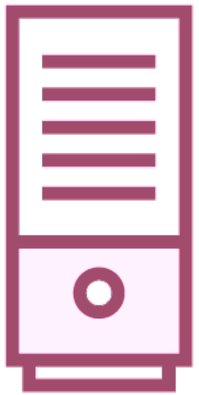
Monolithic

Distributed

Monolithic

One star player

Monolithic



A single powerful server

2x Expense

> 2x Performance

Two Ways to Build a System

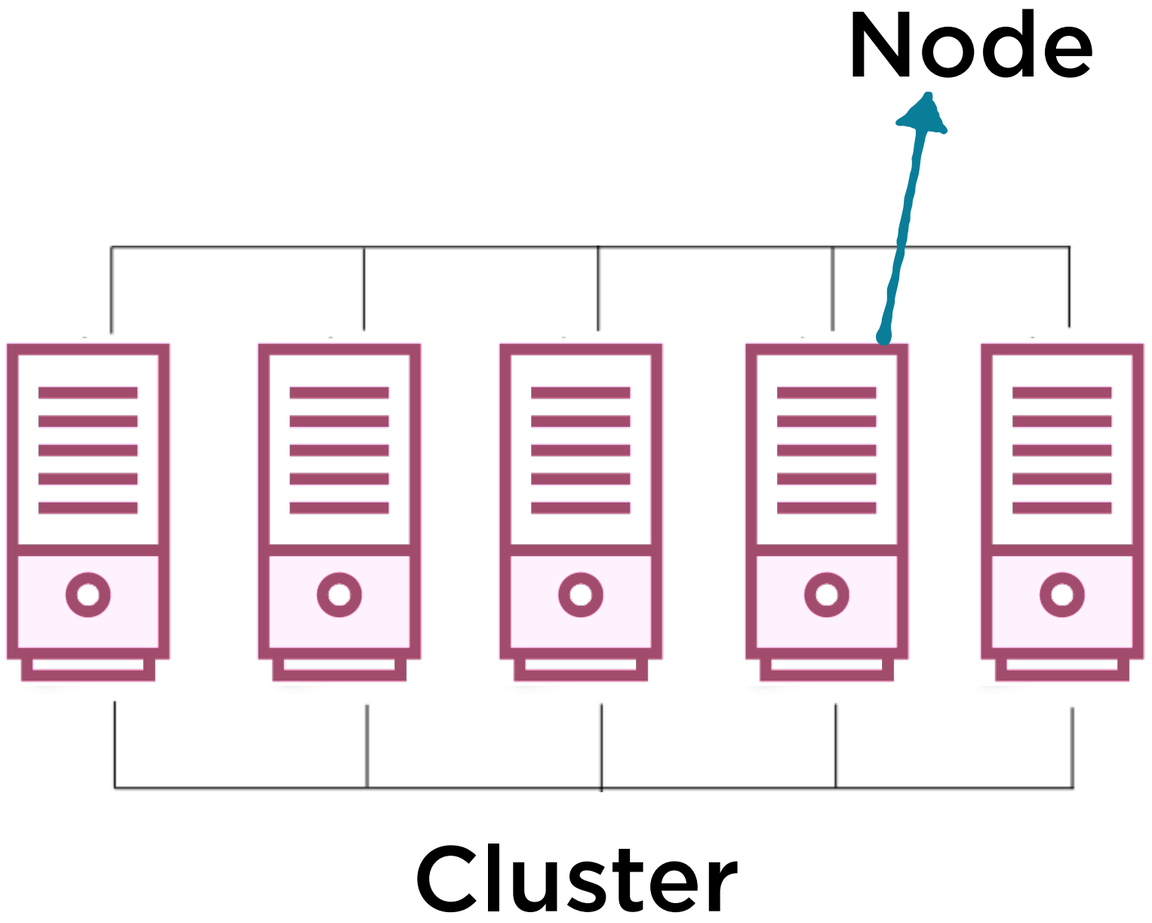
Monolithic

Distributed

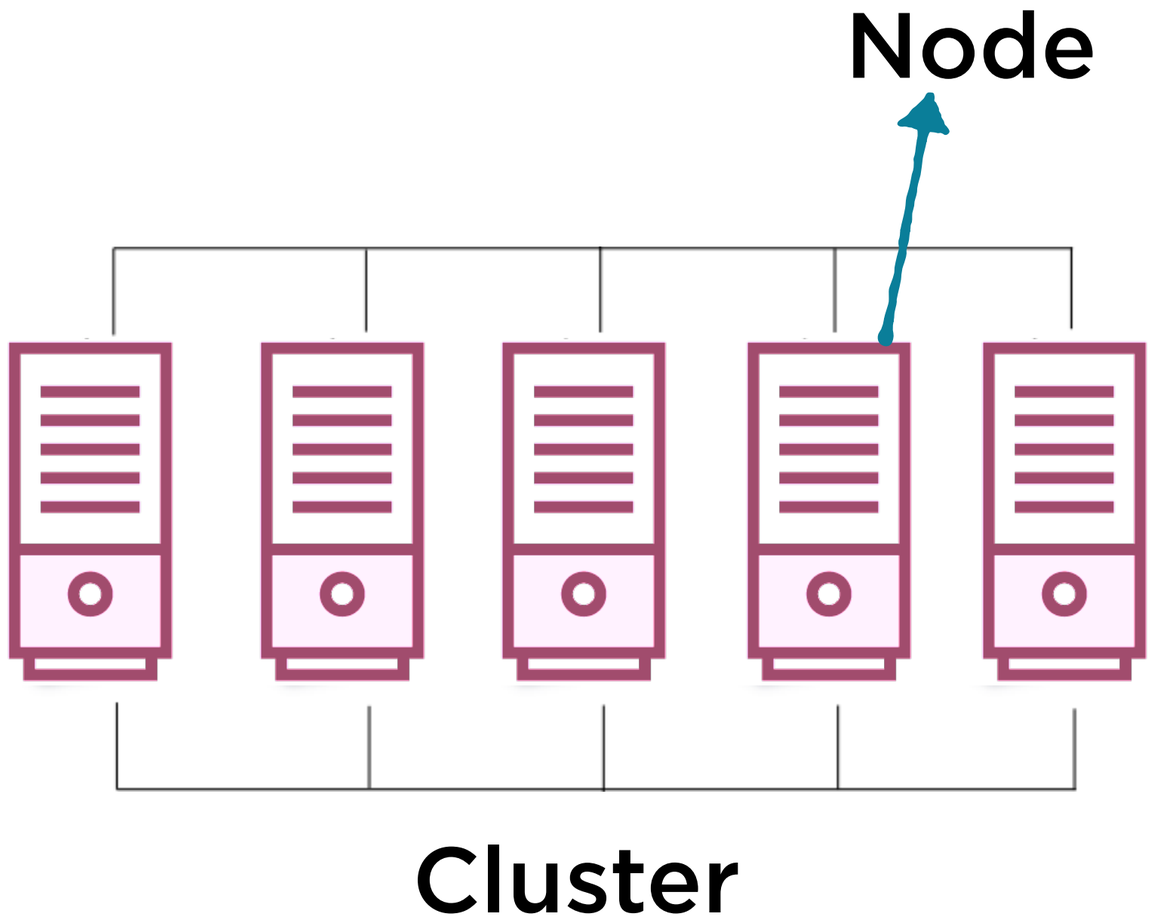
Distributed

A team of good players who know how to pass

Distributed



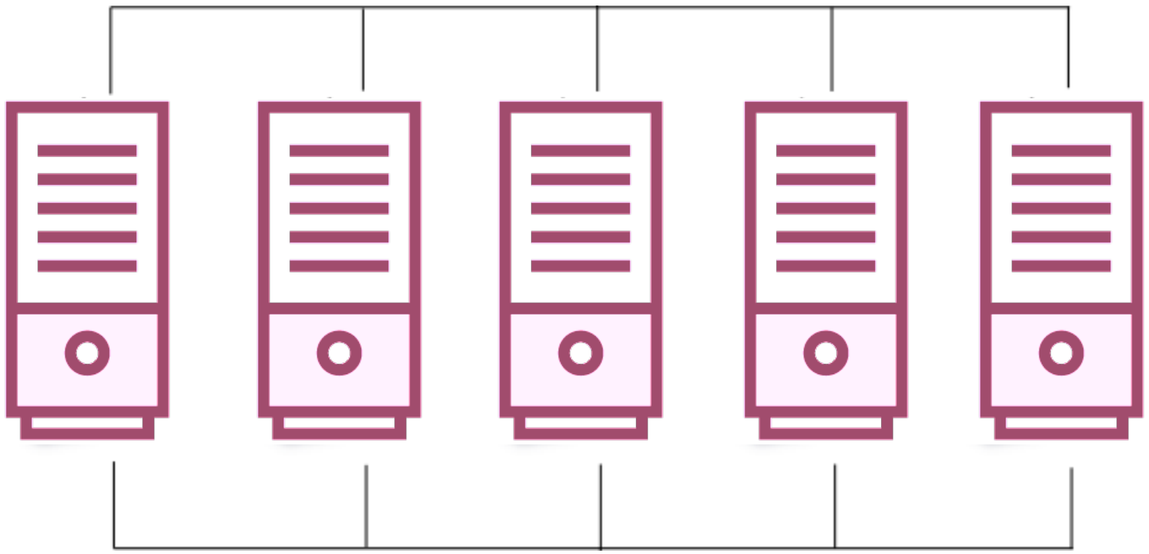
Distributed System



Many small and cheap computers come together...

...to act as a single entity

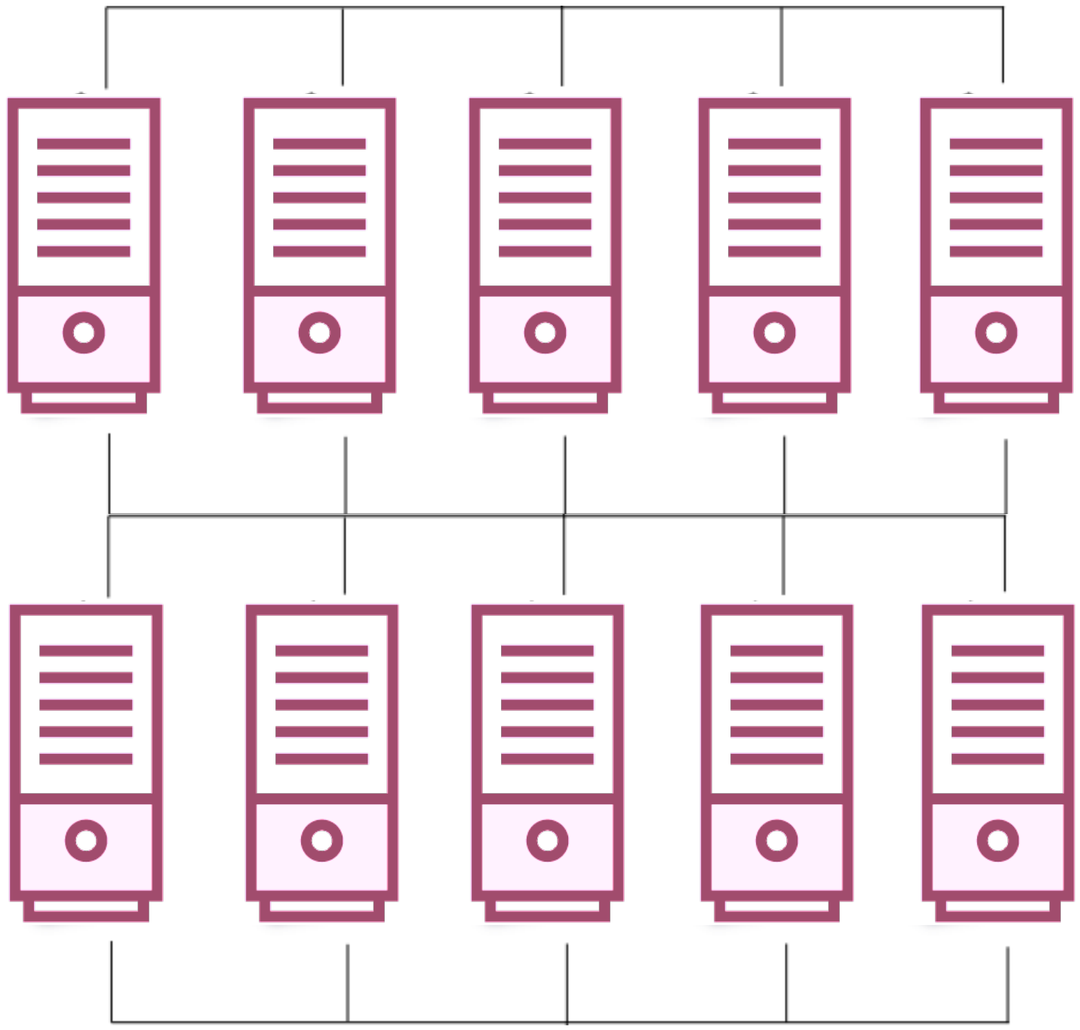
Distributed System



Cluster

Such a system can scale linearly

Distributed System



2x Nodes

2x Storage

~ 2x Speed

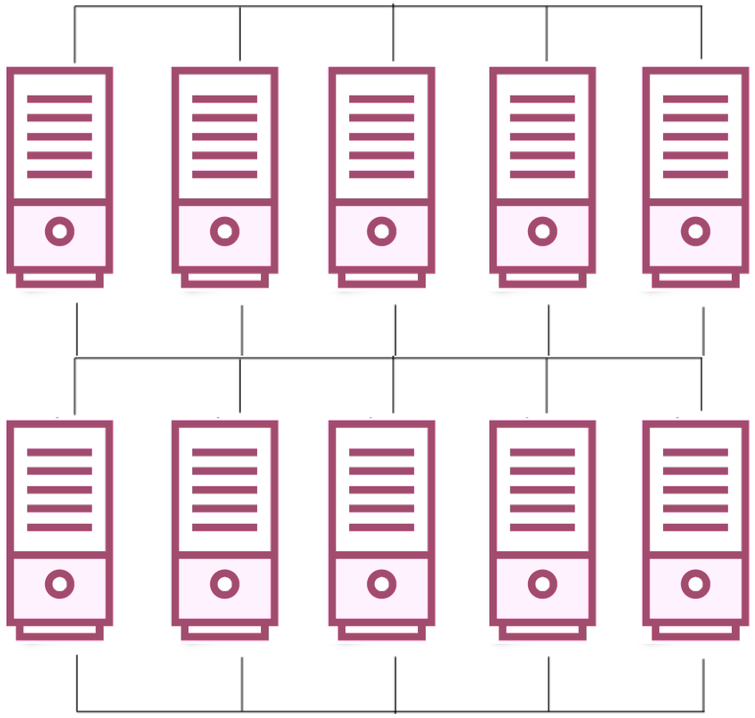
Store **Process** **Scale**

Distributed Computing makes for a lot of complexity

Complexity of Storage and Processing

- **Partitioning**
- **Fault tolerance and recovery**
- **Parallel processing**

Complexity of Storage and Processing



Complexity of coordination across nodes

Complexity of Storage and Processing

Abstraction

A distributed computing framework abstracts away these issues

Complexity of Storage and Processing

The programmer just needs to *specify what processing needs to be performed*

...in parallel

Complexity of Storage and Processing

The programmer just needs to *specify what processing needs to be performed*

...in parallel

MapReduce

MapReduce

The Power of MapReduce

Abstraction

Parallelization

Abstraction

MapReduce abstracts away the complexity into just two operations

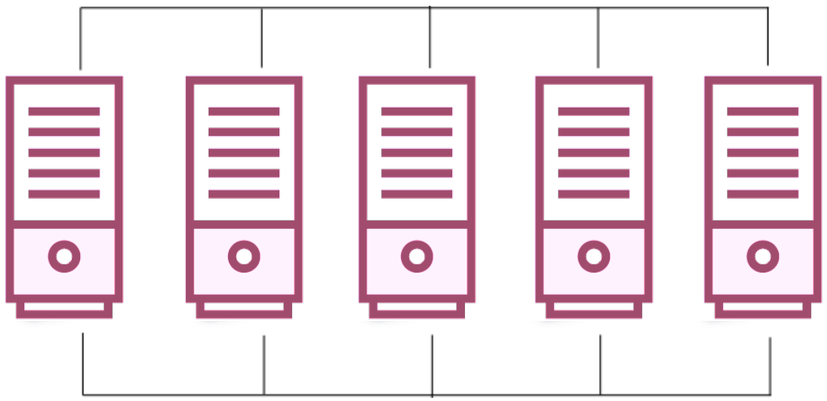
Abstraction

The programmer just defines 2 functions

```
map
reduce
```

Hadoop does the rest - behind the scenes

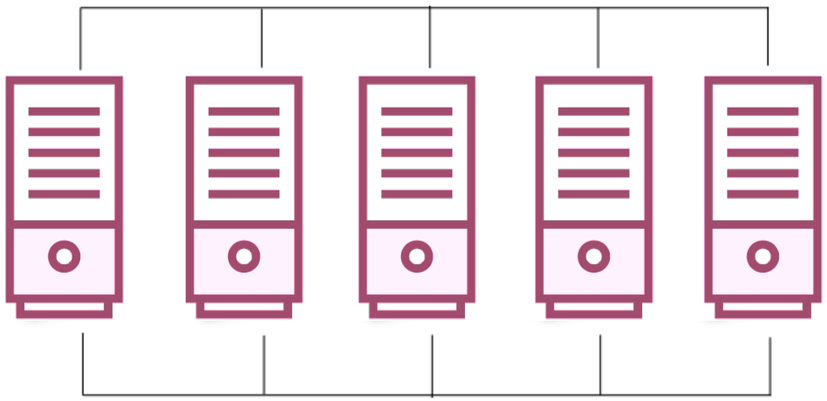
Parallelization



map

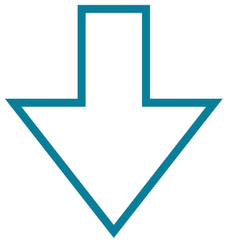
An operation performed in parallel, on small portions of the dataset

Parallelization



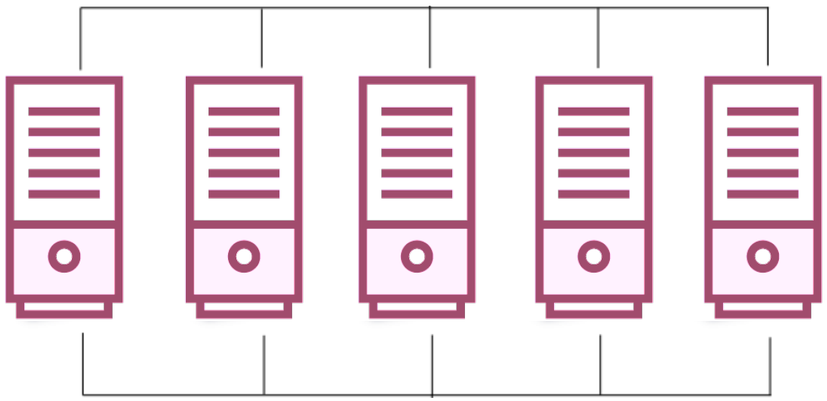
map

One Record



Key-Value Output

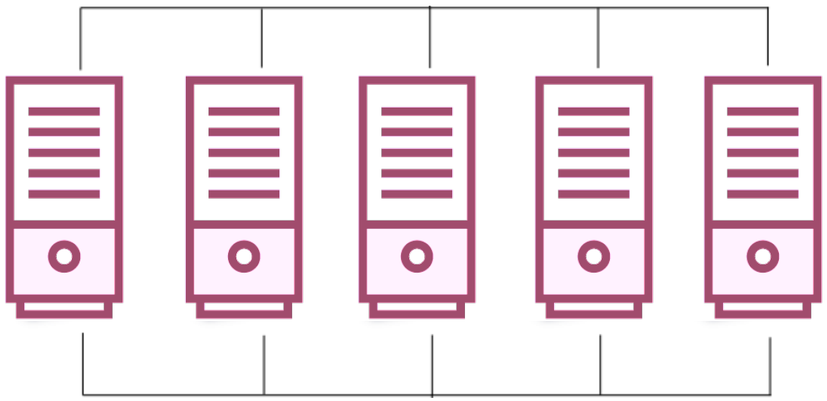
Parallelization



reduce

An operation to combine the results of the map step

Parallelization



reduce

Map Output

Map Output

Map Output

Map Output

Final Output

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

Think Parallel

map A step that can be performed in parallel

reduce A step to combine the intermediate results

Think Parallel

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

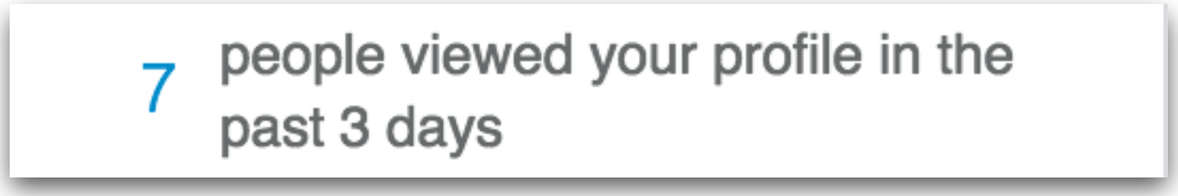
Summary

Distributed Computing scales to deal with data as it grows

MapReduce is an abstraction to express processing tasks in distributed computing

MapReduce helps parallelize expensive tasks like building indexes

LinkedIn displays some interesting metrics when a member visits their profile page



7 people viewed your profile in the past 3 days

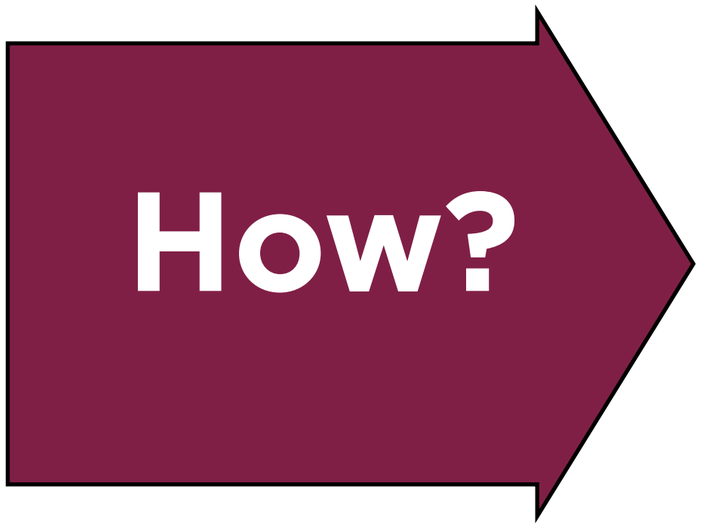
This is powered by an index

Member	# Profile Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

Building a User-ViewCount Map

Data is originally captured in this form

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



User-ViewCount Map

Member	# Profile Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

MapReduce Flow

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep

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

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani

View ID	From Member	To Member
3	Shreya	Pradeep
4	Jitu	Vitthal

View ID	From Member	To Member
5	Shreya	Janani
6	Jitu	Pradeep

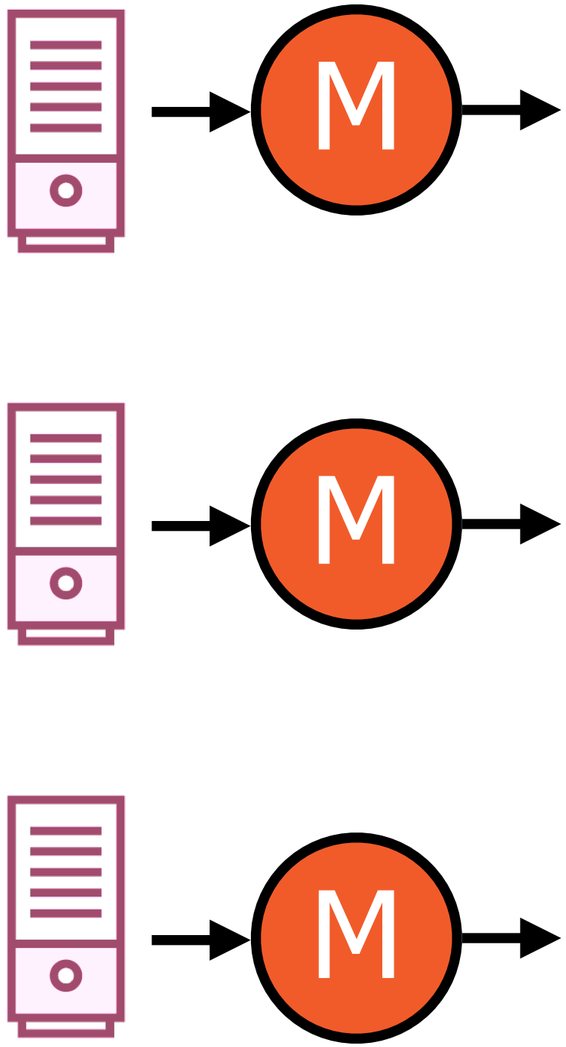
Each partition is given to a different process i.e. to mappers

MapReduce Flow

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani

View ID	From Member	To Member
3	Shreya	Pradeep
4	Jitu	Vitthal

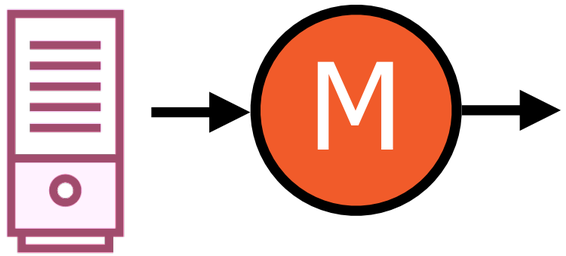
View ID	From Member	To Member
5	Shreya	Janani
6	Jitu	Pradeep



Each mapper works in parallel

Map Flow

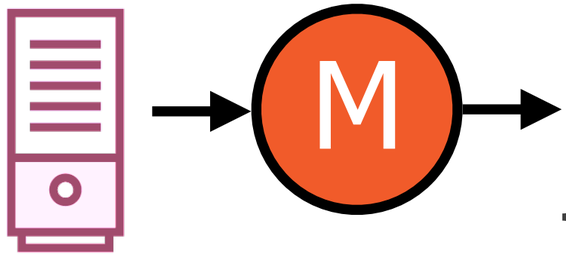
View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani



Within each mapper, the rows are processed serially

Map Flow

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani



Member	# Profile Views

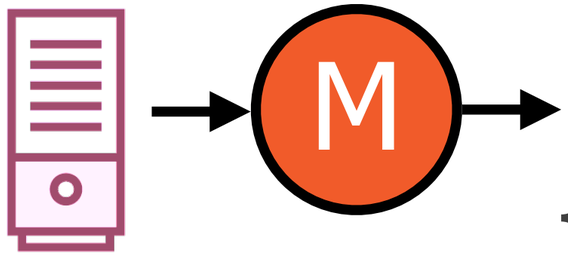
{Jitu, 1}

{Janani, 1}

Each row emits a {key, value} pair

Map Flow

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani



Member	# Profile Views

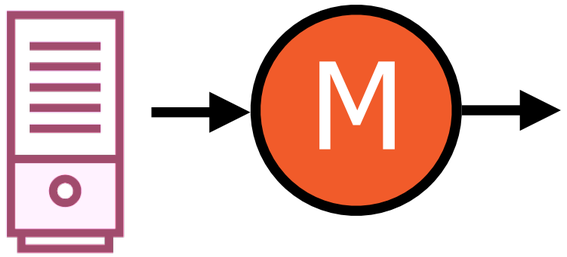
{Jitu, 1}

{Janani, 1}

This operation can occur independently on each mapper

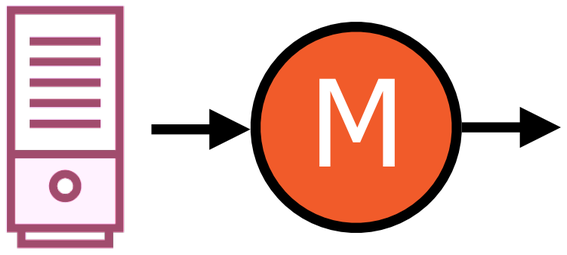
Map Flow

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani



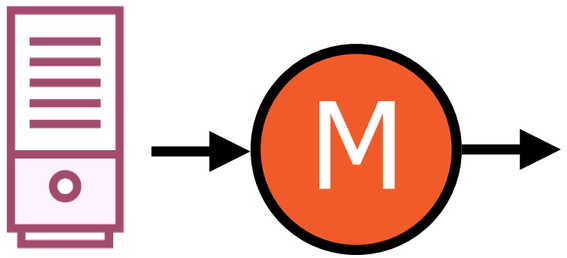

```
{Jitu, 1}
{Janani, 1}
```

View ID	From Member	To Member
3	Shreya	Pradeep
4	Jitu	Vitthal



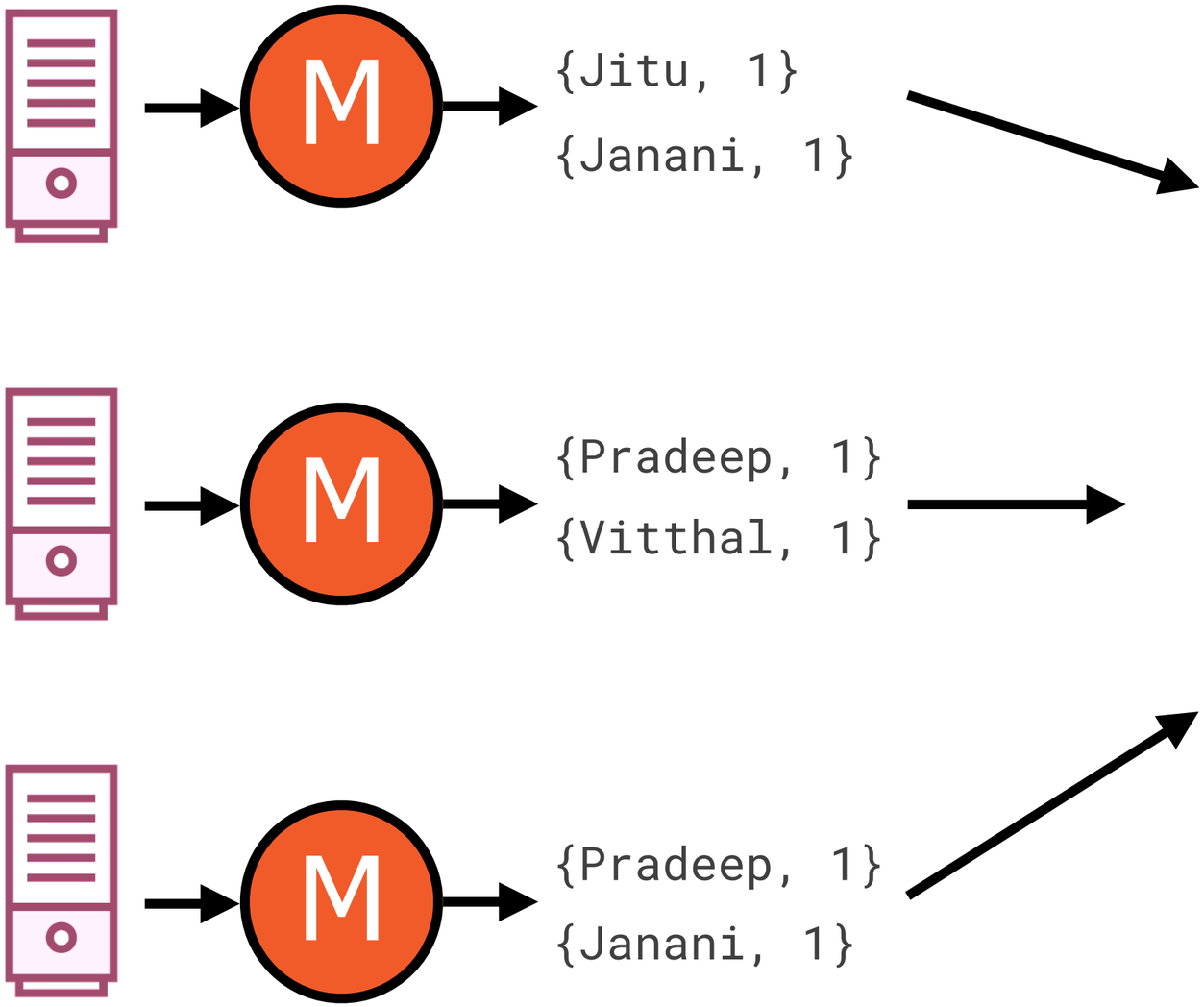

```
{Pradeep, 1}
{Vitthal, 1}
```

View ID	From Member	To Member
5	Shreya	Janani
6	Jitu	Pradeep



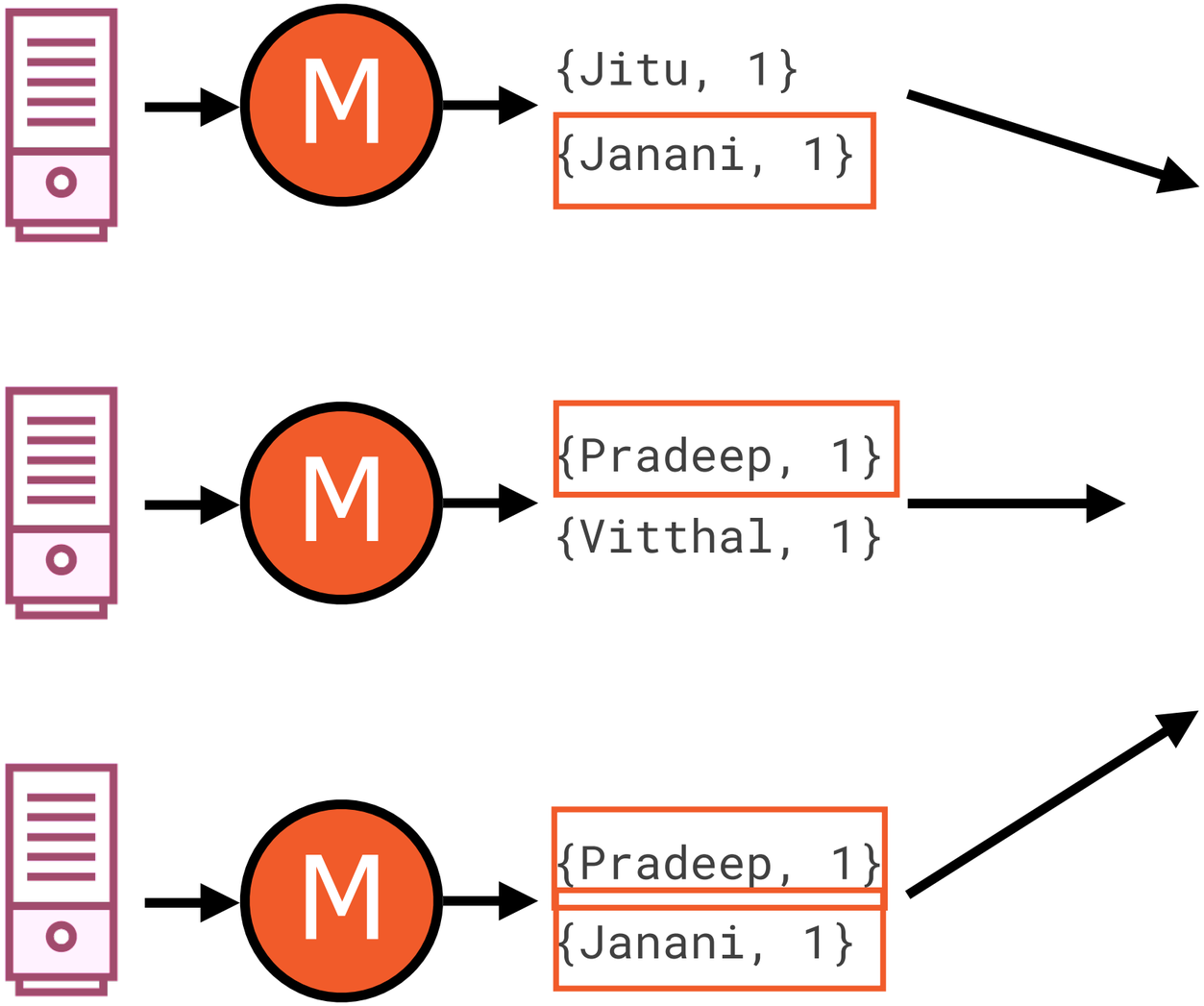

```
{Pradeep, 1}
{Janani, 1}
```

Reduce Flow



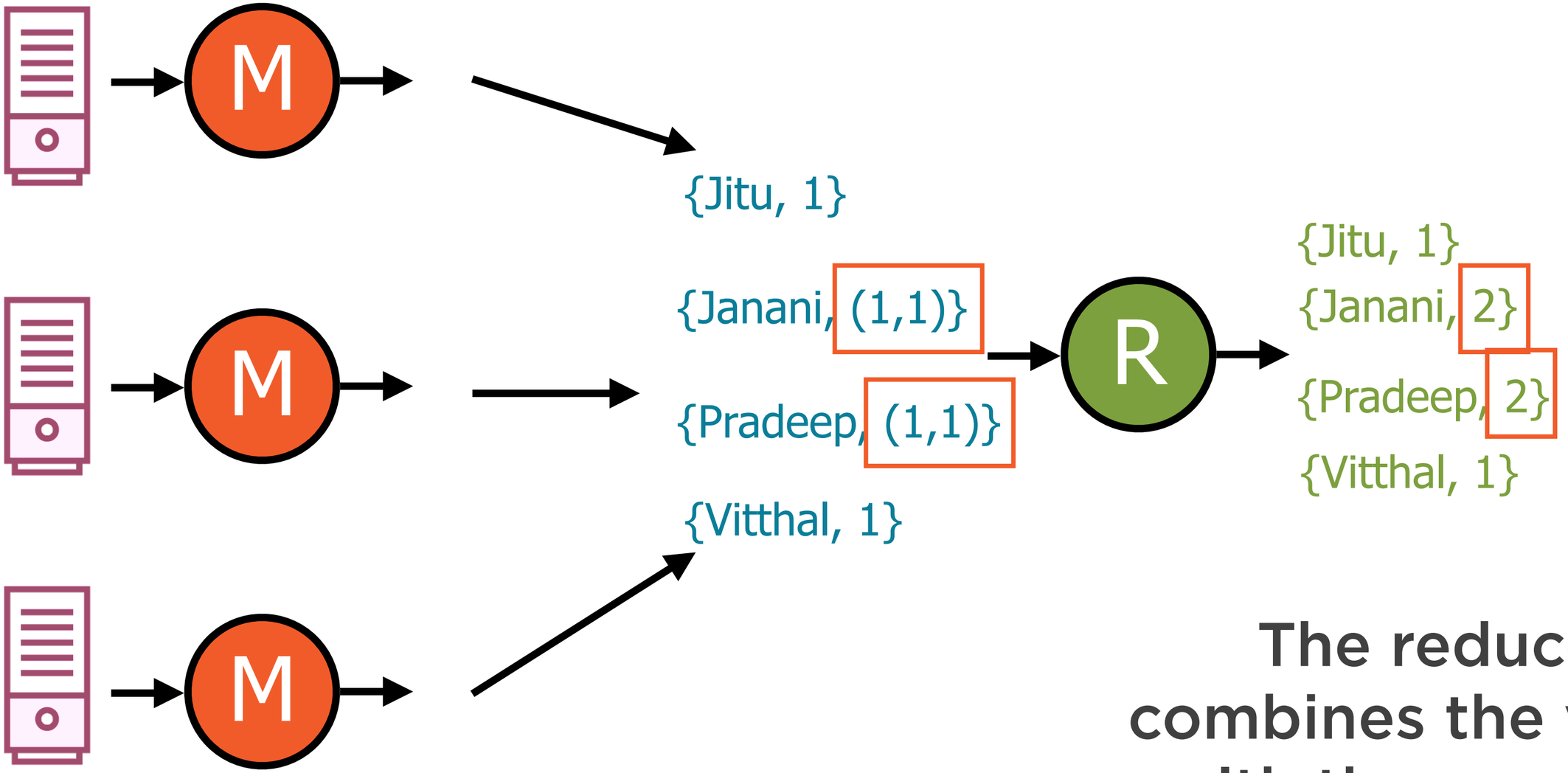
The results are passed on to another process ie. a reducer

Reduce Flow



Values with the same key are collected together

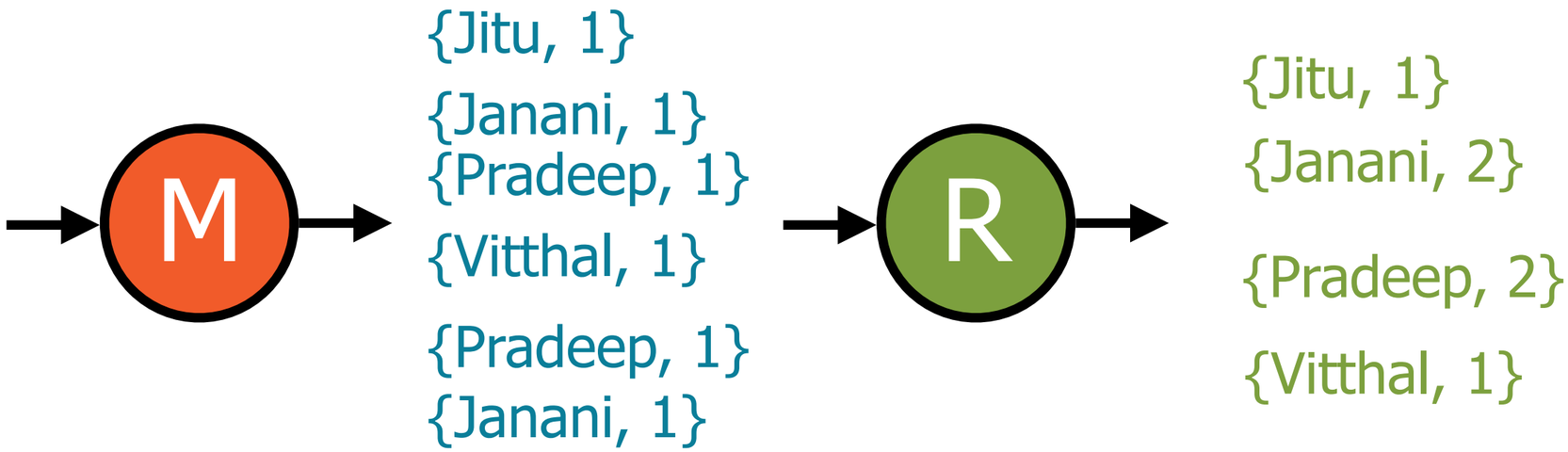
Reduce Flow



The reducer combines the values with the same key

MapReduce Flow

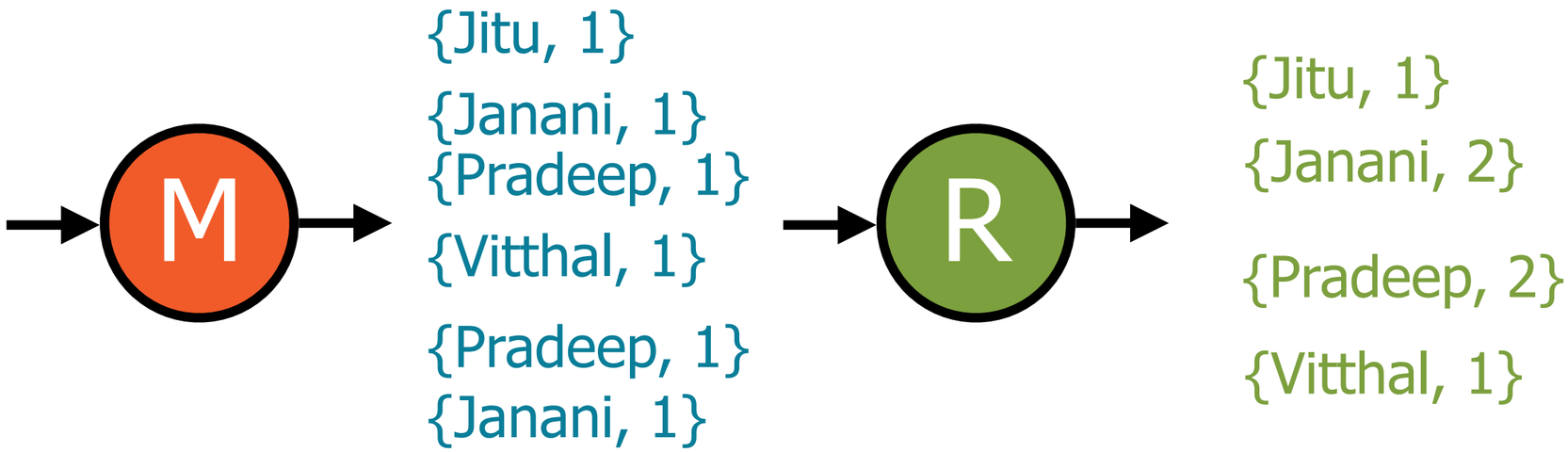
View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



The basic form of EVERY MapReduce task

MapReduce Flow

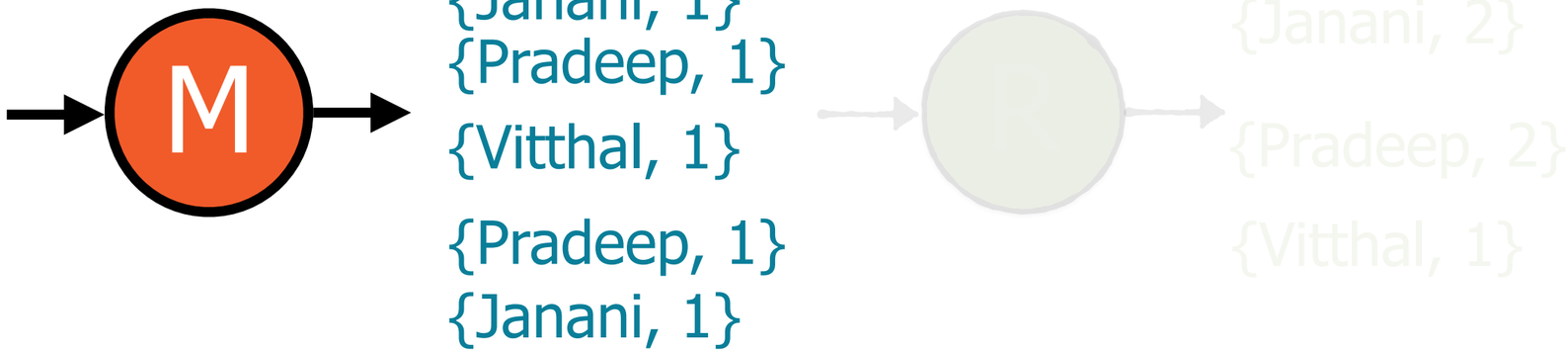
View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



The basic form of EVERY MapReduce task

MapReduce Flow

View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



The Map step produces a set of {key, value} pairs

MapReduce Flow

View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep

{Jitu, 1}
{Janani, 1}
{Pradeep, 1}
{Vitthal, 1}
{Pradeep, 1}
{Janani, 1}

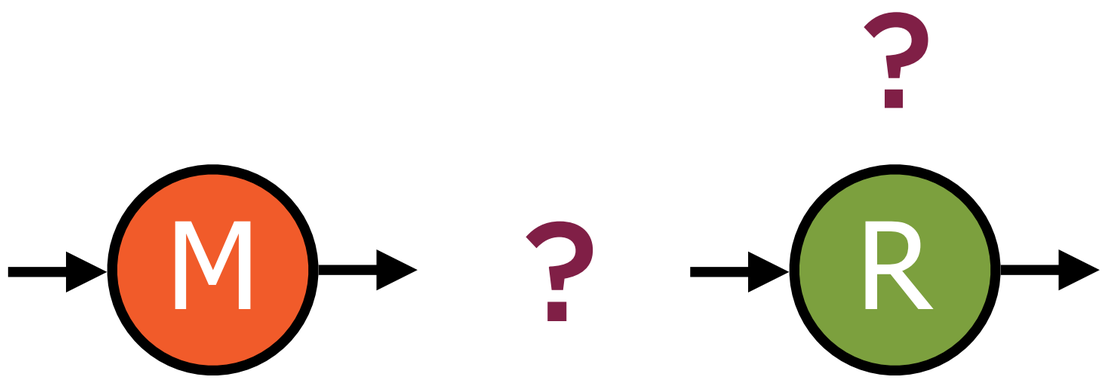
{Jitu, 1}
{Janani, 2}
{Pradeep, 2}
{Vitthal, 1}

The Reduce step combines {key, value} pairs with the same key

Key Insight Behind MapReduce

Many data processing tasks can be expressed in this form

Answer 2 Questions



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

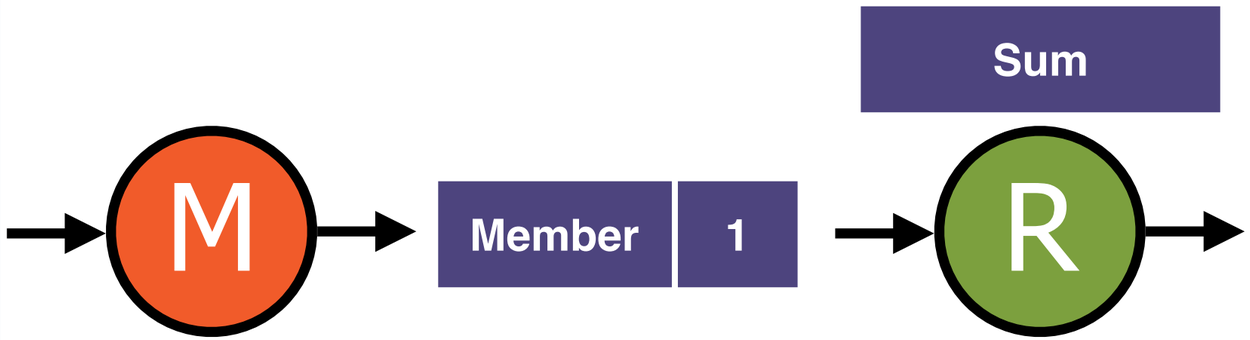
Building a User-ViewCount Map

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep

How many views has each profile had?

Building a User-ViewCount Map

View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



Member	# Views
Janani	50
Swetha	15
Vitthal	22
Shreya	23
Jitu	32
Pradeep	10

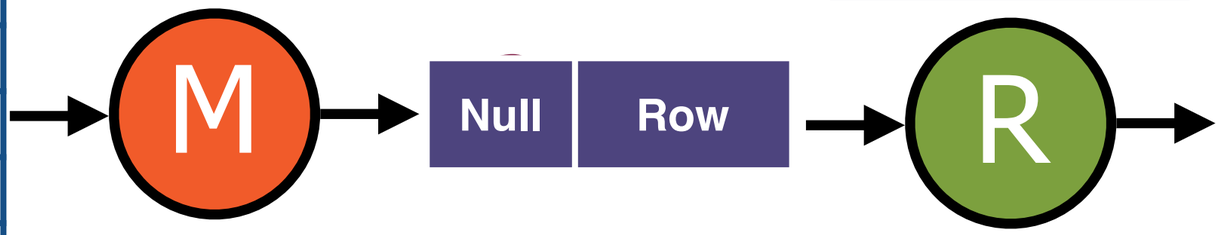
Filter Views for a User

View ID	From Member	To Member
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep

What information regarding page views should be shown to one user?

Filter Views for a User

View	From	To
1	Janani	Jitu
2	Swetha	Janani
3	Shreya	Pradeep
4	Jitu	Vitthal
5	Shreya	Janani
6	Jitu	Pradeep



View ID	From	To
3	Shreya	Pradeep
6	Jitu	Pradeep

If the user matches the given user

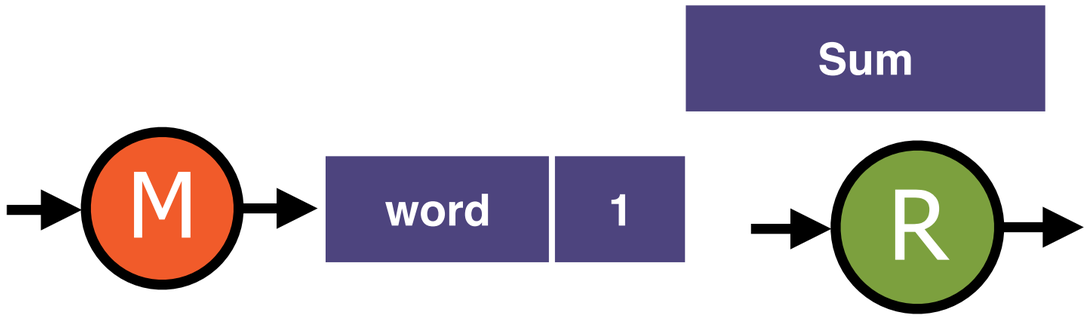
Word Counts in a Document

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

What is the frequency of every word in this document?

Word Counts in a Document

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



For each word in each line

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

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

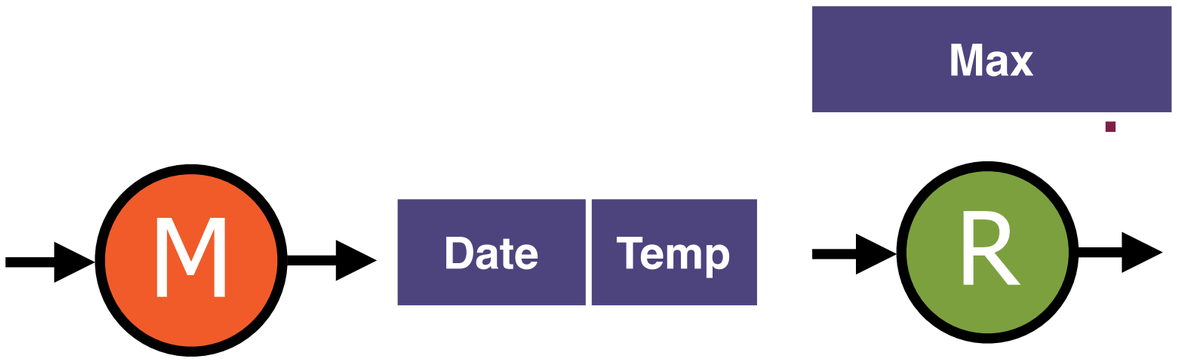
Max Temperature in a Day

Date	Time	Temp
12/12/2015	00:00	50
12/12/2015	01:00	52
12/12/2015	02:00	49
12/12/2015	03:00	48
12/12/2015	04:00	54
12/12/2015	05:00	55

What is the maximum recorded temperature on a given day?

Max Temperature in a Day

Date	Time	Temp
12/12/2015	00:00	50
12/12/2015	01:00	52
12/12/2015	02:00	49
12/12/2015	03:00	48
12/12/2015	04:00	54
12/12/2015	05:00	55



Date	Max Temp

From each row

Answer these to
parallelize any task :)

Summary

Understand the need for Distributed Computing

Understand the role of MapReduce in a distributed computing setup

Spot applications of MapReduce

Know the typical flow of a MapReduce task