Foundations of Data Engineering

Thomas Neumann

About this Lecture

The goal of this lecture is teaching the standard **tools** and **techniques** for **large-scale data processing**.

Related keywords include:

- Big Data
- cloud computing
- scalable data-processing
- ...

We start with an overview, and then dive into individual topics.

Goals and Scope

Note that this lecture emphasizes practical usage (after the introduction):

- we cover many different approaches and techniques
- but all of them will be used in practical manner, both in exercises and in the lecture

We cover both concepts and usage:

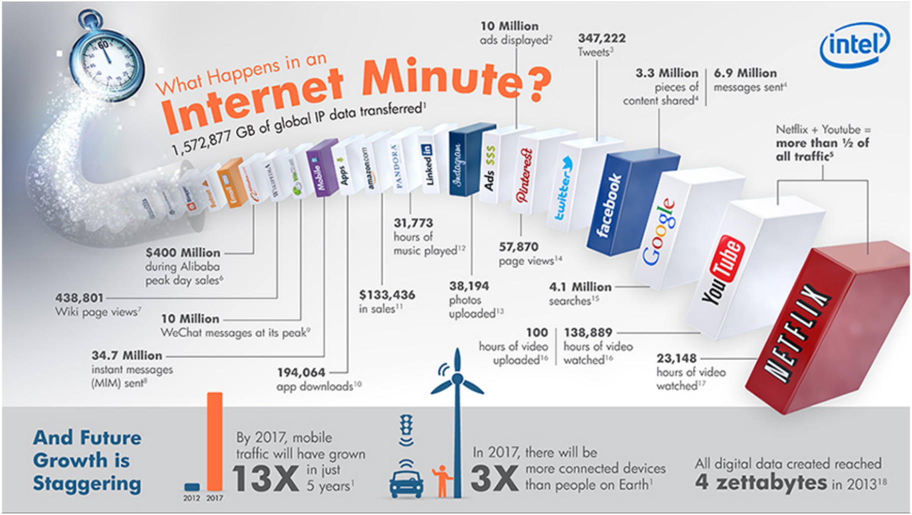
- what software layers are used to handle Big Data?
- what are the principles behind this software?
- which kind of software would one use for which data problem?
- how do I use the software for a concrete problem?

Some Pointers to Literature

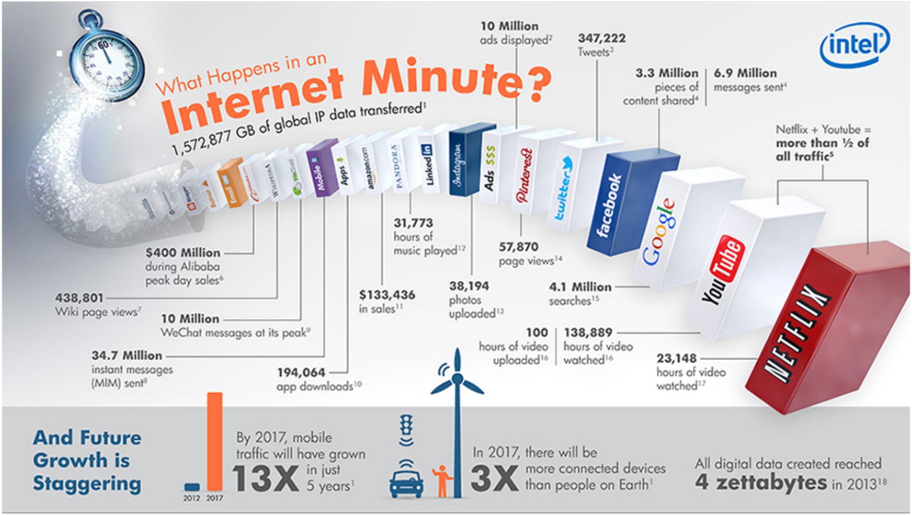
They are not required for the course, but might be useful for reference

- The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines
- Hadoop: The Definitive Guide
- Big Data Processing with Apache Spark
- Big Data Infrastructure course by Peter Boncz

The Age of Big Data

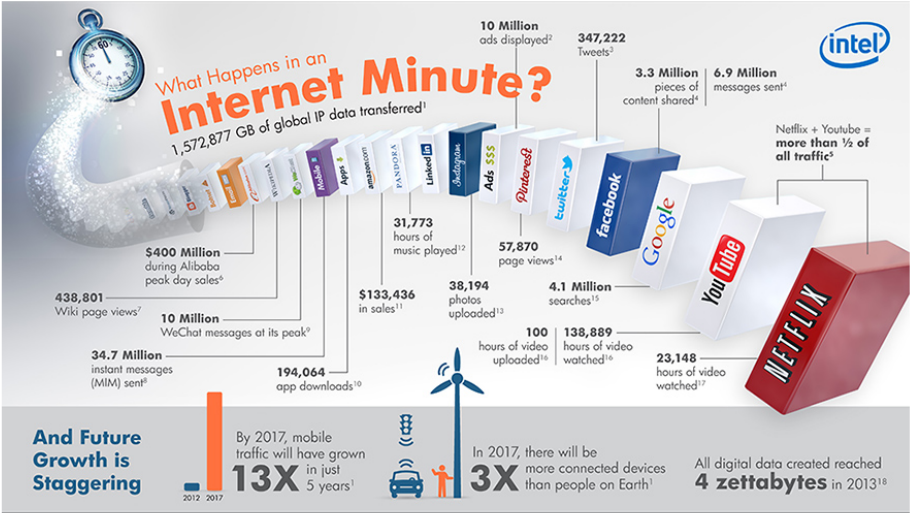


The Age of Big Data



- 1,527,877 GB/m = 1500 TB/m = 1000 drives/m = 20m stack/m

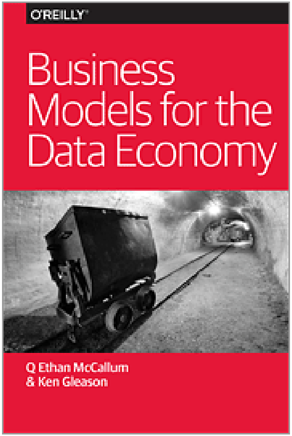
The Age of Big Data



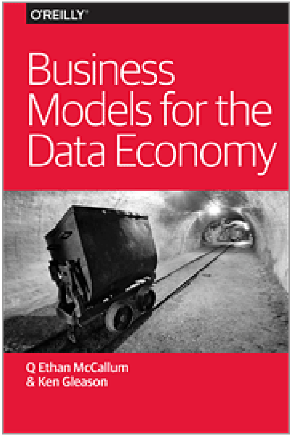
- 1,527,877 GB/m = 1500 TB/m = 1000 drives/m = 20m stack/m
- 4 zetabytes = 3 **billion** drives

"Big Data"

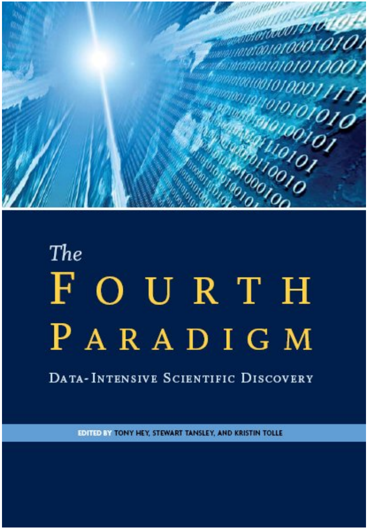
The Data Economy



The Data Economy



Data Disrupting Science



Scientific paradigms:

- Observing
- Modeling
- Simulating
- **Collecting and Analyzing Data**

Big Data

Big Data is a relative term

- if things are breaking, you have Big Data
  - Big Data is not always Petabytes in size
  - Big Data for Informatics is not the same as for Google
- Big Data is often hard to understand
  - a model explaining it might be as complicated as the data itself
  - this has implications for Science
- the game may be the same, but the rules are completely different
  - what used to work needs to be reinvented in a different context

Big Data Challenges (1/3)

- **Volume**
  - data larger than a single machine (CPU,RAM,disk)
  - infrastructures and techniques that scale by using more machines
  - Google led the way in mastering “cluster data processing”
- Velocity
- Variety

Supercomputers?

- take the top two supercomputers in the world today
  - Tianhe-2 (Guangzhou, China)
    - cost: US$390 million
  - Titan (Oak Ridge National Laboratory, US)
    - cost: US$97 million
- assume an expected lifetime of five years and compute cost per hour
  - Tianhe-2: US$8,220
  - Titan: US$2,214
- this is just for the machine showing up at the door
  - not factored in operational costs (e.g., running, maintenance, power, etc.)

Let's rent a supercomputer for an hour!

Amazon Web Services charge US$1.60 per hour for a large instance

- an 880 large instance cluster would cost US$1,408
- data costs US$0.15 per GB to upload
  - assume we want to upload 1TB
  - this would cost US$153
- the resulting setup would be #146 in the world's top-500 machines
- total cost: US$1,561 per hour
- search for: LINPACK 880 server

Supercomputing vs. Cluster Computing

- Supercomputing
  - focus on performance (biggest, fastest).. At any cost!
  - oriented towards the [secret] government sector / scientific computing
  - programming effort seems less relevant
  - Fortran + MPI: months do develop and debug programs
  - GPU, i.e. computing with graphics cards
  - FPGA, i.e. casting computation in hardware circuits
  - assumes high-quality stable hardware
- Cluster Computing
  - use a network of many computers to create a 'supercomputer'
  - oriented towards business applications
  - use cheap servers (or even desktops), unreliable hardware
  - software must make the unreliable parts reliable
  - focus on economics (bang for the buck)
  - programming effort counts, a lot! No time to lose on debugging..

Cloud Computing vs Cluster Computing

- Cluster Computing
  - Solving large tasks with more than one machine
    - parallel database systems (e.g. Teradata, Vertica)
    - NoSQL systems
    - Hadoop / MapReduce
- Cloud Computing

Cloud Computing vs Cluster Computing

- Cluster Computing
- Cloud Computing
  - machines operated by a third party in large data centers
    - sysadmin, electricity, backup, maintenance externalized
  - rent access by the hour
    - renting machines (Linux boxes): Infrastructure as a Service
    - renting systems (Redshift SQL): Platform-as-a-service
    - renting an software solution (Salesforce): Software-as-a-service
- independent concepts, but they are often combined!

Economics of Cloud Computing

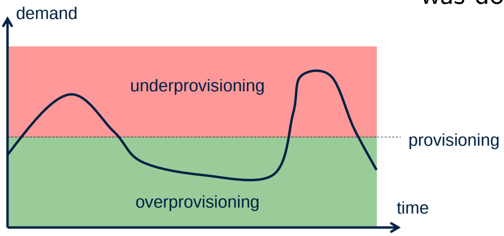
- a major argument for Cloud Computing is pricing:
  - We could own our machines
    - . . . and pay for electricity, cooling, operators
    - . . . and allocate enough capacity to deal with peak demand
  - since machines rarely operate at more than 30% capacity, we are paying for wasted resources
- pay-as-you-go rental model
  - rent machine instances by the hour
  - pay for storage by space/month
  - pay for bandwidth by space/hour
- no other costs
- this makes computing a commodity
  - just like other commodity services (sewage, electricity etc.)
- some caveats though, we look at them later

Cloud Computing: Provisioning

We can quickly scale resources as demand dictates

- high demand: more instances
- low demand: fewer instances

Elastic provisioning is crucial



Target (US retailer) uses Amazon Web Services (AWS) to host target.com

- during massive spikes (November 28 2009 –"Black Friday") target.com is unavailable

Remember your panic when Facebook was down?

Cloud Computing: some rough edges

- some provider hosts our data
  - but we can only access it using proprietary (non-standard) APIs
  - **lock-in** makes customers vulnerable to price increases and dependent upon the provider
  - local laws (e.g. privacy) might prohibit externalizing data processing
- providers may control our data in unexpected ways:
  - July 2009: Amazon remotely removed books from Kindles
  - Twitter prevents exporting tweets more than 3200 posts back
  - Facebook locks user-data in
  - paying customers forced off Picasa towards Google Plus
- anti-terror laws mean that providers have to grant access to governments
  - this privilege can be overused

Privacy and Security

- people will not use Cloud Computing if trust is eroded
  - who can access it?
    - governments? Other people?
    - Snowden is the Chernobyl of Big Data
  - privacy guarantees needs to be clearly stated and kept-to
- privacy breaches
  - numerous examples of Web mail accounts hacked
  - many many cases of (UK) governmental data loss
  - TJX Companies Inc. (2007): 45 million credit and debit card numbers stolen
  - every day there seems to be another instance of private data being leaked to the public

High performance and low latency

- how quickly data moves around the network
  - total system latency is a function of memory, CPU, disk and network
  - the CPU speed is often only a minor aspect
- examples
  - Algorithmic Trading (put the data centre near the exchange); whoever can execute a trade the fastest wins
  - simulations of physical systems
  - search results
    - Google 2006: increasing page load time by 0.5 seconds produces a 20% drop in traffic
    - Amazon 2007: for every 100ms increase in load time, sales decrease by 1%
    - Google's web search rewards pages that load quickly

Big Data Challenges (2/3)

- Volume
- **Velocity**
  - endless stream of new events
  - no time for heavy indexing (new data arrives continuously)
  - led to development of data stream technologies
- Variety

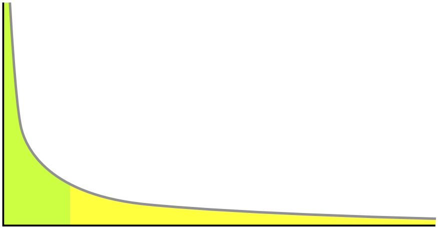
Big Streaming Data

- storing it is not really a problem: disk space is cheap
- efficiently accessing it and deriving results can be hard
- visualising it can be next to impossible
- repeated observations
  - what makes Big Data big are repeated observations
  - mobile phones report their locations every 15 seconds
  - people post on Twitter $>$ 100 million posts a day
  - the Web changes every day
  - potentially we need unbounded resources
    - repeated observations motivates streaming algorithms

Big Data Challenges (3/3)

- Volume
- Velocity
- **Variety**
  - dirty, incomplete, inconclusive data (e.g. text in tweets)
  - semantic complications:
    - AI techniques needed, not just database queries
    - Data mining, Data cleaning, text analysis
    - techniques from other DEA lectures should be used in Big Data
  - technical complications:
    - skewed value distributions and "Power Laws"
    - complex graph structures, expensive random access
    - complicates cluster data processing (difficult to partition equally)
    - localizing data by attaching pieces where you need them makes Big Data even bigger

Power laws



- Big Data typically obeys a power law
- modelling the head is easy, but may not be representative of the full population
  - dealing with the full population might imply Big Data (e.g., selling all books, not just block busters)
- processing Big Data might reveal power-laws
  - most items take a small amount of time to process
  - a few items take a lot of time to process
- understanding the nature of data is key

Skewed Data

- distributed computation is a natural way to tackle Big Data
  - MapReduce encourages sequential, disk-based, localised processing of data
  - MapReduce operates over a cluster of machines
- one consequence of power laws is uneven allocation of data to nodes
  - the head might go to one or two nodes
  - the tail would spread over all other nodes
  - all workers on the tail would finish quickly.
  - the head workers would be a lot slower
- power laws can turn parallel algorithms into sequential algorithms

Summary

Introduced the notion of Big Data, the three V's
Explained Super/Cluster/Cloud computing

We will come back to that in the lecture, but we will start simple

- given a complex data set, what should you do to analyze it?
- start with simple approaches, become more and more complex
- we finish with cloud-scale computing, but not always appropriate
- Big Data is not the same for everybody

Notes on the Technical Side

We will use a lot of tools during this lecture

- we concentrate on free and/or open source tools
- in general available for all major platforms
- we strongly suggest to use a **Linux** system, though
- ideally a recent Ubuntu/Debian system
- other systems should work, too, but you are on your own
- using a Virtual Machine is ok, might be easier than a native Linux system