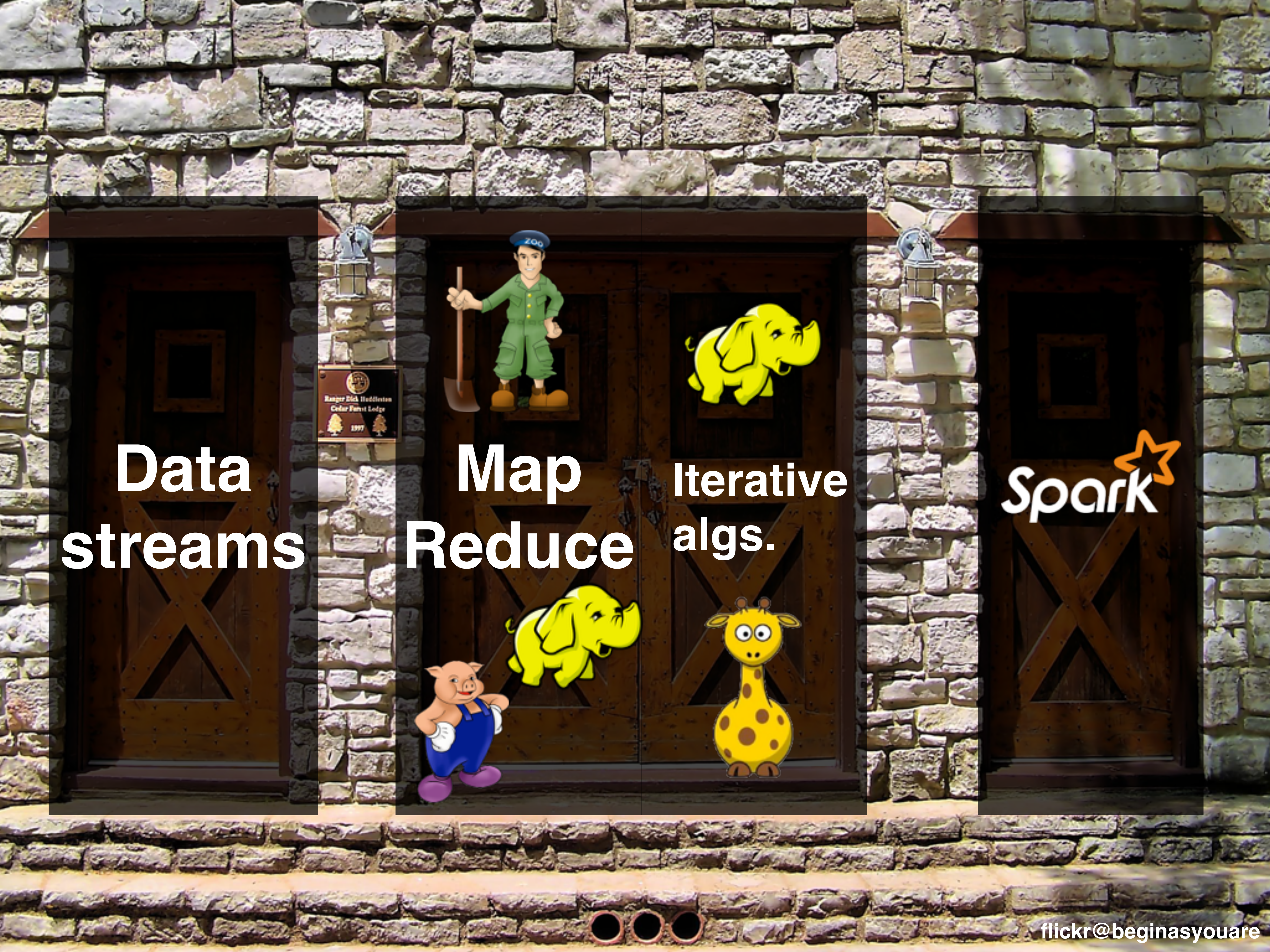
TI2736-B

Big Data Processing

Claudia Hauff

ti2736b-ewi@tudelft.nl

TUDelft



Data streams
Map Reduce
Iterative algs.
Spark
flickr@beginasyouare

Course objectives

- **Explain** the ideas behind the “big data buzz”
- **Understand** and **describe** the four different paradigms covered in class
- **Code productively** in one of the most important big data software frameworks we have to date: Hadoop (and tools building on it)
- **Transform** big data problems into **sensible** algorithmic solutions

Final exam

75%

Nothing is mandatory! To pass: **overall grade >=5.75**

Assignments

25%

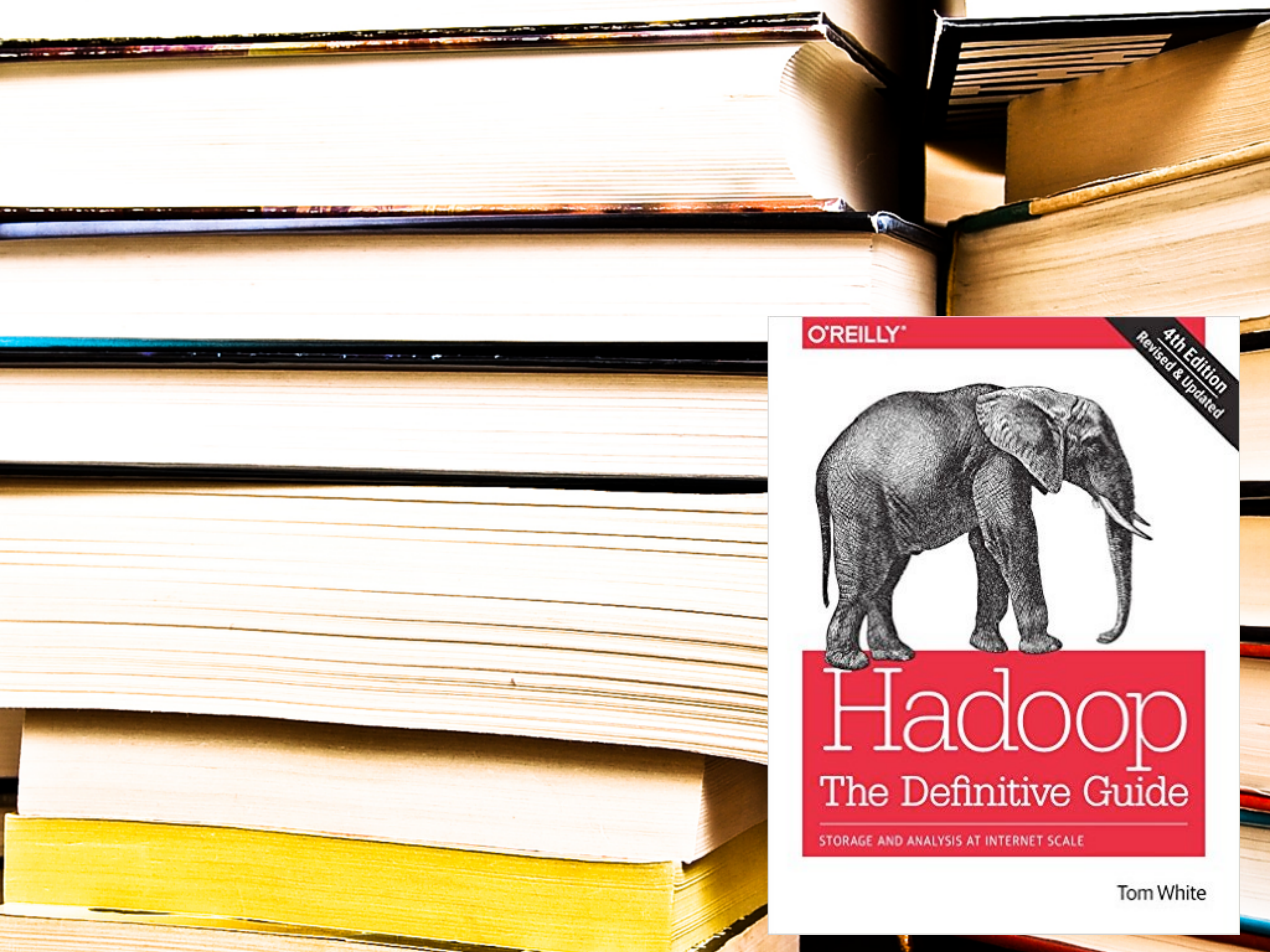
Quizzes

+1

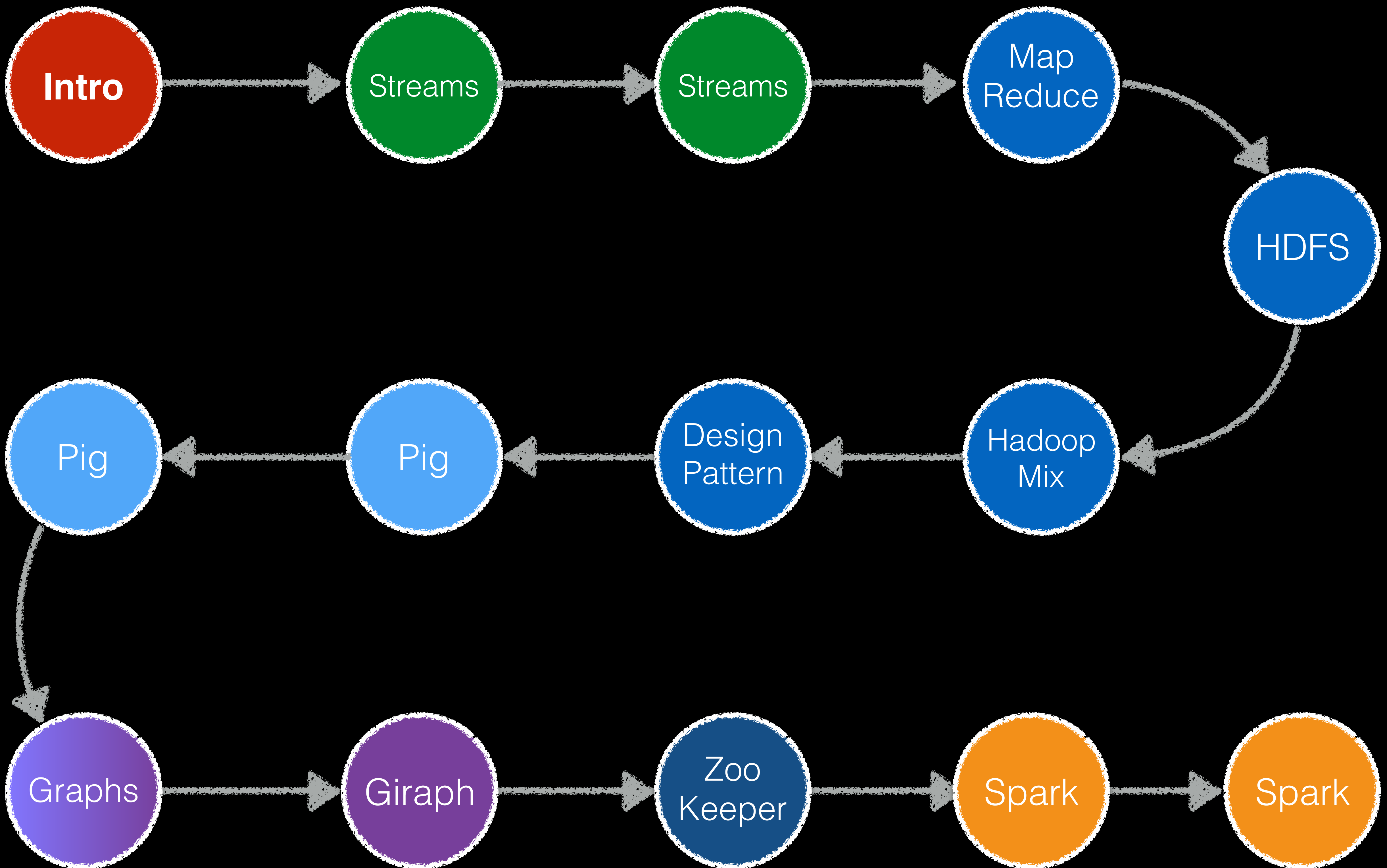
flickr@jakerust

If you need help with your assignments, attend the lab sessions (week 2+).

flickr@gottgraphicsdesign


O'REILLY
4th Edition
Revised & Updated
Hadoop
The Definitive Guide
STORAGE AND ANALYSIS AT INTERNET SCALE
Tom White

Big Data


Intro
Streams
Streams
Map Reduce
HDFS
Hadoop Mix
Design Pattern
Pig
Pig
Graphs
Giraph
Zoo Keeper
Spark
Spark

Today’s learning objectives

- **Explain** and **recognise** the V’s of big data
- **Explain** the main differences between data streaming and MapReduce algorithms
- **Identify** the correct approach (streaming vs. MapReduce) to be taken in an application setting

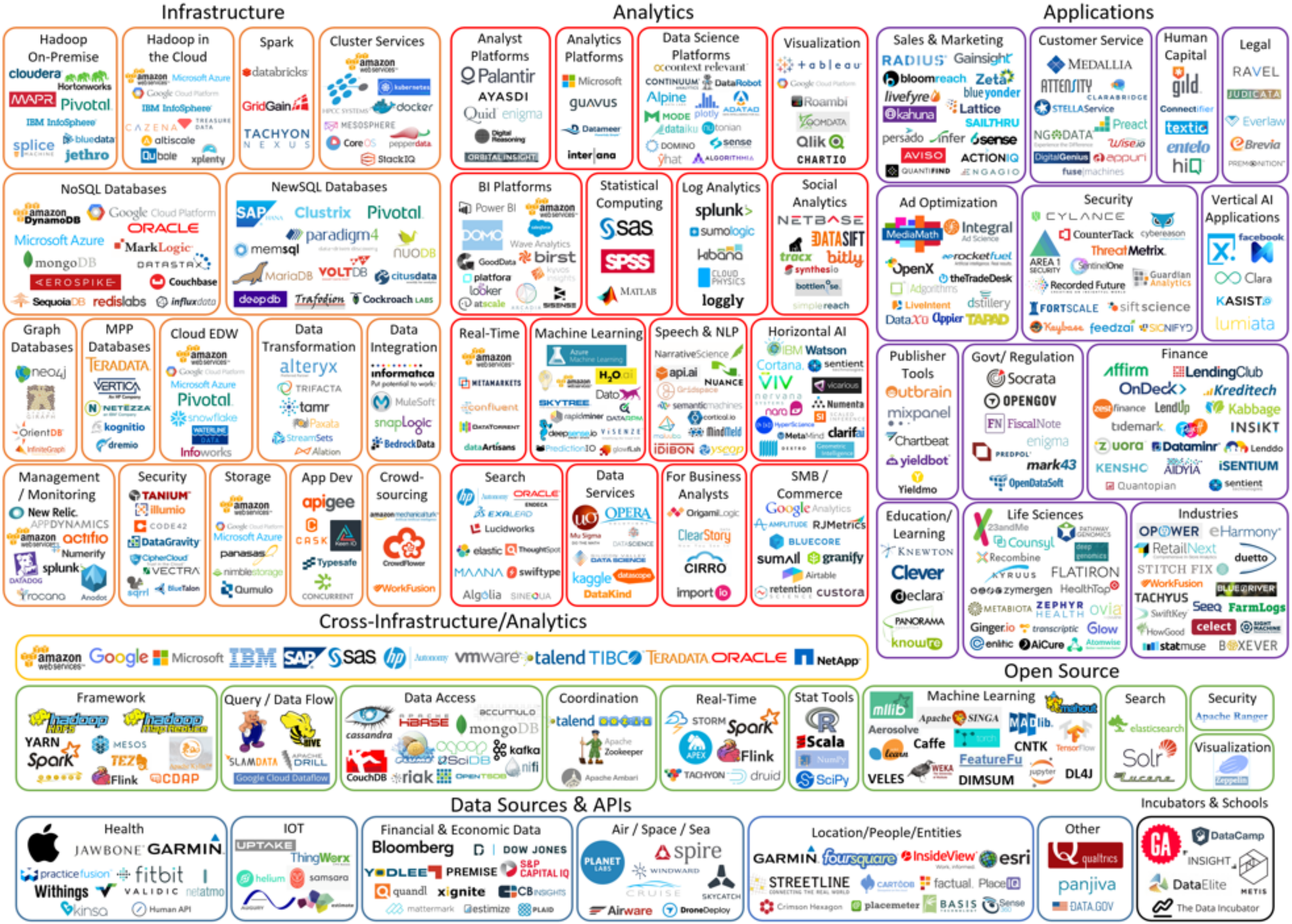
What is “big data”?

- A buzz word, fuzzy boundaries

“**Massive** amounts of **diverse**, **unstructured** data produced by **high-performance** applications.”

“Data **too large & complex** to be effectively handled by standard database technologies currently found in most organisations.”

Big Data Landscape 2016



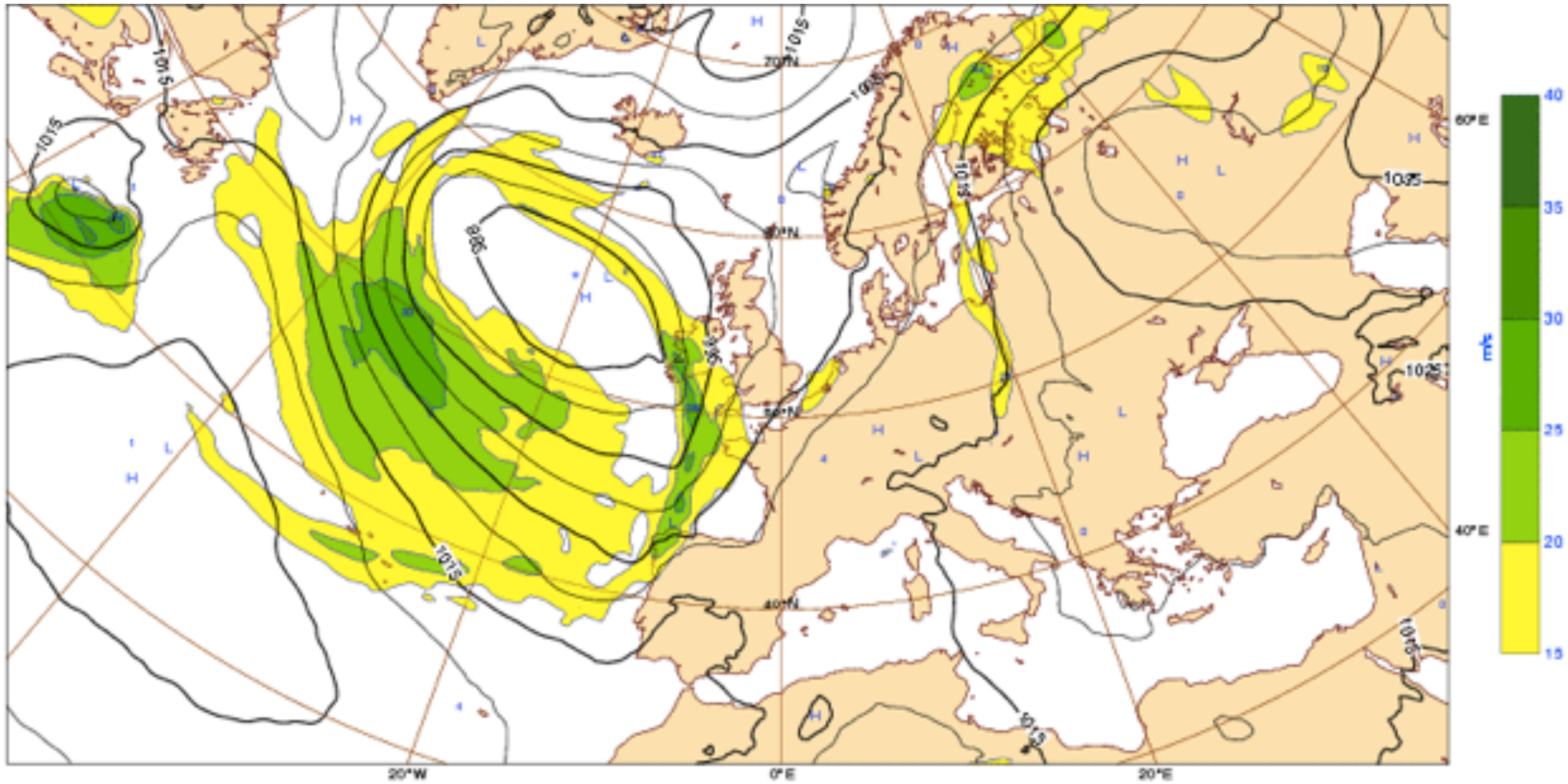
© Matt Turck (@mattturck), Jim Hao (@jimrhao), & FirstMark Capital (@firstmarkcap)

FIRSTMARK

Large-scale computing is not new

Weather forecasting has been a long-term scientific challenge

- Supercomputers were already used in the 1970s
- Equation crunching

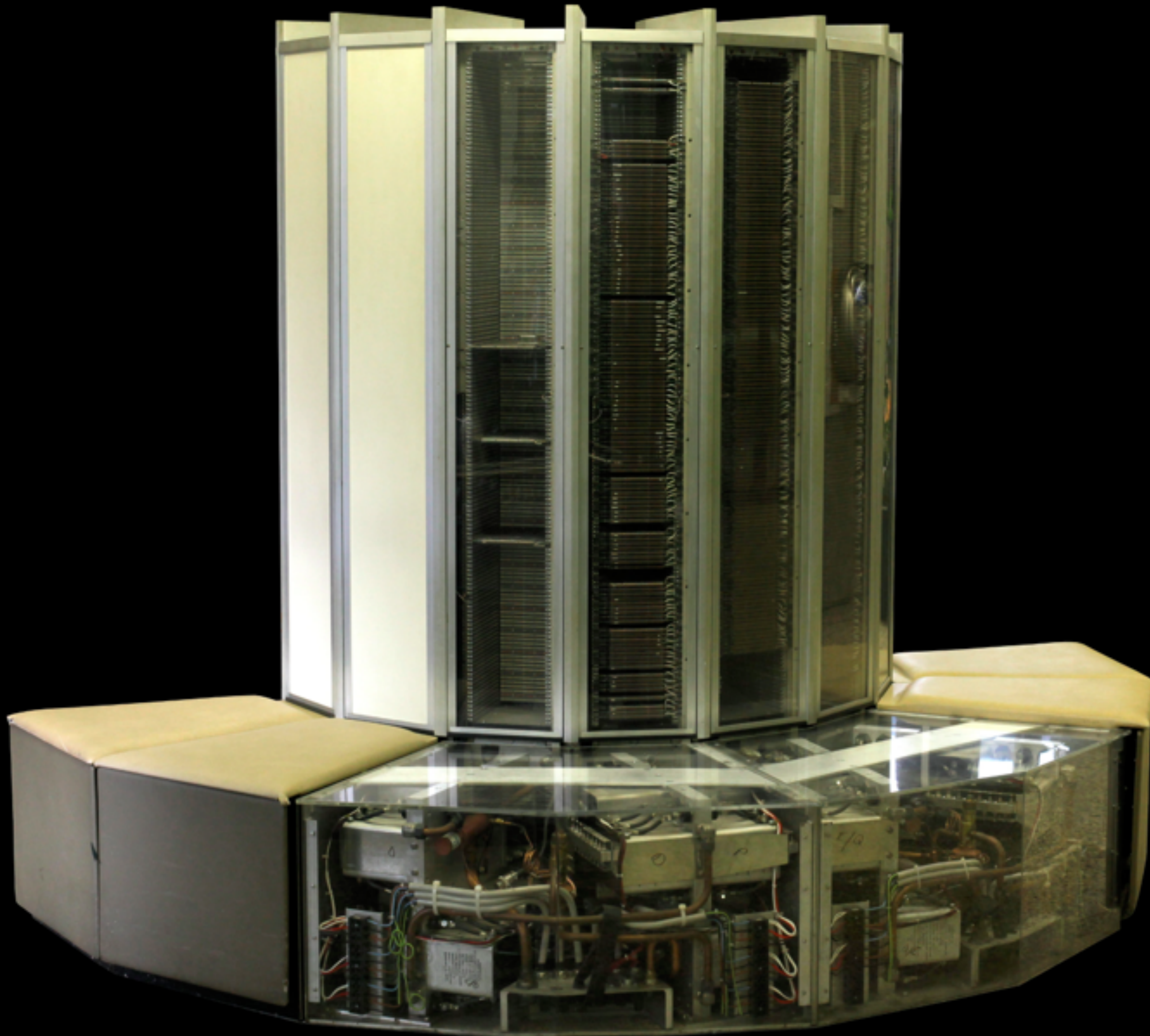


Large-scale computing is not new

Weather forecasting has been a long-term scientific challenge

- Supercomputers were already used in the 1970s
- Equation crunching

Specification	Cray-1A	IBM POWER7 System	Approx Ratio
Year installed	1978	2012	
Architecture	Vector processor	Dual Cluster of scalar CPUs	
Number of Cores	1	~49,000	49,000:1
Clock Speed	12.5 nsec (80 MHz)	0.26 nsec (3.83 GHz)	49:1
Peak perf per Core	160 MFLOPS	30 GFLOPS	190:1
Peak perf per system	160 MFLOPS	~1.5PFLOPS	9,200,000:1
Sustained performance	~50 MFLOPS	~70 TFLOPS	1,400,000:1
Memory	8 MiBytes	~106 TiBytes	13,900,000:1
Disk Space	2.5 GBytes	~3.1 PBytes	1,250,000:1



Cray-1A

Image: Wikipedia

Large-scale computing is not new

Weather forecasting has been a long-term scientific challenge

- Supercomputers were already used in the 1970s
- Equation crunching

Specification	Cray-1A	IBM POWER7 System	Approx Ratio
Year installed		2012	
Architecture		Dual Cluster of scalar CPUs	
Number of Cores		~49,000	49,000:1
Clock Speed		0.26 nsec (3.83 GHz)	49:1
Peak perf per Core		30 GFLOPS	190:1
Peak perf per system		~1.5PFLOPS	9,200,000:1
Sustained performance		~70 TFLOPS	1,400,000:1
Memory		~106 TiBytes	13,900,000:1
Disk Space		~3.1 PBytes	1,250,000:1

IBM Power7

Image: IBM

Big data processing

- So-called big data technologies are about **discovering patterns** (in semi/unstructured data)

- Main focus is on **how** to make computations on big data **feasible** without a supercomputer
  - Cluster(s) of **commodity hardware**

- Q3: Data Mining course focuses on **how to discover** those patterns

Just an academic exercise?

- **Cloud computing**: “Anything running inside a browser that gathers and stores user-generated content”
- **Utility computing**
  - Computing as a **metered service**
  - A “cloud user” buys any amount of computing power from a “cloud provider” (**pay-per-use**)
  - Virtual machine instances
  - IaaS: **infrastructure as a service**
  - **Amazon Web Services** is the dominant provider

Just an academic exercise?

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
GPUs p2.16xlarge	64	188	732	EBS Only	$14.4 per Hour
t2.micro	1	Variable	1	EBS Only	$0.013 per Hour
t2.small	1	Variable	2	EBS Only	$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	$0.052 per Hour
t2.large	2	Variable	8	EBS Only	$0.104 per Hour
m4.large	2	6.5	8	EBS Only	$0.12 per Hour
m4.xlarge	4	13	16	EBS Only	$0.239 per Hour
m4.2xlarge	8	26	32	EBS Only	$0.479 per Hour

You can run your own big data experiments!

Progress often driven by industry

- Development of **big data standards** & (open source) software commonly driven by companies such as Google, Facebook, Twitter, Yahoo! …

- Why do they care about big data?

More data → **More knowledge** → **More money**

- More knowledge leads to:
  - better customer engagement
  - fraud prevention
  - new products

4 minutes

Big data analytics: IBM pitch

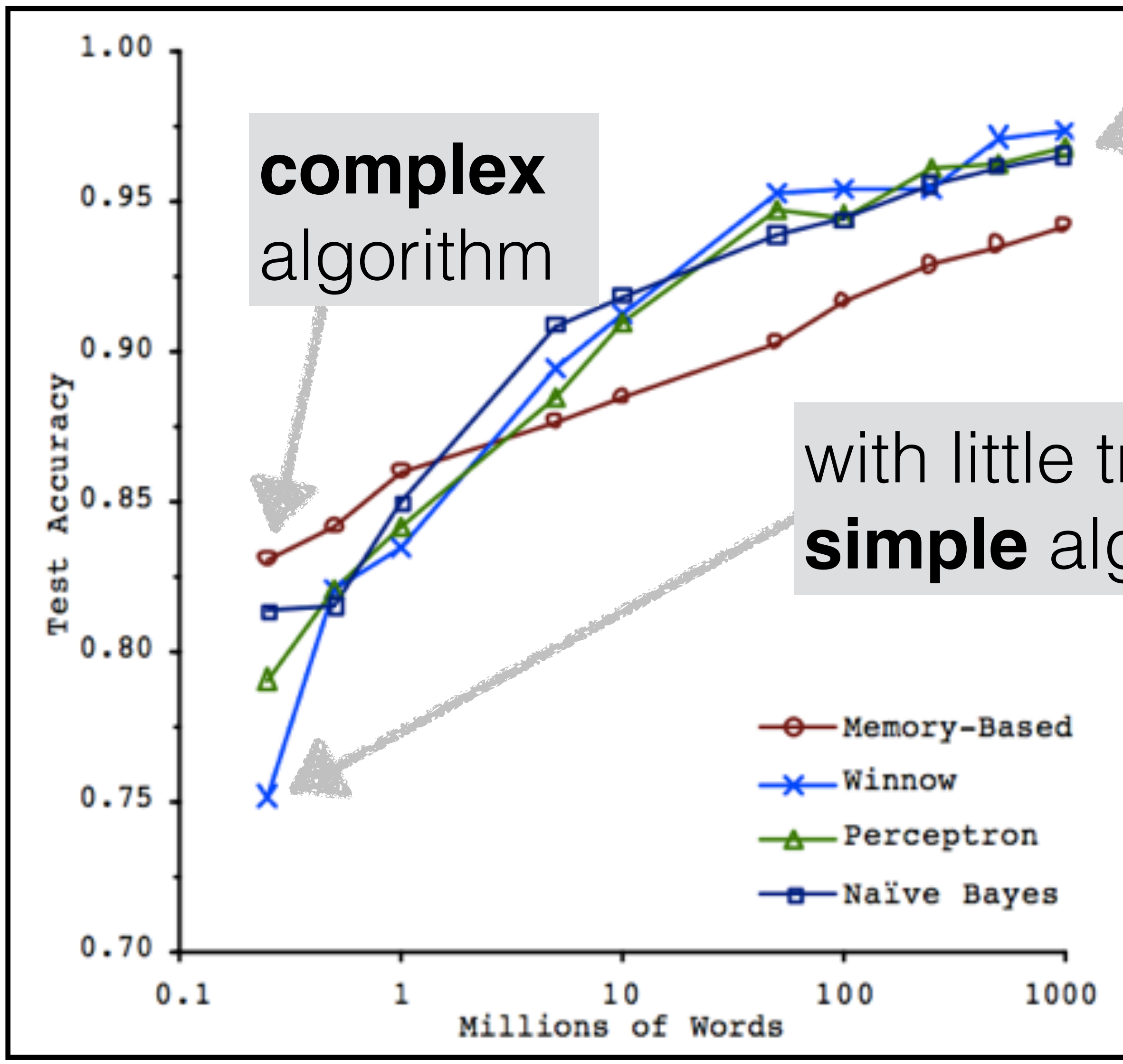
https://www.youtube.com/watch?v=1RYKgj-QK4I

Big data vs. small data

perfect

quality

poor



complex algorithm

with little training **simple** algorithms fail

but with **a lot of data** …

little ← training data → **a lot**

Task: *confusion set disambiguation*

then vs. **than**

to vs. **two** vs. **too**

Scaling to very very large corpora for natural language disambiguation.
M. Banko and E. Brill, 2001.

The 5 V's

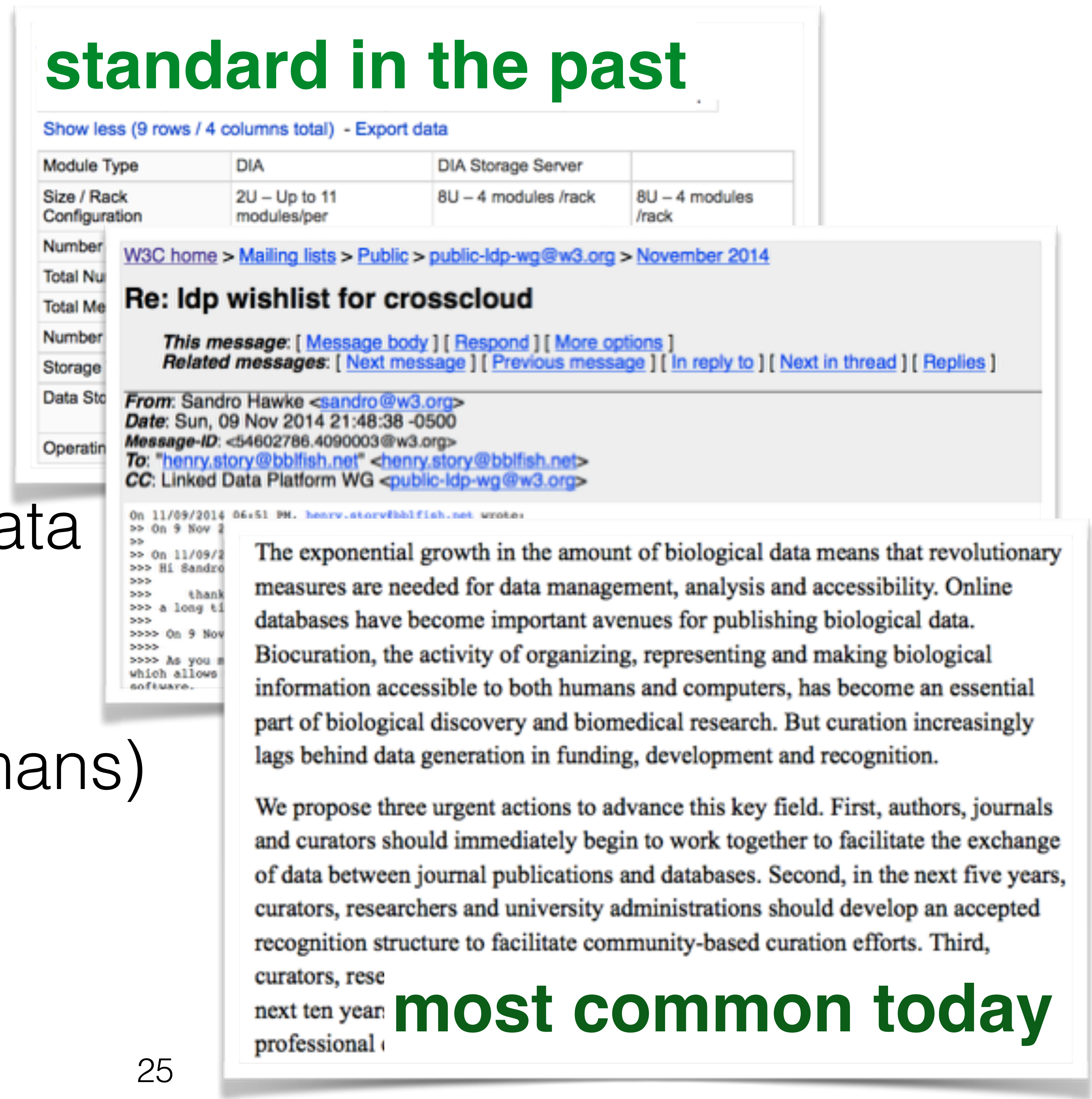
- **Volume**: large amounts of data
- **Variety**: data comes in many different forms from diverse sources
- **Velocity**: the content is changing quickly
- **Value**: data alone is not enough; how can value be derived from it?
- **Veracity**: can we trust the data? how accurate is it?

Velocity types

- **Batch processing**: running a series of computer programs without human intervention
- **Near real-time**: brief delay between the data becoming available and it being processed
- **Real-time**: guaranteed responses between the data becoming available and it being processed

Variety types

- **Structured** data
(well defined fields)
- **Semi-structured** data
- **Unstructured** data
(by humans for humans)

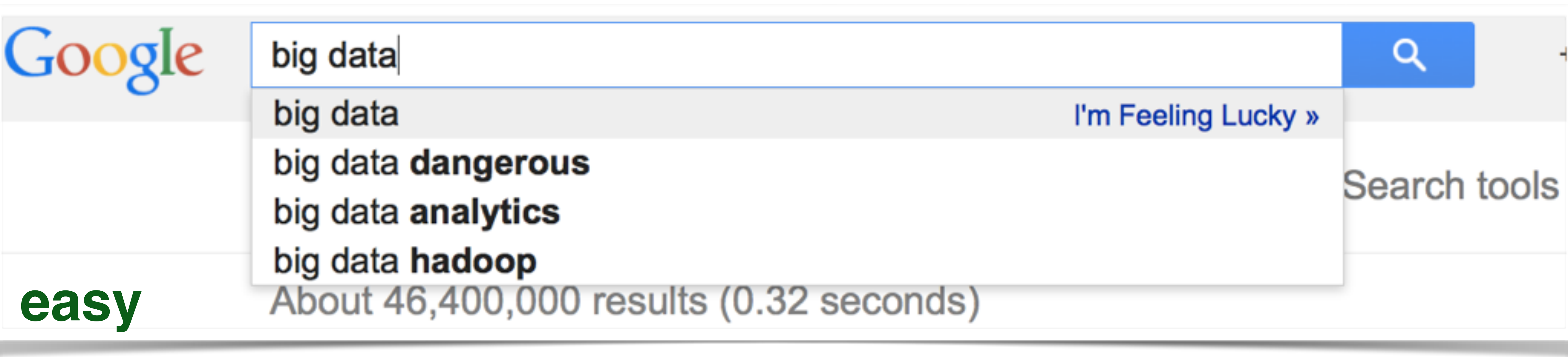


Unstructured text

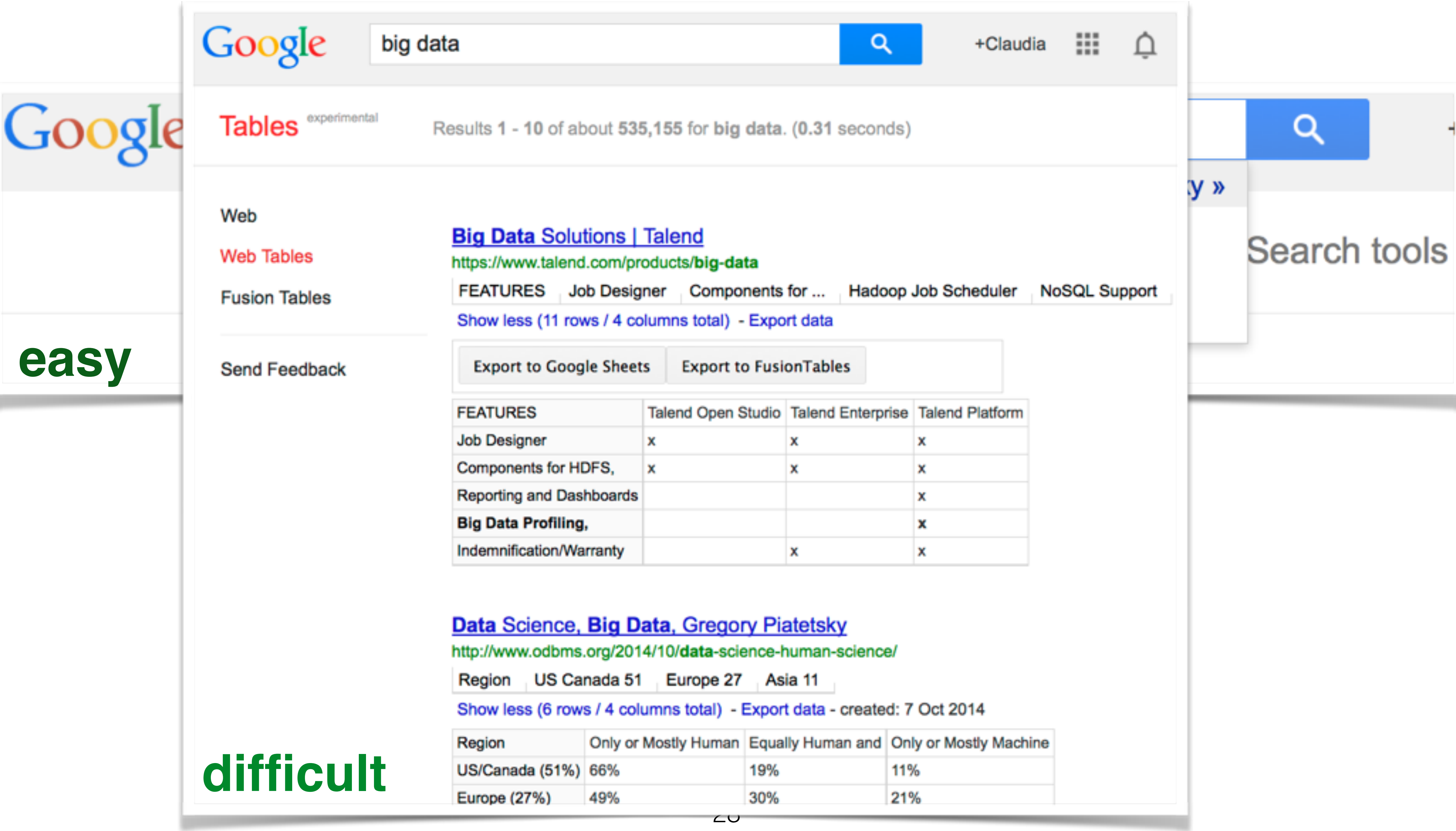
- To get value out of unstructured text we need to impose structure **automatically**
  - Parse text
  - **Extract meaning** from it (can be easy or difficult)

- Amount of data we create is more than doubling every two years, most new data is unstructured or at most semi-structured

Extracting meaning



Extracting meaning

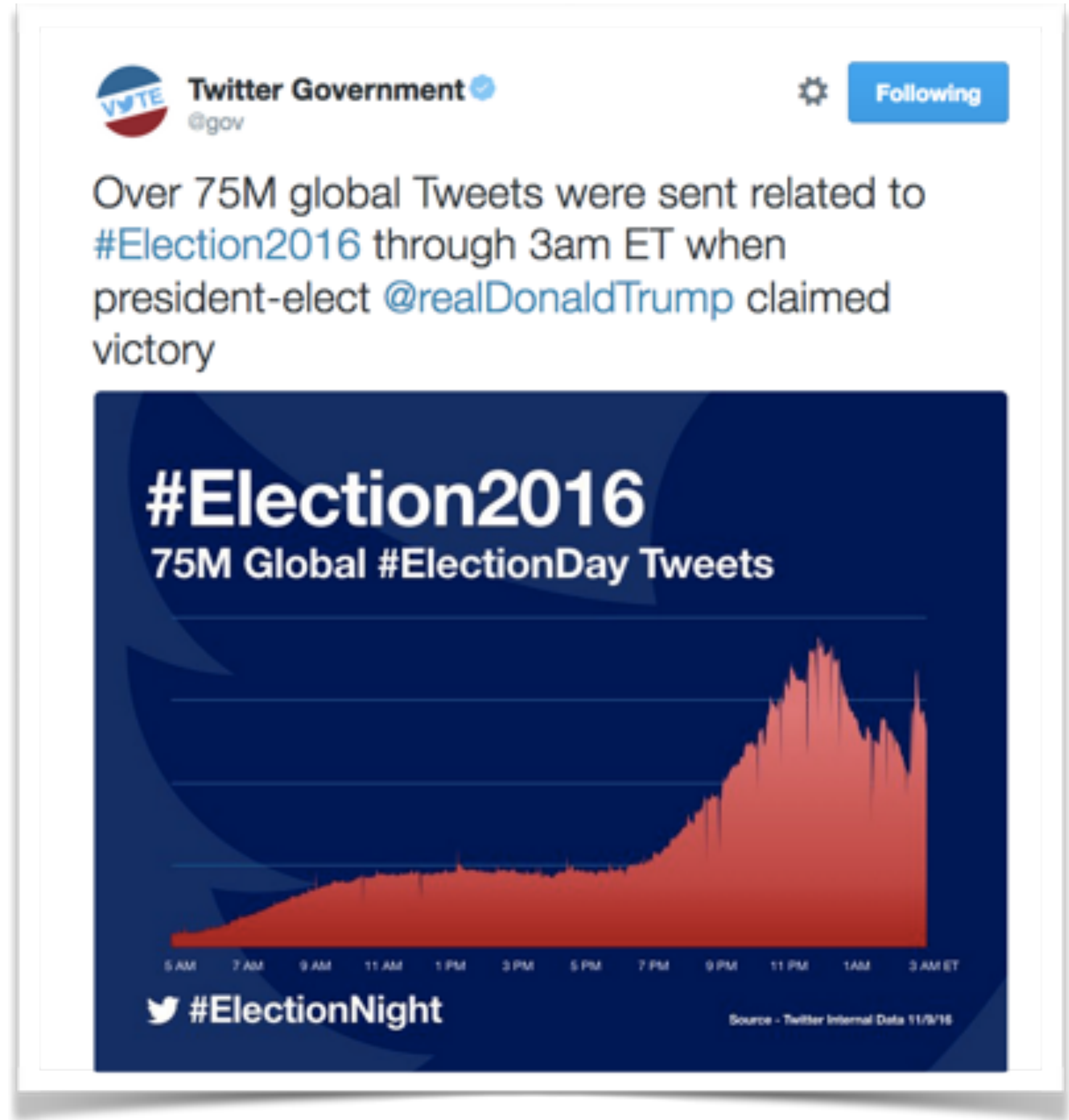


easy

difficult

Examples of Volume & Velocity: Twitter

- **>500 million** tweets a day
- **>300 million** active users / day
- On average >6000 tweets a second
- Peaks of >100,000 tweets a second
  - Super Bowl
  - US election (**75M** tweets in 24h)
  - New Year's Eve
  - Football World Cup (**672M** tweets in total #WorldCup)



- Messages are instantly accessible for **search**
- Messages are used in **post-hoc** analyses to gather insights

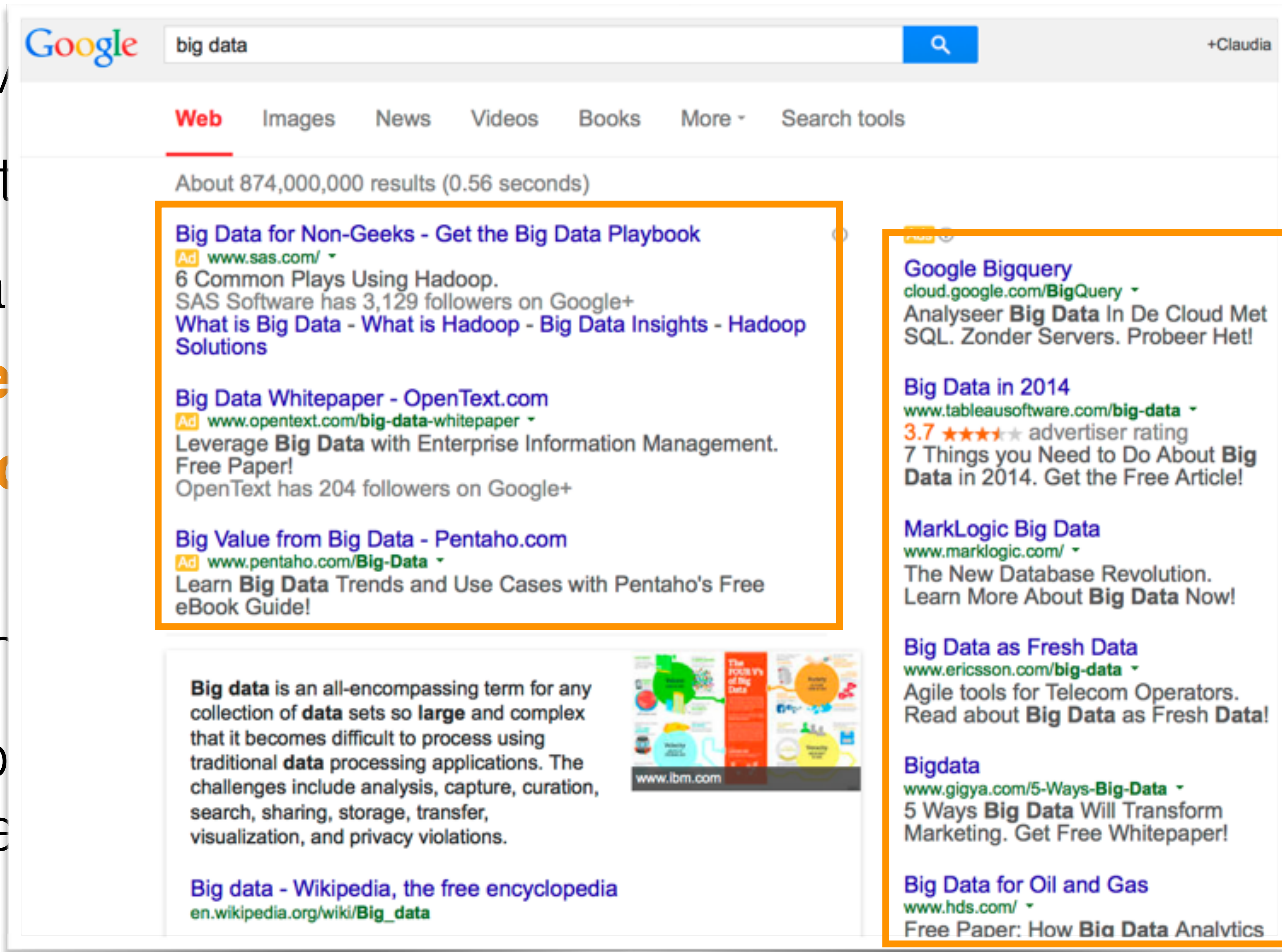

WORLDWIDE LHC COMPUTING GRID (WLCG)
WORLD WIDE WEB SERVERS
ENGINEERING SERVICES
70

Examples of Velocity: targeted advertising on the Web

- US revenues in 2013: ~$40 billion
- Advertisers usually get paid per click
- For each search request, search engines decide
  - **whether** to show an ad
  - **which** ad to show
- Users willing at best to wait **2 seconds** for their search results
- Feedback loop via user clicks, user searches, mouse movements, etc.

Examples of Velocity: targeted advertising on the Web

- US rev
- Advert
- For ea
  - **whe**
  - **whi**
- Users search
- Feedb mouse



More interactions/data on the Web

- **YouTube**: 4 billion views a day, one hour of video uploaded every second
- **Facebook**: 483 million daily active users (*Dec. 2011*); 300 Petabytes of data
- **Google**: >1 billion searches per day (*March 2011*)
- **Google** processed 100 Terabytes of data per day in *2004* and 20 Petabytes data/day in *2008*
- **Internet Archive**: contains 2 Petabytes of data, grows 20 Terabytes per month (*2011*)

Movie recommendations

Movie recommendations contd.

- **Ignore the data**, use **experts** instead (movie reviewers); assumes no large subscriber/reviewer divergence
- Use all data but **ignore individual preferences**; assumes that most users are close to the average
- Lump people into **preference groups** based on shared likes/dislikes; compute group-based average score per movie
- Focus computational effort on **difficult** movies

The research field of **recommender systems** is concerned with this issue.

Movie recommendations contd.

- **Netflix Prize**: open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings (>**100 million** ratings by ~0.5 million users for ~18,000 movies)
- First competitor to improve over Netflix's baseline **by 10%** receives $1,000,000
- Competition started in 2006, price money was paid out in 2009 (winner was **20 minutes faster** runner up)

Thousands of **(research) teams** competed.
Innovation driven by industry again!

Example of Variety: Restaurant Locator

- **Task:** Given a person's location anywhere in the world, list the top five restaurants in his immediate neighbourhood.
- **Required data**:
  - World map, list of all restaurants in the world (opening hours, GPS coordinates, menu, special offers)
  - Reviews/ratings
  - Optional: social media stream(s)
- Data is continuously **changing** (restaurants close, new ones open, data formats change, etc.)

Society can benefit too…

- Accurate **predictions of natural disasters** and diseases
- Better responses to **disaster discovery**
  - Timely & effective decisions
  - Provide resources where need the most

- Complete disease/genomics databases to enable **biomedical discoveries**
- Accurate models to support **forecasting** of ecosystem developments

Idea: earthquake warnings

- Social sensors: users (humans) that use Twitter, Facebook, Instagram, i.e. portals with **real-time** posting abilities

people in the area tweet about it

- Challenges: how to detect when a tweet is about an **actual earthquake**, **which** earthquake is it about and **where** is the centre

Idea: earthquake warnings contd.

It is possible!

- **Goal**: warning should reach people earlier than the seismic waves
- Travel times of seismic waves: 3-7km/s; arrival time of a wave 100km away: **20 seconds**
- Speed of an existing system (**Sakaki et al., 2010**):

Twitter-based

Date	Magnitude	Location	Time	E-mail sent time	#tweets within 10 min	Announce of JMA
Aug. 18	4.5	Tochigi	6:58:55	7:00:30	35	07:08
Aug. 18	3.1	Suruga-wan	19:22:48	19:23:14	17	19:28
Aug. 21	4.1	Chiba	8:51:16	8:51:35	52	8:56
Aug. 25	4.3	Uraga-oki	2:22:49	2:23:21	23	02:27
Aug. 25	3.5	Fukushima	22:21:16	22:22:29	13	22:26
Aug. 27	3.9	Wakayama	17:47:30	17:48:11	16	17:53
Aug. 27	2.8	Suruga-wan	20:26:23	20:26:45	14	20:31
Aug. 31	4.5	Fukushima	00:45:54	00:46:24	32	00:51
Sep. 2	3.3	Suruga-wan	13:04:45	13:05:04	18	13:10
Sep. 2	3.6	Bungo-suido	17:37:53	17:38:27	3	17:43

earthquake **traditional warning system**

A brief introduction to

Streaming & MapReduce

A brief introduction to

Streaming

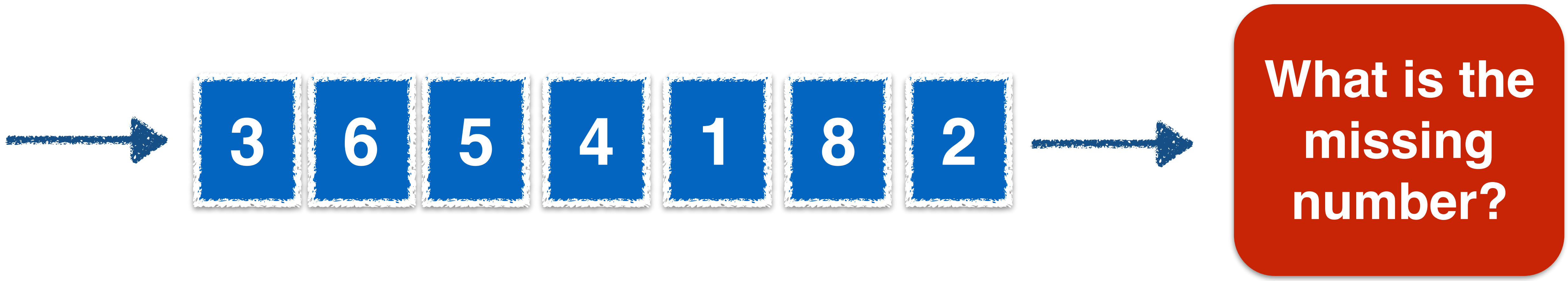
Data streaming scenario

- **Continuous** and **rapid** input of data ("stream of data")
- **Limited memory** to store the data - less than linear in the input size
- **Limited time** to process each data item - sequential access
- Algorithms have **one** (**or very few** passes) over the data

We go for the practical setup!

- Can be approached from a **practical** or **mathematical** point of view: metric embedding, pseudo-random computations …

Data streaming example



stream of n numbers; permutation from 1 to n;
one number is missing;
we are allowed **one pass** over the data

Solution 1: memorise all number seen so far;
memory requirements: n bit (impractical for large n)

Data streaming example

→ 3 6 5 4 1 8 2 →

What is the missing number?

stream of n numbers; permutation from 1 to n;
one number is missing;
we are allowed **one pass** over the data

Solution 2:
(closed form)

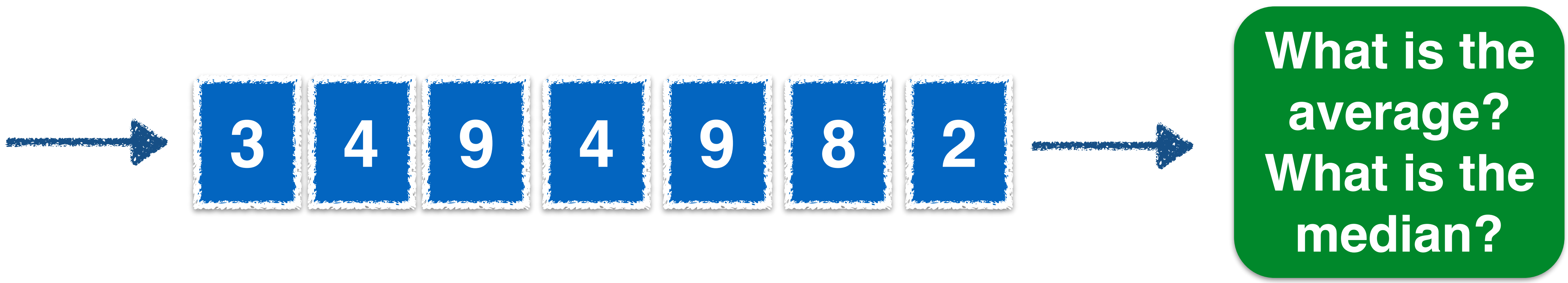
$$s = \boxed{\frac{n(n+1)}{2}} - \boxed{\sum_{j \leq i} \pi[j]}$$

subtract seen numbers

sum of all numbers from 1 to n

memory: $2logn$

Data streaming example



we are allowed **one pass** over the data, can only **store three numbers**

- **Average**: can be computed by keeping track of two numbers (sum and #numbers seen)
- **Median**: sample data points - but how?

Data streaming contd.

- Typically: **simple functions** of the stream are computed and used as input to other algorithms
  - Median
  - Number of distinct elements
  - Longest increasing sequence
  - …
- **Closed form solutions** are rare
- Common approaches are **approximations** of the true value: **sampling**, **hashing**

A brief introduction to

MapReduce

MapReduce is an industry standard

Hadoop is the open-source implementation of MapReduce framework.

Scaling Pinterest
5:20pm - 6:10pm
By: **Yash Nelapati** , **Marty Weiner**
Infrastructure Engineer, Pinterest-- Engineer, Pinterest

Pinterest grew to one of the world's largest social networks in just a few years. The first year and half was a scalability rocket ship. We had to grow the architecture hyper fast without much sleep, and had the opportunity to try lots of things, and make LOTS of mistakes before starting to get things under control.

Stop by and ask anything.

We'll give a quick overview of our architecture, some of the new systems we're building (Pinball, Frontdoor), and talk about some of the tech we use / used for databases (MySQL, MongoDB, Casandra, etc), caching (Memcache, Redis), logging (Flume, Kafka), map reduce (EMR, Qubole, Redshift), logic (Python, Java, Go, Nutcracker), load balancing / HA (haproxy, nginx, Varnish, ZooKeeper), server management (Puppet), and others. We'll keep the presentation relatively short and open the floor for any and all questions.

Data & Infrastructure at Airbnb
1:35pm - 2:25pm
By: **Brenden Matthews**
Software Engineer at Airbnb

At Airbnb, we want to change the way people travel. To accomplish that, we need to change the way we think about infrastructure. By leveraging Mesos, we'v built out our next generation of infrastructure to support several frameworks like Hadoop and Storm. Mesos paves the way for application level distributed computing, and is poised to become the chassis of distributed computing for the future.

Attendees will gain insight into building, deploying ar running a Mesos cluster with several frameworks, su as Hadoop and Storm.

Scaling AncestryDNA using Hadoop and HBase
2:50pm - 3:40pm
By: **Bill Yetman** , **Jeremy Pollack**
Senior Director of Engineering at Ancestry.com-- Senior Software Engineer, Ancestry.com

What do you get when you take Bioinformatics Scientists with PhDs and mix them up with Software Engineers? Why Ancestry DNA on Hadoop and HBase! Get the whole story from both the management (Bill Yetman, Sr. Director) and developer (Jeremy Pollack, Principle Engineer/Team Lead) points of view. Find out how this unique cast of characters took academic programs and created an industrial, scalable, DNA processing pipeline (a real Big Data problem) using Hadoop and HBase. The final implementation provided a 1700% performance improvement.

QCon 2013 (San Francisco)

"QCon empowers software development by facilitating the spread of knowledge and innovation in the developer community. A practitioner-driven conference …"

Industry is moving fast

Monday, 3 November

Architectures You've Always Wondered about
The newest and biggest Internet architectures

Real World Functional
Putting functional programming concepts to work in the real world.

The Future of Mobile
The future of mobile and performance improvements

Continuous Delivery: From Heroics to Becoming Invisible
Continuous Delivery philosophies, cultures, hiccups, and best practices.

Unleashing the Power of Streaming Data
This track explores a variety of use-cases, platforms, and techniques for processing and analyzing stream data from the companies deploying them at scale!

Tuesday, 4 November

Engineering for Product Success
Architectures that make products more successful

Reactive Service Architecture
Reactive, Responsive, Fault Tolerant and More.

Modern CS In the Real World
How modern CS tackles problems in the real world.

Applied Machine Learning and Data Science
Understand your big big data!

Deploying at Scale
Containerizing Applications, Discovering Services, and Deploying to the Grid.

Wednesday, 5 November

Beyond Hadoop
Emerging Big Data Frameworks and Technology

Scalable Microservice Architectures
This track addresses the ways companies with hundreds of fine-grained web-services (e.g. Netflix, LinkedIn) manage complexity!

Java at the Cutting Edge
The latest and greatest in the Java ecosystem

Engineering culture
Successes and failures in creating an engineering culture.

Next gen HTML5 and JS
How Web Components, the Future of CSS, and more are changing the web.

QCon 2014 (San Francisco)

Industry is moving fast

Monday Nov 16	Tuesday Nov 17	Wednesday Nov 18
Architectures You've Always Wondered About Silicon Valley to Beijing: Exploring some of the world's most intrigiuing architectures	**Containers in Practice** Build resilient, reactive systems one service at a time.	**Streaming Data @ Scale** Real-time insights at Cloud Scale & the technologies that make them happen!
Applied Machine Learning How to start using machine learning and data science in your environment today. Latest and greatest best practices.	**Architecting for Failure** Your system will fail. Take control before it takes you with it.	**Taking Java to the Next Level** Modern, lean Java. Focuses on topics that push Java beyond how you currently think about it.
Browser as a platform (Realizing HTML5) Exciting new standards like Service Workers, Push Notifications, and WebRTC are making the browser a	**Modern CS in the Real World** Real-world Industry adoption of modern CS ideas	**The Dark Side of Security** Lessons from your enemies
Modern Languages in Practice The rise of 21st century languages: Go, Rust, Swift	**The Amazing Potential of .NET Open Source** From language design in the open to Rx.NET, there is amazing potential in an Open Source .NET	**Taming Distributed Architecture** Reactive architectures, CAP, CRDTs, consensus systems in practice

QCon 2015 (San Francisco)

MapReduce

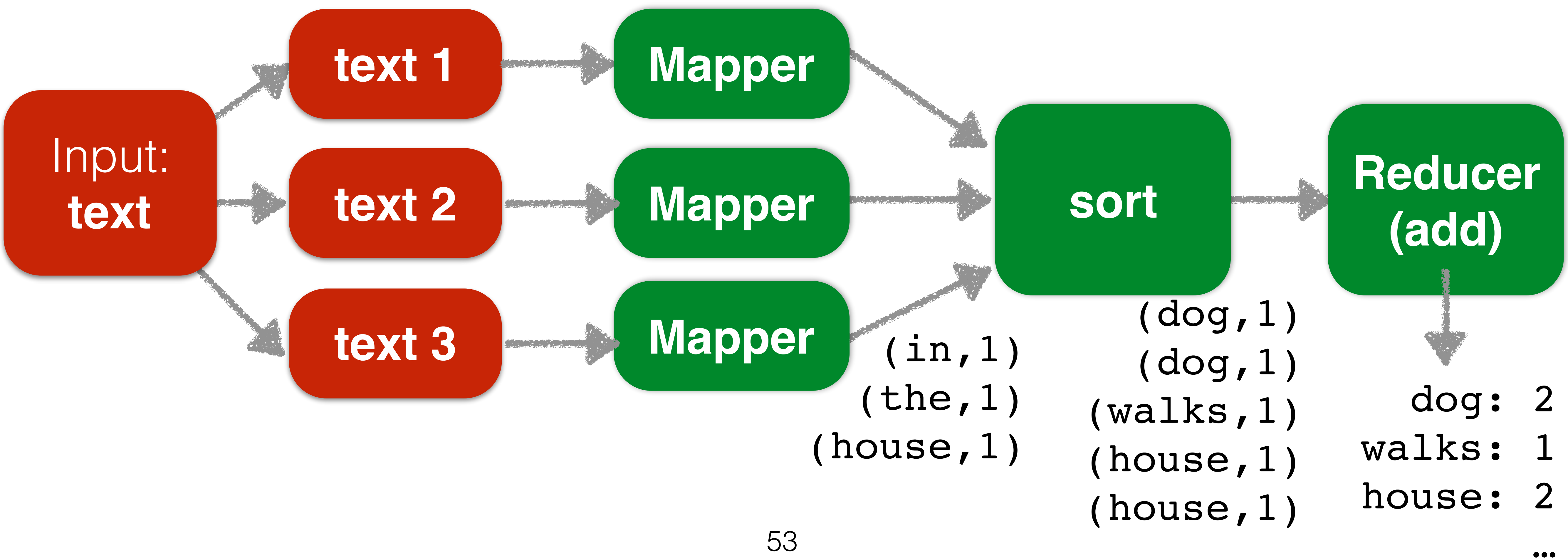
- Designed for **batch processing** over large data sets
- **No limits** on the number of passes, memory or time
- Programming model for distributed computations inspired by the **functional programming paradigm**

MapReduce example

WordCount: given an input text, determine the frequency of each word.

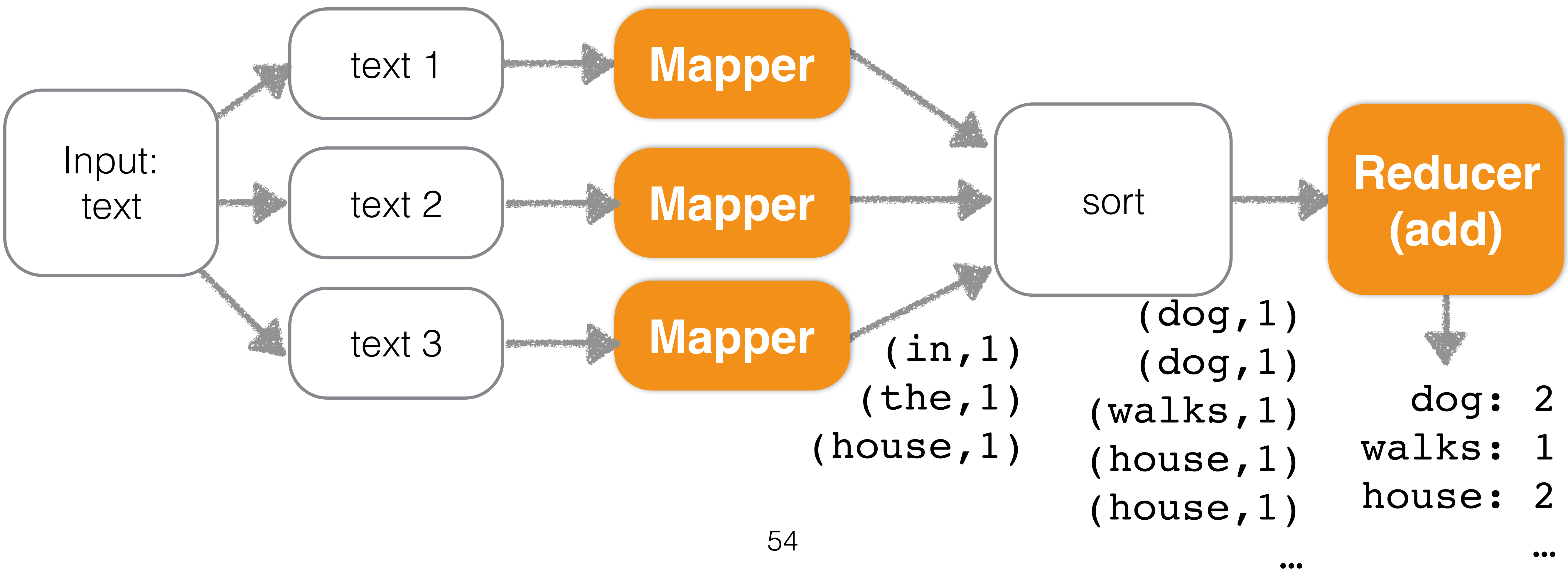
The "Hello World" of the MapReduce realm.

Input text: *The dog walks around the house. The dog is in the house.*



MapReduce example

We implement the **Mapper** and the **Reducer**.
Hadoop (and other tools) are responsible for the “rest”.



Summary

- What are the **characteristics** of “big data”?
- **Example use cases** of big data
- A brief introduction of **data streams** and **MapReduce**

Reading material

Required reading

None.

Recommended reading

Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information by Jules Berman. Chapters 1, 14 & 15.

THE END