W4121
Computer Systems for Data Science
Spring 2018

Roxana Geambasu, Sambit Sahu, Eugene Wu

https://w4121.github.io/

1

Data

2

Data
is for serious business

3

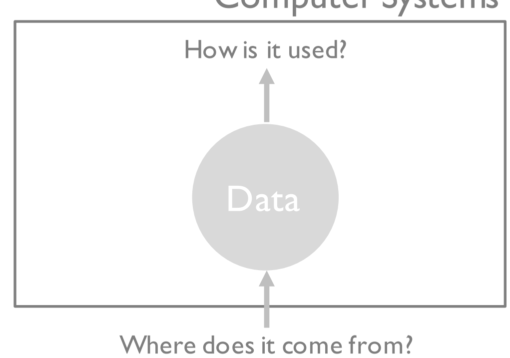
Data
is at the center of most things.

4

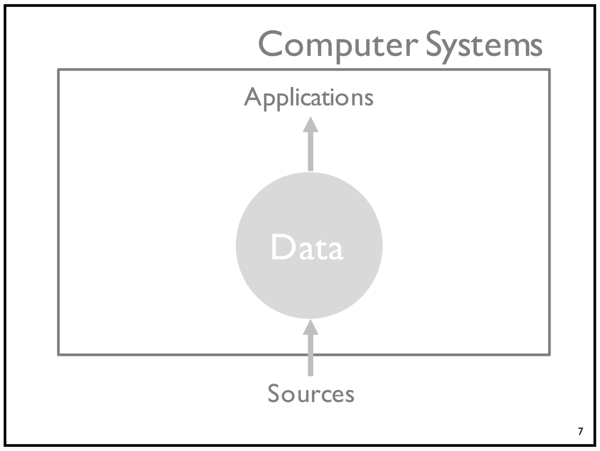
Data
is at the center of *everything*

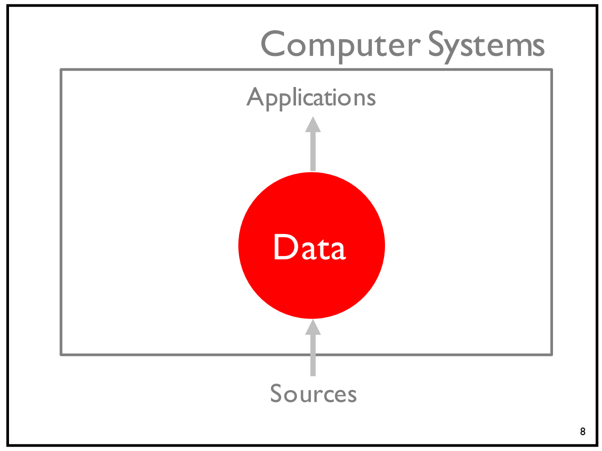
5

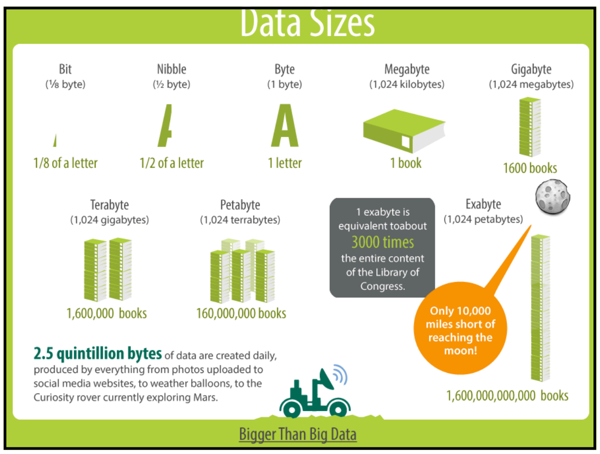
Computer Systems

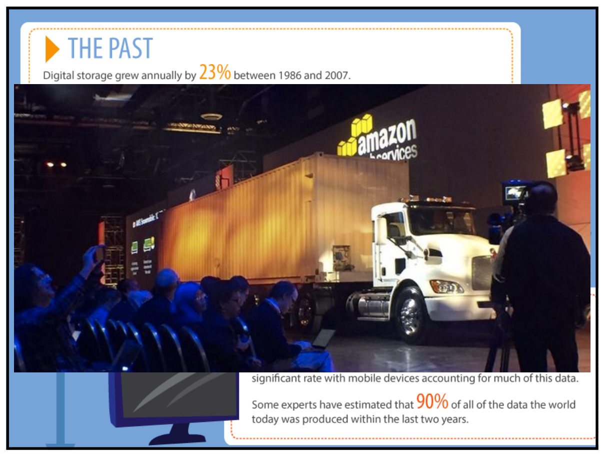


6





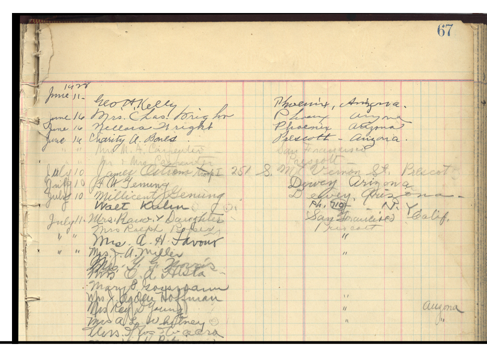




How did we get here?

11

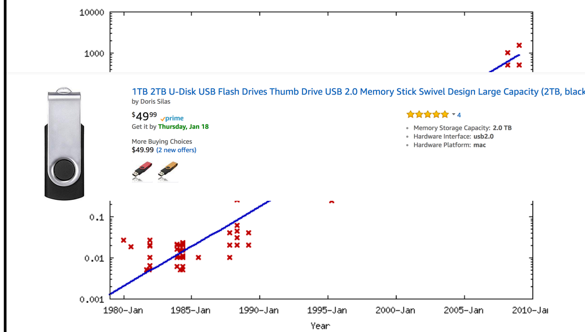
Data was *Manual*



12

Data was *Expensive*

Data is *Cheap*



Data is *Automated*

Physical devices

Data is *Automated*

Physical devices
Software logs

Data is *Ubiquitous*

Physical devices
Software logs
Phones

Data is *Ubiquitous*

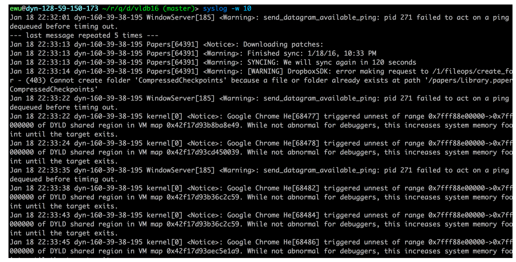
Physical devices
Software logs
Phones
GPS/Cars

Data is *Everywhere*

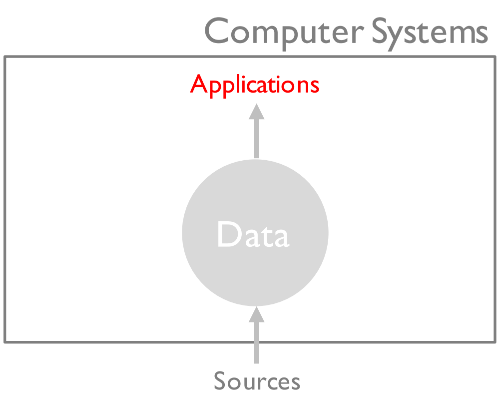
Physical devices
Software logs
Phones
GPS/Cars
Internet of *Things*

19

Data is *Temporal*


```
ewu@dyn-128-59-150-173 ~/r/q/d/vldb16 (master)> syslog -w 10
Jan 18 22:32:01 dyn-160-39-38-195 WindowServer[185] <Warning>: send_datagram_available_ping: pid 271 failed to act on a ping
dequeued before timing out.
--- last message repeated 5 times ---
Jan 18 22:33:13 dyn-160-39-38-195 Papers[64391] <Notice>: Downloading patches:
Jan 18 22:33:13 dyn-160-39-38-195 Papers[64391] <Warning>: Finished sync: 1/18/16, 10:33 PM
Jan 18 22:33:13 dyn-160-39-38-195 Papers[64391] <Warning>: SYNCING: We will sync again in 120 seconds
Jan 18 22:33:14 dyn-160-39-38-195 Papers[64391] <Warning>: [WARNING] DropboxSDK: error making request to /1/fileops/create_fo
r - (403) Cannot create folder 'CompressedCheckpoints' because a file or folder already exists at path '/papers/Library.paper
CompressedCheckpoints'
Jan 18 22:33:22 dyn-160-39-38-195 WindowServer[185] <Warning>: send_datagram_available_ping: pid 271 failed to act on a ping
dequeued before timing out.
Jan 18 22:33:22 dyn-160-39-38-195 kernel[0] <Notice>: Google Chrome He[68477] triggered unnest of range 0x7fff88e00000->0x7ff
000000 of DYLD shared region in VM map 0x42f17d93b8ba8e49. While not abnormal for debuggers, this increases system memory foo
int until the target exits.
Jan 18 22:33:24 dyn-160-39-38-195 kernel[0] <Notice>: Google Chrome He[68478] triggered unnest of range 0x7fff88e00000->0x7ff
000000 of DYLD shared region in VM map 0x42f17d93cd450039. While not abnormal for debuggers, this increases system memory foo
int until the target exits.
Jan 18 22:33:35 dyn-160-39-38-195 WindowServer[185] <Warning>: send_datagram_available_ping: pid 271 failed to act on a ping
dequeued before timing out.
Jan 18 22:33:38 dyn-160-39-38-195 kernel[0] <Notice>: Google Chrome He[68482] triggered unnest of range 0x7fff88e00000->0x7ff
000000 of DYLD shared region in VM map 0x42f17d93b36c2c59. While not abnormal for debuggers, this increases system memory foo
int until the target exits.
Jan 18 22:33:43 dyn-160-39-38-195 kernel[0] <Notice>: Google Chrome He[68484] triggered unnest of range 0x7fff88e00000->0x7ff
000000 of DYLD shared region in VM map 0x42f17d93b36c2c59. While not abnormal for debuggers, this increases system memory foo
int until the target exits.
Jan 18 22:33:45 dyn-160-39-38-195 kernel[0] <Notice>: Google Chrome He[68486] triggered unnest of range 0x7fff88e00000->0x7ff
000000 of DYLD shared region in VM map 0x42f17d93aec5e1a9. While not abnormal for debuggers, this increases system memory foo
```

20



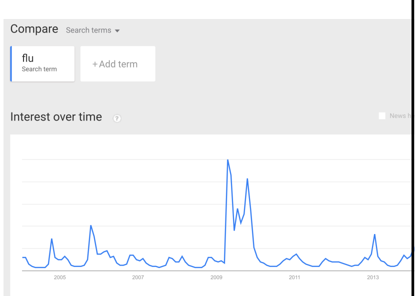
21

Data Science Applications

Health

Data Science Applications

Health



https://www.google.org/flutrends/

23

Data Science Applications

Health

Thank you for stopping by.

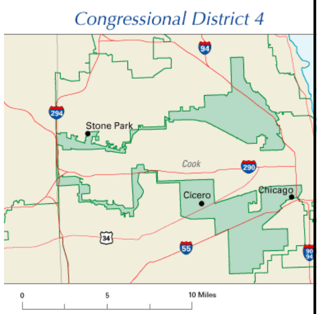
Google Flu Trends and Google Dengue Trends are no longer publishing current estimates of Flu and Dengue fever based on search patterns. The historic estimates produced by Google Flu Trends and Google Dengue Trends are available below. It is still early days for

https://www.google.org/flutrends/

What are we doing with data?

Health

Politics

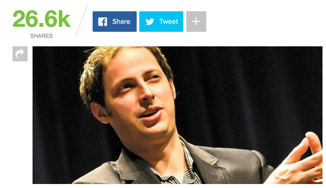


What are we doing with data?

Health

Politics

Triumph of the Nerds: Nate Silver Wins in 50 States

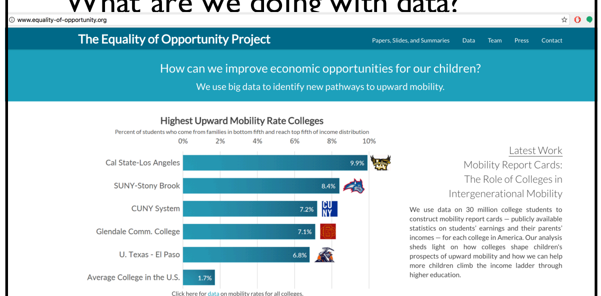


What are we doing with data?

Health

Politics

What are we doing with data?



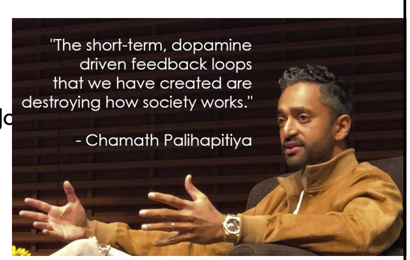
What are we doing with data?

Health

Politics

Investigative Jo

Society



https://nogreport.com/2017/12/23/chamath-palihapitiya-problem-facebook-presents-society/

What are we doing with data?

Epidemiological modeling of online social network dynamics

John Cannarella[1], Joshua A. Spechler[1,*]

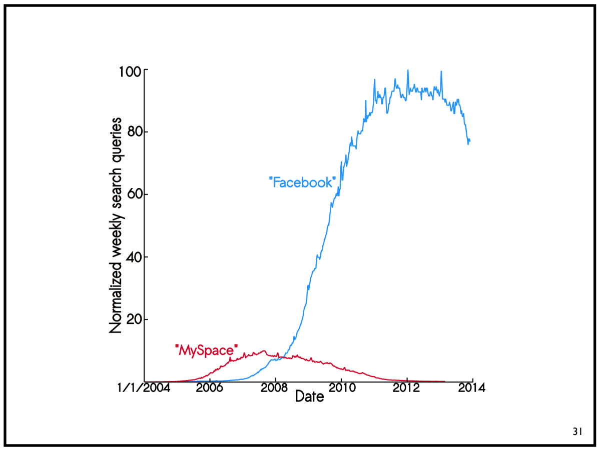
1 Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA

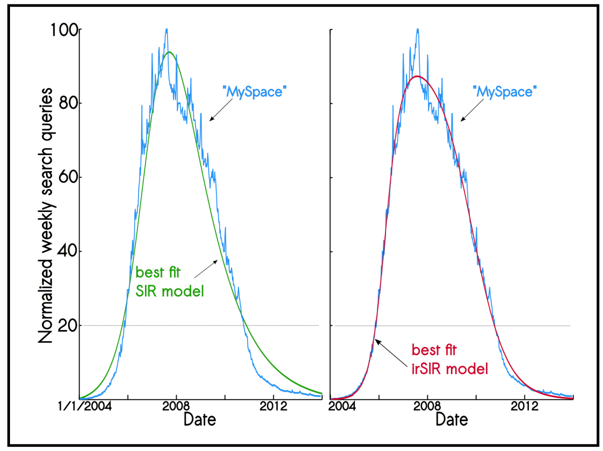
*** E-mail: Corresponding spechler@princeton.edu**

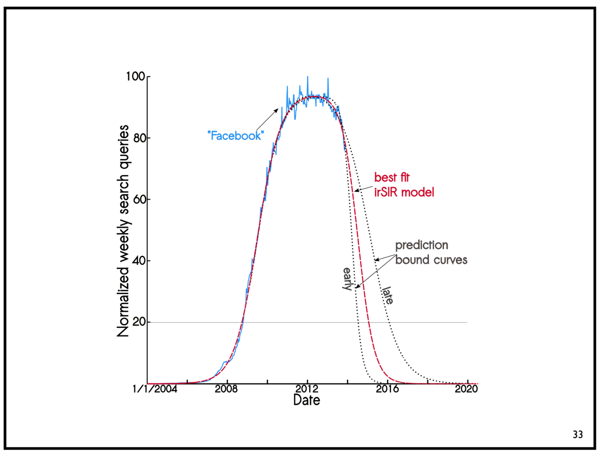
Abstract

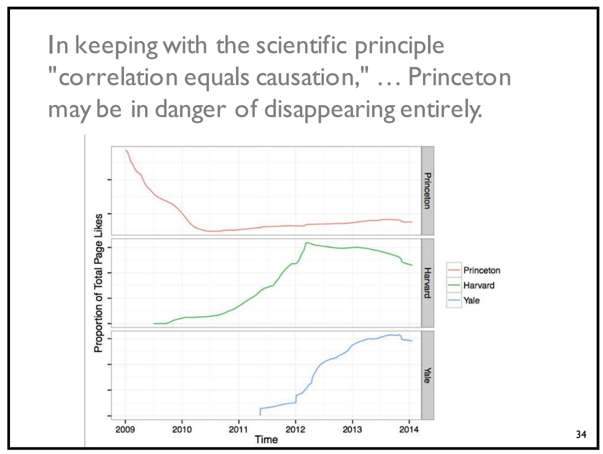
The last decade has seen the rise of immense online social networks (OSNs) such as MySpace and Facebook. In this paper we use epidemiological models to explain user adoption and abandonment of OSNs, where adoption is analogous to infection and abandonment is analogous to recovery. We modify the traditional SIR model of disease spread by incorporating infectious recovery dynamics such that contact between a recovered and infected member of the population is required for recovery. The proposed infectious recovery SIR model (irSIR model) is validated using publicly available Google search query data for "MySpace" as a case study of an OSN that has exhibited both adoption and abandonment phases. The irSIR model is then applied to search query data for "Facebook," which is just beginning to show the onset of an abandonment phase. Extrapolating the best fit model into the future predicts a rapid decline in Facebook activity in the next few years.

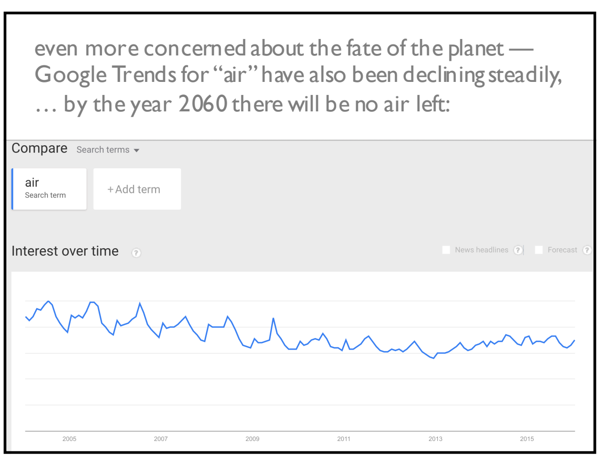
Extrapolating the best fit model into the future predicts a rapid decline in Facebook activity in the next few years

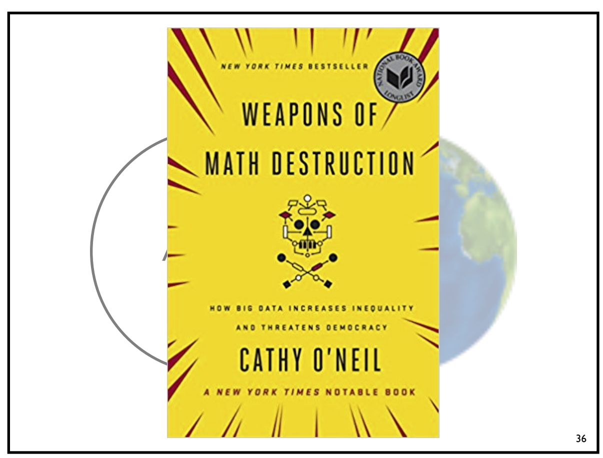

Normalized weekly search queries
100
80
60
40
20
"Facebook"
"MySpace"
1/1/2004
2006
2008
2010
2012
2014
Date
31

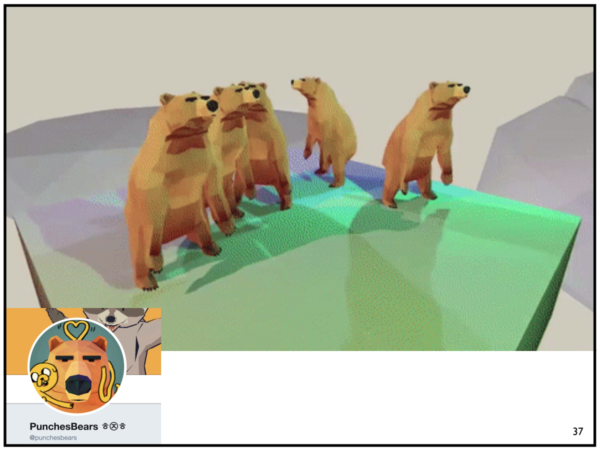

Normalized weekly search queries
100
80
60
40
20
"MySpace"
best fit
SIR model
1/1/2004
2008
2012
Date
"MySpace"
best fit
IrSIR model
2004
2008
2012
Date


Normalized weekly search queries
100
80
60
40
20
"Facebook"
best fit
IrSIR model
prediction
bound curves
early
late
1/1/2004
2008
2012
2016
2020
Date
33


In keeping with the scientific principle "correlation equals causation," … Princeton may be in danger of disappearing entirely.
Proportion of Total Page Likes
Princeton
Harvard
Yale
2009
2010
2011
2012
2013
2014
Time
34

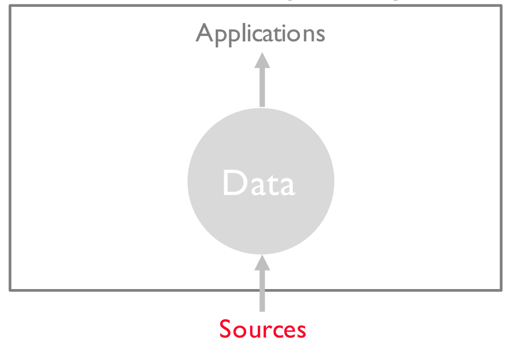

even more concerned about the fate of the planet — Google Trends for "air" have also been declining steadily, … by the year 2060 there will be no air left:
Compare
Search terms
air
Search term
+ Add term
Interest over time


NEW YORK TIMES BESTSELLER
WEAPONS OF MATH DESTRUCTION
HOW BIG DATA INCREASES INEQUALITY AND THREATENS DEMOCRACY
CATHY O'NEIL
A NEW YORK TIMES NOTABLE BOOK
36



37

Computer Systems



38

Autogenerated – record every…

Mouse click
Car drive
Ad impression
Webpage visit
Billing transaction
Network message
Error
Video stream

39

User generated

WhatsApp

40

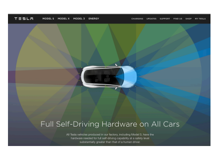
Then there's the Q50, a smart watch for children. Marketed as a way to help parents easily communicate with and keep track of their kids, bugs in the watch would allow hackers to "intercept all communications, remotely listen to the child's surroundings and spoof the child's location," according to a report by Top10VPN, a consumer research company this month.

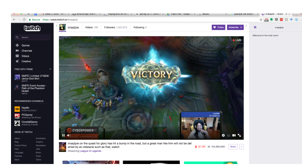
And the BB-8 droid, which was released with "The Last Jedi" this month, also had an insecure

https://twitter.com/internetofshit?lang=en

41

Video

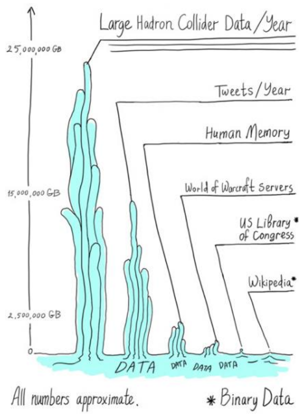




42

Scientific

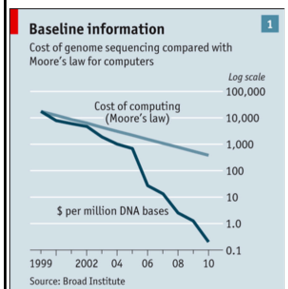
Only limited by the precision of the instrument

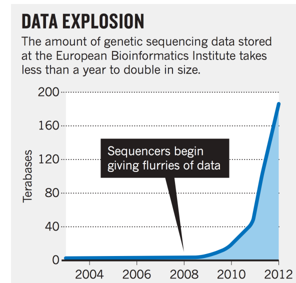


http://www.symmetrymagazine.org/article/august-2012/particle-physics-tames-big-data

43

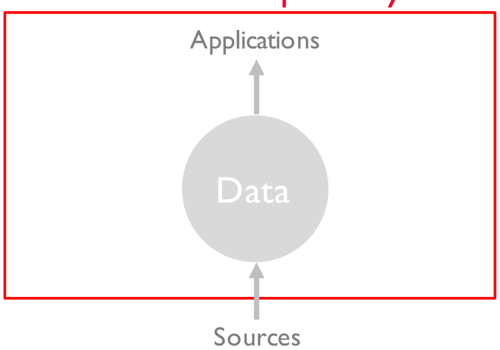
Health/Bioinformatics





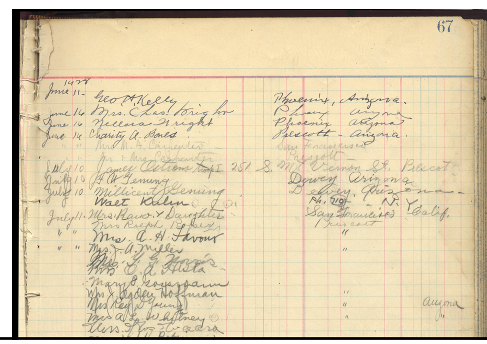
44

Computer Systems



45

Pen and Paper



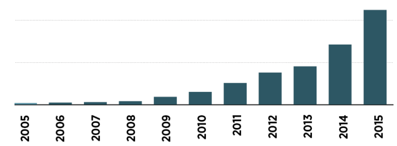
46

http://www.theatlantic.com/technology/archive/2013/10/computing-power-used-to-be-measured-in-kilo-girls/280633/

47

https://en.wikipedia.org/wiki/IBM_PC_compatible

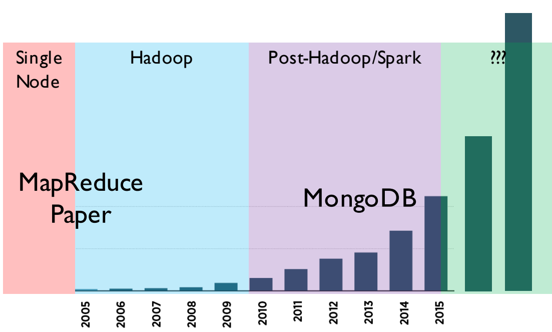
Data Volume Over Time



Note: Post-2013 figures are predicted. Source: UNECE

50

Big Data Systems Over Time



Note: Post-2013 figures are predicted. Source: UNECE

51

Post-Spark: ML Systems?

Lines of code in google's ML system

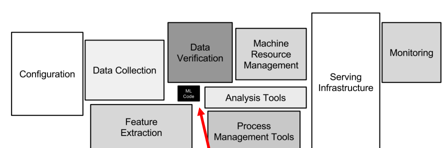


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

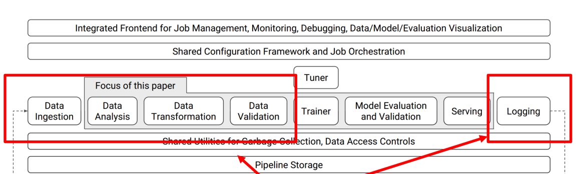
ML Training

Hidden technical debt in machine learning systems

52

Post-Spark: ML Systems?

TFX: Google's TensorFlow ML Platform



Data Problems

TFX

53

Post-Spark: ML Systems?

Massive data management *to support* ML

Many data problems

collection, cleaning, merging, validation, analysis, monitoring, processing, finding, versioning, sharing

54

Course Goal

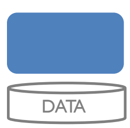
Understand fundamental principles behind large-scale data science systems

Data Processing Techniques for “big” data

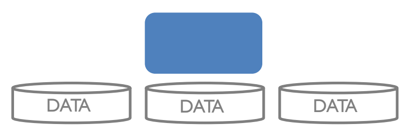
Experience with modern data science tools (Spark)

55

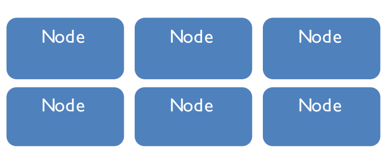
57



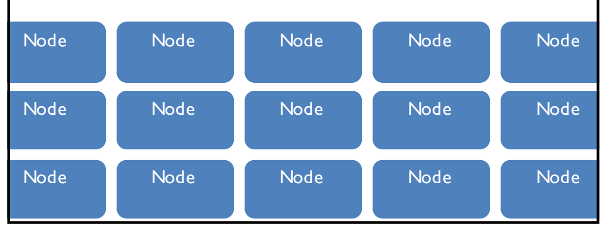
58



59

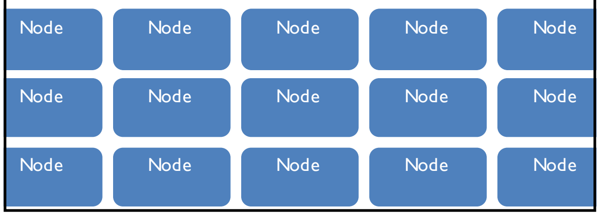


60



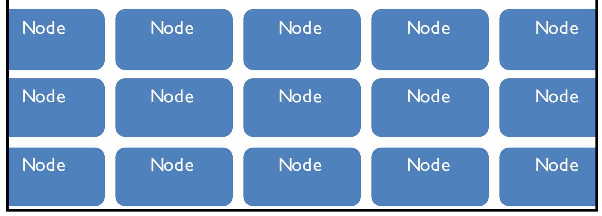
Fundamental Issues

Applies to *any* multi-node system



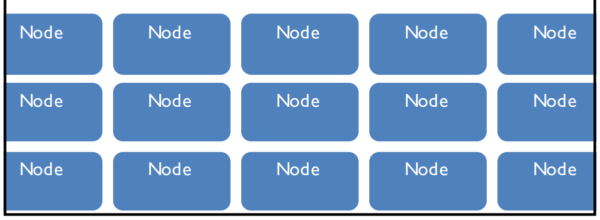
Data Processing Issues

How to lay out, index, structure data to answer queries quickly, correctly



Spark/Application Issues

How to use modern distributed computing system?



Big Data Systems in the Wild

Spark
Google Cloudflow
Azure Cloud
AWS/Redshift
Tensorflow
Cloudera
…

65

Course Structure Overview

Three key modules and focus areas:

1. Data modeling and visualization (Wu)
   * Various data models and storage
   * Graph processing and big-data visualization
2. Storage at Scale (Geambasu)
   * Challenges and core techniques for scalability and fault tolerance
   * Distributed transactions on sharded databases
   * Replication architectures and protocols
   * Design and implementation of Spanner, Google's geo-distributed, transactional store
3. Processing at Scale (Sahu)
   * Batch processing with Map Reduce and higher level programming construct
   * Real-time responsive analytics with Spark and Spark Streams

Designing Machine Learning Systems with Big Data

66

Course Administrative Details

Course materials

1. Primarily lecture notes. Additional reference readings will be provided as needed based on the lecture topics including research papers.

All course related submissions will be done using courseworks.

Important deadlines and communications will be done using Courseworks Announcement.

3-4 TAs will be available to assist in the course. We will announce their contact emails.

Good programming background in one of the languages Python/Java

w4121.github.io

67

Grading and Project

Grading
1. 60% Homework
2. 30% Tests/Quizzes
3. 10% Participation (ask/answer questions)

Optional Project
1. 0-40% Extra credit (does not affect curve)

A+s
1. Hand selected by instructors for *exceptional* work

68

Logistics

Register with piazza
We will not answer direct emails

69

Collaboration Policy

Read Syllabus on course site for allowed conduct

CS Dept academic honesty policies
http://www.cs.columbia.edu/education/honesty

We will not tolerate *any* cheating
Cheating = Failing grade

70

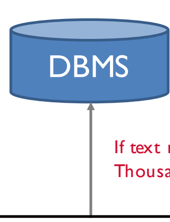
Module 1: Data Modeling Topics

Data models
Data cleaning
Data wrangling, Entity Resolution, Explanation
Large scale analytics
Visualizations and scaling them

71

How does data get into a DBMS?

Entity resolution
Data extraction
Missing data
…



Extract
Transform
Load

If text matches XXX, then…
Thousands of rules

72

How does data get into a DBMS?

Entity resolution
Data extraction
Missing data
…

Extract
Transform
Load

verizon

The New York Times

Technology

SUBSCRIBE | LOG IN

Medicaid's Data Gets an Internet-Era Makeover

By STEVE LOHR
JANUARY 9, 2017

Jini Kim's relationship with Medicaid is business and personal.

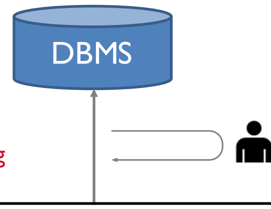
Her San Francisco start-up, Nuna, while working with the federal government, has built a cloud-computing database of the nation's 74 million Medicaid patients and their treatment.

Medicaid, which provides health care to low-income people, is administered state by state. Extracting, cleaning and curating the information from so many disparate and dated computer systems was an extraordinary achievement, health and technology specialists say. This new collection of data could inform the coming debate on Medicaid spending.

Andrew M. Slavitt, acting director of the Centers for Medicare and Medicaid Services, described the cloud database as "near historic." Largely because Medicaid information resides in so many state-level computing silos, Mr. Slavitt explained, "we've never had a systemwide view across the program."

74

How does data get into a DBMS?



Interactive data cleaning

75

How does data get into a DBMS?

Text → data records

Building sensor IDs – no consistency, arbitrary

BLDA1C600A_ART

BLDC1C2____TMR

Automated Metadata Construction To Support Portable Building Applications

Arka Bhattacharya, David Culler
Electrical Engineering and Computer Sciences, UC Berkeley
arka,culler@eecs.berkeley.edu

Dezhi Hong, Kamin Whitehouse
University of Virginia
dh5gm,whitehouse@virginia.edu

Jorge Ortiz
IBM Research
jjortiz@us.ibm.com

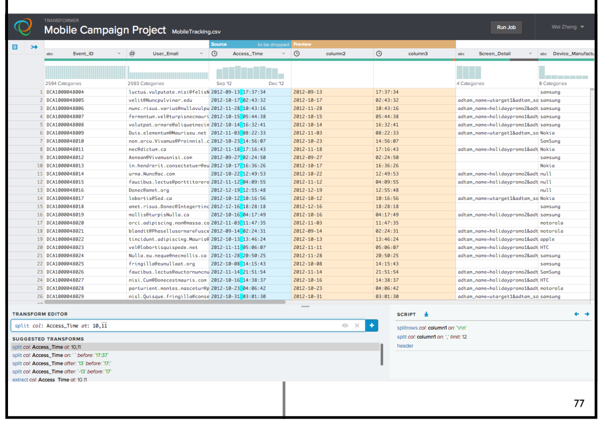
Eugene Wu
Computer Science, Columbia University
ewu@cs.columbia.edu

ABSTRACT

Commercial buildings consume nearly 19% of delivered en-

active topic for emerging systems, it has long been a core challenge in the legacy setting. Often, a critical step in the

76



77

Large scale analytics

Data volumes too large to even scan once

How to deal with this?

- Spend more time
- Concurrency
- Reduce data size
- Read less data
- Do less work
- Waste less time doing work

78

Large scale analytics

Columnar databases

In-memory databases

Intermediate results

Graph "databases"

Sketching and sampling

79

Visualization

How to think about and approach visualization
Modern visualization tools
How to scale visualizations

80

Module 2: Storage at Scale

- Two key reasons for distributed systems:
  - Scaling: system capacity grows proportionally with # of machines.
  - Fault tolerance: being able to continue operation despite failures, which can happen constantly in a large system.
- But achieving scale and fault tolerance (at scale) is hard.
  - Consistency, coherence, semantics are one challenge.
  - Fault tolerance requires coordination, which limits scalability.
- The second model will teach key techniques and protocols for scaling and fault tolerance, with a particular focus on one system: Google's Spanner storage system.

81

Module 3: Processing at Scale

Computation on huge amount of data is not a luxury – it is a necessity!

Imagine Facebook logs for logins. FB wants to compute how many people are logging in from which continents for each hour.

How to compute?

82

What's the big deal?

- How big is the data?
- Huge data – the data file does not fit into single server's disks…how do you compute if data does not even fit into server's storage?
- Data is on multiple servers – on a cluster of servers. So how do you compute and where do you compute what?
- How do we compute the final results?
- Who takes care of some machine or computing failure?
- How do you automate such computations spread across machines?

83

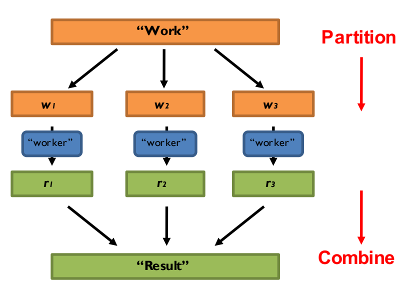
Computing Models for Big Data

We will learn two computing paradigm for big data on a cluster of machines

1. Batch processing with Map Reduce
   1. Idea is to divide and conquer the task – compute partial results on smaller chunks of data and then merge the partial results to compute final result
   2. Move computing task to where data is
- Real-time processing with Spark
  - Map Reduce is great but too slow due to lot of disk based operations
  - Spark computes with in-memory data

84

Divide and Conquer



85

So what is Hadoop/Map Reduce

Hadoop/Map Reduce is a computing system on a cluster of machines that provide at the minimum the following

- Storage across a cluster of machines (HDFS)
- A computation model to divide-conquer a task (map-reduce)
- A runtime to enable map-reduce style of computation

86

Why MapReduce not efficient for iterative computations?

- MapReduce is an excellent computing model that scales for log processing type of computations described earlier.
- What about iterative models that use the same data again and again?
  - Every operation is to read and write to disk. So every iteration requires reading and writing to disk. Too many disk based operations for iterative computing.
  - Many machine learning based computations are iterative in nature.
- So what is the solutions? Can the data be somehow kept in memory until all the operations on it completes…
- **Spark Model**: Resilient Distributed Datasets (RDD)
  - Recent computing model that is 100x faster and more suitable for iterative and real-time analytics
  - We will learn how to write real-time analytics using Spark and Spark Streams.

87