CC5212-1

Procesamiento Masivo de Datos

Otoño 2023

Lecture 1

Introduction

Aidan Hogan

aidhog@gmail.com

The value of data

Soho, London, 1854

A COURT FOR KING CHOLERA.

Cholera: What we know now …

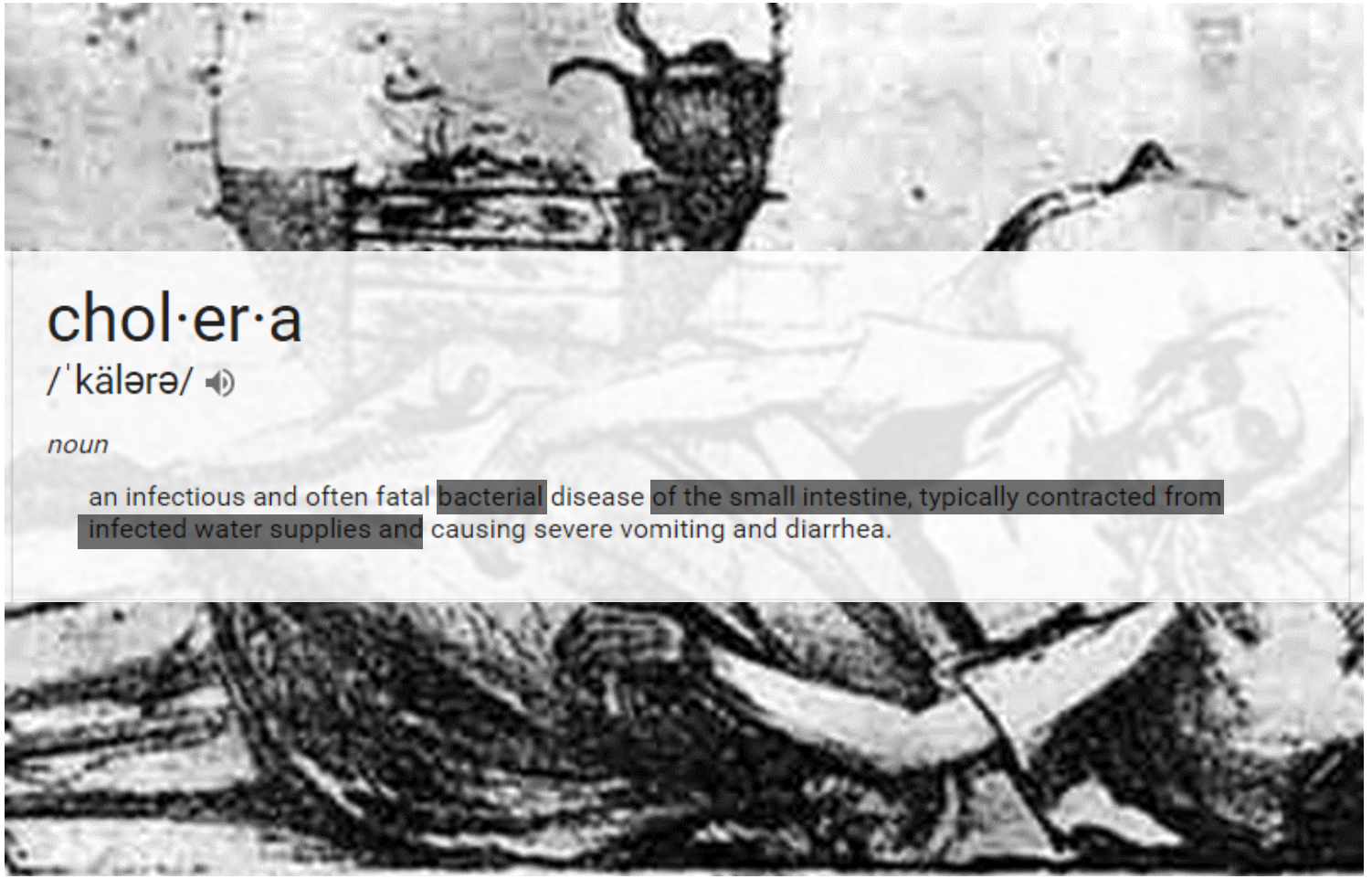
chol·er·a

/ˈkälərə/

noun

an infectious and often fatal bacterial disease of the small intestine, typically contracted from infected water supplies and causing severe vomiting and diarrhea.

Cholera: What we knew in 1854



chol·er·a

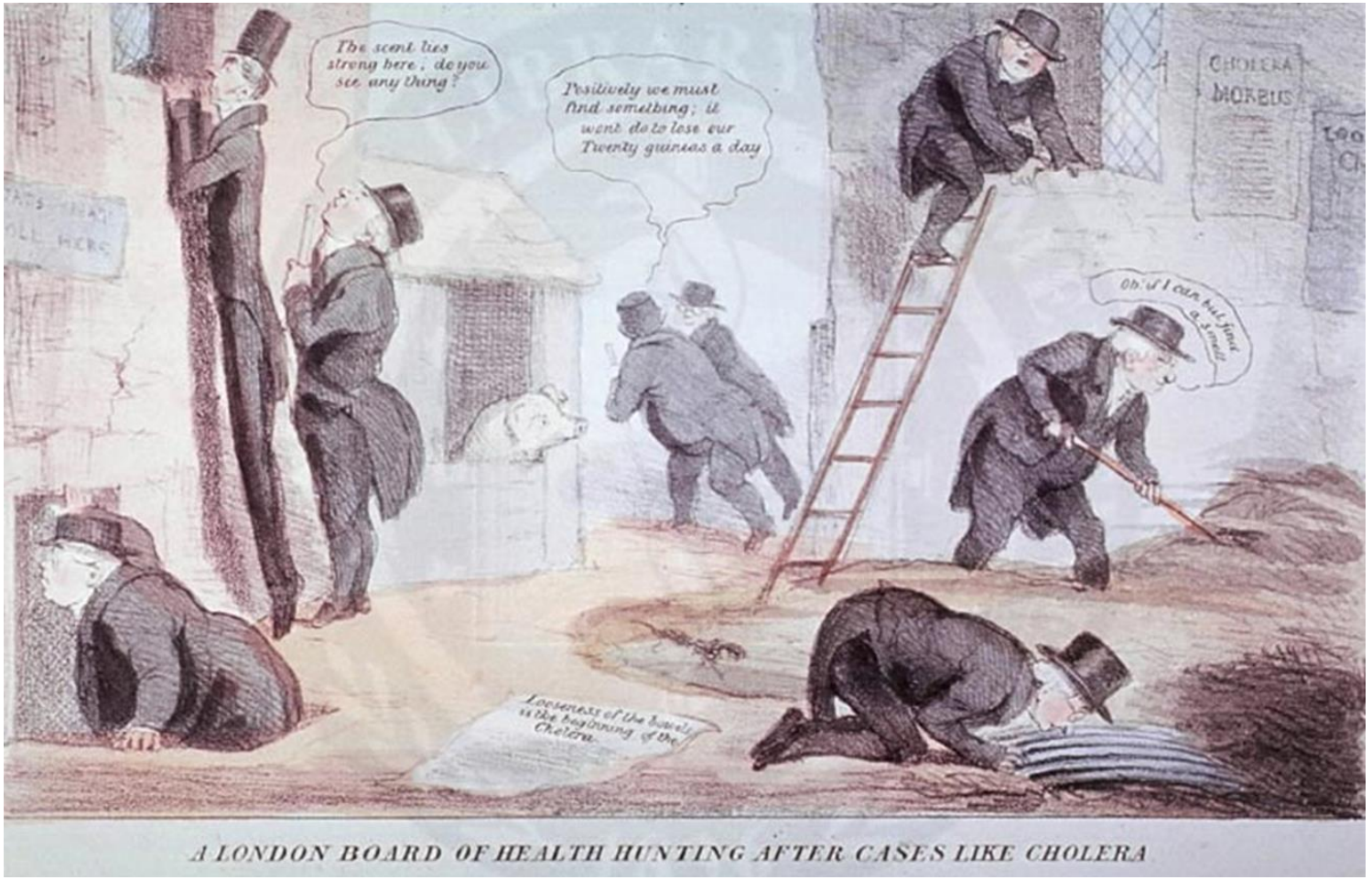
/ˈkälərə/

noun

an infectious and often fatal bacterial disease of the small intestine, typically contracted from infected water supplies and causing severe vomiting and diarrhea.

1854: Galen's miasma theory of cholera

1854: The hunt for the invisible cholera



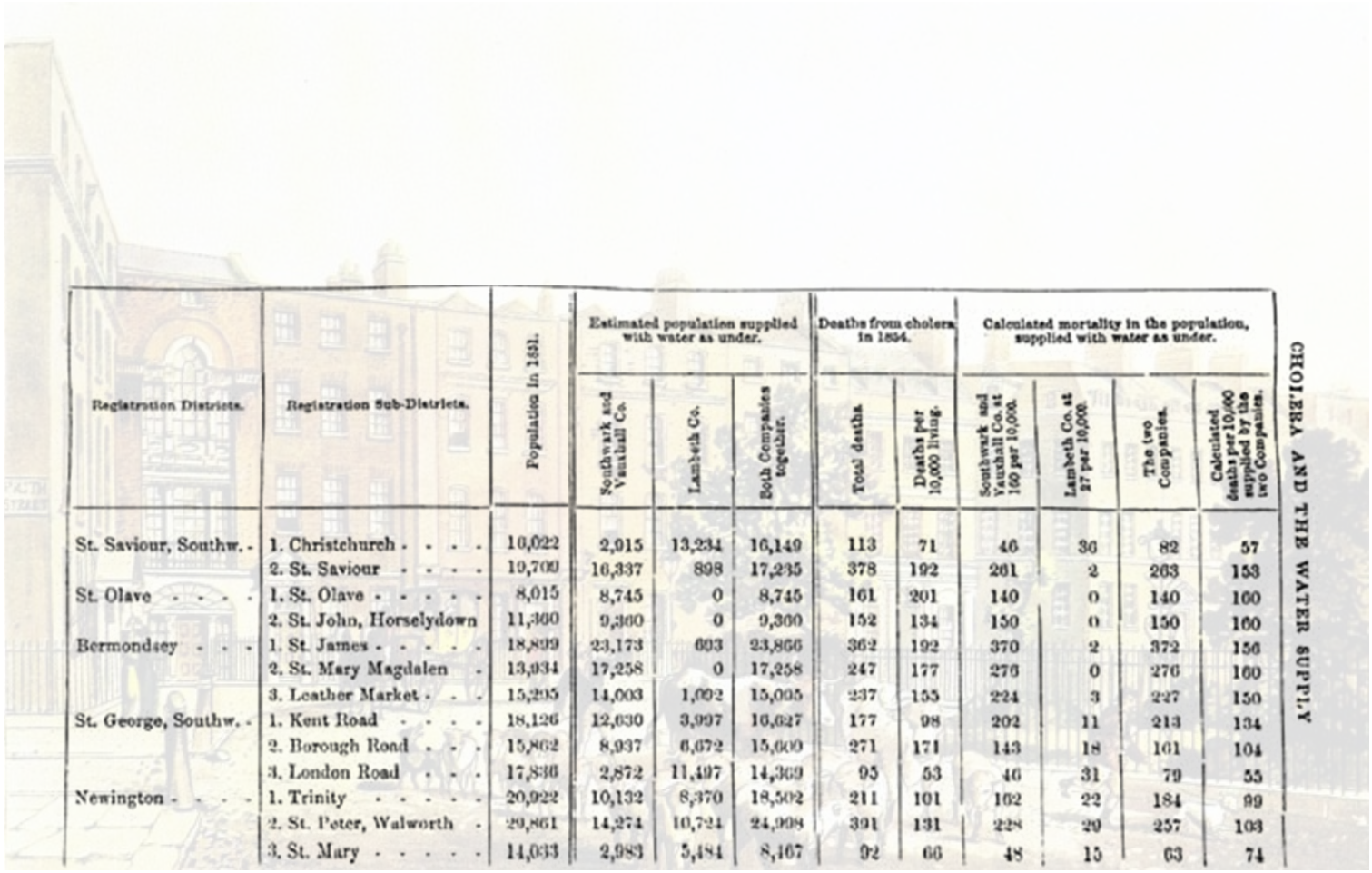
A LONDON BOARD OF HEALTH HUNTING AFTER CASES LIKE CHOLERA

John Snow: 1813–1858

John Snow: 1813–1858

The survey of Soho

The survey of Soho

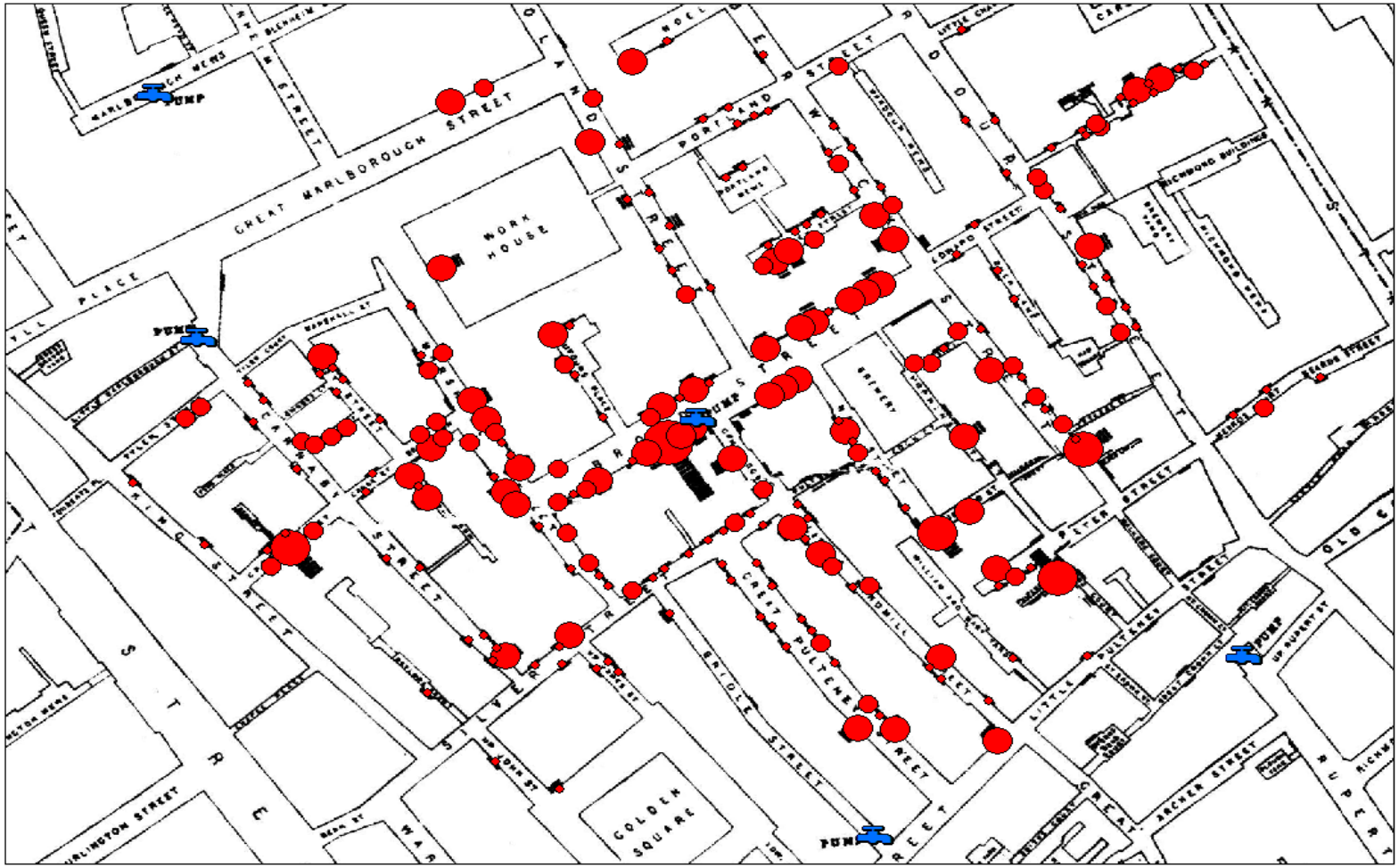


CHOLERA AND THE WATER SUPPLY.

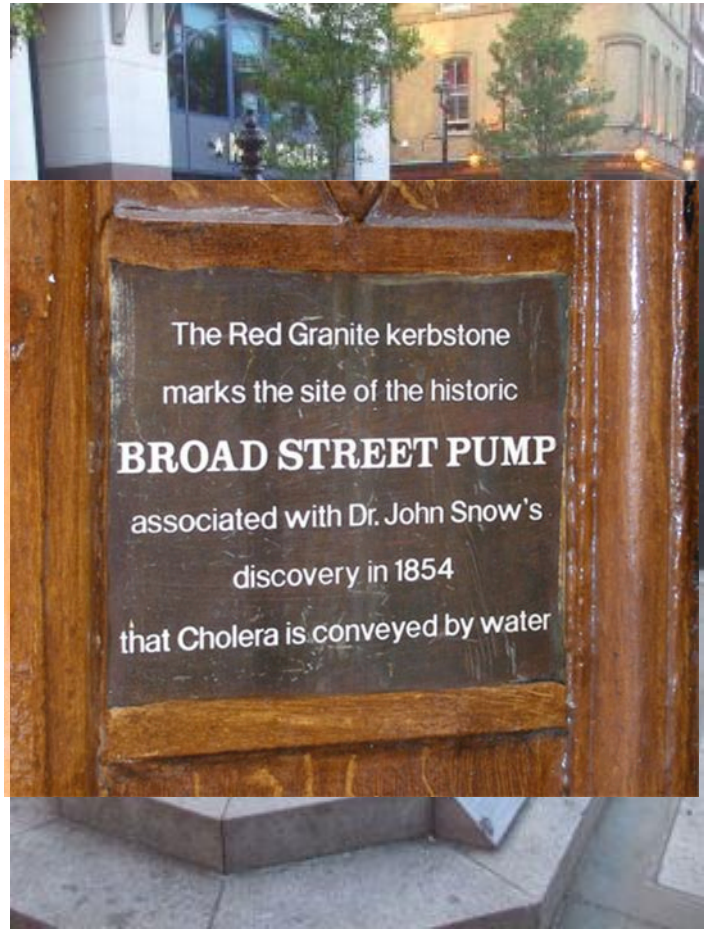
Registration Districts.	Registration Sub-Districts.	Population in 1851.	Estimated population supplied with water as under.			Deaths from cholera in 1854.		Calculated mortality in the population, supplied with water as under.			
			Southwark and Vauxhall Co.	Lambeth Co.	Both Companies together.	Total deaths	Deaths per 10,000 living.	Southwark and Vauxhall Co. at 150 per 10,000.	Lambeth Co. at 37 per 10,000.	The two Companies.	Calculated deaths per 10,000 supplied by the two Companies.
St. Saviour, Southw.	1. Christchurch	16,022	2,915	13,234	16,149	113	71	46	36	82	57
	2. St. Saviour	19,709	16,337	898	17,235	378	192	261	2	263	153
St. Olave	1. St. Olave	8,015	8,745	0	8,745	161	201	140	0	140	160
	2. St. John, Horselydown	11,360	9,360	0	9,360	152	134	150	0	150	160
Bermondsey	1. St. James	18,899	23,173	693	23,866	362	192	370	2	372	156
	2. St. Mary Magdalen	13,034	17,258	0	17,258	247	177	276	0	276	160
	3. Leather Market	15,295	14,003	1,092	15,095	237	155	224	3	227	150
St. George, Southw.	1. Kent Road	18,126	12,630	3,997	16,627	177	98	202	11	213	134
	2. Borough Road	15,862	8,937	6,672	15,609	271	171	143	18	161	104
	3. London Road	17,836	2,872	11,497	14,369	95	53	46	31	79	55
Newington	1. Trinity	20,922	10,132	8,370	18,502	211	101	162	22	184	99
	2. St. Peter, Walworth	29,861	14,274	10,724	24,998	391	131	228	29	257	103
	3. St. Mary	14,033	2,983	5,484	8,467	92	66	48	15	63	74

What the data showed …

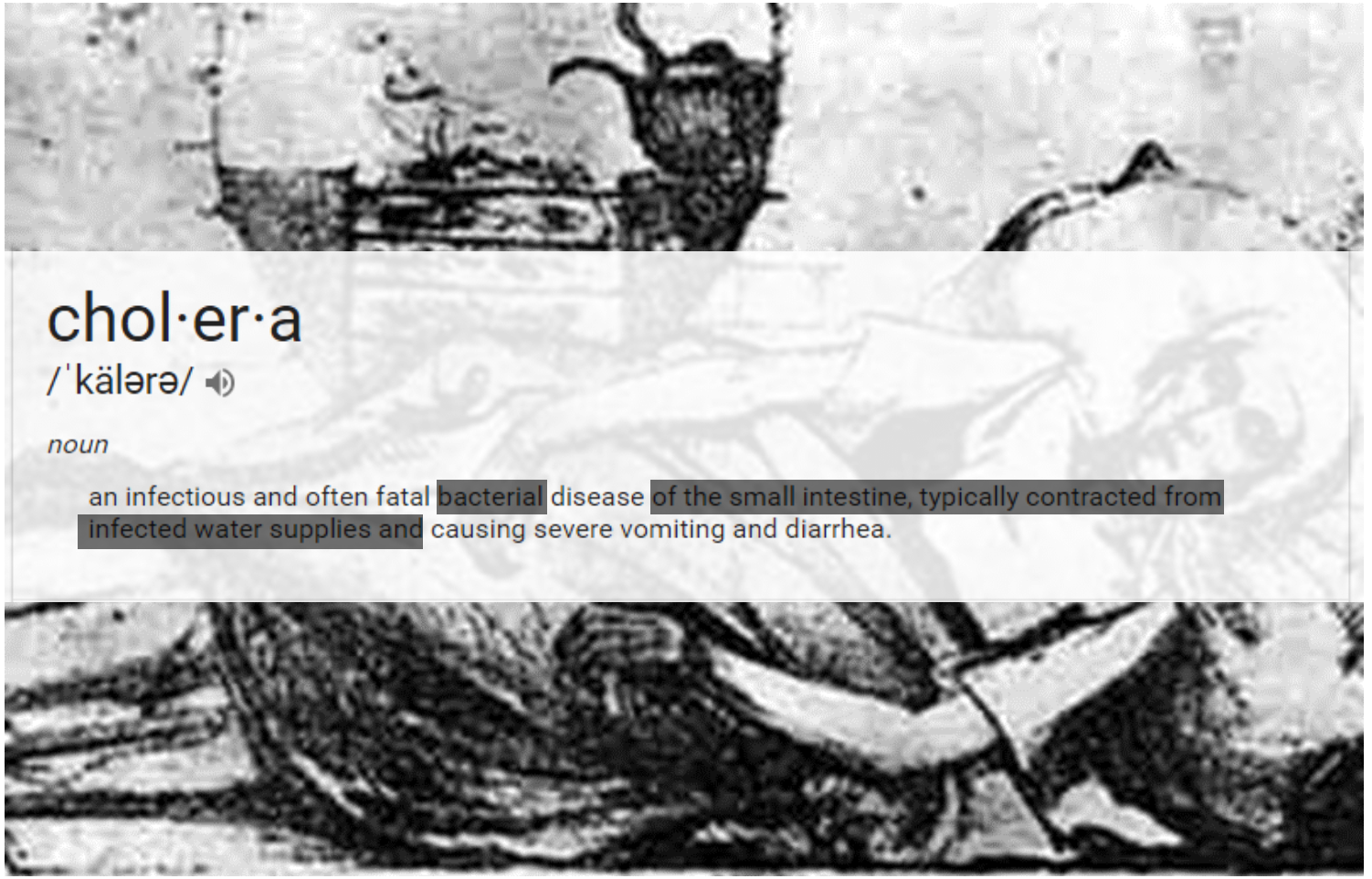
What the data showed …



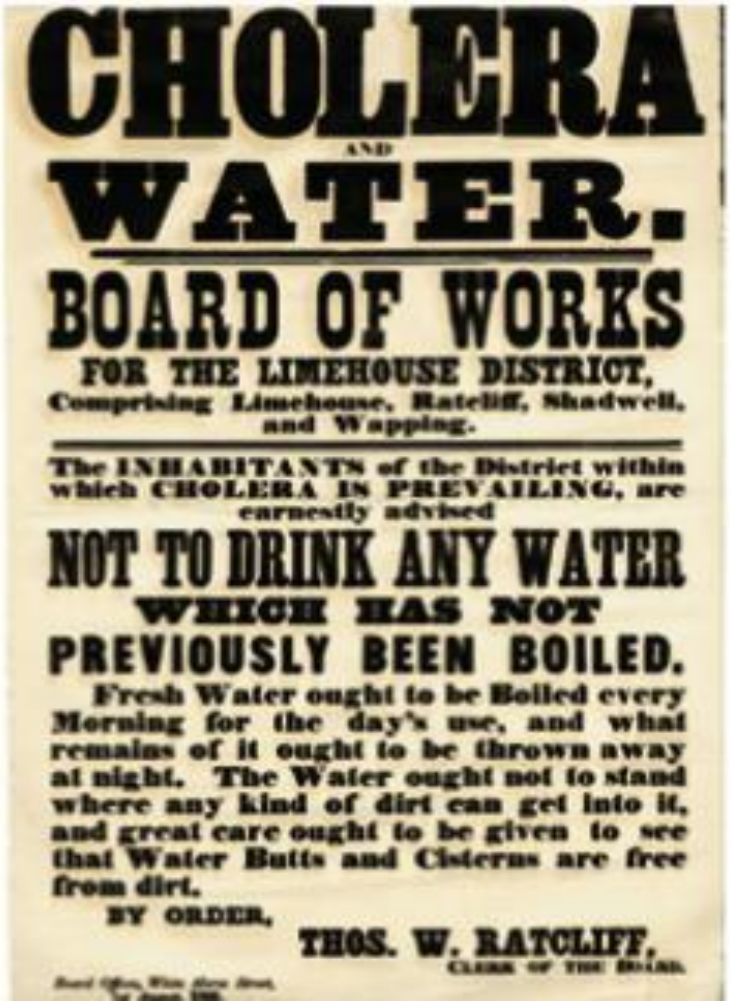
616 deaths, 8 days later …



Cholera: What we knew in 1855



Cholera boil notice ca. 1866



CHOLERA
AND
WATER.

BOARD OF WORKS
FOR THE LIMEHOUSE DISTRICT,
Comprising Limehouse, Ratcliff, Shadwell, and Wapping.

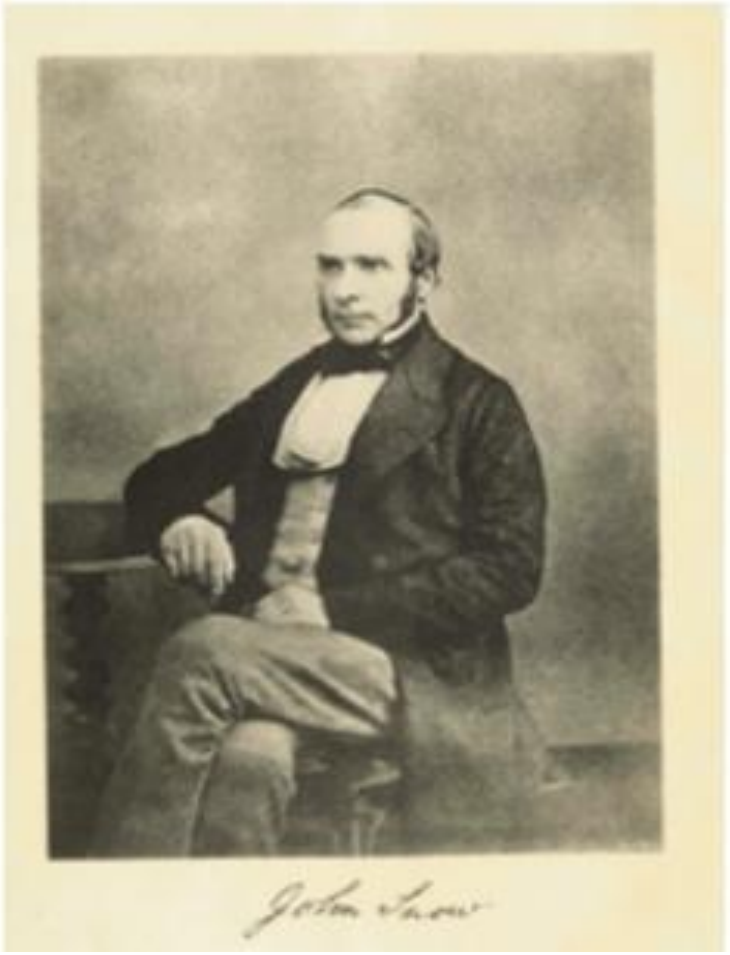
The INHABITANTS of the District within which CHOLERA IS PREVAILING, are earnestly advised

NOT TO DRINK ANY WATER
WHICH HAS NOT
PREVIOUSLY BEEN BOILED.

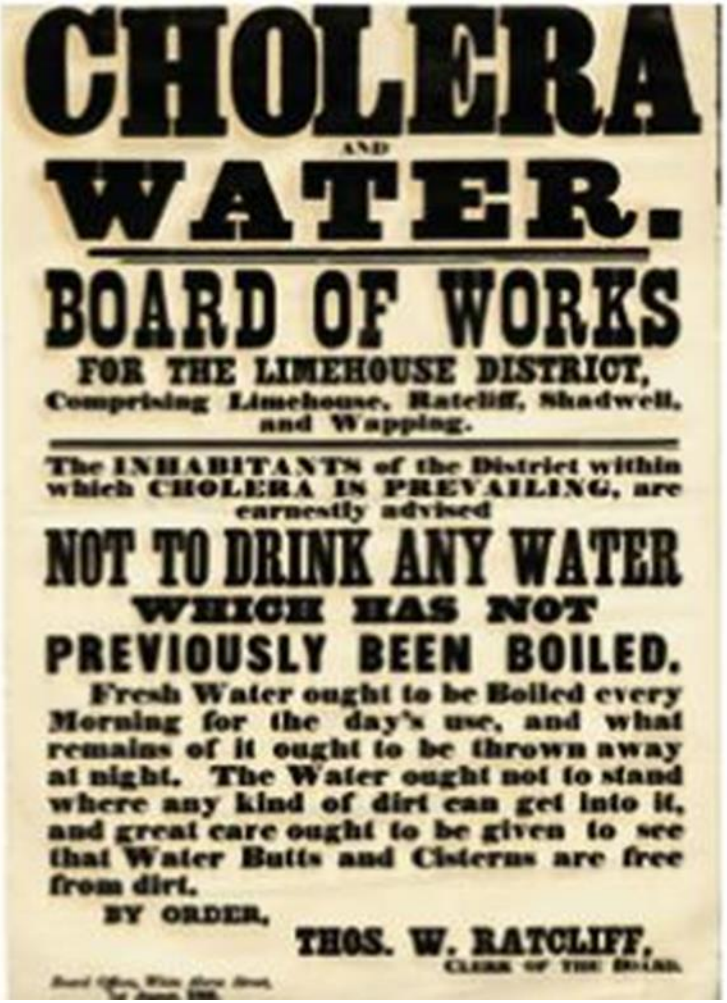
Fresh Water ought to be Boiled every Morning for the day's use, and what remains of it ought to be thrown away at night. The Water ought not to stand where any kind of dirt can get into it, and great care ought to be given to see that Water Butts and Cisterns are free from dirt.

BY ORDER,

THOS. W. RATCLIFF,
CLERK OF THE BOARD.



Cholera boil notice ca. 1866



CHOLERA
AND
WATER.

BOARD OF WORKS
FOR THE LIMEHOUSE DISTRICT,
Comprising Limehouse, Ratcliff, Shadwell, and Wapping.

The INHABITANTS of the District within which CHOLERA IS PREVAILING, are earnestly advised

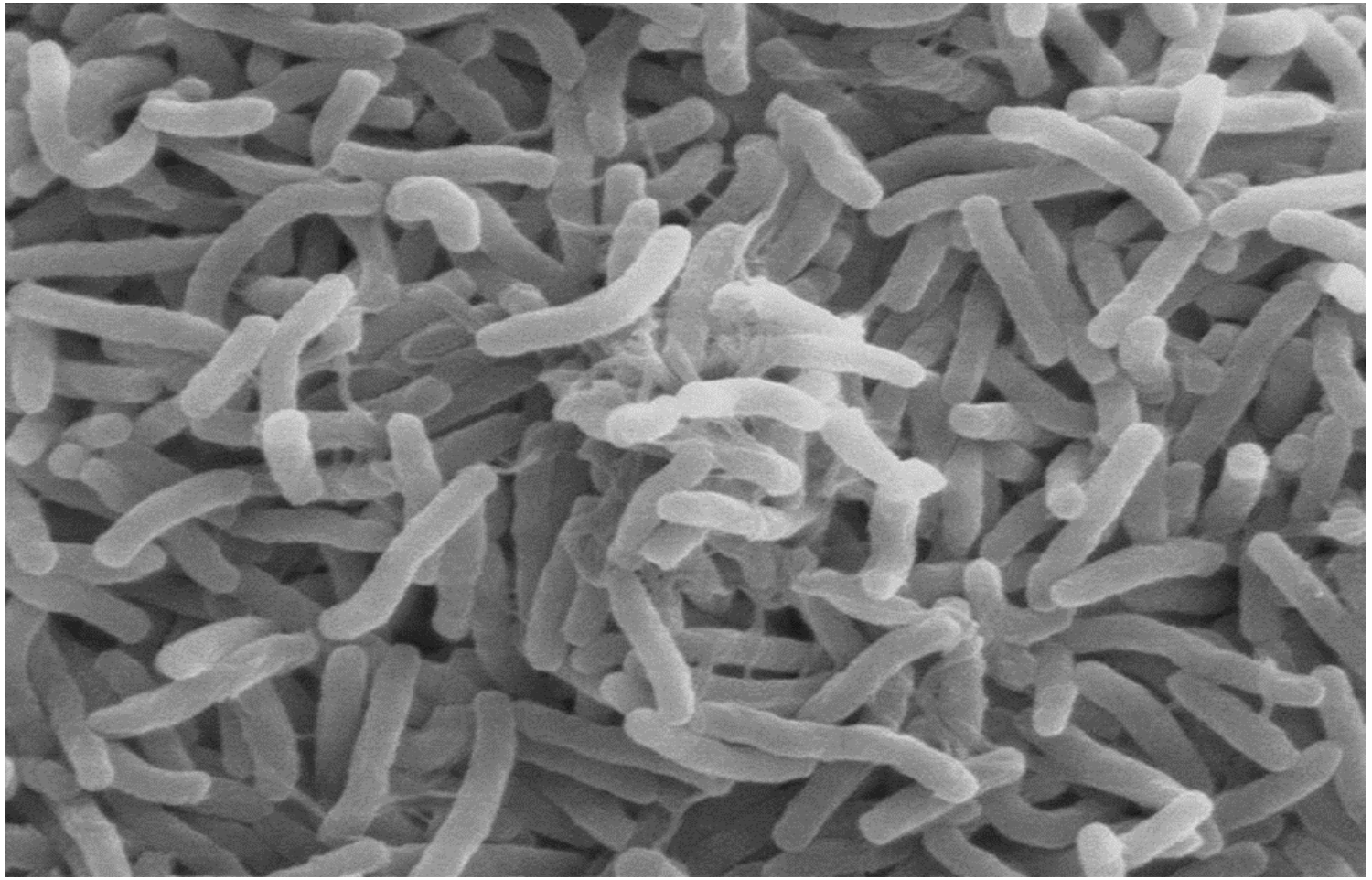
NOT TO DRINK ANY WATER
WHICH HAS NOT
PREVIOUSLY BEEN BOILED.

Fresh Water ought to be Boiled every Morning for the day's use, and what remains of it ought to be thrown away at night. The Water ought not to stand where any kind of dirt can get into it, and great care ought to be given to see that Water Butts and Cisterns are free from dirt.

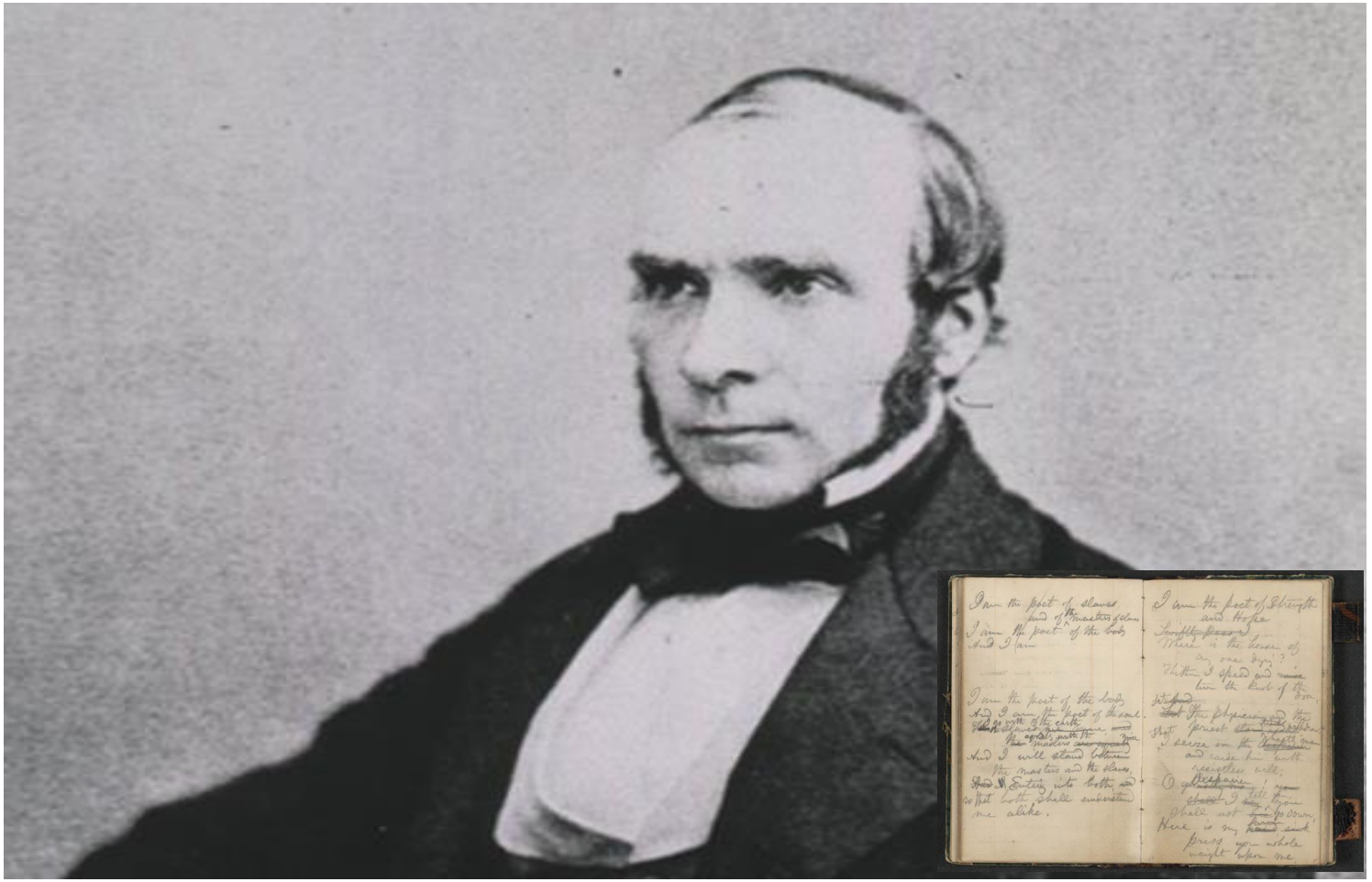
BY ORDER,

THOS. W. RATCLIFF,
CLERK OF THE BOARD.

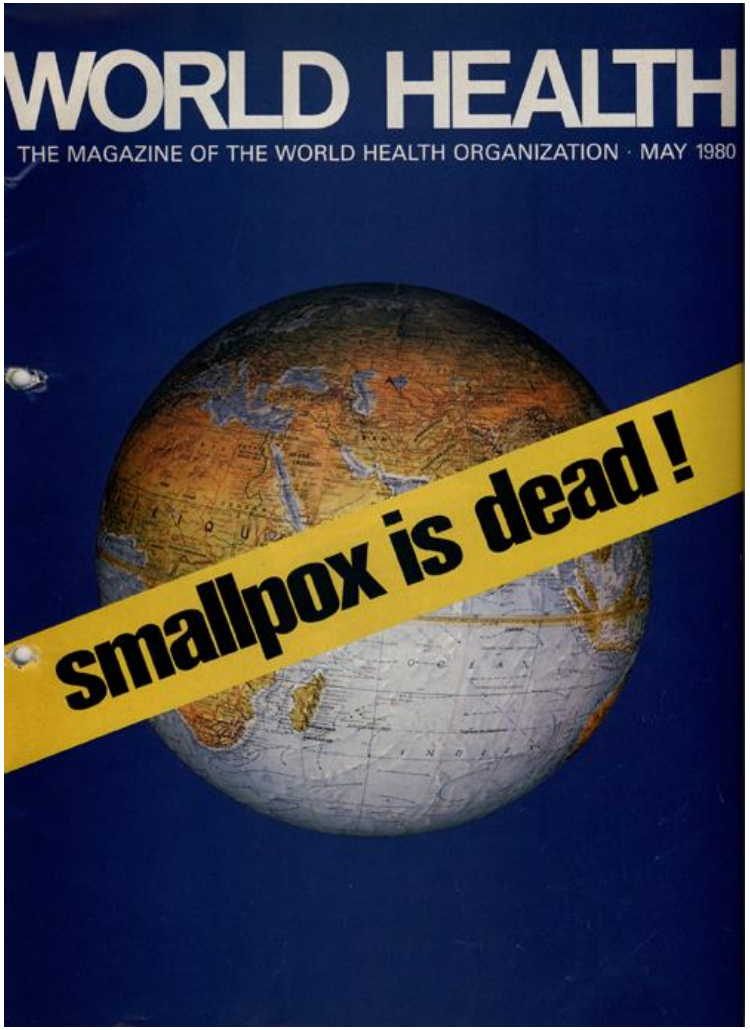
Thirty years before discovery of *V. cholerae*



John Snow: Father of Epidemiology



Epidemiology's Success Stories

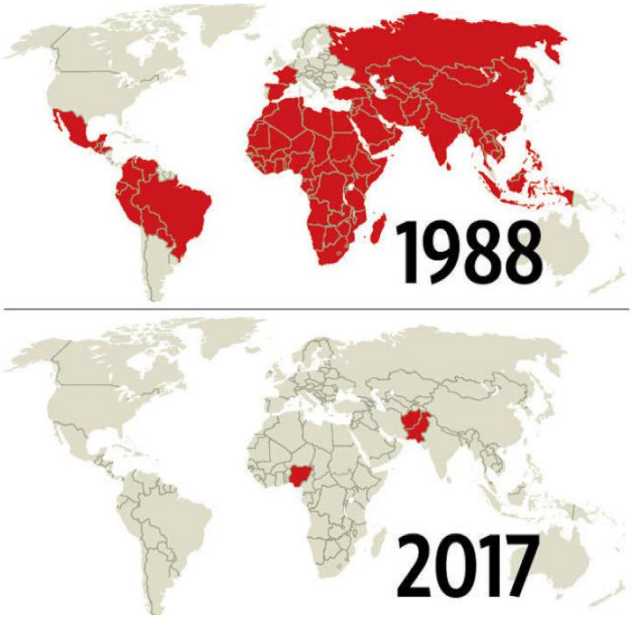


The polio endgame

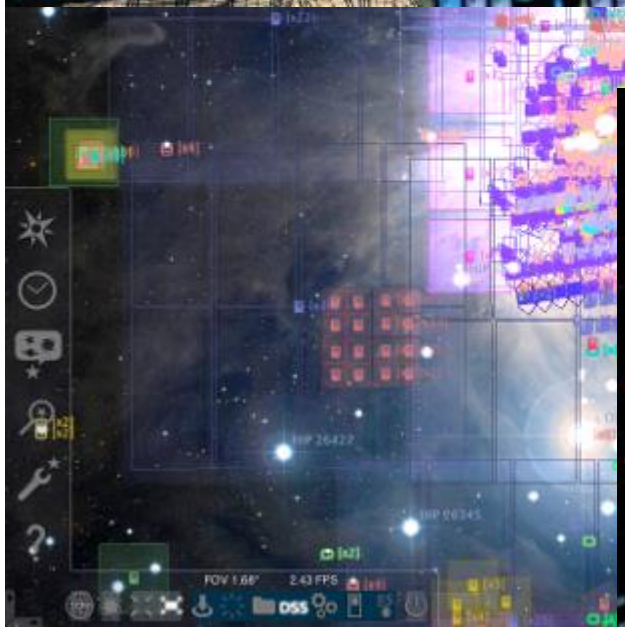
Since 1988, when the WHO resolved to eradicate polio, its footprint has shrunk dramatically. It is only considered endemic in Afghanistan, Pakistan and Nigeria (which hasn't seen a case since 2016). Last year there were only 22 new cases reported.

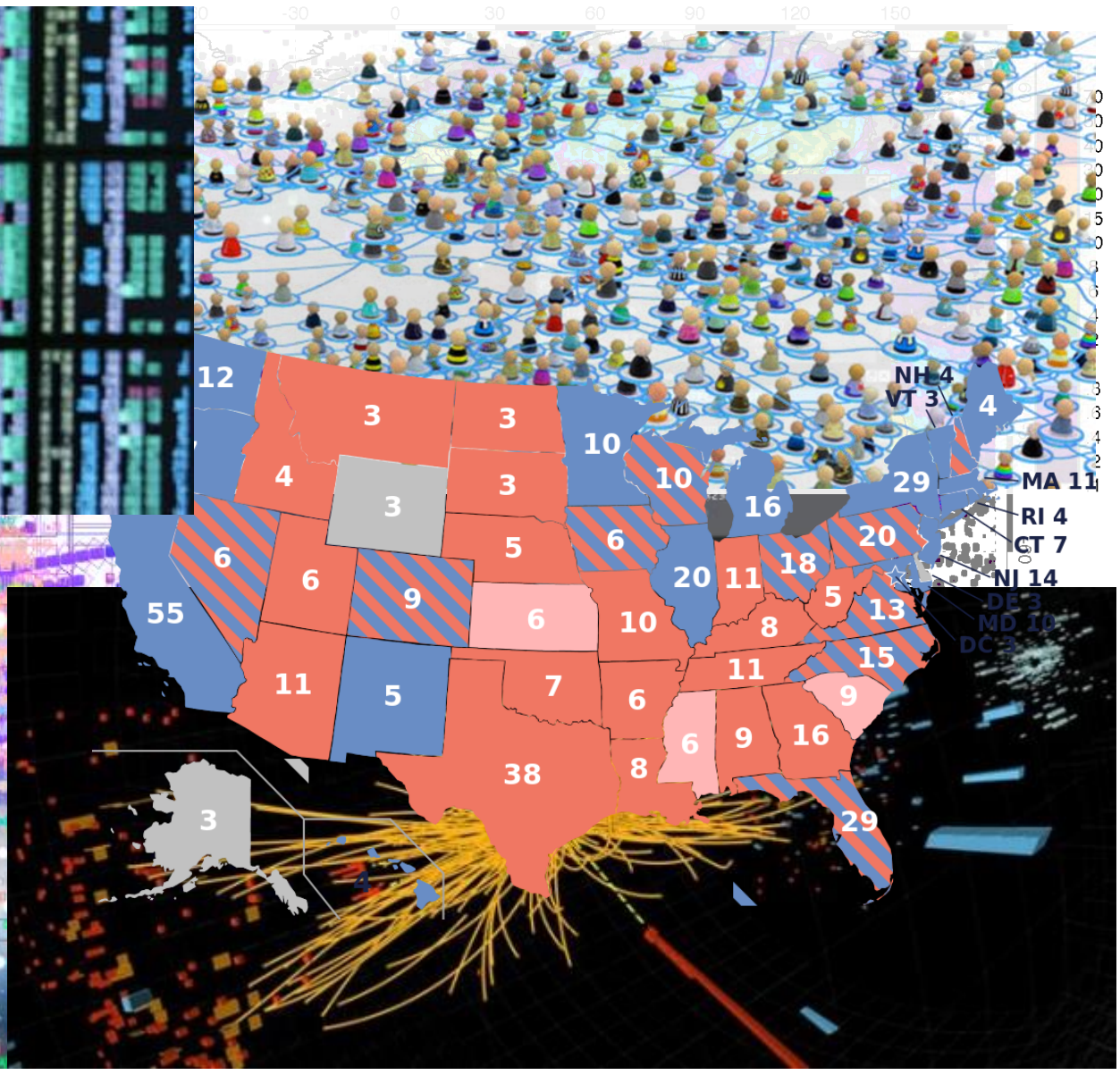
	1988	2017
Endemic countries	125	3

SOURCE: World Health Organization
TORONTO STAR GRAPHIC

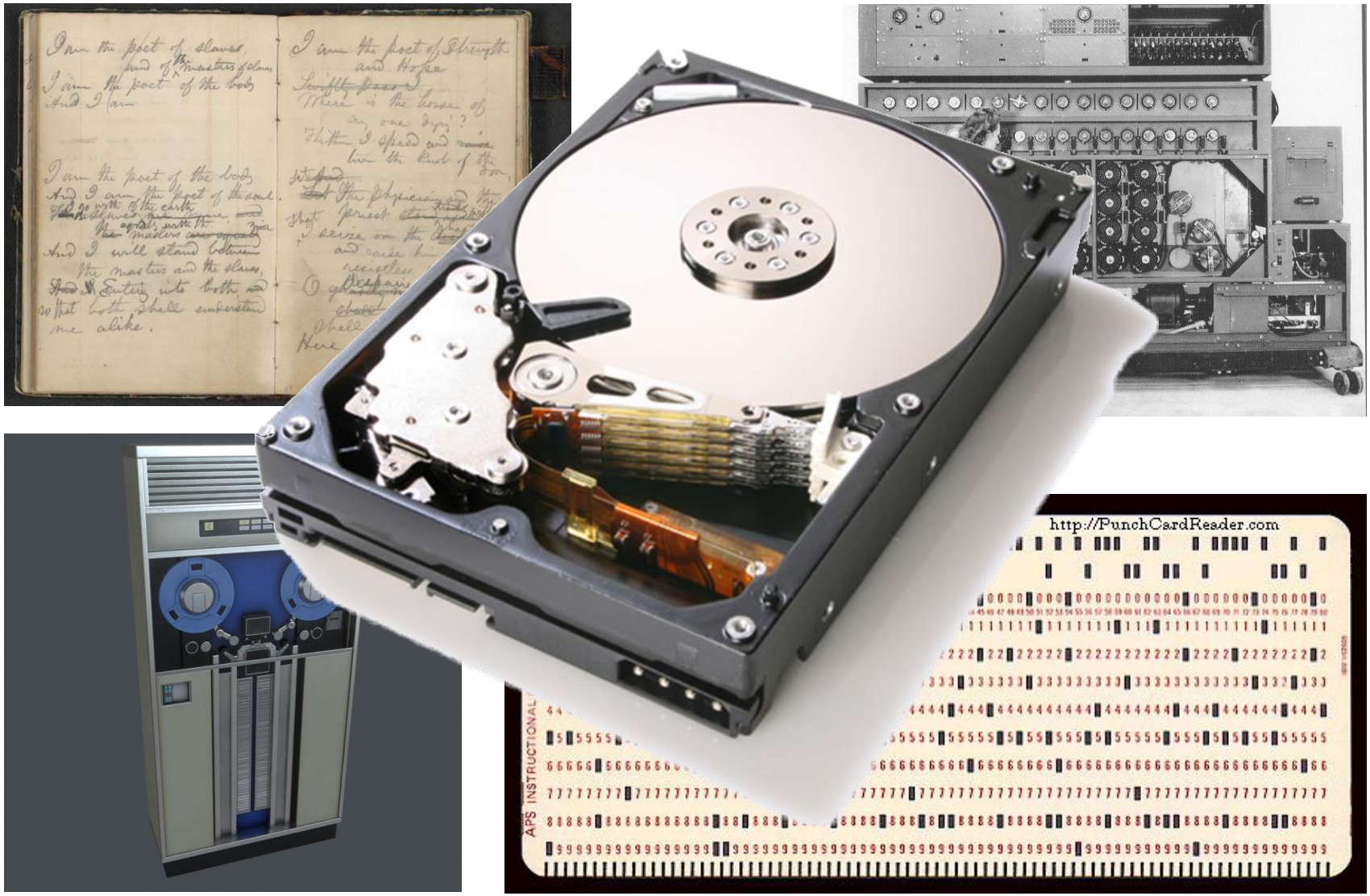


Value of data: Not just epidemiology





(Paper) Notebooks no longer good enough



The Growth of Data

“Big Data”

English Wikipedia

≈ 51 GB of data

(2015 dump)

(Text; No edit history)

(XML, uncompressed)

WIKIPEDIA

The Free Encyclopedia

1 Wiki = 1 Wikipedia

“Big Data”

Wikimedia Commons

≈ 24 TB of data

≈ 470.6 Wiki

(2014 dump)

“Big Data”

Twitter

≈ 8 TB / day

≈ 157 Wiki / day

(2013, generated)

“Big Data”

Large Synoptic Survey Telescope

≈ 15 TB / day (night)

≈ 294 Wiki / day

(2020, generated)

“Big Data”

Facebook

≈ 600 TB / day

≈ 11,764 Wiki / day

(2014, incoming Hive data)

“Big Data”

Large Hadron Collider

≈ 1 PB / day

≈ 19,607 Wiki / day

(2017, filtered data)

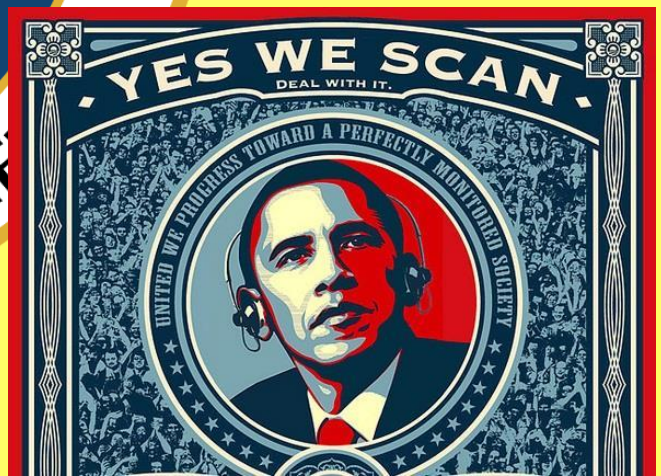
“Big Data”

PRISM: NSA Surveillance

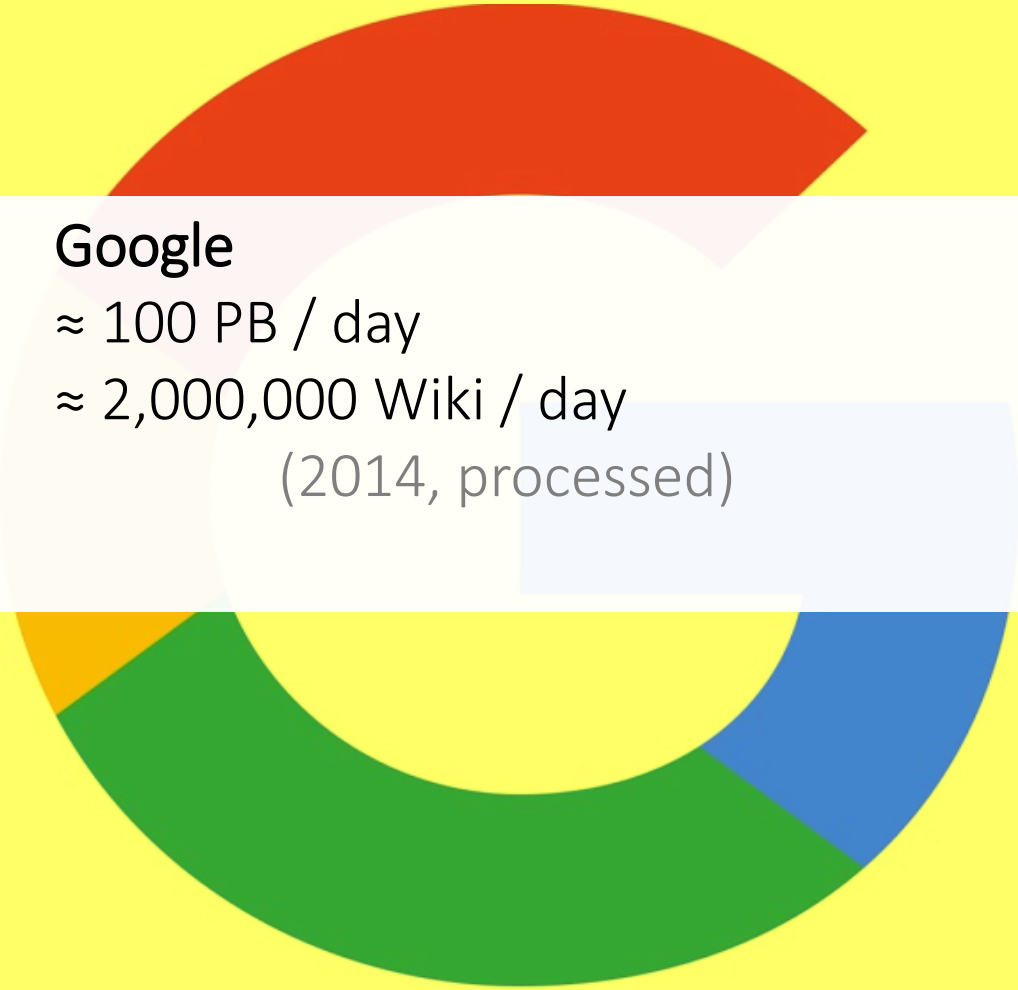
≈ 29 PB / day

≈ 568,627 Wiki / day

(2013, processed)



“Big Data”



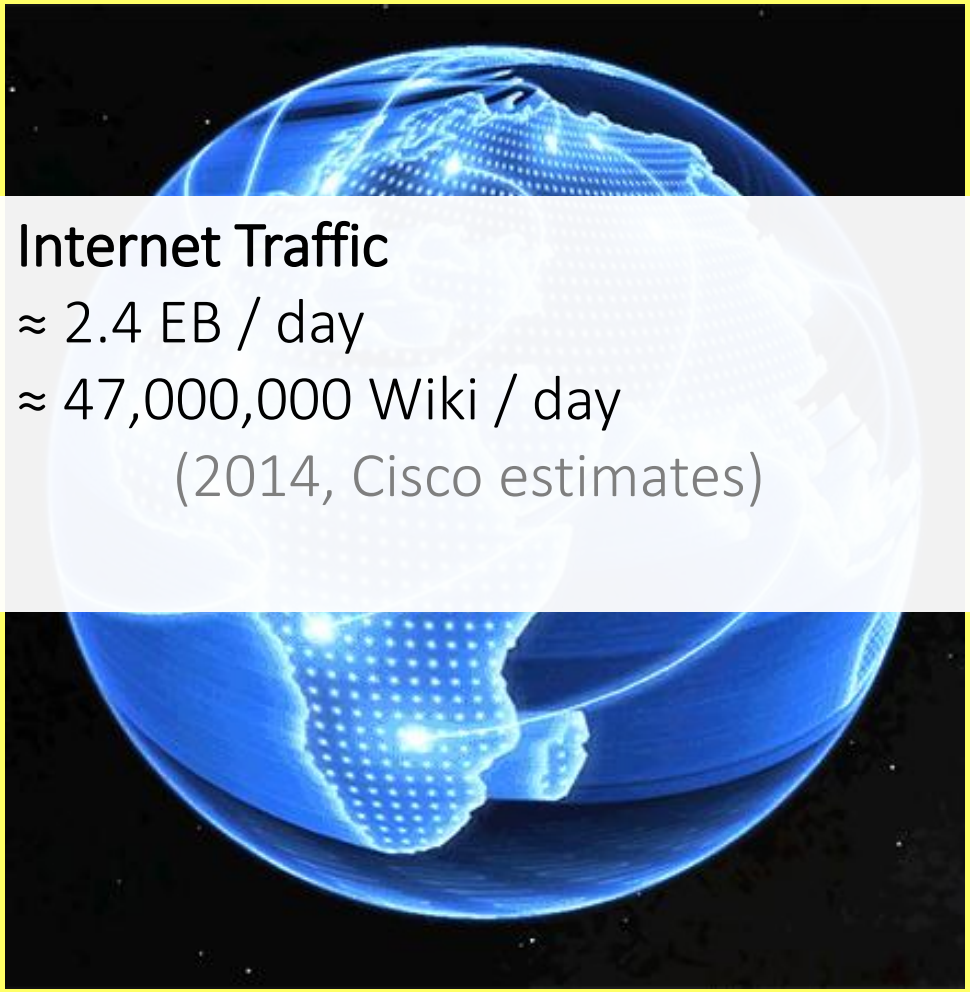
Google

≈ 100 PB / day

≈ 2,000,000 Wiki / day

(2014, processed)

“Big Data”



Internet Traffic

≈ 2.4 EB / day

≈ 47,000,000 Wiki / day

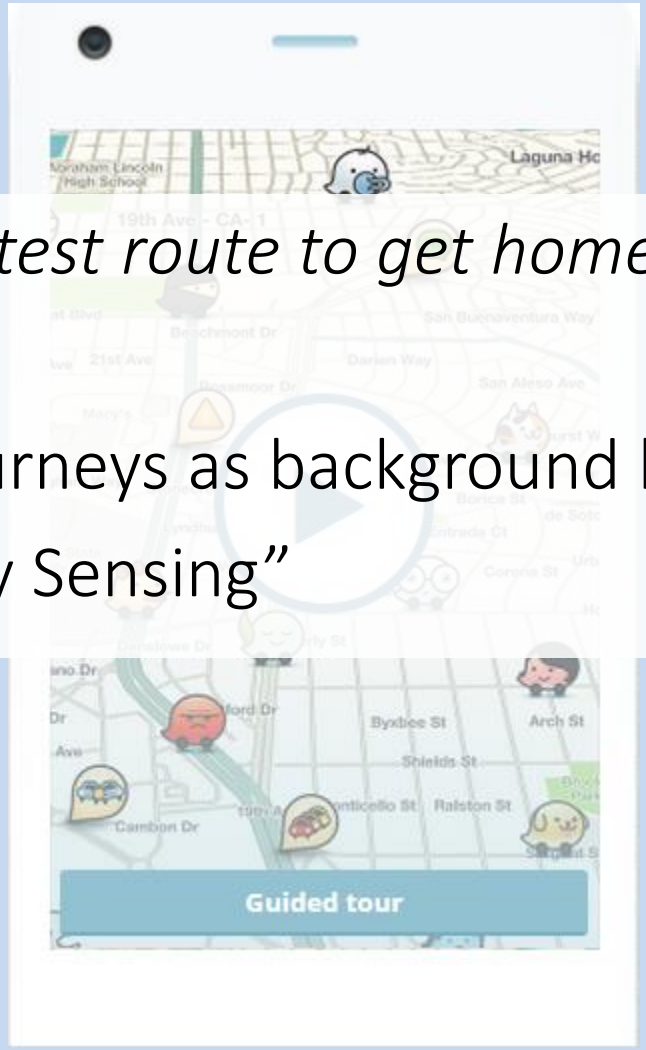
(2014, Cisco estimates)

Data: A Modern-day Bottleneck?

The ‘V’s of “Big Data”

“Big Data” in action …

Getting Home (Waze)



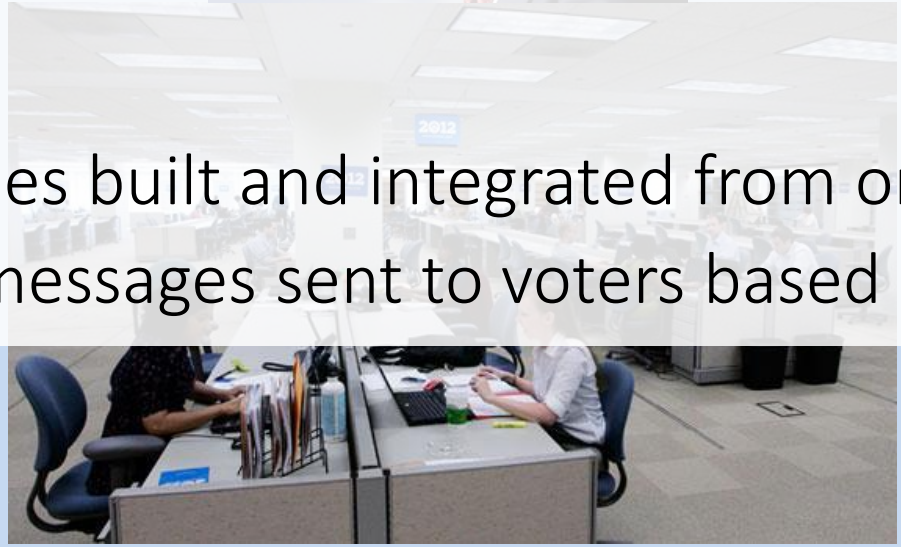
"What's the fastest route to get home right now?"

- Processes journeys as background knowledge
- "Participatory Sensing"

Getting Elected President (Narwhal)

"Who are the undecided voters and how can I convince them to vote for me?"

- User profiles built and integrated from online sources
- Targeted messages sent to voters based on profile



Getting Elected President (Narwhal)

"Who are the undecided voters and how can I convince them to vote for me?"

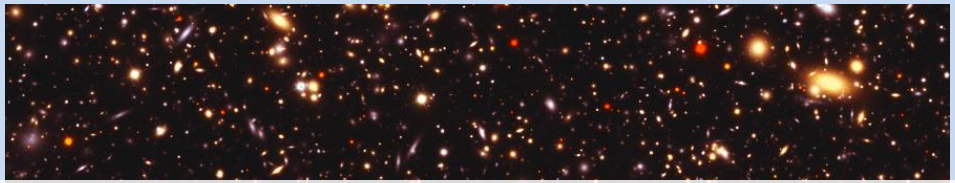
- Us
- Ta

Winning Jeopardy (IBM Watson)

"Can a computer beat human experts at Jeopardy?"

- Indexed 200 million pages of content
- An ensemble of 100 processing techniques

Surveying the Night Sky (LSST)



"What astronomic bodies are changing in the sky?"

- Surveys the visible sky each week (15 TB / day)
- Queries external catalogues for known information

Processing lots of data ...

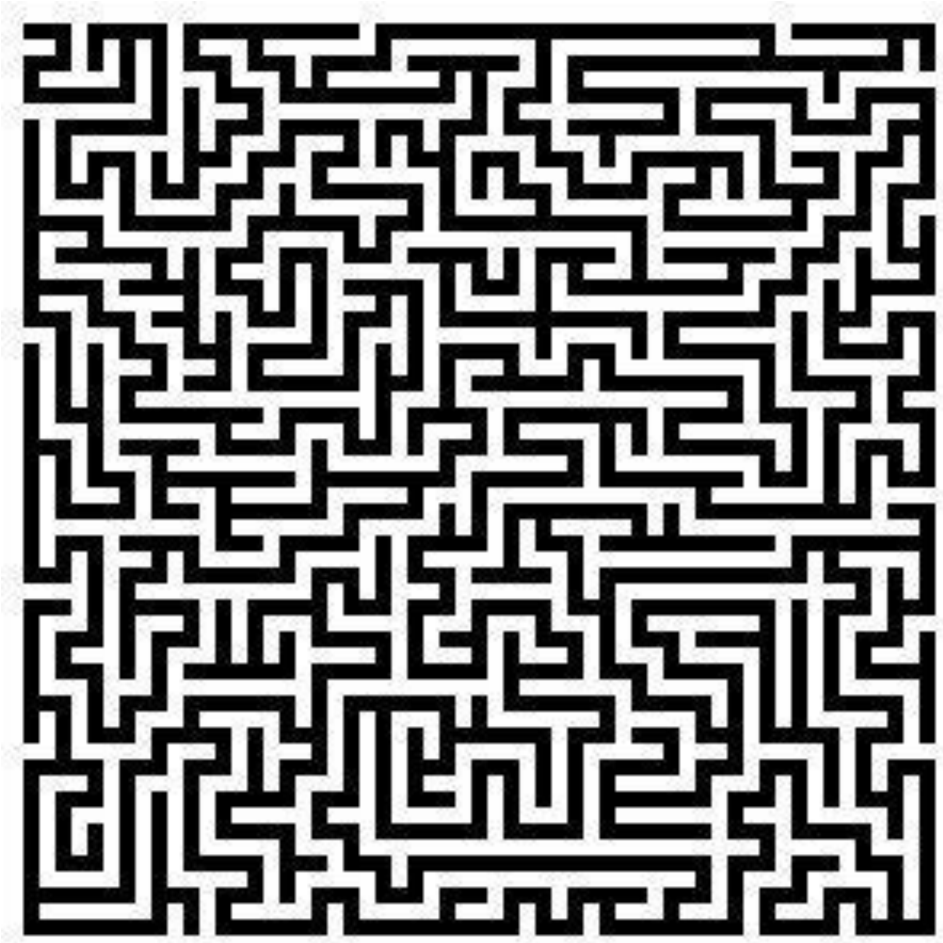
Every Application is Different …

- Data can be
  - (Semi-)Structured data
    - (Relational DBs, JSON, XML, CSV, HTML form data)
  - Unstructured data
    - (text document, comments, tweets)
  - And everything in-between!

Every Application is Different …

- Processing can involve:
  - Data Management
    - (indexing, querying, joins, aggregation)
  - Natural Language Processing
    - (keyword search, topic extraction, entity recognition, machine translation, etc.)
  - Data Mining and Statistics
    - (pattern recognition, classification, regression, recommendations, etc.)
  - Or something else / A mix

So where to start?



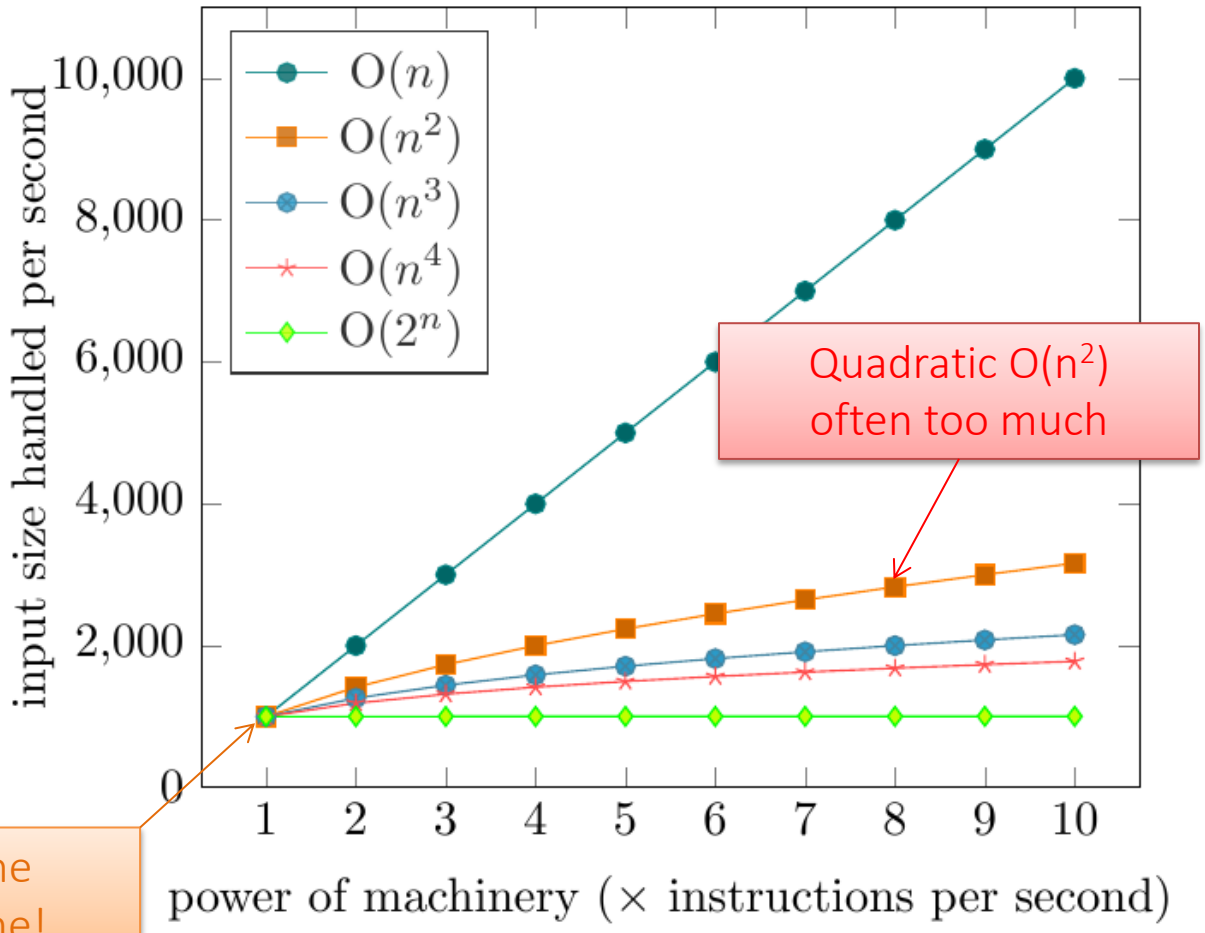
Scale is a Common Factor …

I have an algorithm.

I have a machine that can process 1,000 input items in an hour.

If I buy a machine that is *n* times as powerful, how many input items can I process in an hour?

Depends on what the algorithm is!!



Scale is a Common Factor …

- One machine that's *n* times as powerful?

vs.

- *n* machines that are equally as powerful?

Scale is a Common Factor …

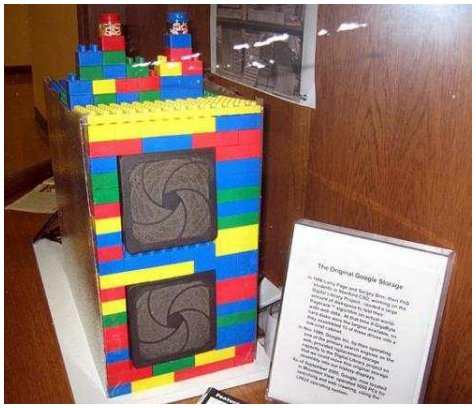
- Data-intensive
  - Inexpensive algorithms / Large inputs
  - e.g., Google, Facebook, Twitter
- Compute-intensive
  - More expensive algorithms / Smaller inputs
  - e.g., climate simulations, chess games, combinatorials
- No black and white!

Distributed Computing …

Distributed Computing

- Lots of data? Need more than one machine!

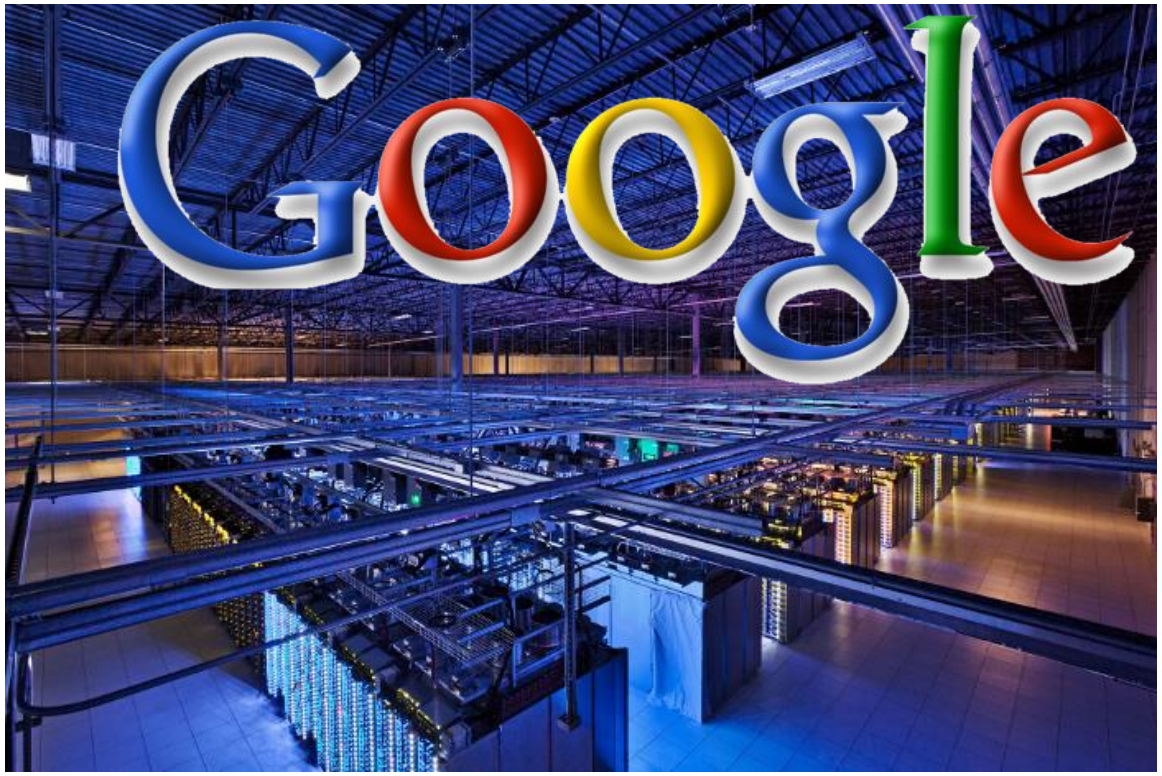
- Google ca. 1998:



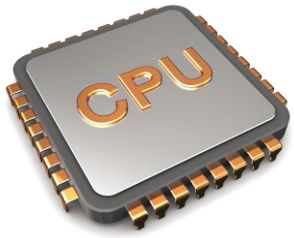
Distributed Computing

- Lots of data? Need more than one machine!

- Google ca. 2014:



Data Transport Costs (typical figures)



Important to minimise the network overhead!

- The network gives additional cost
- The network is shared across many machines

(bandwidth)

30 GB/s	600 MB/s	100 MB/s	1.25 GB/s	5 GB/s
Main Memory	Solid State Disk	Hard-disk	Network (same *rack*)	Network (across *racks*)
50–150 ns	10–100 µs	5–15 ms	300–600 ns	1–15 µs

(latency)

Data Placement

- Need to think carefully about where to put what data!

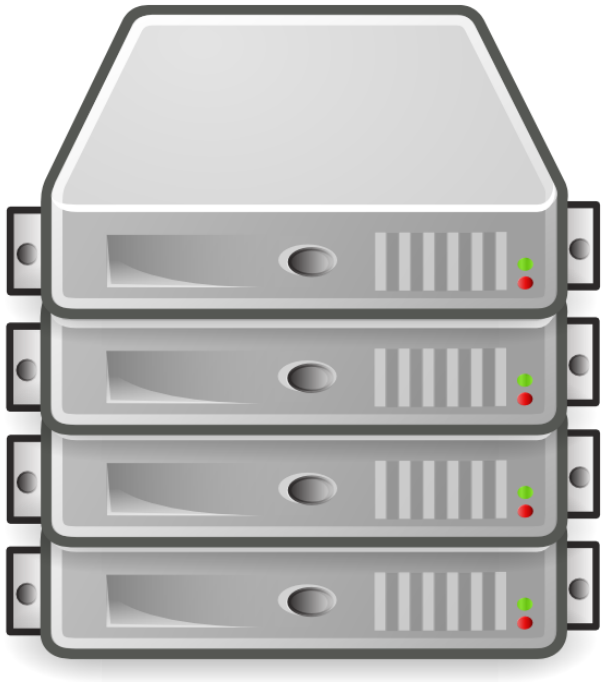
I have four machines to run a website. I have 10 million users.

Each user has personal profile data, photos, friends and games.

How should I split the data up over the machines?

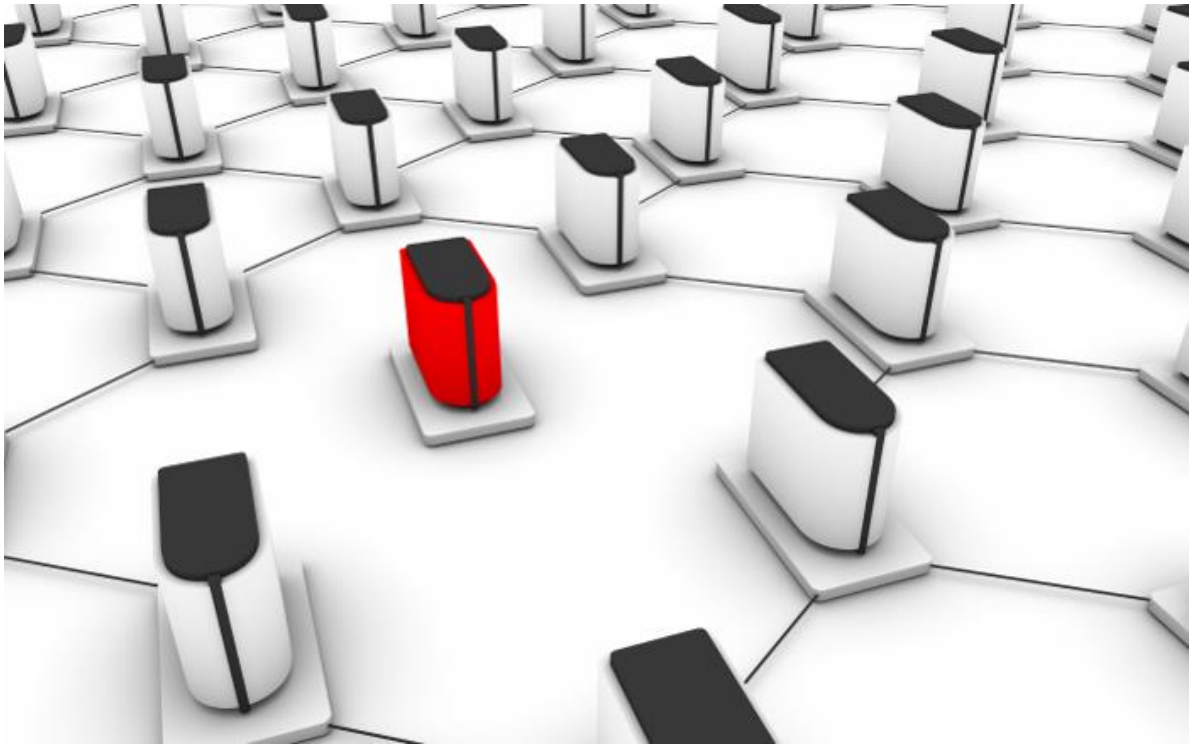
Depends on the application!

But some general principles and design choices apply.



Network/Node Failures

- Need to think about failures!



Network/Node Failures

- Need to think (even more!) carefully about where to put what data!

I have four machines to run a website. I have 10 million users.

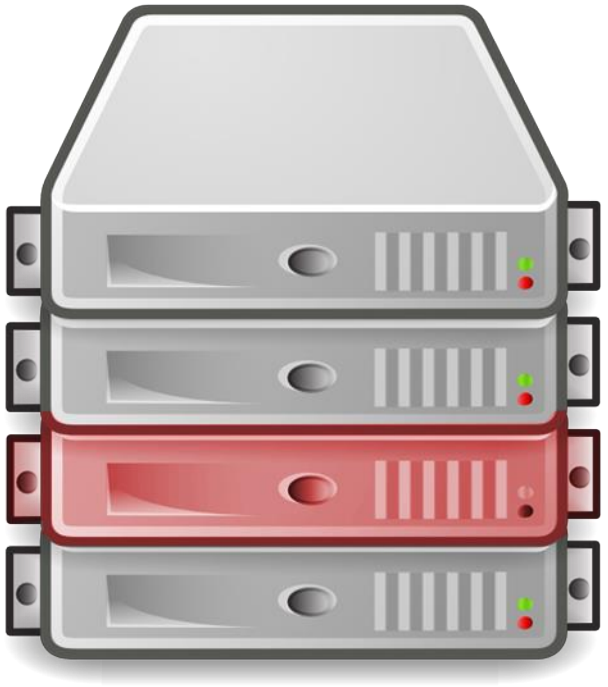
Each user has personal profile data, photos, friends and games.

How should I split the data up over the machines?

(Again)

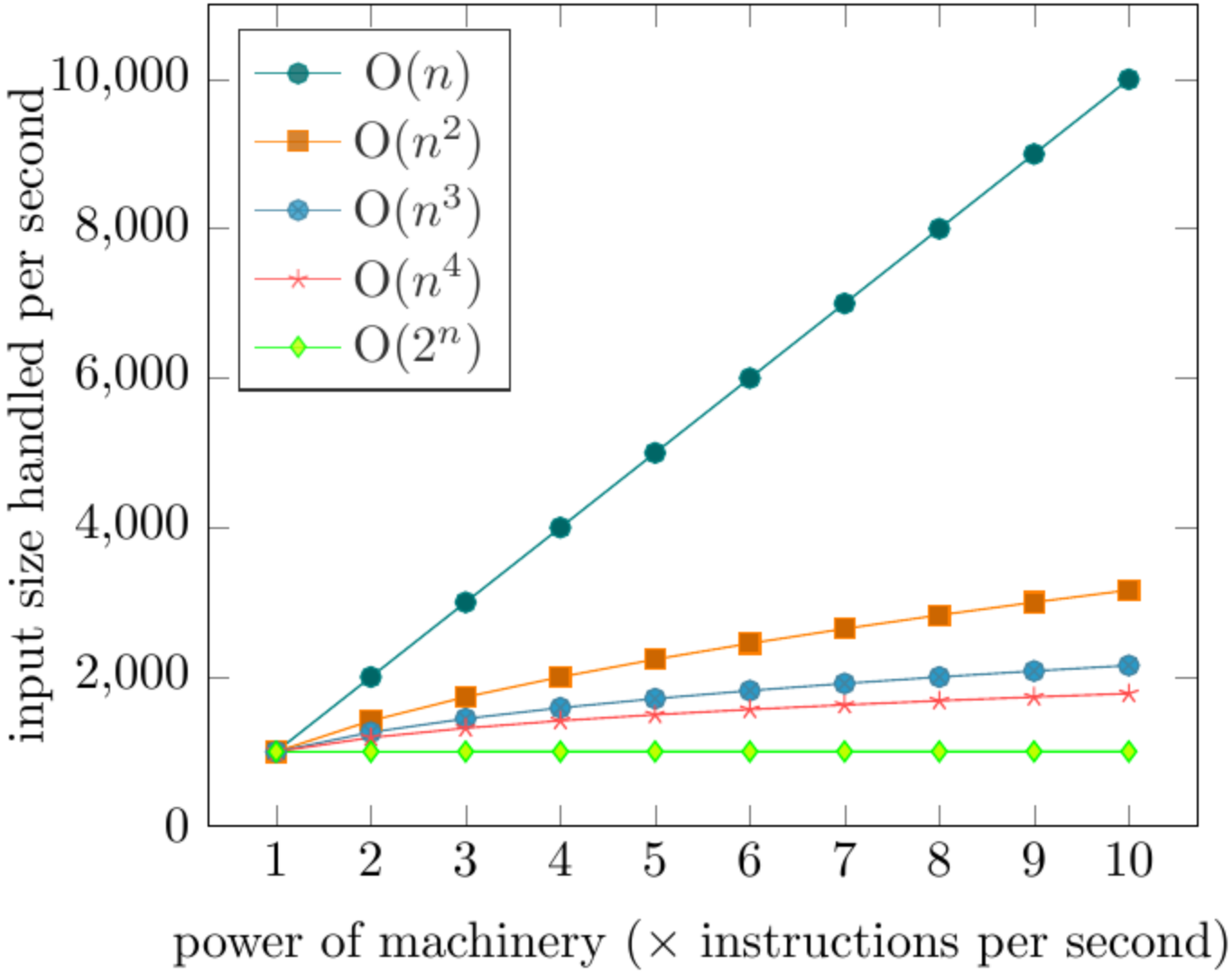
Depends on the application!

But some general principles and design choices apply.



Human Distributed Computation

Distribution Not Always Applicable!



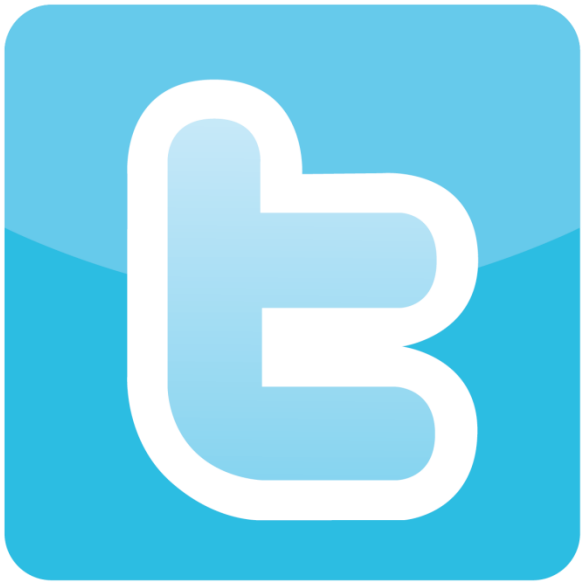
Distributed Development Difficult

- Distributed systems can be complex
- Multiple machines; need to take care of
  - Data in different locations
  - Logs and messages in different places
  - Different users with different priorities
  - Different network capabilities
  - Need to balance load!
  - Need to handle failures!
- Tasks may take a long time!
  - Bugs may not become apparent for hours
  - Lots of data = lots of counter-examples

Frameworks/Abstractions can Help

- For Distrib. Processing

- For Distrib. Storage

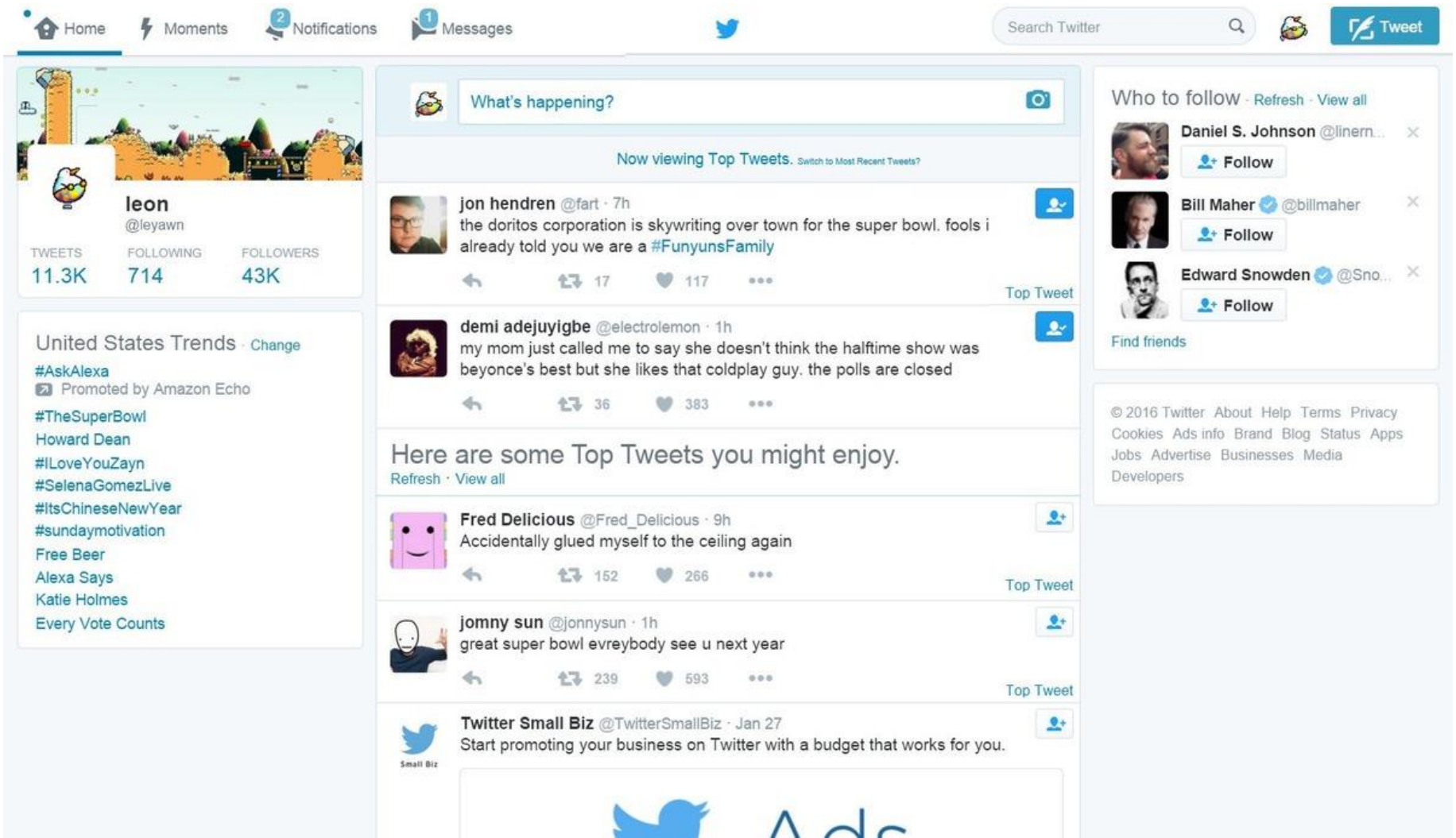


INSIDE TWITTER …

Based on 2013 slides by Twitter lead architect: Raffi Krikorian

"Twitter Timelines at Scale"

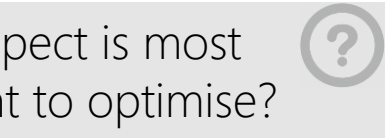
Twitter Timeline



Big Data at Twitter

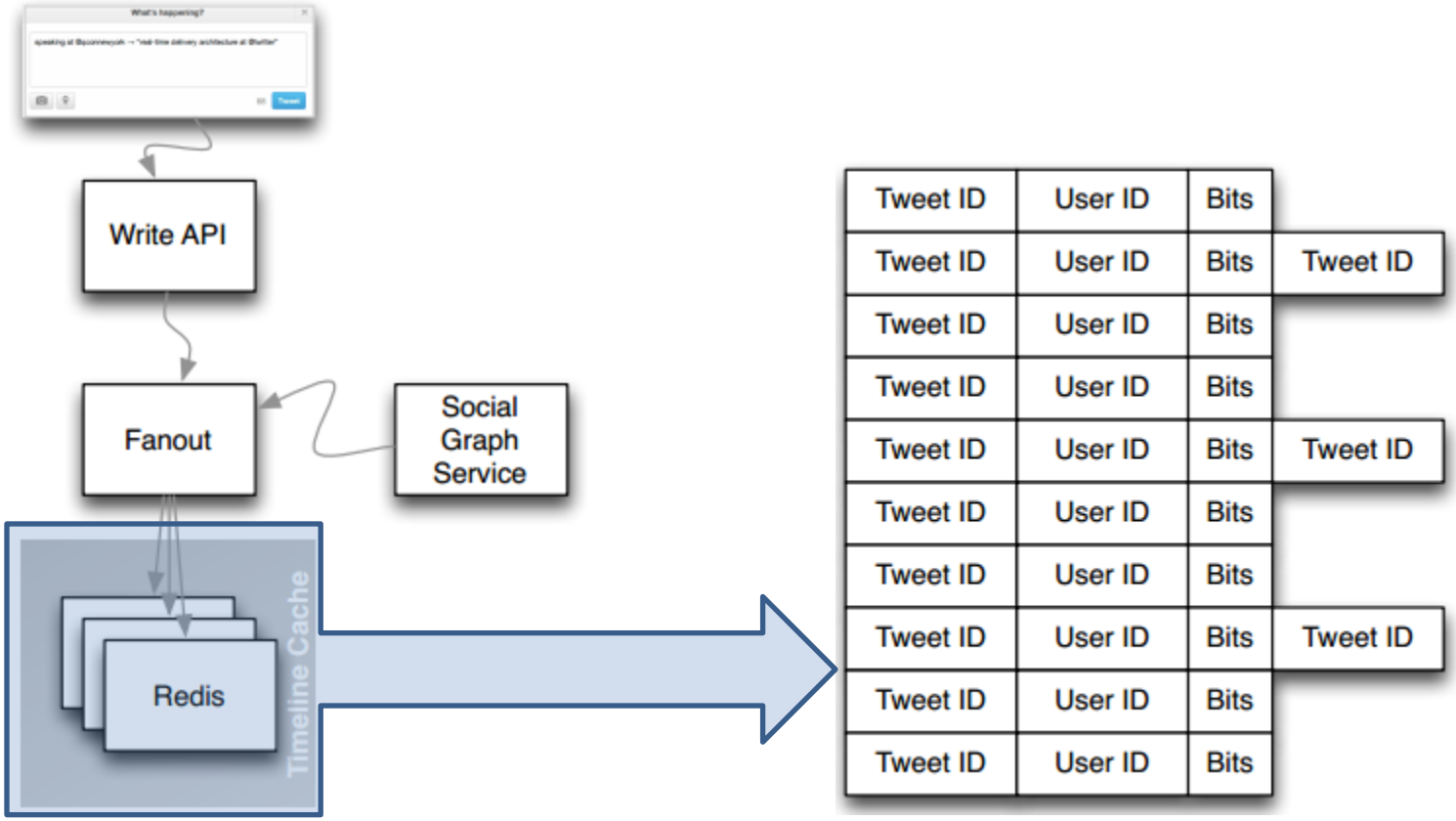
- 150 million active worldwide users
- 400 million tweets written per day
  - mean: 4,600 tweets/second
  - max: 150,000 tweets/second
- 300,000 queries/second for user timelines
- 6,000 queries/second for custom search

Which aspect is most important to optimise?

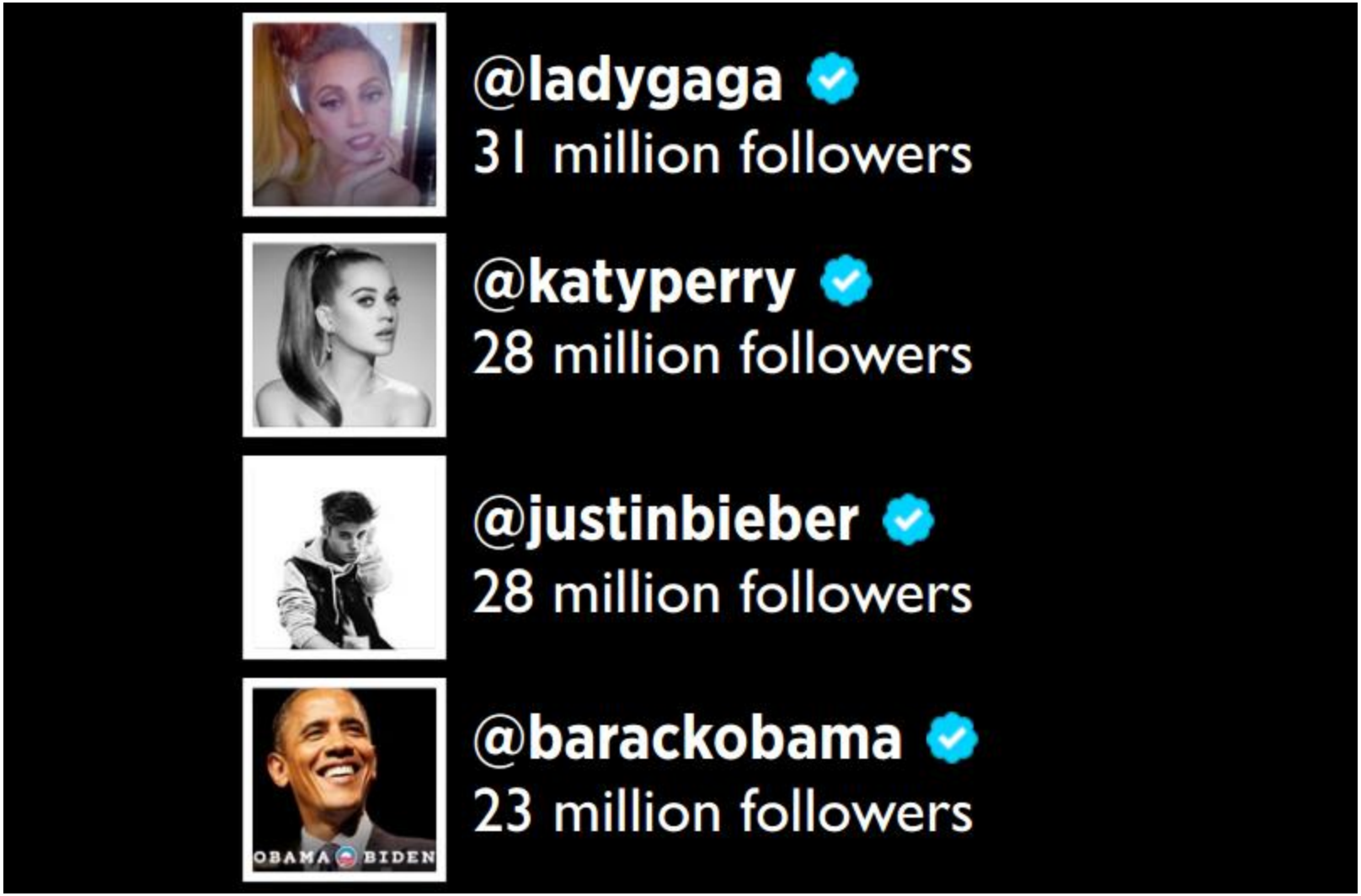


Supporting timelines: Write

- mean: 4,000 tweets/second

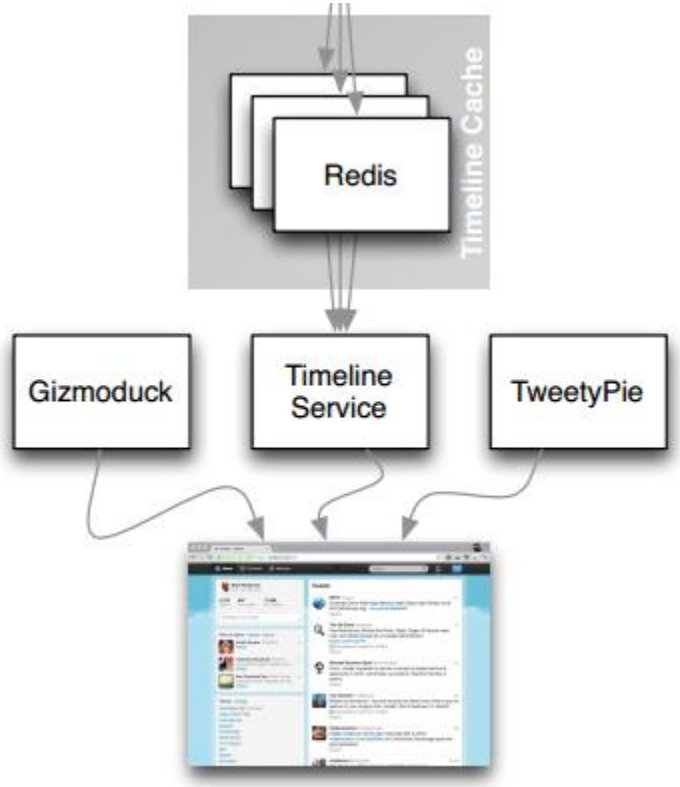


High-fanout


@ladygaga
31 million followers
@katyperry
28 million followers
@justinbieber
28 million followers
@barackobama
23 million followers
OBAMA BIDEN

Supporting timelines: Read

- 300,000 queries/second

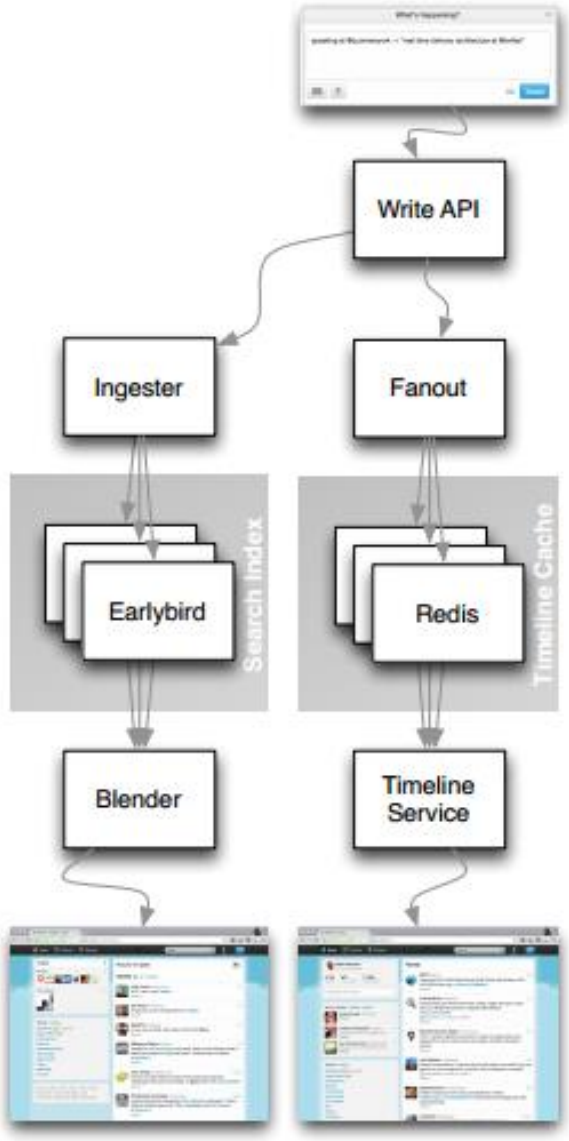


Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	Tweet ID
Tweet ID	User ID	Bits	
Tweet ID	User ID	Bits	

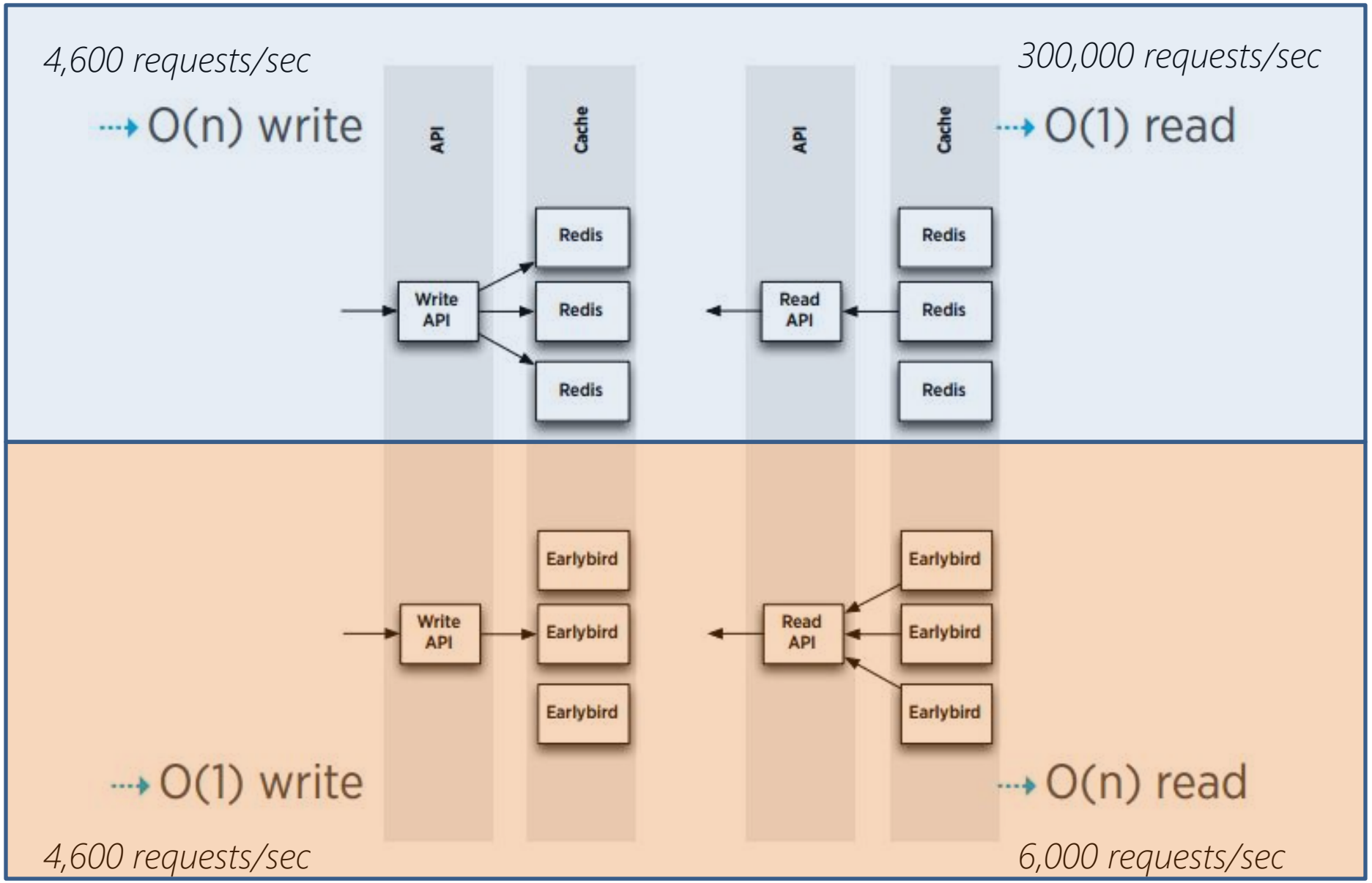
1ms @p50
4ms @p99

Supporting text search

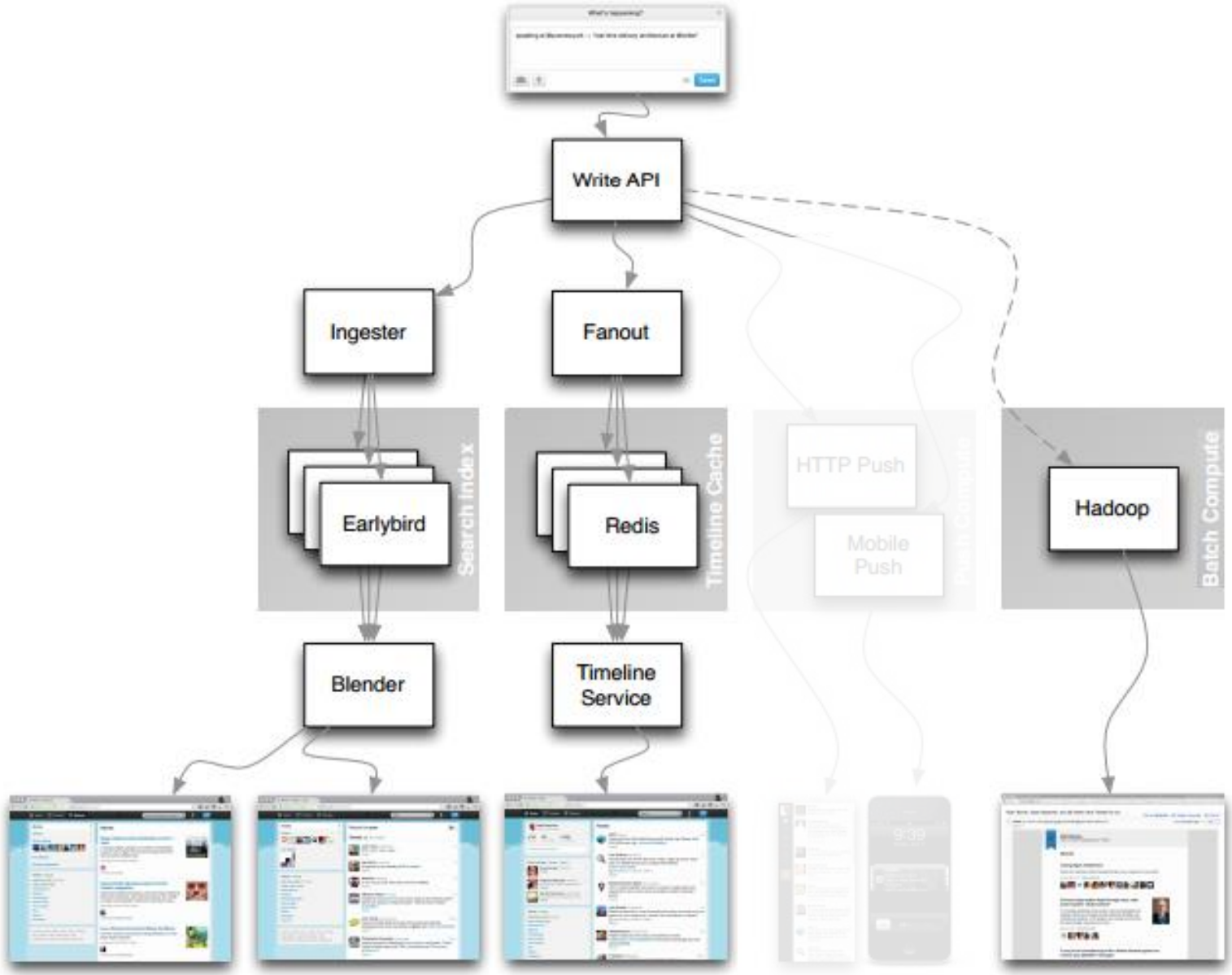
- Information retrieval
  - Earlybird: Lucene clone
  - Write once
  - Query many



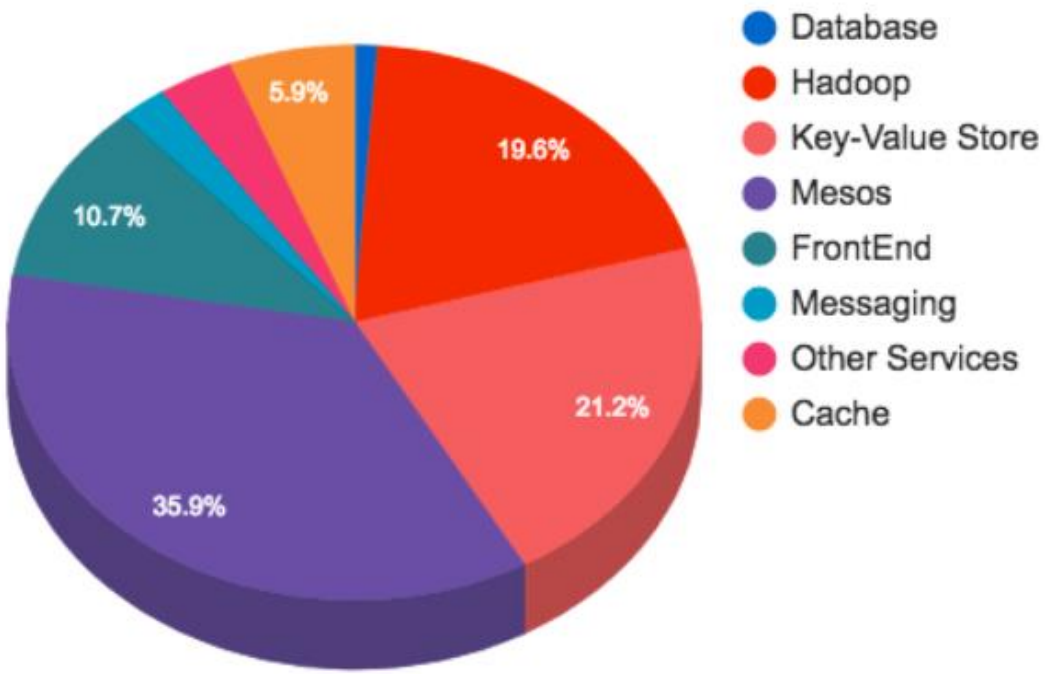
Timeline vs. Search



Twitter: Full architecture



Twitter en ~~2023~~ 2017?



https://blog.twitter.com/engineering/en_us/topics/infrastructure/2017/the-infrastructure-behind-twitter-scale.html

In-memory vs. Disk processing

Task: Count words and phrases

- Find the most frequent words and phrases in all of Wikipedia

How would you do this?

Count words/phrases

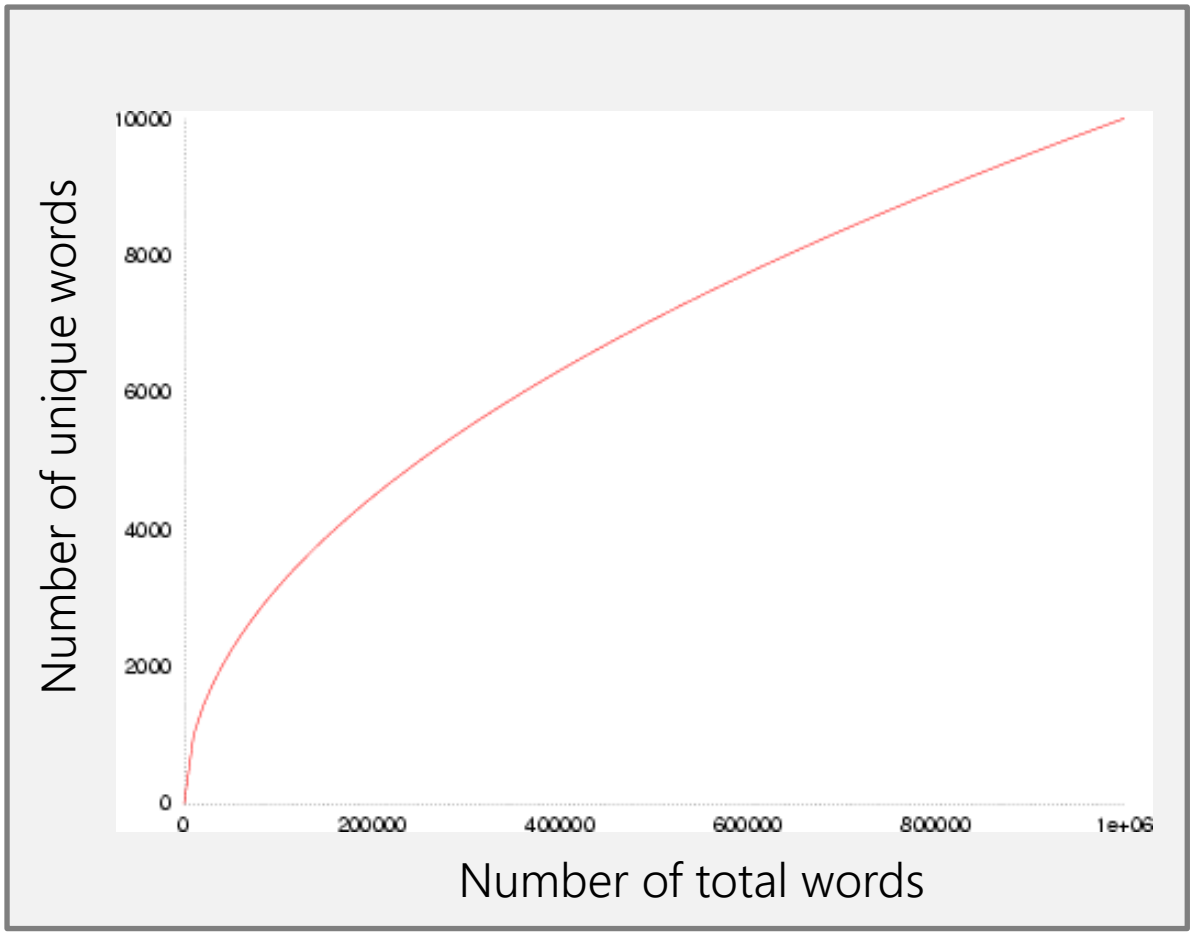
```
INPUT: file (input text file), k (print top-k most frequent)

map = new Map()

for(word w: file)
    count = map.get(w)
    if(count is null)
        map.put(w,1)
    else
        map.put(w,count+1)

list = sortByValueDescending(map)
print(list[1,k])
```

Any issues?

Count words in-memory

We can count words over very large corpora in memory

- Heap's law:



And if the phrases don’t fit in memory?

What can we do then?

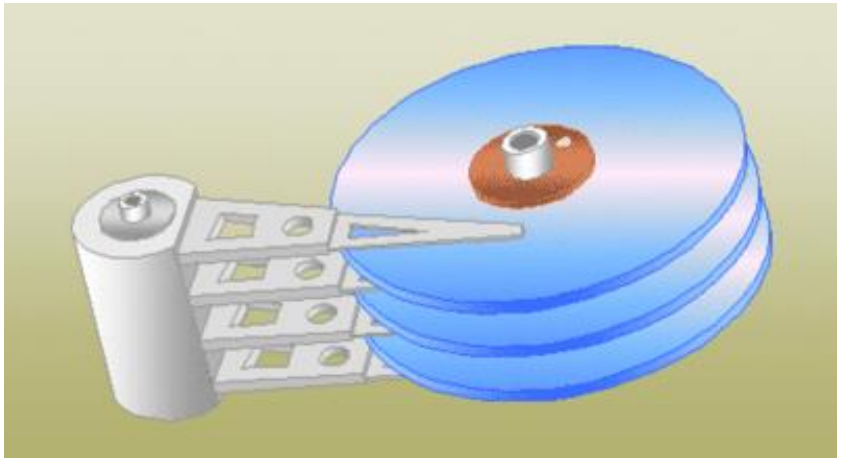
Count words/phrases on disk

```
INPUT: file (input text file), k (print top-k most frequent)

map = new Map()

for(word w: file)
    count = map.get(w)    ← random access
    if(count is null)
        map.put(w,1)    ← random access
    else
        map.put(w,count+1)    ← random access

list = sortByValueDescending(map)
print(list[1,k])
```

Can we do the same storing the map on-disk?

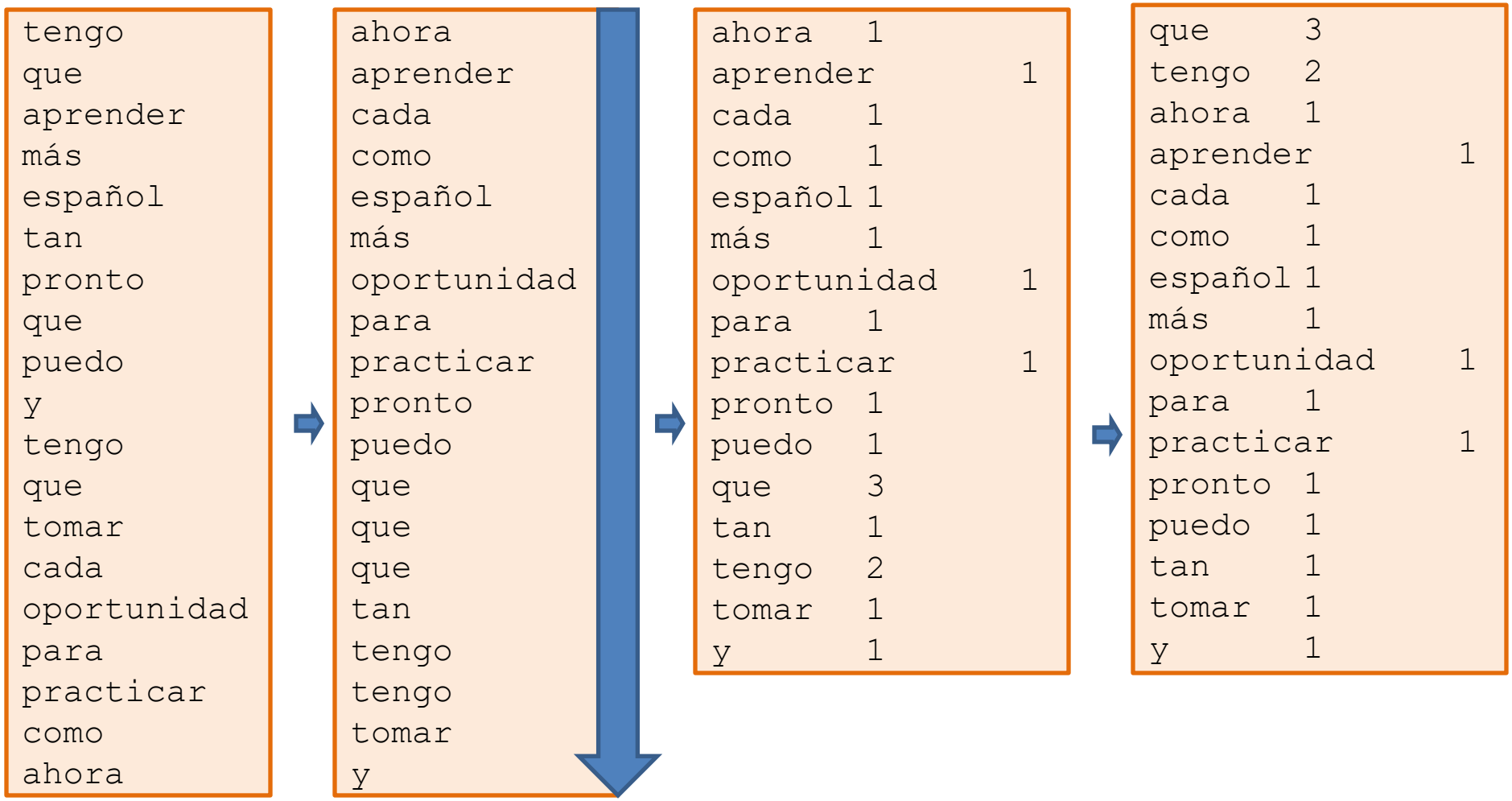
Physical disk arm

- Takes time to physically move the arm
  - Needed for random accesses
  - Around 10 ms seek time
  - About 3 hours to do one million seeks



Can we count words/phrases on-disk using more sequential access and less random access?

Order the words



How can we order the words on disk?

External Sort 1: Sort Batches

- Order the data in batches

Input (disk)
(Size: *n*)

es que
de la
por qué
de la

ha sido
no hay
de la
y de

ha sido
de la
así es
ha sido

Order (in memory)
(Batch size: *b*)

Intermediate output (disk)
(⌈*n*/*b*⌉ batches)

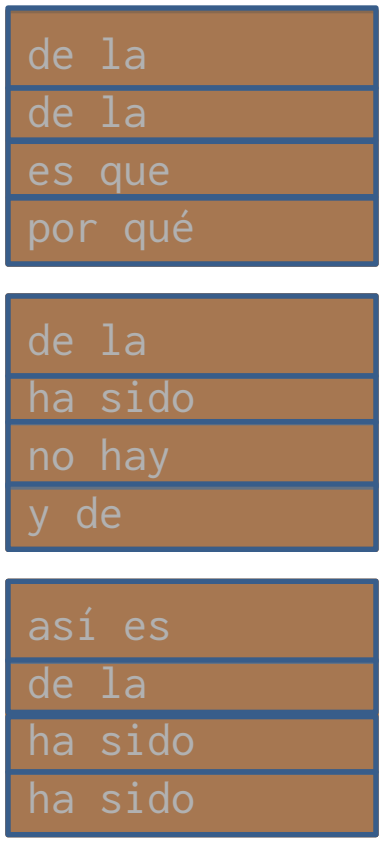
de la
de la
es que
por qué

de la
ha sido
no hay
y de

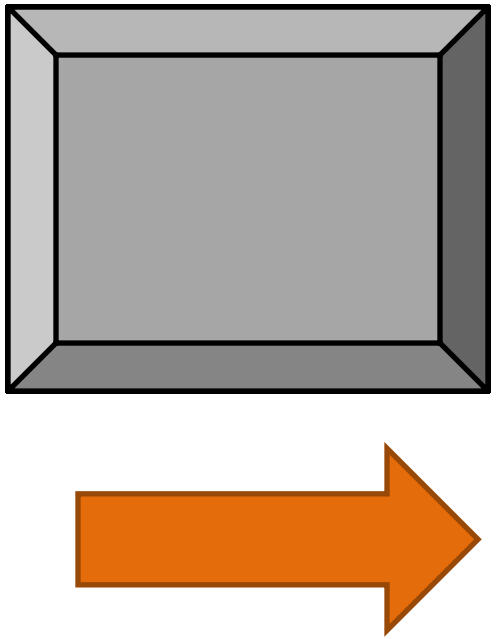
así es
de la
ha sido
ha sido

External Sort 2: Merge-sort Batches

Intermediate output (disk)
(⌈*n*/*b*⌉ batches)



Order (in memory)
(Size: ⌈*n*/*b*⌉)

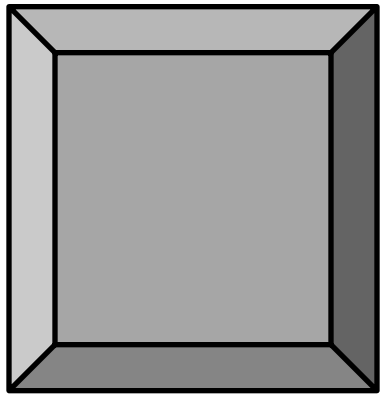


Output (disk)
(Size: *n*)

así es
de la
de la
de la
de la
es que
ha sido
ha sido
ha sido
no hay
por qué
y de

Count

```
así es
de la
de la
de la
de la
es que
ha sido
ha sido
ha sido
no hay
por qué
y de
```



If we want, we can order again by frequency using the same method on-disk

```
así es,    1
de la,     4
es que,    1
ha sido,   3
no hay,    1
por qué,   1
y de,      1
```

Scale further?

About the course …

What the Course Is/Is Not

- Data-intensive not compute-intensive
- Distributed tasks not networking
- Commodity hardware not supercomputers
- General methods not specific algorithms
- Practical methods with a little theory

What the Course Is

- Distributed Computing [1 week]
- Distributed Processing Frameworks [5 weeks]
- Information Retrieval [2 weeks]
- Distributed Databases [3 weeks]
- Projects [1–2 weeks]

Course Structure

- Mondays: Lecture
- Wednesdays: Lab
  - You can work in groups if you attend the session
  - Otherwise you must work alone
- Fridays: Auxiliar
  - Continuation of the labs
  - Resolve questions/doubts

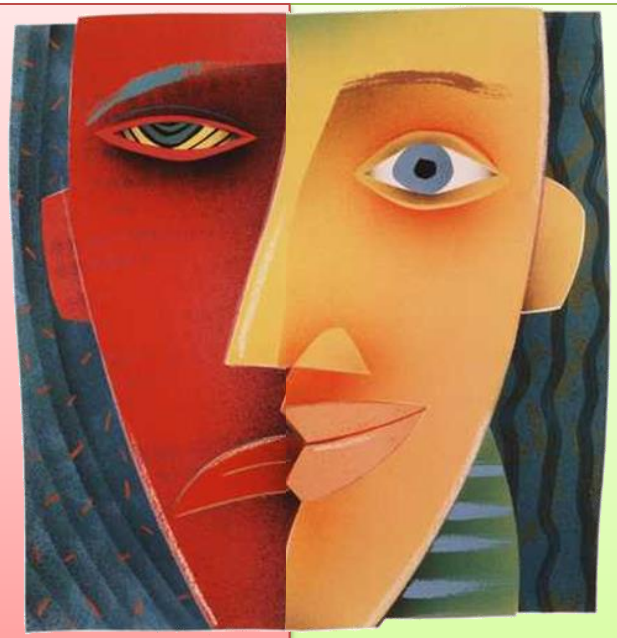
http://aidanhogan.com/teaching/cc5212-1-2023/

Course Marking

- 80% for Weekly Labs
  - 11 labs total, best 9 count
- 20% for Class Project

Assignments each week

Working in groups

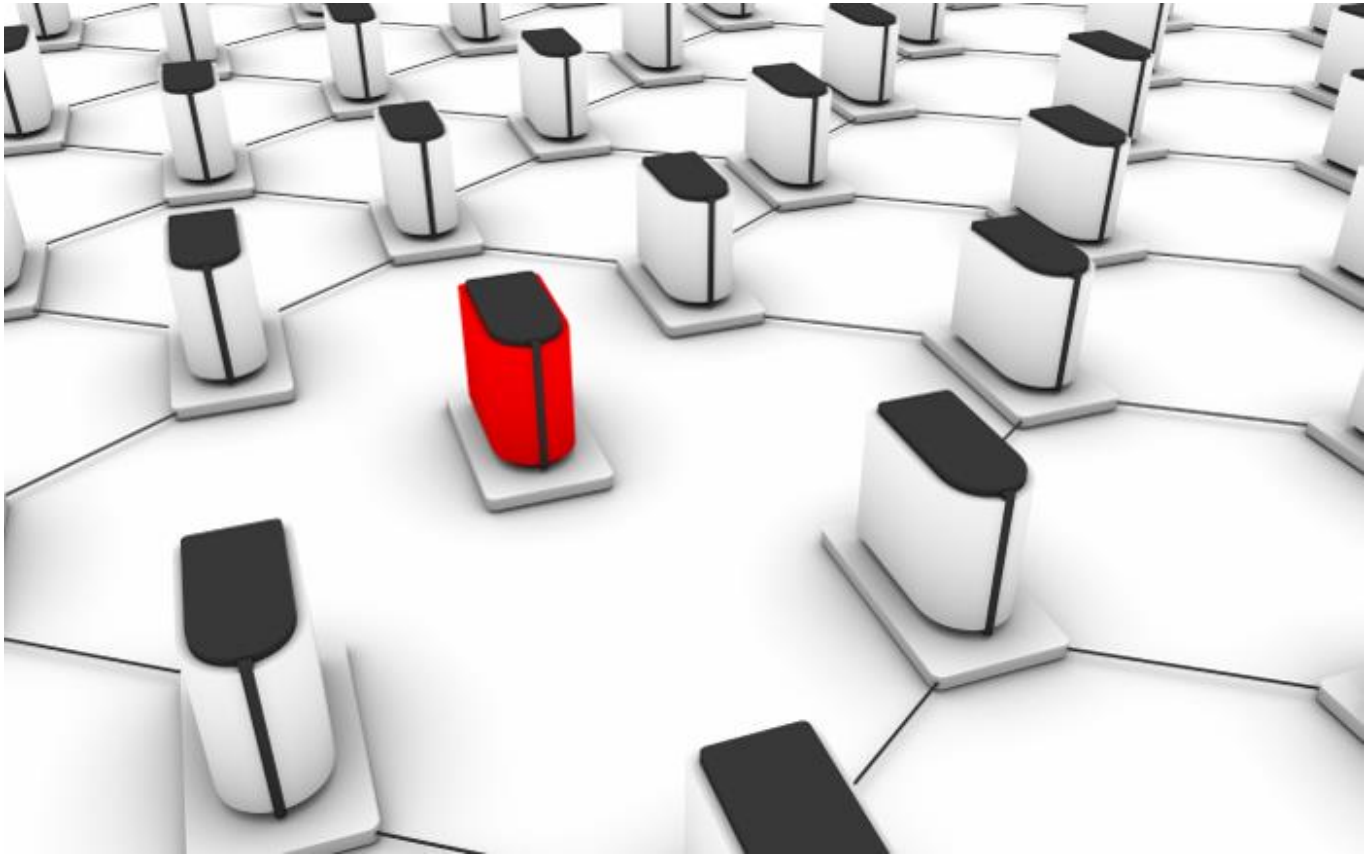


Hands-on each week!

Working in groups!

Outcomes!

Outcomes!



Outcomes!

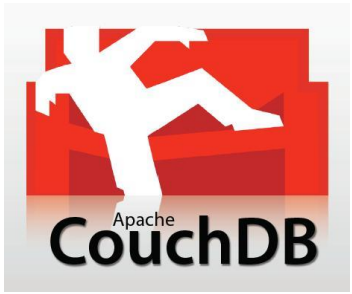

neo4j


HBASE


APACHE
Spark


cassandra


hadoop


Apache
CouchDB


Apache
Solr


mongoDB


APACHE
GIRAPH


accumulo

Outcomes!

Questions?