CAS Information Engineering

Big Data

Concept & Slides: Prof. Dr. Kurt Stockinger

Course Instructor: Dr.-Ing. habil. Josef Spillner <spio@zhaw.ch>

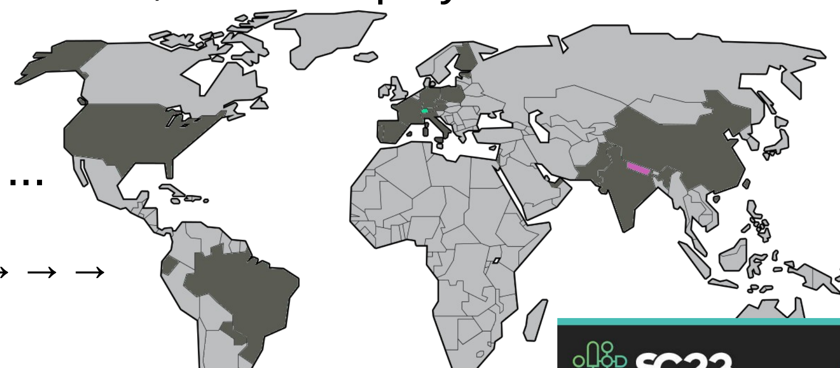
Course Coach: Oliver Cvetkovski <cvek@zhaw.ch>

Our Background

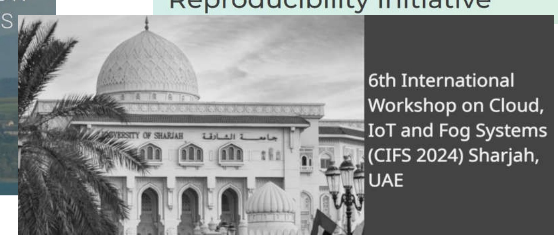
Supporting (via ZHAW joint innovation projects) Swiss companies and institutions in R&D/innovation around

- big data & cloud applications, cloud-native/serverless technologies
- IoT, messaging, service/API design, data service architectures
- data science & data integration frameworks, model deployment
- digital infrastructures
- domains: mobility, retail, agriculture, ...

Our global R&D/innovation experience → → →

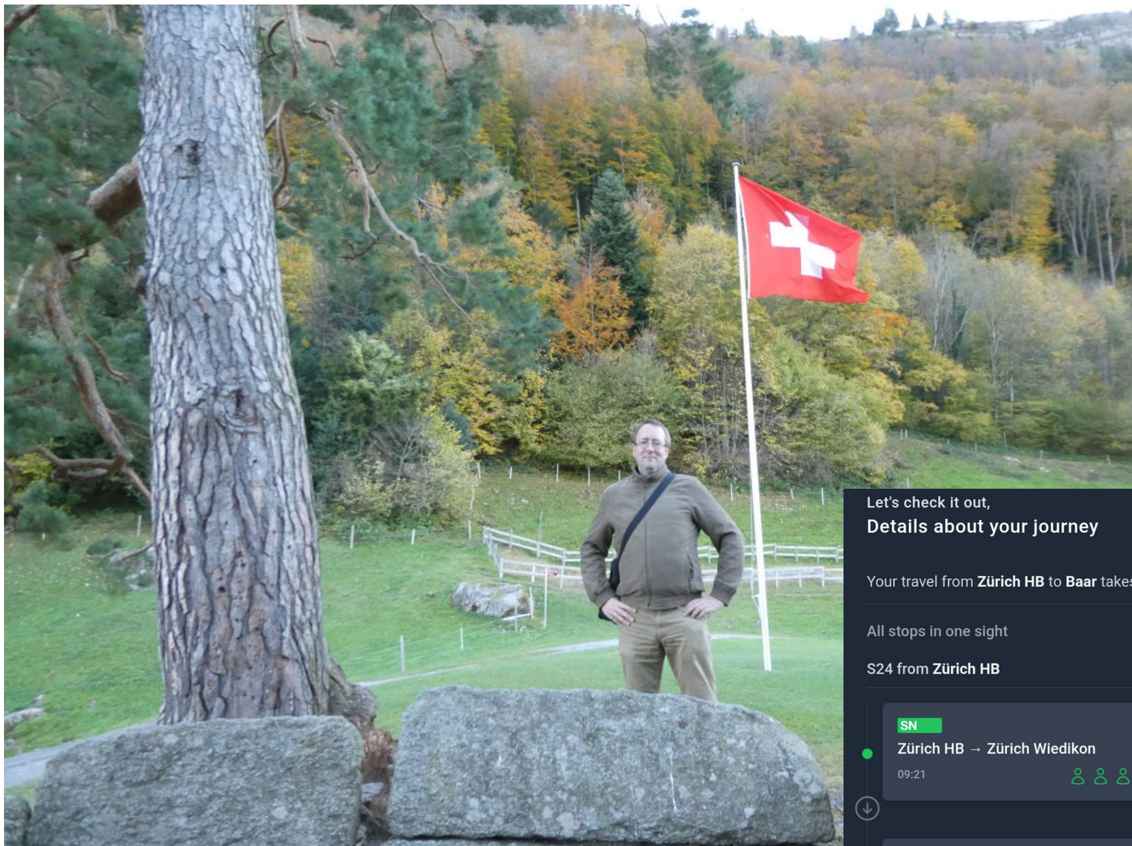


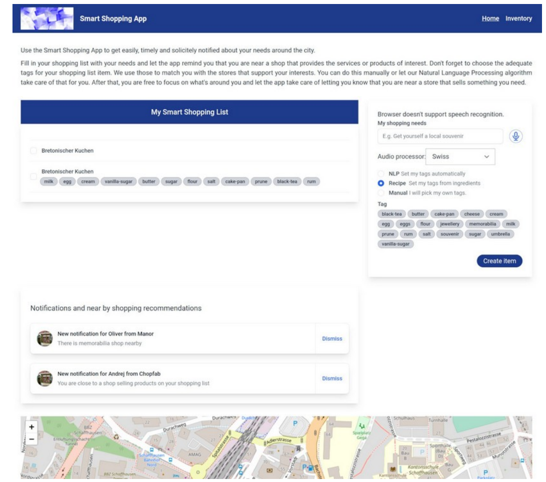
Our local activities →

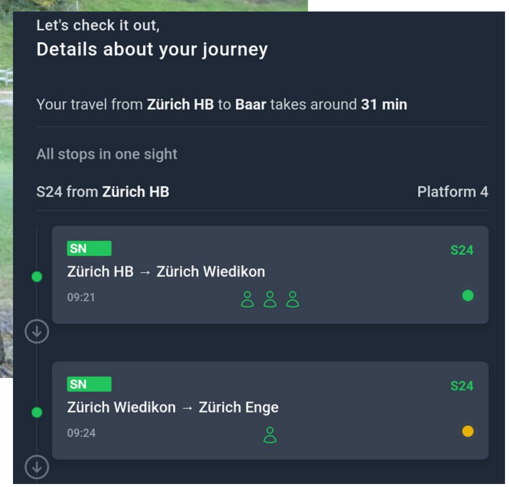


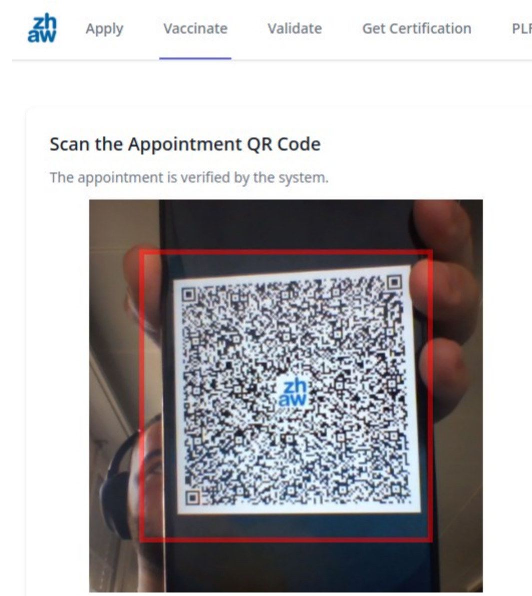
Our Background (Societal Significance)

Ensuring the availability of nation-scale & resilient big data applications.



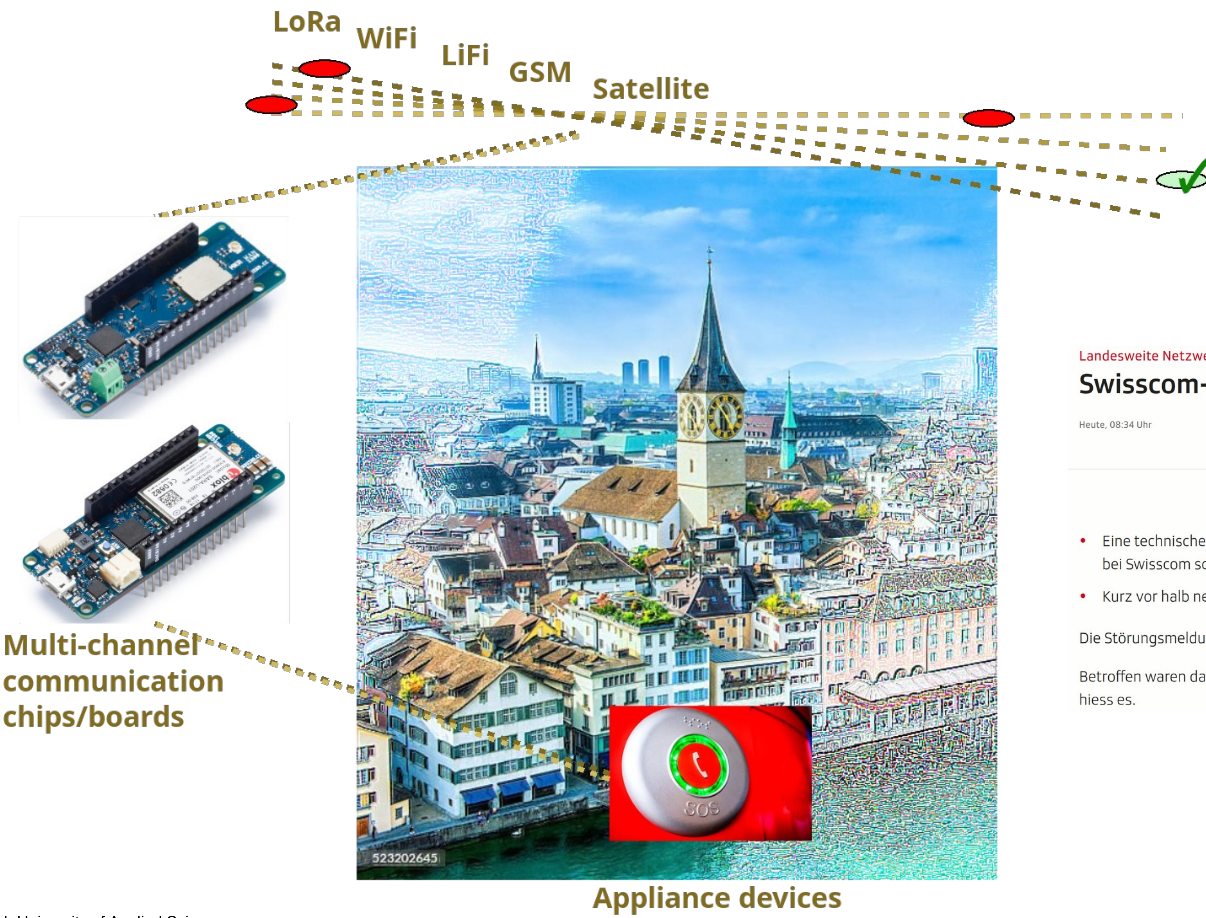






Our Background (Societal Significance)

Ensuring the availability of nation-scale & resilient big data applications.



Der Schock über die Folgen des Extremwetterereignisses sitzt auch am Dienstag noch tief. Die Notrufzentrale Neuchâtel (CNU) erhielt am Montag 1186 Telefonanrufe. Der Tag werde «unauslöschliche Spuren hinterlassen», sagte Stadtpräsident Jean-Daniel Jeanneret (53) am Montagabend.

Landesweite Netzwerkprobleme

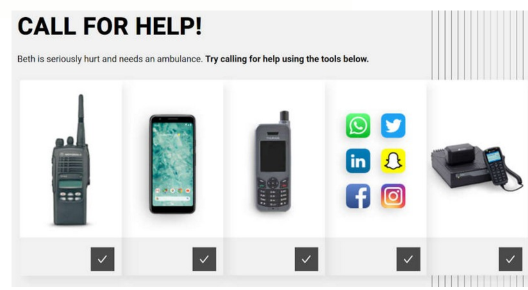
Swisscom-Netz war schweizweit gestört

Heute, 08:34 Uhr

- Eine technische Störung hat am Dienstagmorgen den Zugriff aufs Internet bei Swisscom schweizweit vorübergehend beeinträchtigt.
- Kurz vor halb neun Uhr war die Störung behoben.

Die Störungsmeldung wurde um 07.53 Uhr aufgeschaltet.

Betroffen waren das Internet über das Festnetz sowie über das Mobile-Netz, hiess es.

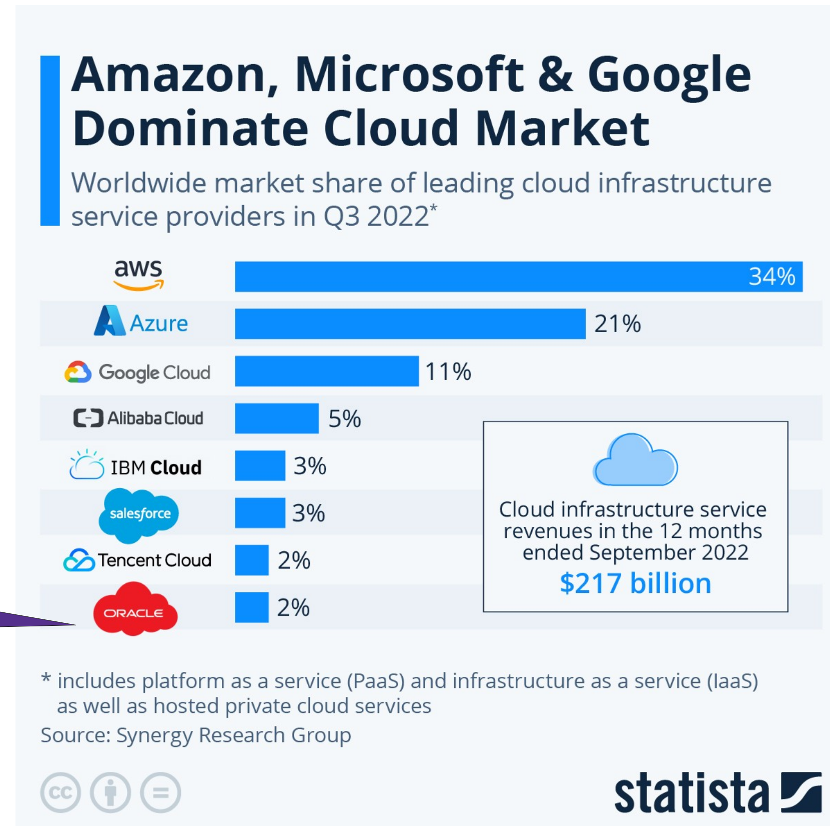


Zurich University
of Applied Sciences

Our Background (Societal Significance)

Can you / can the country master all these <u>big data processing</u> technologies?

Are clouds and big data lakes the next “too big to fail“ problem for Switzerland?



Totalausfall von Passkontroll-Automaten am Flughafen Zürich

Wochenlang keine Post

Post erklärt Ehemann vier Mal für tot

Jemand ändert im System der Post den Status eines Mannes wiederholt auf «tot». Die Post kann angeblich nichts tun.

Panne bei BAG-App

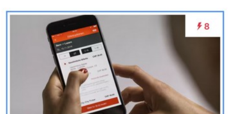
Covid-Zertifikate waren eine Stunde lang ungültig

Die BAG-App zum Nachweis einer Impfung oder eines negativen Tests war von einer technischen Störung betroffen. Als Gründe wurden Wartungsarbeiten genannt.

ALERTSWISS

Hochwasser-Alarm

Alertswiss-App: Bedenken wegen Abhängigkeit vom Ausland

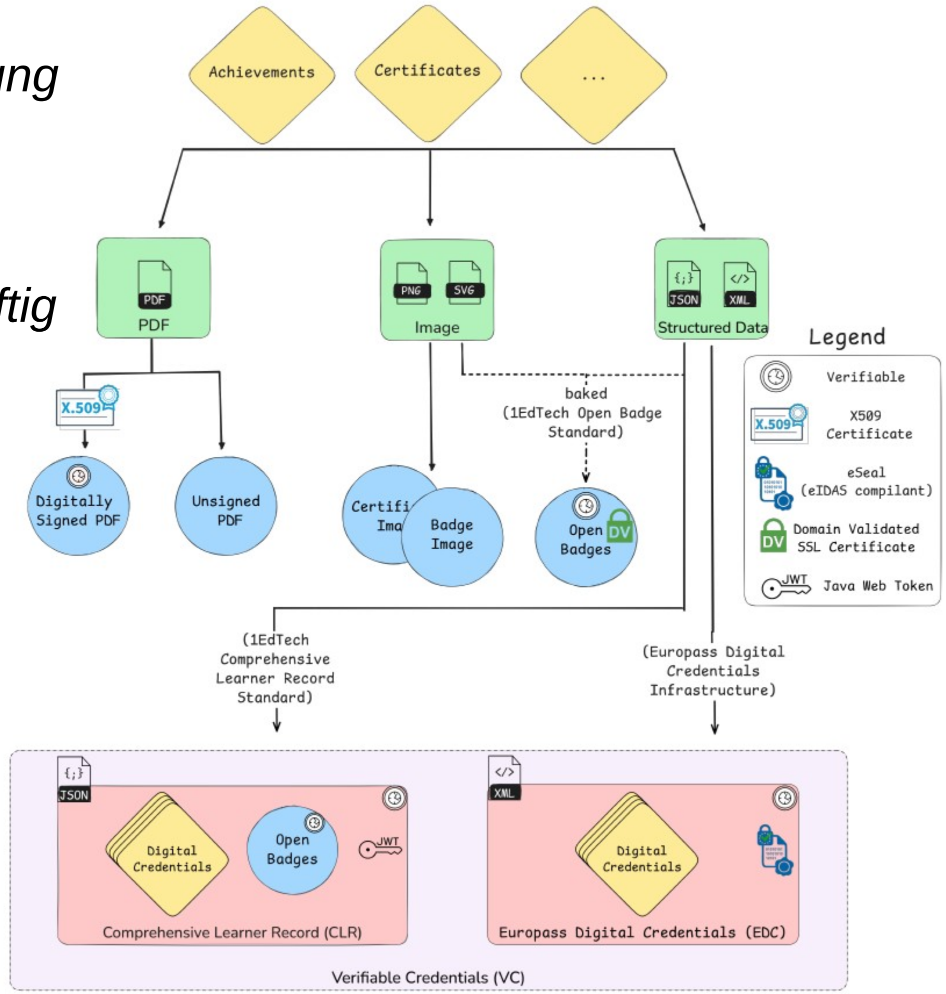


Technische Probleme – Billet-Kauf über SBB-App derzeit nicht möglich

Our Background (Societal Significance)

New digital infrastructures for digital certification - **`credentials.zhaw.ch`**

«Die Richtline MCs in der Weiterbildung wurde inhaltlich von der Kommission Bildung grundsätzlich gutgeheissen. D.h. für externe WBK-Teilnehmende (von WBKs mit ECTS) wird es zukünftig verpflichtend einen MC geben, für Personen welche WBKs im Rahmen eines CAS etc. besuchen regeln die Departemente, ob diese einen MC erhalten.»



Logistics, Terms and Conditions

- Dates
  - Lecture: 90 minutes theory with example (13:30-15:00)
  - Break: 30 minutes (15:00-15:30)
  - Lab: 90 minutes of self-study with coach support (15:30-17:00)
    Module days always* onsite, passive streaming provided

- Self-study: Read & experiment as much as possible at home

- Material: Find everything on the course e-learning platform Moodle
  Since 2023: Complementary mini-script + book

- Grading: Passed / not passed, based on presentation of a project (see next-next slide)

- ***Tech support: on demand during lab sessions***
  + during project phase by e-mail

"Install a patch for the update of the new version. If that doesn't work, install the new version of the update for the patch. If all else fails, install a patch for the new version of the update."

Working Environment

«Sparky» hosting Jupyter, terminal, Spark

```
ubuntu@sparky-collab:~$ pyspark
Python 3.11.4 (main, Jun  9 2023, 07:59:55) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
23/11/13 05:34:07 WARN NativeCodeLoader: Unable to load native-hadoop
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.4.0
      /_/

Using Python version 3.11.4 (main, Jun  9 2023 07:59:55)
Spark context Web UI available at http://sparky-collab:4040
Spark context available as 'sc' (master = local[*], app id = local-169
SparkSession available as 'spark'.
>>>
```

```python
floods = sqlContext.read.format("csv").option("header",
"true").option("inferSchema", "true").load(r"/home/jovyan/work/team5/Data/NFIP/nfip-flood-policies.csv")
```

```python
floods.count()
```

```
50406943
```

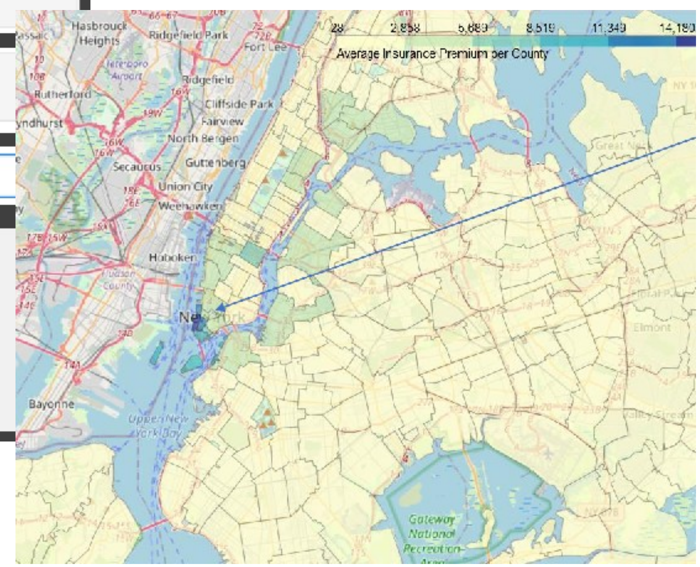
```python
floods.createOrReplaceTempView("floods")
```

```python
average_prem_per_state = sqlContext.sql("SELECT propertystate, AVG(totalinsurancepremiumofthepolicy) AS avg_prem /
                                          FROM floods GROUP BY propertystate")
```

```python
flood_state_pd_df = average_prem_per_state.toPandas()
```

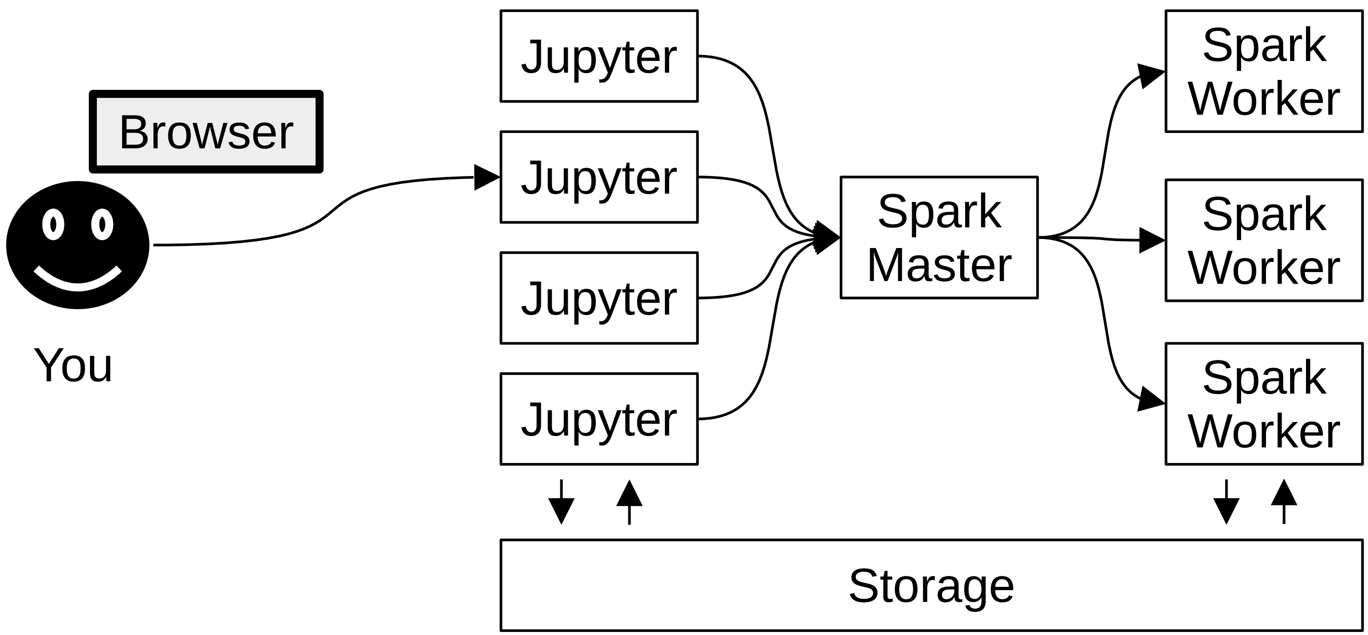
Read in data and aggregation process

- Read in 15 GB csv file
- Create TempView for basic SQL queries
- Aggregate data per state (resp. county)
- Transform spark dataframe to pandas dataframe



Working Environment

Infrastructure sketch



Project Work

- Start (16.12.2024):
  - Team up in groups of 3 people (teams of 2 is also ok, but not recommended, and needs confirmation)

- Project:
  - Implement a project of your own choice by using a distributed big data framework, and *boldly go where noone went before...*
      - e.g. exploit GPU support, Koalas, ... (details later)
  - With public CH datasets or data from your business (if fulfilling “big data“ criteria); might include real-world data acquisition

- Grading:
  - Sunday, Jan 12, 2025 @ 23:59:
    - Analysis code + reference to data set
    - PDF slides of presentation (max. 10 slides; no PPT due to format issues)
    - Upload to Moodle

Content

Day	Lecture	Exercise
11.11.2024	Big Data Introduction	MapReduce in Python
18.11.2024	Spark Introduction + DataFrames (DF)	DF over CSV data
25.11.2024	Spark Resilient Distributed Data Set (RDD)	RDD over textual/numeric data
2.12.2024	Data Storage: Hadoop Distributed File System + Parquet	Text Analytics RDD → DF + Project Outline
9.12.2024	Query Optimization	Performance Analysis (Encodings, Joins)
16.12.2024	Spark Best Practices and Applications	Project Implementation
Merry x-mas & a happy new year		
6.01.2025	NoSQL Systems/Approaches Beyond Spark	Project Implementation
13.01.2025	Project Presentations	---

Feedback? Feedback!

2023/24 improvement suggestions…

- Der Unterschied zwischen Bigdata Shell und Sparky Collab war nicht wirklich klar (unterschiedliche Zugriff auf Spark, etc.)
- Die Infrastruktur könnte ein bisschen datailierter beschrieben werden.
- Die Jupyter-Umgebung hätte schon zu Beginn genutzt werden können.

  ☑ Sparky is ready to serve + well documented
- Zu wenig Zeit für die Projektarbeit

  ☑ 2 extra weeks starting in 2023
- Abgabe eines Skripts/Handouts wäre hilfreich gewesen.

  ☑ complementary script covering especially weeks 1-3 exists
- Übungsbeispiele zu wenig didaktisch

  ☑ exercises revamped with more explanations
- Der Einsteig war etwas "steil" evtl. könnte man da Leute, welche noch nie mit Kommandozeile etc. gearbeitet haben besser abholen.

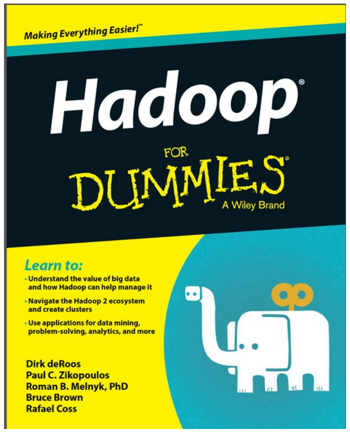
  ☑ OSI book available to learn more about operating systems

Educational Objectives for Today

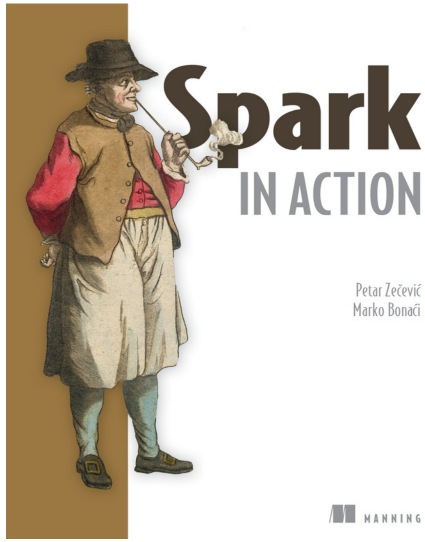
- Learn about the main definitions of big data and its historic background
  - … and what matters in data engineering
- Understand the main concepts of map/reduce data processing
- Understand fundamentals of Hadoop as one early map/reduce system
-
- See possibilities but also limits with vertical scalability in Python

Literature

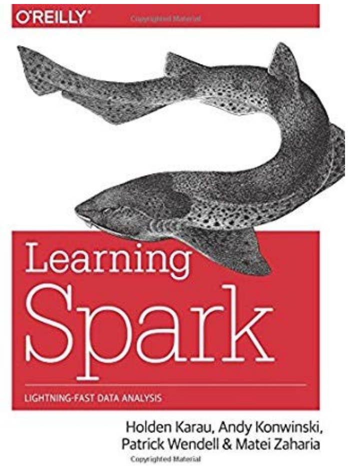
2014



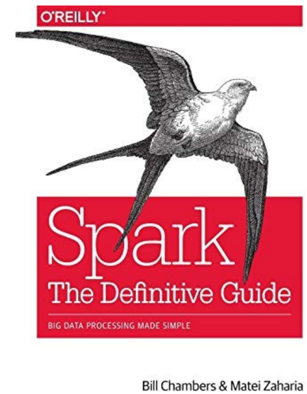
2016



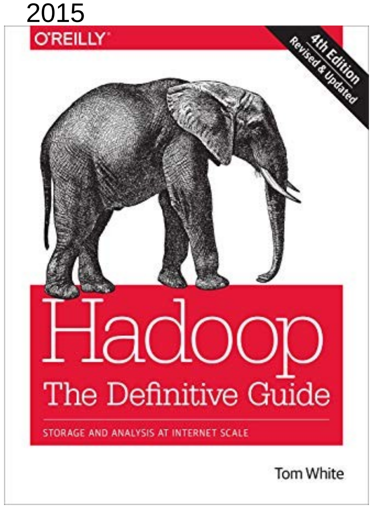
2nd edition, 2020 → Spark 3.0



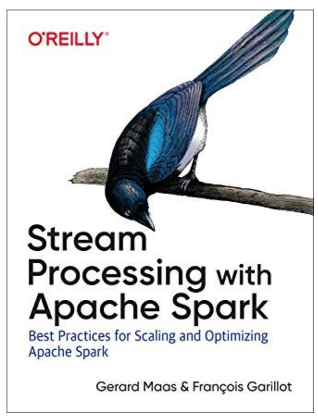
2018



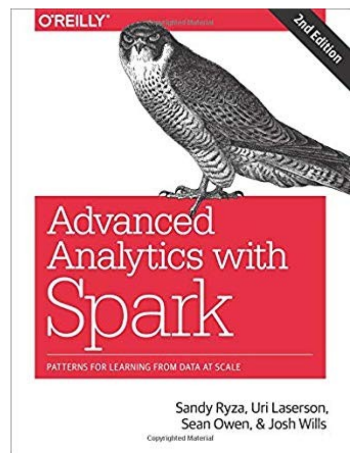
2015



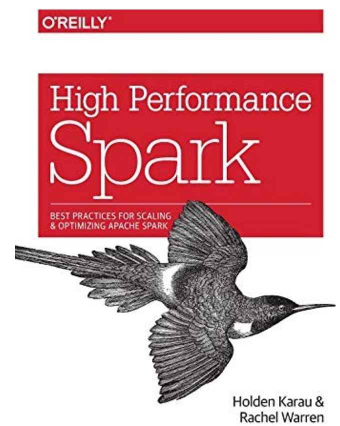
2019



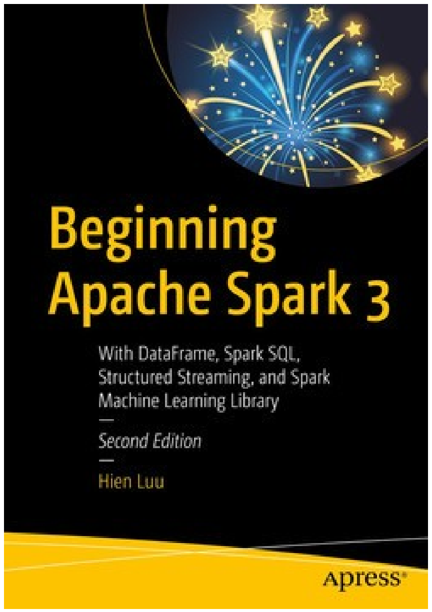
2017

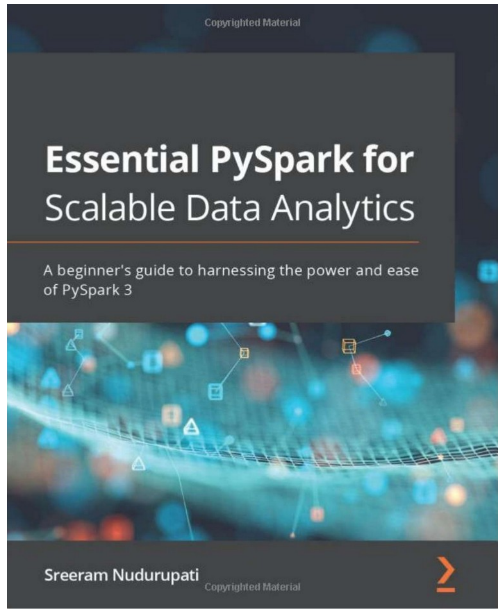


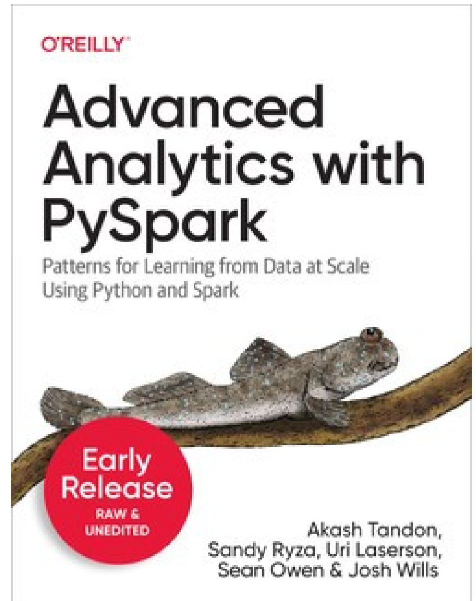
2017



Literature, more recent...







Mini-Script + Book


Parallele und verteilte Verarbeitung grosser Datenmengen – Komplementärskript zum Weiterbildungsmodul 'Big Data' im CAS INFE

Dr.-Ing. habil. Josef Spillner
Zürcher Hochschule für Angewandte Wissenschaften
spio@zhaw.ch

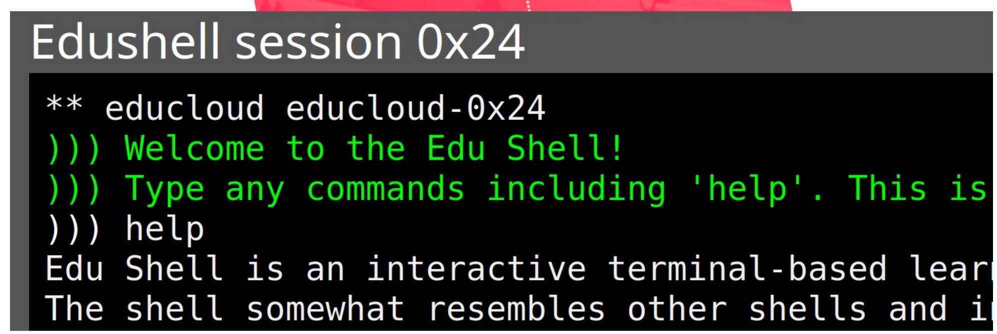
23. Januar 2023

1 Einführung in Big Data

Seit über 10 Jahren ist 'Big Data' ein Trendwort in der Informatik und im Bereich Informationssysteme[1]. In den letzten Jahren hat auch die praktische Bedeutung in Projekten mit IT-Bezug zugenommen. Logdaten, Sensordaten, Transaktionen – die Zunahme des Datenvolumens und der zuweilen damit einhergehenden Datenkomplexität aus allen Quellen erfordert oft eine Neuausrichtung der Verwaltung, Verarbeitung und Auswertung dieser Daten. Dies betrifft viele Bereiche, die zunehmend Entscheidungen anhand von aus Daten gewonnenen Aussagen treffen und diese Entscheidungsfindung zudem noch automatisieren. Ob in der Energiebranche, in der Mobilität, bei Maschinenherstellern oder in Behörden auf allen administrativen Ebenen – wann immer sich grosse Datenbestände nicht vermeiden lassen, werden Methoden und Werkzeuge benötigt, um mit diesen umgehen zu können und insbesondere Abfragen darauf stellen zu können.

Dieses Komplementärskript soll Teilnehmerinnen und Teilnehmern des Big-Data-Moduls im CAS Information Engineering an der Zürcher Hochschule für Angewandte Wissenschaften in Ergänzung der Vorlesungsfolien einen breiten Hintergrund vermitteln. Es ist vor dem Hintergrund der mehrjährigen Durchführung des Moduls entstanden und zielt vor allem auf die Wissensnivellierung für den Einstieg in die ersten beiden Modulwochen ab. Das Skript beginnt mit Betrachtungen zur Datenverarbeitung, motiviert darüber dedizierte Anwendungsdienste für die Verarbeitung verteilter Daten, und geht anschliessend auf Apache Spark als einen solchen Dienst ein. Spark wird dabei aus Systemsicht beleuchtet, während die konkrete Programmierung in

→ in Moodle



→ via vdf (free e-book,
39 Fr printed book with goodies)

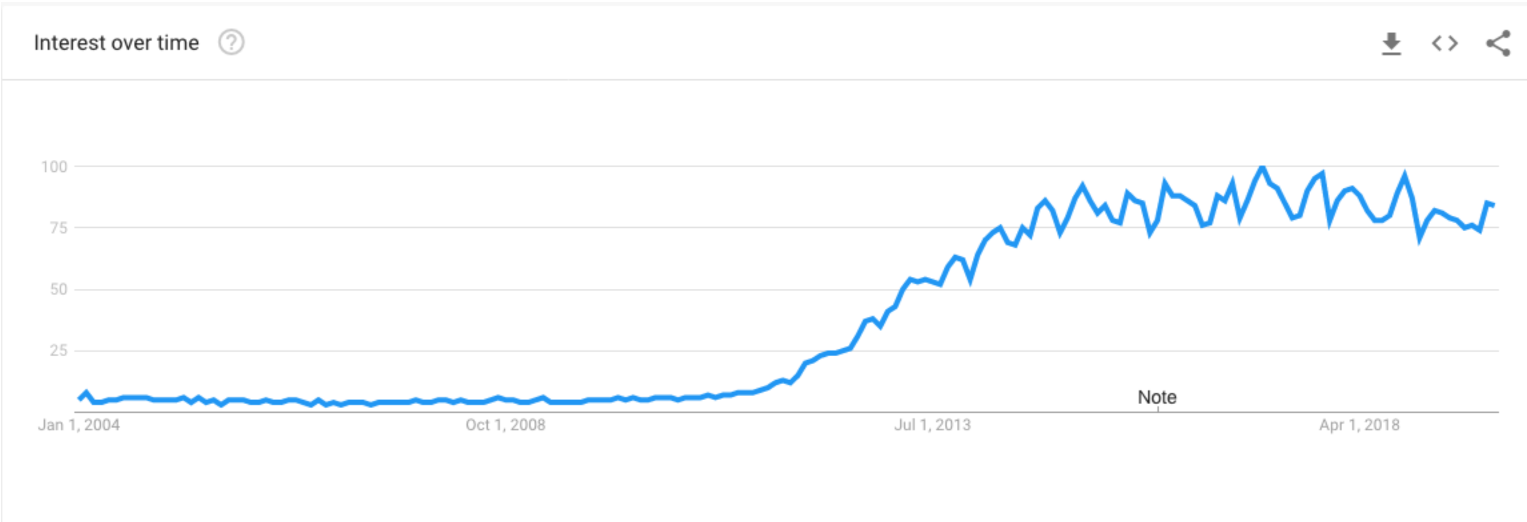
https://vdf.ch/operating-systems-and-infrastructure-in-data-science.html

Big Data

- What is big data for you?

“Big Data” in Google Trends Worldwide

- Since 2011 we see a significant rise



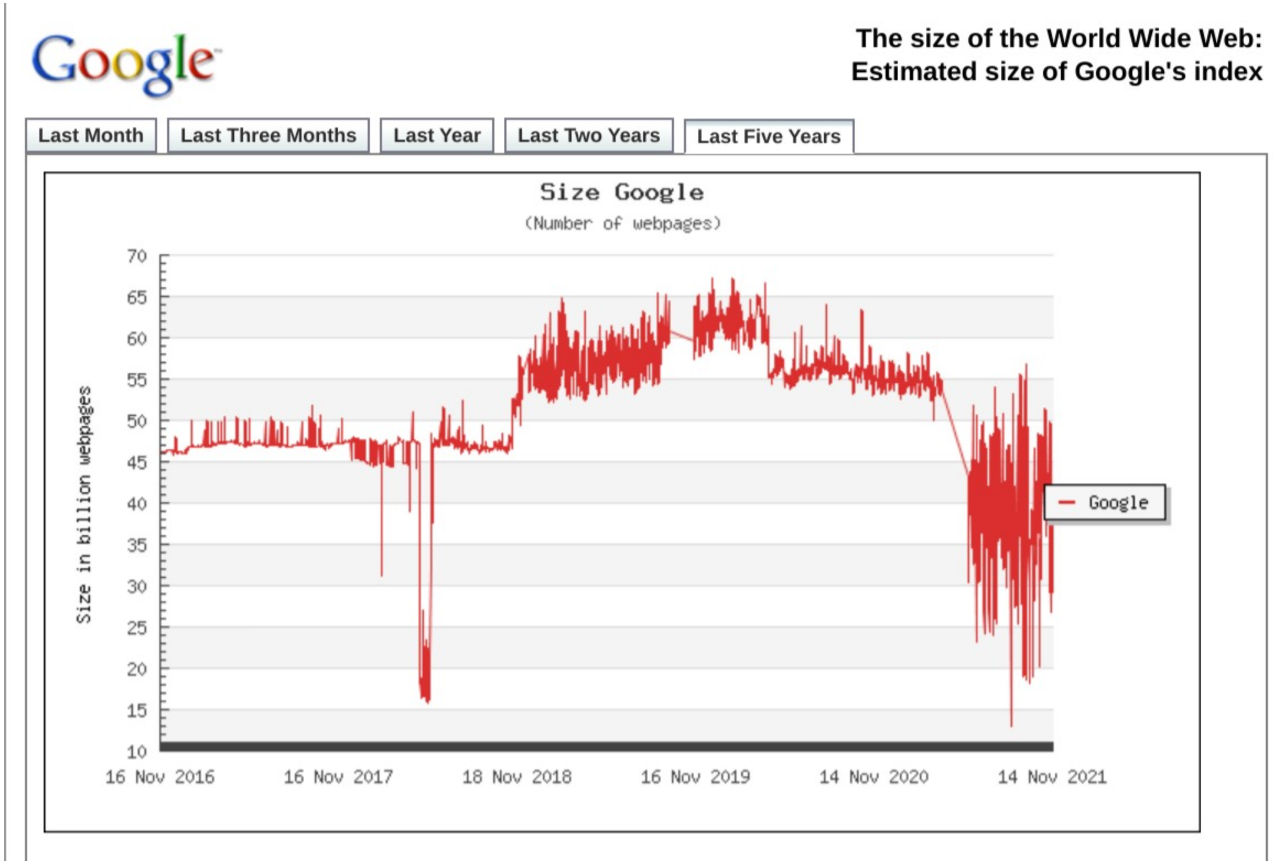
Big Data: Largest Library in the World

- Library of Congress in Washington

- Contains more than ~~36~~ ~~38~~ ~~39~~ ~~40~~ 51 million books
  [Source: https://www.loc.gov/about/general-information/]
  [ca. 20 TB]

- Quiz:
  - How long do you need to count, if you count one book per second?

Big Data: Number of Pages on the Internet

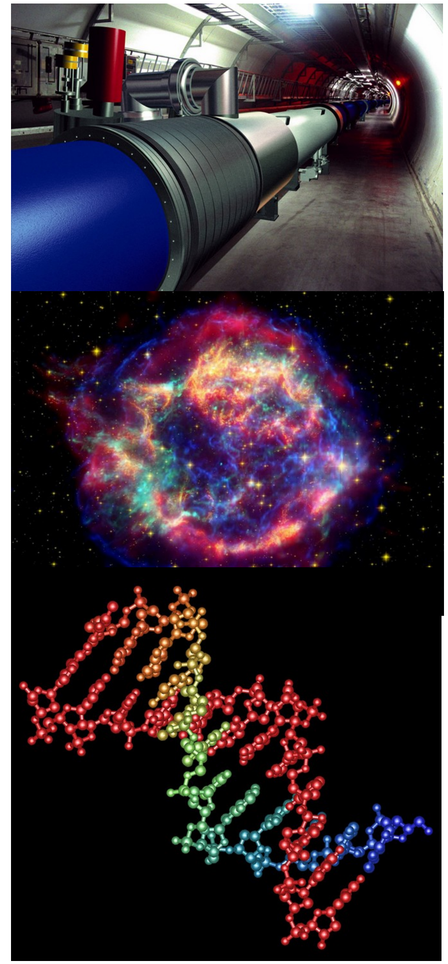


[Source: http://www.worldwidewebsize.com/]

- How long do you need to count now?

Major Waves of Data Tsunami

- Wave 1: Big research
  - E.g. CERN
- Wave 2: Companies
  - E.g. Amazon, eBay, Google, Yahoo
- Wave 3: Social networks
  - E.g. Facebook, Twitter, LinkedIn
- Wave 4: Machine-generated data
  - E.g. mobile phone, industry 4.0

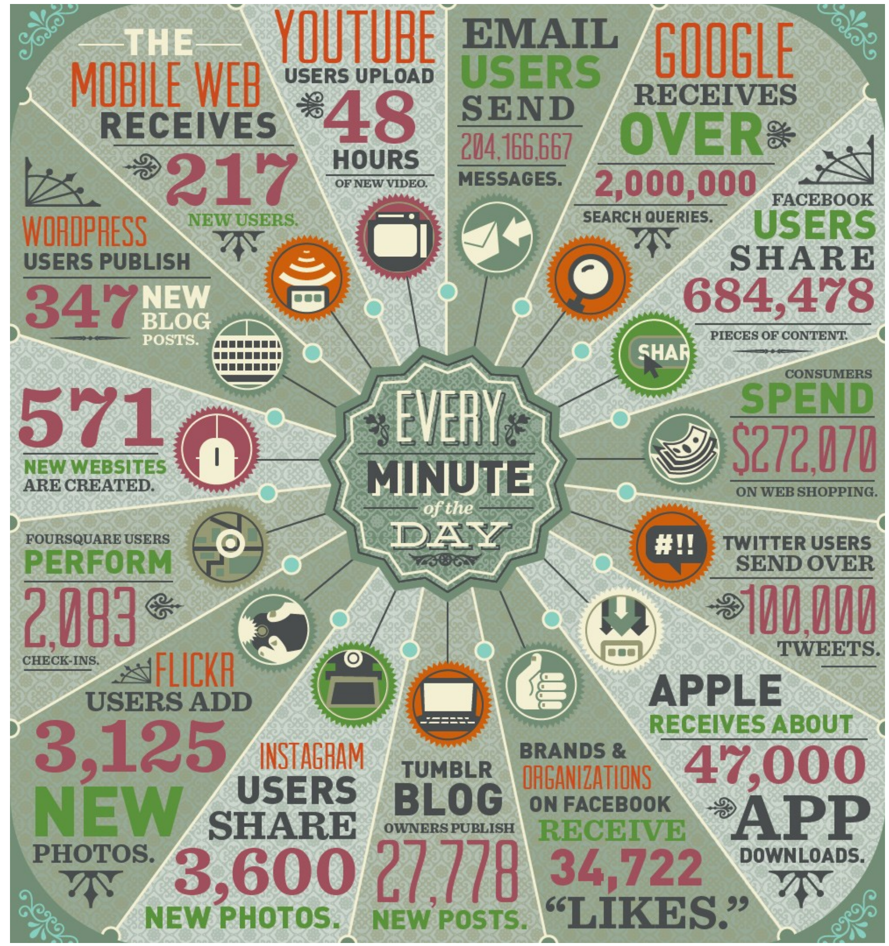


What about your company? Current + potential use:

Machine data (predictive maintenance), log files/metrics/traces (application delivery), business indicators, public open data, ...

How Large is Big Data?

- 1 TB:
  - 1000 Gigabytes
  - 300 hours of high quality videos
  - Encyclopedia Britannica
- 10 TB:
  - All books of the Library of Congress
- 1 PB:
  - 1000 TB
  - Data of a large bank
- 100 PB:
  - Data at CERN



Big Data Definition – the 5 Vs

- Volume (big):
  - “Large” amounts of data

«Does not fit into a computer anymore»
→ Requires distributed architectures

- Velocity (fast):
  - Streams of data need to be processed fast
- Variety (different data sources):
  - Text, images, videos, databases, blogs, social network data
- Veracity (quality):
  - Data of different quality
- Value:
  - Some data are more valuable than users (customer records vs. product description)
- Quiz: Which V is the hardest to manage?

What Matters in (Big) Data Engineering...

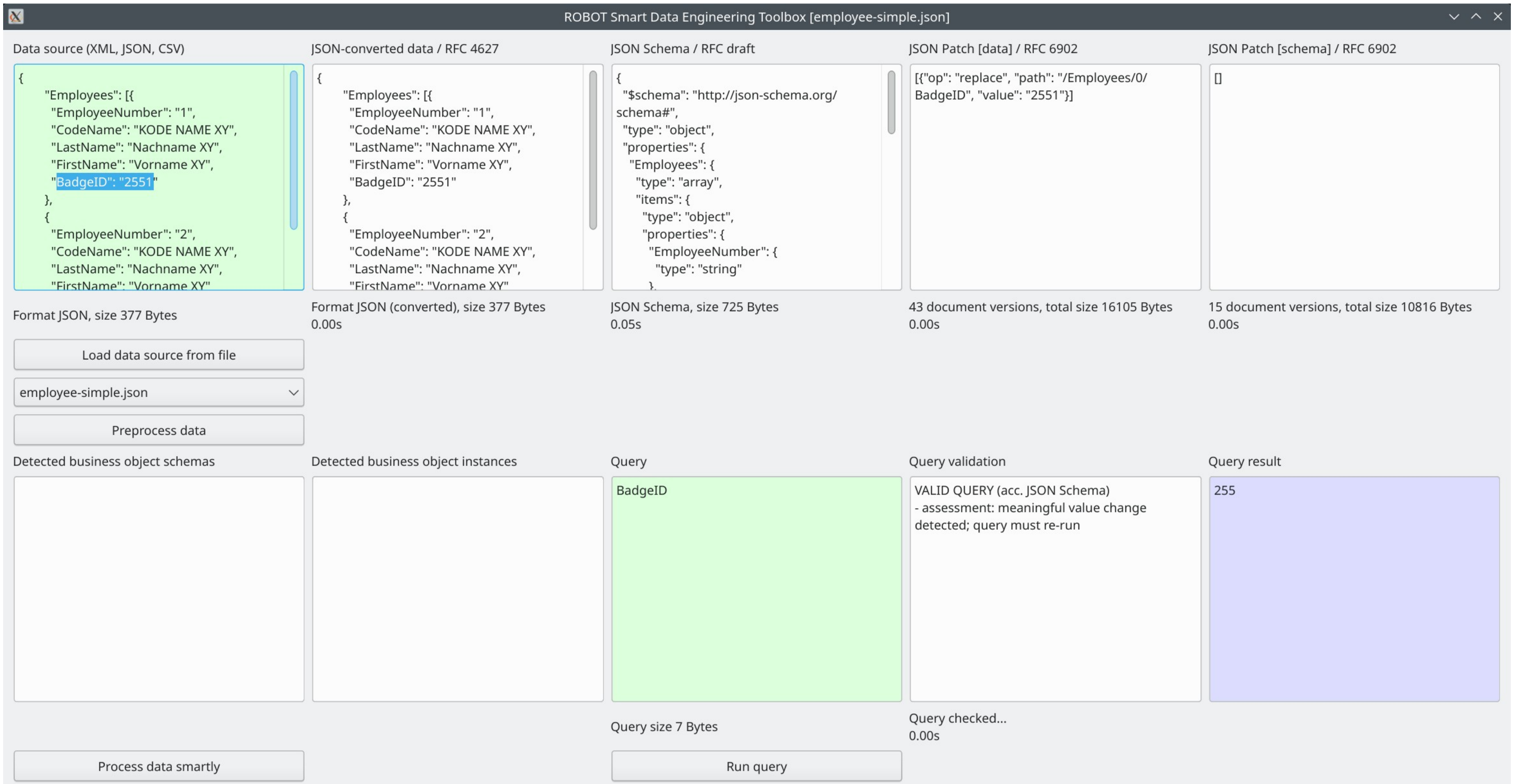
- Data production: follow good conventions
  - File formats & encodings, schemas, metadata/documentation, microformats (e.g. dates)
  - Data delivery: static, differential, queryable…
  - As small as possible! (including compression)
    - Secondary effects: cost, electricity, ecology…

- Data consumption:
  - Mechanical sourcing (big, fast, tedious)
  - Data integration, ETL
  - Smart processing (adaptation, caching, logic order etc.)
  - Confidence / truth

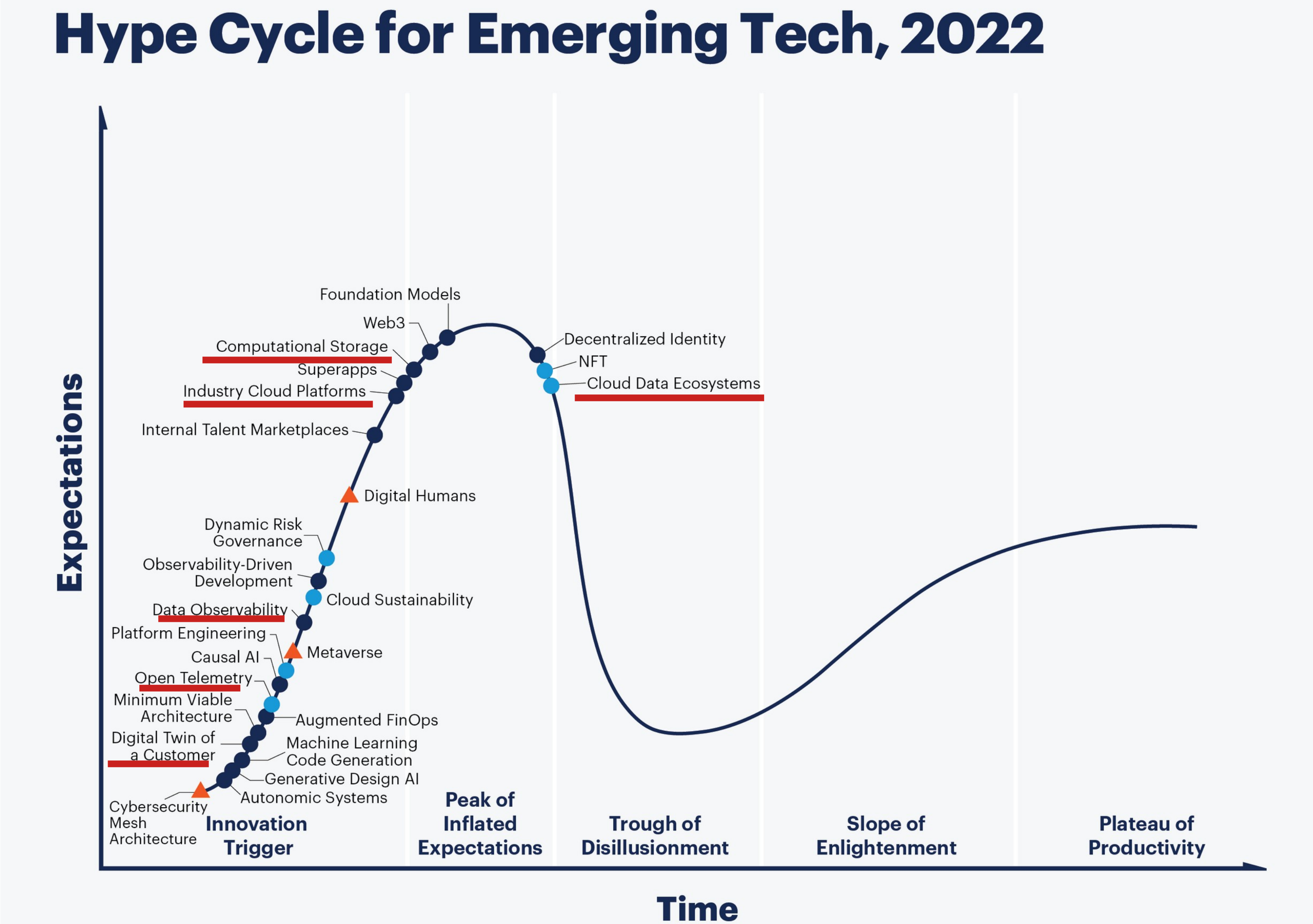
Zurich University
of Applied Sciences

What Matters in (Big) Data Engineering...

→ in this module: focus on volume & velocity
→ do not hesitate to ask if your interests are in other aspects, e.g. variety!

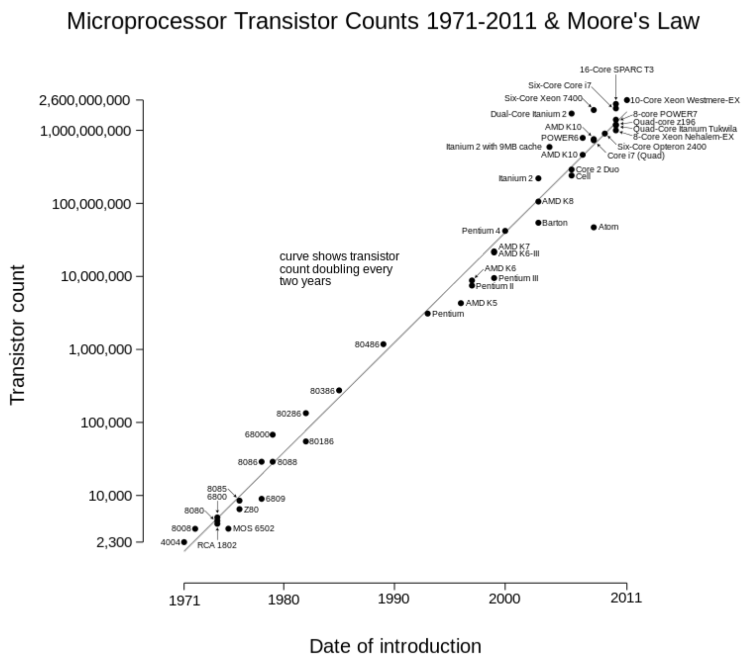


Is Big Data a Hype Topic?



Moore’s Law

- Number of transistors on chips doubles every 18 months (originally every 24 months)
- Exponential growth




SECTIONS HOME SEARCH The New York Times

OPINIONATOR | PRIVATE LIVES Shopping for Antiques, Finding My Mother

EDITORIAL A Tiny Crack in the Russian Ice

CHARLES M. BLOW The President, Fox News and the Poor

GAIL COLLINS Wow, Jeb Bush I

The Opinion Pages | OP-ED COLUMNIST

Moore’s Law Turns 50

MAY 13, 2015

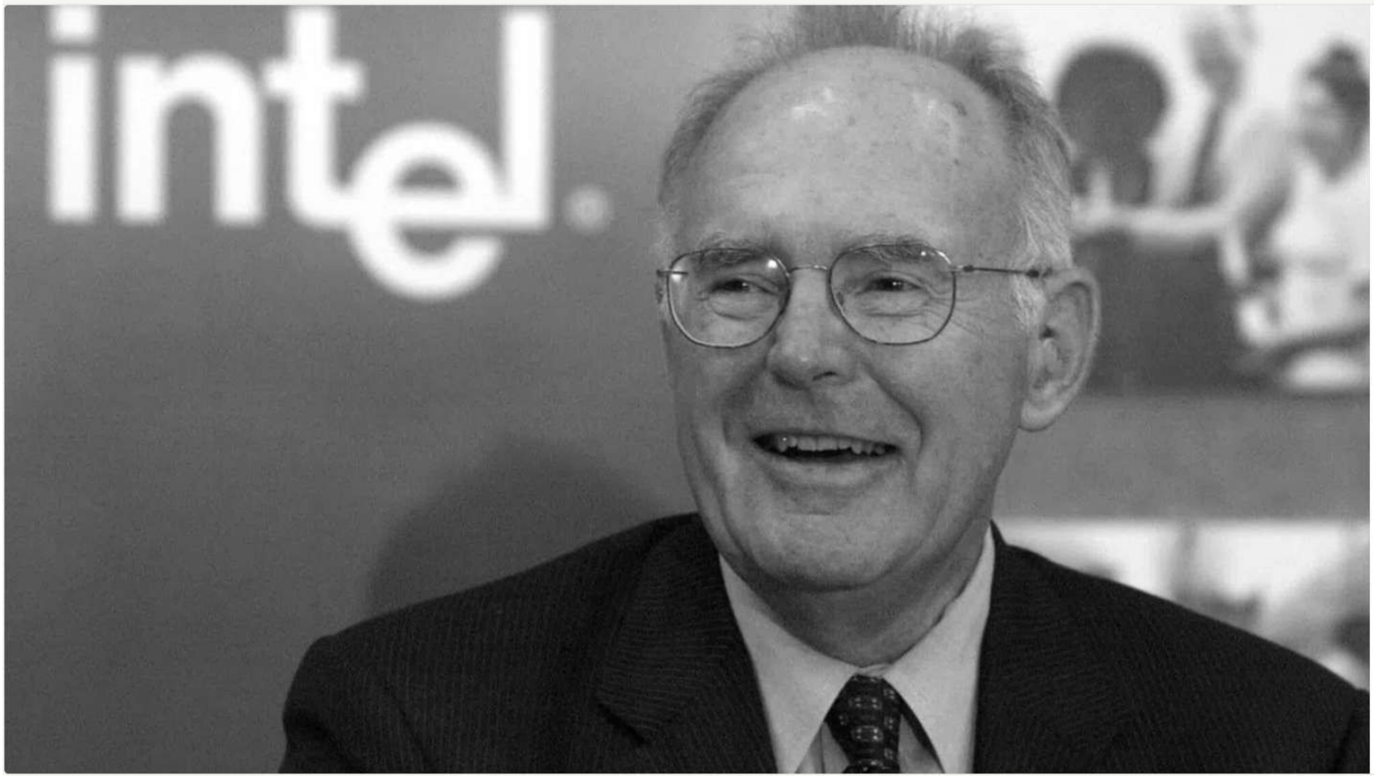
Thomas L. Friedman

Email
Share
Tweet

SAN FRANCISCO — On April 19, 1965, just over 50 years ago, Gordon Moore, then the head of research for Fairchild Semiconductor and later one of the co-founders of Intel, was asked by Electronics Magazine to submit an article predicting what was going to happen to integrated circuits, the heart of computing, in the next 10 years. Studying the trend he’d seen in the previous few years, Moore predicted that every year we’d double the number of transistors that could fit on a single chip of silicon so you’d get twice as much computing power for only slightly more money. When that came true, in 1975, he modified his prediction to a doubling roughly every two years. “Moore’s Law” has essentially held up ever since — and, despite the skeptics, keeps chugging along, making it probably the most remarkable example ever of sustained exponential growth of a technology.

RIP Gordon Moore † 24.03.2023

Gesellschaft >



Vater von Moore's Law

Intel-Mitbegründer und PC-Pionier Gordon Moore stirbt 94-jährig

Unter seiner Mit-Führung erfand Intel die Mikroprozessoren. Diese ebneten den Weg für den PC.

Aktualisiert · Mit Audio

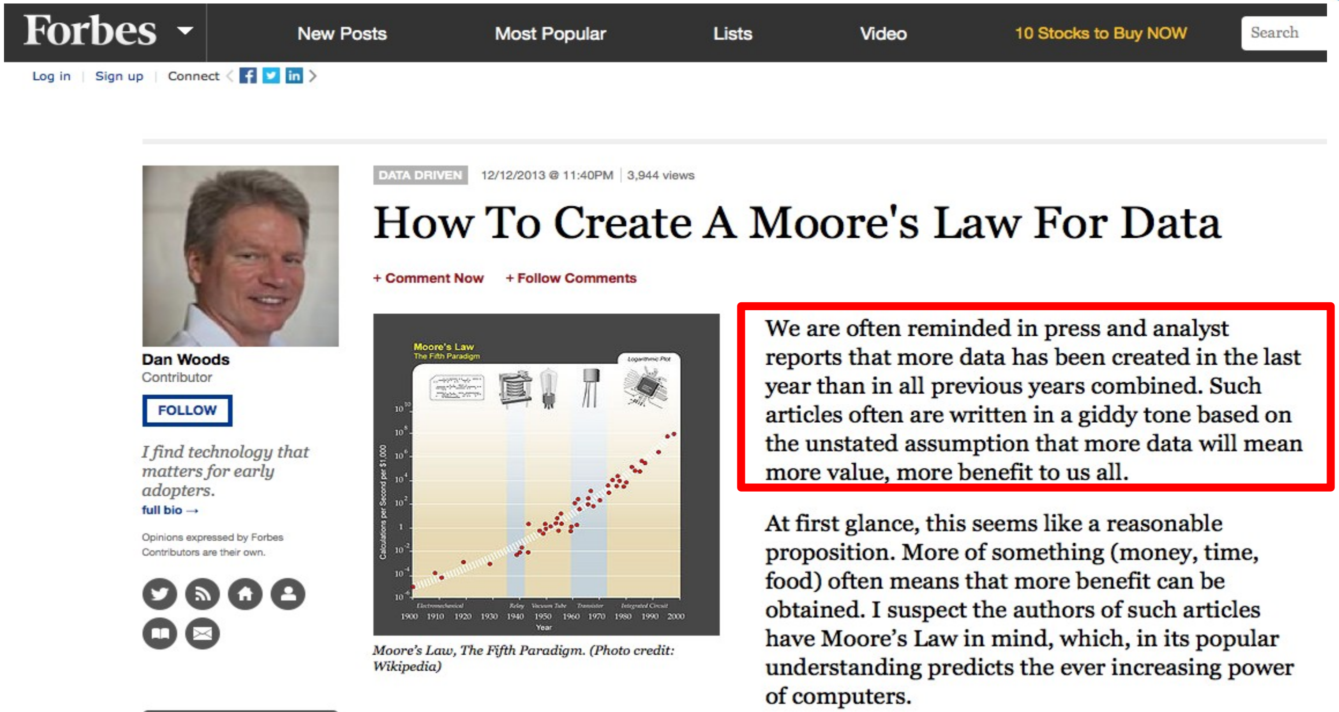
Moore's Law Applied to VW Beatle

- Thought experiment:
  - Assume a 1971 VW Beatle
  - What would be the speed and gas consumption if the VW had developed according to Moore's law?

Moore’s Law Applied to VW Beatle

- Thought experiment:
  - Speed: ~ 480,000 km/h
  - Gas consumption: ~ 0.00011875 liters per 100 km

Moore’s Law - Data



Forbes | New Posts | Most Popular | Lists | Video | 10 Stocks to Buy NOW | Search

Log in | Sign up | Connect

Dan Woods
Contributor
FOLLOW

I find technology that matters for early adopters.
full bio →

Opinions expressed by Forbes Contributors are their own.

DATA DRIVEN 12/12/2013 @ 11:40PM | 3,944 views

How To Create A Moore's Law For Data

+ Comment Now + Follow Comments

Moore's Law, The Fifth Paradigm. (Photo credit: Wikipedia)

We are often reminded in press and analyst reports that more data has been created in the last year than in all previous years combined. Such articles often are written in a giddy tone based on the unstated assumption that more data will mean more value, more benefit to us all.

At first glance, this seems like a reasonable proposition. More of something (money, time, food) often means that more benefit can be obtained. I suspect the authors of such articles have Moore’s Law in mind, which, in its popular understanding predicts the ever increasing power of computers.

But a closer look at the world of data shows that there is no Moore’s Law in effect.

[..] More data means more costs for storage, for governance and having too much unorganized data may make it more difficult to find what you need. In other words more data can mean less value.

Source: http://www.forbes.com/sites/danwoods/2013/12/12/how-to-create-a-moores-law-for-data/

Hardware Trends

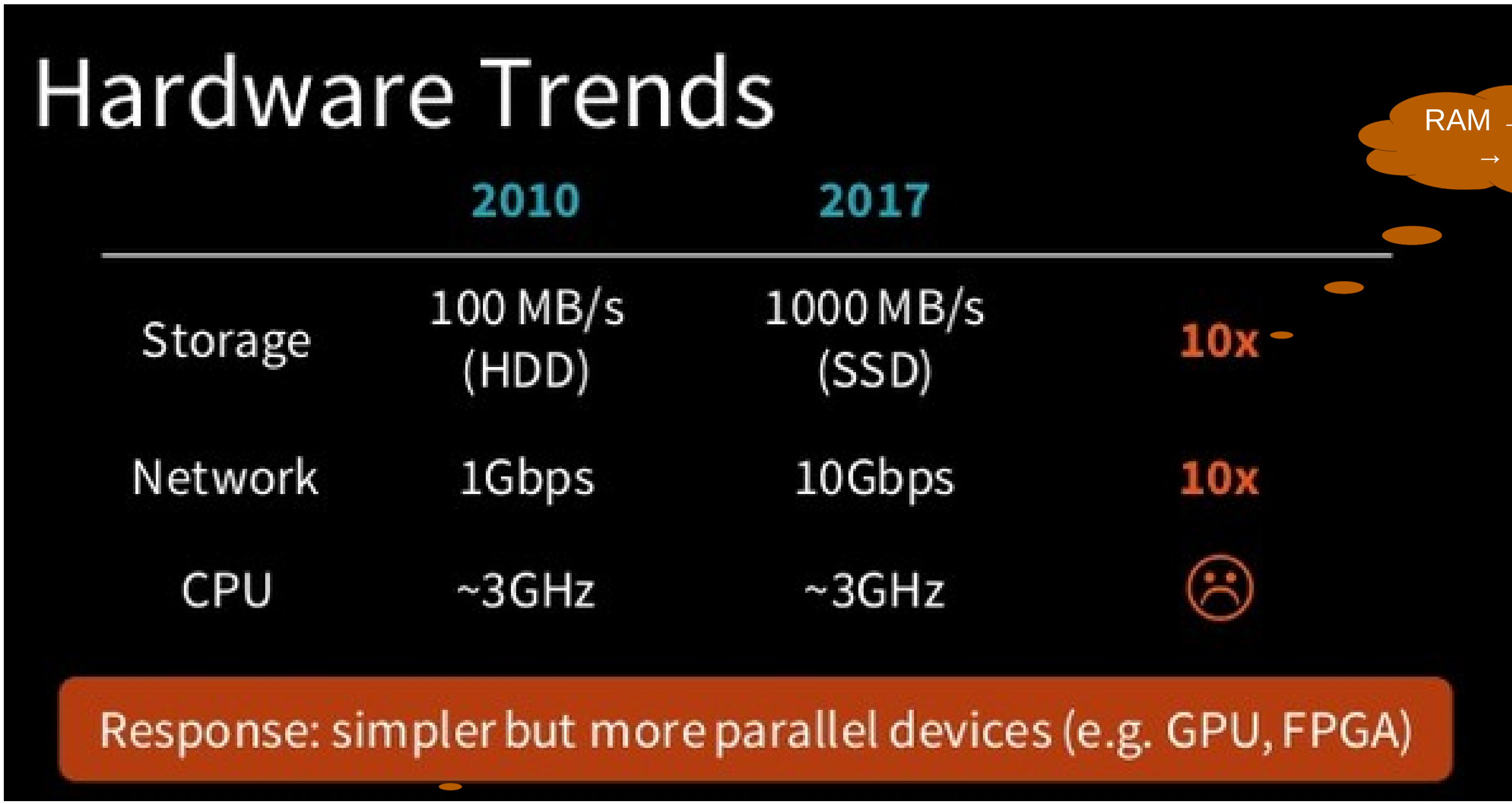
Hardware Trends

	2010	2017	*2023
Storage	100 MB/s (HDD)	1000 MB/s (SSD)	
Network	1Gbps	10Gbps	
CPU	~3GHz	~3GHz	Intel® Core™ i9-13900K Processor 36M Cache, up to 5.80 GHz

(overclocked at -250°C: up to 9 GHz)

Matei Zaharia, Spark Summit East 2018

Hardware Trends #2



Hardware Trends

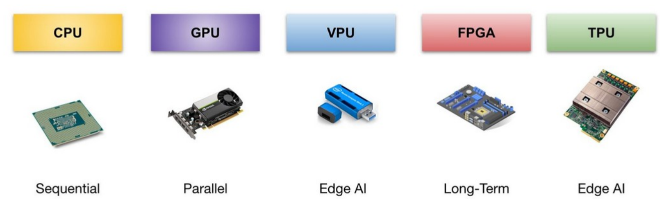
	2010	2017	
Storage	100 MB/s (HDD)	1000 MB/s (SSD)	10x
Network	1Gbps	10Gbps	10x
CPU	~3GHz	~3GHz	☹

Response: simpler but more parallel devices (e.g. GPU, FPGA)

RAM → Pmem → SSD

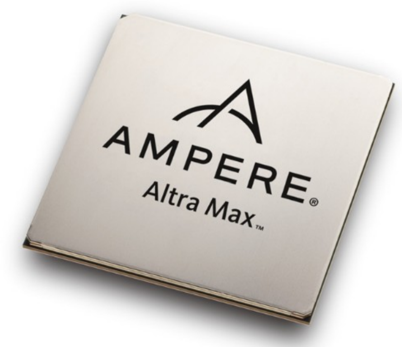
Matei Zaharia, Spark Summit East 2018

CPU, GPU, TPU, DPU, VPU, FPGA, ...



New Hardware Trend: Multicore

How to exploit (homogeneous/heterogeneous) parallelism easily?

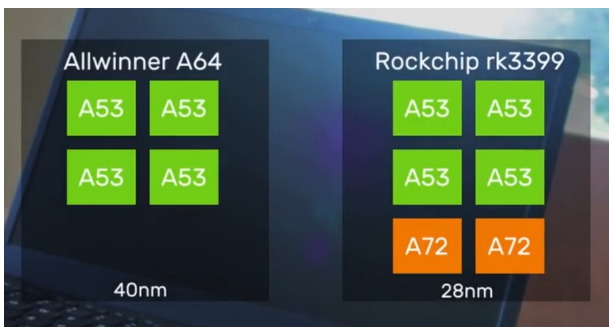


128 cores

Ampere® Altra® Max 64-Bit Multi-Core Processor

Designed to meet the requirements of modern datacenters, Ampere Altra Max delivers predictable performance, high scalability, and power efficiency for datacenter deployments from hyperscale cloud to the edge cloud.

Drive efficiency in your datacenter infrastructure workloads, including data analytics, artificial intelligence, database storage, telco stacks, edge computing, and web hosting.



Intel says AI is overwhelming CPUs, GPUs, even clouds – so all Meteor Lakes get a VPU

Movidius tech already sprinkled on some 13th-gen Core silicon goes mainstream in next generation

… on-demand, whenever we need even more parallelism?

Zurich University
of Applied Sciences

Scaling Beyond Hardware Boundaries

Application level	Not suitable - why mingle business logic with resource management?
Middleware level	Biggest potential - flexible interface between fast-paced apps & underlying OS/resources
OS level	Some approaches (e.g. MOSIX) but no widespread support in languages, tools, providers

Systems?
- SQL
- NoSQL
- MapReduce
- ...

The New Kid on the Block: NoSQL

- What is NoSQL?

NoSQL – Not Only SQL: New Paradigm for Big Data

- Traditional databases don't scale for Big Data:
  - Good performance, when indexes are used. If not…
  - Not efficient for many small inserts
  - Consistency guarantees not necessary for all types of problems

- Adding faster or larger hardware is not a solution

- About 80% of data in a company is unstructured and not suitable for relational database management systems (RDBMS)

- New solution: NoSQL (Not Only SQL):
  - Focus on horizontal scalability (shared nothing architecture)
  - Support only a subset of traditional RDBMS
  - Data is stored as key-value pairs rather than tables

Examples of NoSQL Systems (in wider sense)

Year	System/ Paper	Scale to 1000s	Primary Index	Secondary Indexes	Transactions	Joins/ Analytics	Integrity Constraints	Views	Language/ Algebra	Data model	my label
1971	RDBMS	o	✔	✔	✔	✔	✔	✔	✔	tables	sql-like
2003	memcached	✔	✔	o	o	o	o	o	o	key-val	nosql
2004	MapReduce	✔	o	o	o	✔	o	o	o	key-val	batch
2005	CouchDB	✔	✔	✔	record	MR	o	✔	o	document	nosql
2006	BigTable/Hbase	✔	✔	✔	record	compat. w/MR	/	o	o	ext. record	nosql
2007	MongoDB	✔	✔	✔	EC, record	o	o	o	o	document	nosql
2007	Dynamo	✔	✔	o	o	o	o	o	o	ext. record	nosql
2008	Pig	✔	o	o	o	✔	/	o	✔	tables	sql-like
2008	HIVE	✔	o	o	o	✔	✔	o	✔	tables	sql-like
2008	Cassandra	✔	✔	✔	EC, record	o	✔	✔	o	key-val	nosql
2009	Voldemort	✔	✔	o	EC, record	o	o	o	o	key-val	nosql
2009	Riak	✔	✔	✔	EC, record	MR	o			key-val	nosql
2010	Dremel	✔	o	o	o	/	✔	o	✔	tables	sql-like
2011	Megastore	✔	✔	✔	entity groups	o	/	o	/	tables	nosql
2011	Tenzing	✔	o	o	o	o	✔	✔	✔	tables	sql-like
2011	Spark/Shark	✔	o	o	o	✔	✔	o	✔	tables	sql-like
2012	Spanner	✔	✔	✔	✔	?	✔	✔	✔	tables	sql-like
2013	Impala	✔	o	o	o	✔	✔	o	✔	tables	sql-like

2006 Hadoop

2023 still relevant?

MapReduce

- Who has heard about it?

History of MapReduce

- 2003 Google publishes paper on Google File System (GFS)

- 2004 Google publishes MapReduce (MR) programming paradigm based on GFS:
  - GFS and MR written in C++ (closed source)
  - Python and Java-APIs only for Googlers

- 2006: Yahoo works on Hadoop:
  - Open source Java-implementation of MR

- 2008: Hadoop is released as independent Apache project

Properties of MapReduce

- Enables parallel and fault-tolerant data processing on commodity hardware

- `map()` and `reduce()` functions taken from functional programming

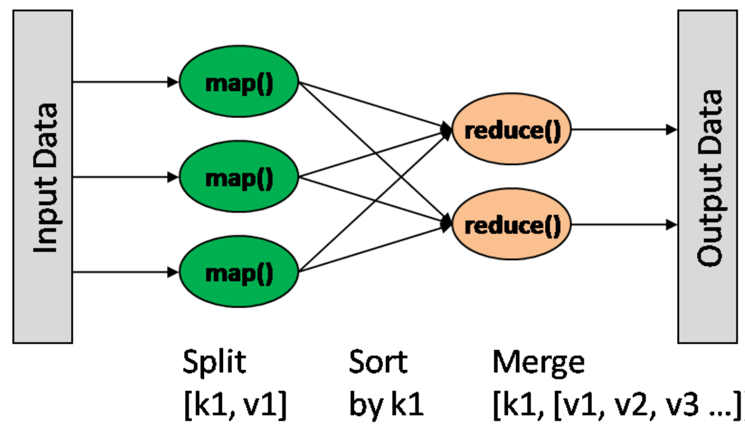
  but also including semantically similar functions: filter =~ map, join =~ reduce ...

MapReduce Programming Model

- Read data → map-phase:
  - Extract data out of each record
  - Map: [key, value] → list([another key, another value])
  - Every node in cluster processes parts of the data

- Sort data:
  - Group by keys

- Write results → reduce-phase:
  - Aggregation of data
  - Reduce: [key, list(value)] → [key, value]
  - Every node processes parts of the keys



Example: Calculate Number of Words in Documents

- Assume we have three documents D1, D2 and D3 with the following content:

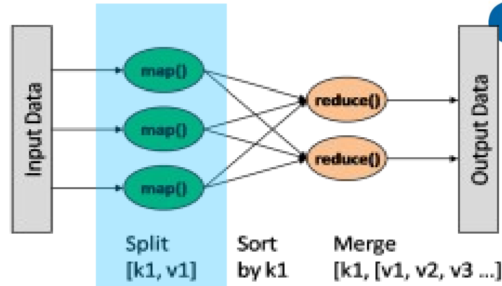
- D1: Heute ist Montag.
- D2: Heute ist Sechseläuten in Zürich.
- D3: Kann der Böögg das Wetter vorhersagen?

- The problem shall be solved with three worker nodes (that may be deployed in Uri).




Map Output of Each Worker-Node: (Key: Value)-Pair

Zurich University of Applied Sciences



(D1: Heute ist Montag.)
(D2: Heute ist Sechseläuten in Zürich.)
(D3: Kann der Böögg das Wetter vorhersagen?)

Worker 1:
(Heute: 1), (ist: 1), (Montag: 1)

Worker 2:
(Heute: 1), (ist: 1), (Sechseläuten: 1), (in: 1), (Zürich: 1)

Worker 3:
(Kann: 1), (der: 1), (Böögg: 1), (das: 1), (Wetter: 1), (vorhersagen: 1)

Reduce Input (Sorted alph.)

Worker 1:
(Böögg: 1)
(das: 1)
(der: 1)
(Heute: 1), (Heute: 1)

Worker 2:
(in: 1)
(ist: 1), (ist: 1)
(Kann: 1)
(Montag: 1)
(Sechseläuten: 1)

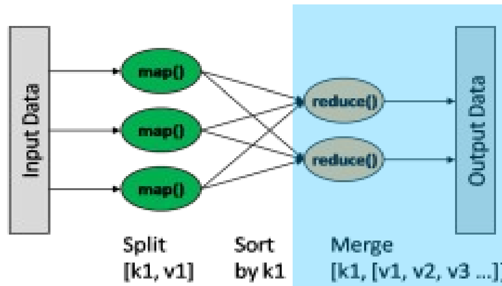
Worker 3:
(vorhersagen: 1)
(Wetter: 1)
(Zürich: 1)

Reduce Output (Sorted + aggr.)

Worker 1:
(Böögg: 1)
(das: 1)
(der: 1)
(Heute: 2)

Worker 2:
(in: 1)
(ist: 2)
(Kann: 1)
(Montag: 1)
(Sechseläuten: 1)

Worker 3:
(vorhersagen: 1)
(Wetter: 1)
(Zürich: 1)



MapReduce: Parallel Programming

- Main requirements for efficient usage of MapReduce:
  - “Large” amounts of data
  - Processing of independent tasks

- Which problems would you solve with MapReduce and which ones not?

MapReduce vs. Traditional RDBMS

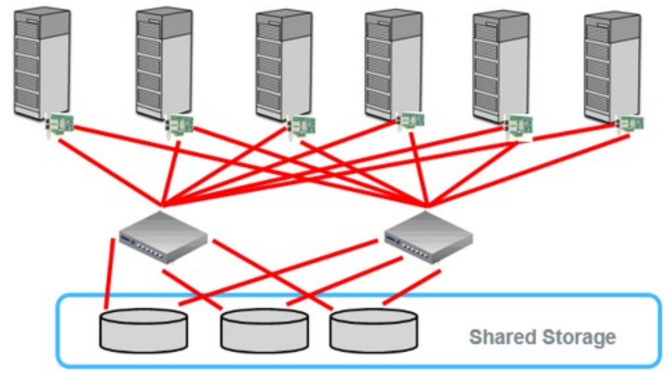
	MapReduce	Traditional RDMS
Data volume	Terabytes - Petabytes	Gigabytes - Terabytes
Access	Batch	Interactiv and batch
Updates	Write once, read many times	Read and write many times
Structure	Dynamic schema	Static schema
Integrity	Low	High (normalized data)
Scaling	Linear	Non-linear

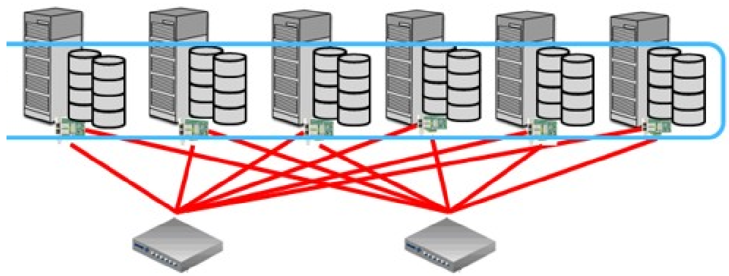
Hadoop

- Open source Apache project for scalable, fault-tolerant and distributed programming
- Implementation of MapReduce paradigm [Hadoop MapReduce part]
- Major distributors:
  - Cloudera
  - Horton Works
  - Microsoft
  - MapR

Major Storage Architectures

- Shared storage:
  - Central data sever
  - Storage attached network (SAN)
  - Commonly used in data center and cloud
  - Does not scale well for big data





- Shared-noting storage:
  - Every server has local disk array
  - Scales well with HDFS
    (Hadoop Distributed File System)

Network (red) : 10GbE => ca. 1000 MB/s throughput
1 disk : 100-200 MB/s throughput
1 SSD : 500-2000 MB/s throughput

Architecture Goals of Hadoop #1

- Resilient against hardware failures:
  - Data distribution and replication across nodes
  - Automatic error detection and correction

- Management of large data:
  - File sizes of Gigabytes to Terabytes
  - Millions of files per instances

- Simple coherence model:
  - Write-once-read many access
  - Data can't be changed an more

- Portability:
  - Across hardware and software

Architecture Goals of Hadoop #2

- Linear scalability:
  - More nodes can perform more work within the same time
  - Linear in data size and resources

- Computation near data:
  - Minimize expensive data transfer across network
  - Big data, small programs
  - „Bring the code to the data“ / „Processing at the edge“

- Streamed data access:
  - Avoid random access
  - Read large data blocks

Hadoop Usage

- Good:
  - Large data
    - File can be larger than disk
  - Streaming (write-once-read-many):
    - 128 MB blocks
  - Commodity hardware (fault tolerant)

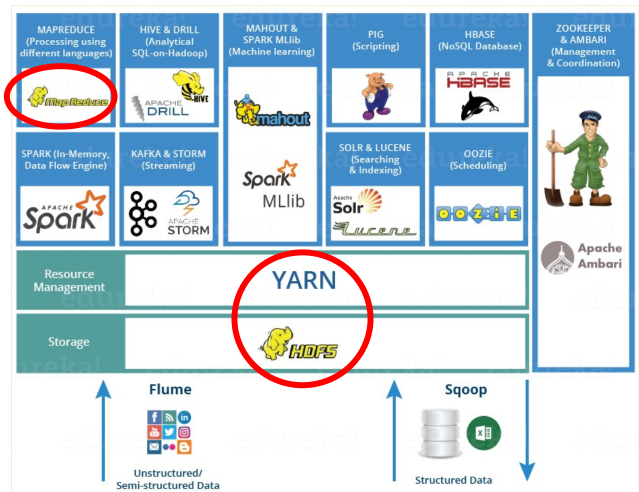
- Bad:
  - Many small files
  - Low latency access (fast response times)
  - Many inserts at different positions in file (random access)

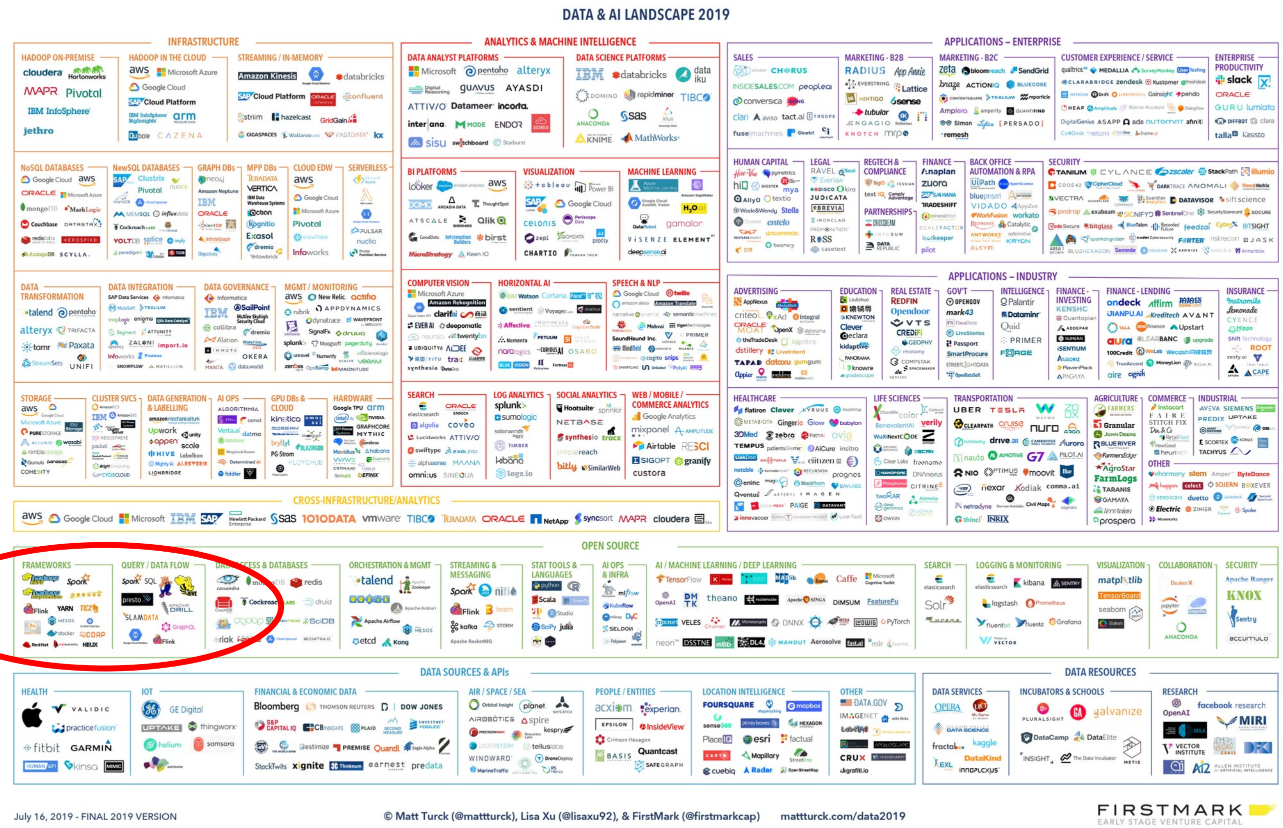
Hadoop Family

Name	Description
Pig	High-level Data Flow Language and parallelel execution framework
Hive	Distributed data warehouse
HBase	Distributed, column-oriented database
Zookeeper	Distributed coordination service
Scoop	Bulk transfer beteween RDBMS (structured data) und HDFs
Mahout	Machine Learning Library
BigTop	Packaging and testing

Hadoop Today

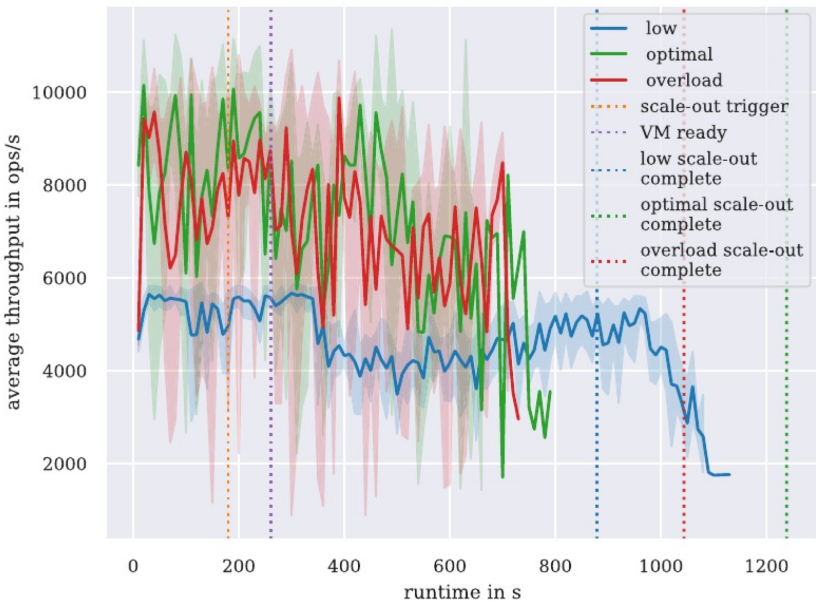
- "Grand father" of open source big data technology
- Main principles remain but better implementation

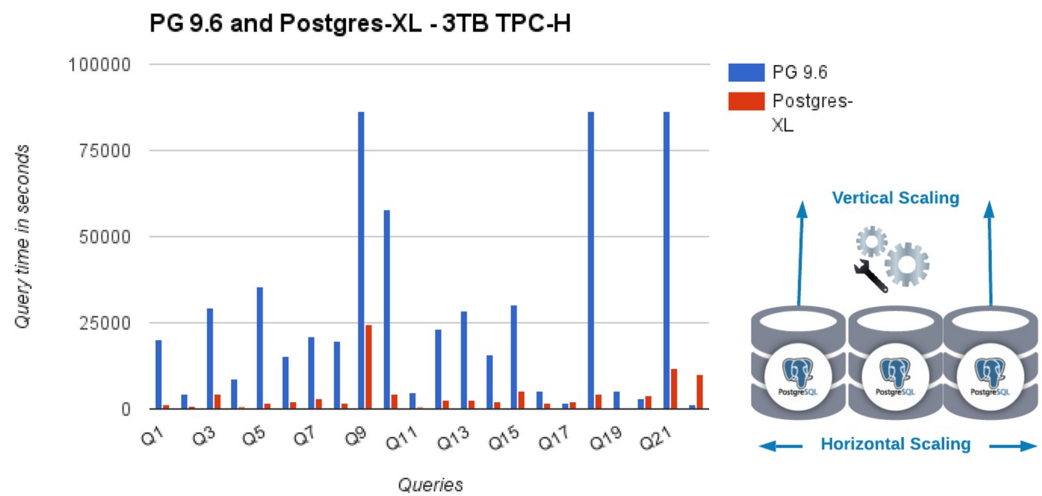




(RDBMS Today)

- Horizontally scalable + “cloud-native“ database management
  - semi-ACID due to CAP theorem
- MySQL/MariaDB Galera, PostgreSQL hot standby, containers (Crate), ...
- but: Big Data requirements beyond relations (streams, pipelines, models, time series, semantics/business objects...)





[Seybold et al. Kaa: Evaluating Elasticity of Cloud-Hosted DBMS, CloudCom 2019: Couchbase elastic scale-out causes too much overhead]

Programming Perspective

Why do I need to learn Big Data tools? I would just like to continue in Python...
... but still go scalable!
→ Try this out in Python, no external libraries needed.

```python
import multiprocessing as mp
import time
import math

def heavycompute():
    for i in range(99999999):
        i += math.sqrt(i)

print(f"Processing on {mp.cpu_count()} cores...")
procs = []
for i in range(mp.cpu_count()):
    procs.append(mp.Process(target=heavycompute))
    procs[-1].start()

t = time.time()
for proc in procs:
    proc.join()
print(f"Processing finished in {time.time() - t:.1f}s.")
```

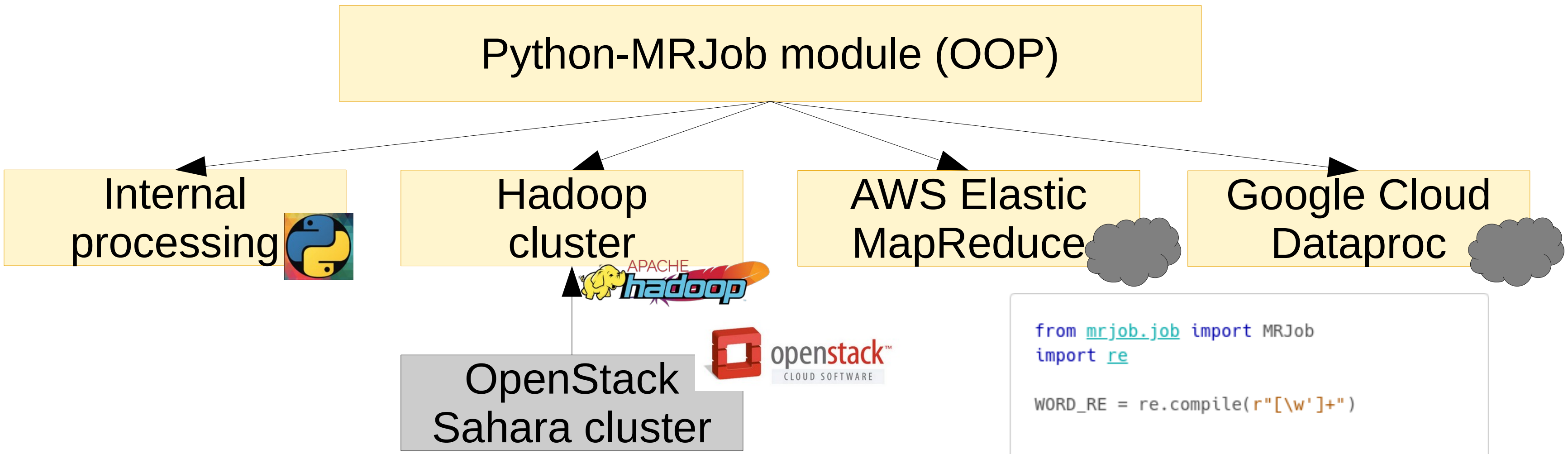
Programming – Multiprocessing Results

- All CPU cores maxed out. Limit: 1 computer.
- Good for independent calculations, but not for workflows/correlations/reducers.
- More capable Python tools needed - e.g. MrJob, Modin/Koalas, PySpark.

```
  1  [||||||||||||||||||||||||100.0%]    4  [||||||||||||||||||||||||100.0%]    7  [||||||||||||||||||||||||100.0%]
  2  [||||||||||||||||||||||||100.0%]    5  [||||||||||||||||||||||||100.0%]    8  [||||||||||||||||||||||||100.0%]
  3  [||||||||||||||||||||||||100.0%]    6  [||||||||||||||||||||||||100.0%]    9  [||||||||||||||||||||||||100.0%]
  Mem[|||||||||||||||||||||||||||||||||||||||||||||||||            5.61G/31.0G]   Tasks: 194, 791 thr; 12 running
  Swp[                                                                0K/976M]   Load average: 7.39 2.69 1.62
                                                                                 Uptime: 14 days, 11:48:05

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
11284 spio       20   0 17124  9392  3880 R 99.8  0.0  0:05.18 python3 mp.py
11285 spio       20   0 17124  9396  3880 R 99.8  0.0  0:04.62 python3 mp.py
11276 spio       20   0 17124  9388  3880 R 98.6  0.0  0:05.03 python3 mp.py
11282 spio       20   0 17124  9388  3880 R 97.3  0.0  0:05.10 python3 mp.py
11283 spio       20   0 17124  9388  3880 R 96.7  0.0  0:05.06 python3 mp.py
11278 spio       20   0 17124  9388  3880 R 96.7  0.0  0:04.83 python3 mp.py
11281 spio       20   0 17124  9388  3880 R 95.4  0.0  0:05.01 python3 mp.py
11287 spio       20   0 17124  9460  3944 R 94.8  0.0  0:04.84 python3 mp.py
11280 spio       20   0 17124  9388  3880 R 93.5  0.0  0:04.99 python3 mp.py
11277 spio       20   0 17124  9388  3880 R 89.0  0.0  0:04.73 python3 mp.py
11279 spio       20   0 17124  9388  3880 R 87.8  0.0  0:04.85 python3 mp.py
11286 spio       20   0 17124  9396  3880 R 82.7  0.0  0:04.77 python3 mp.py
```

Programming – Parallelisation across Nodes

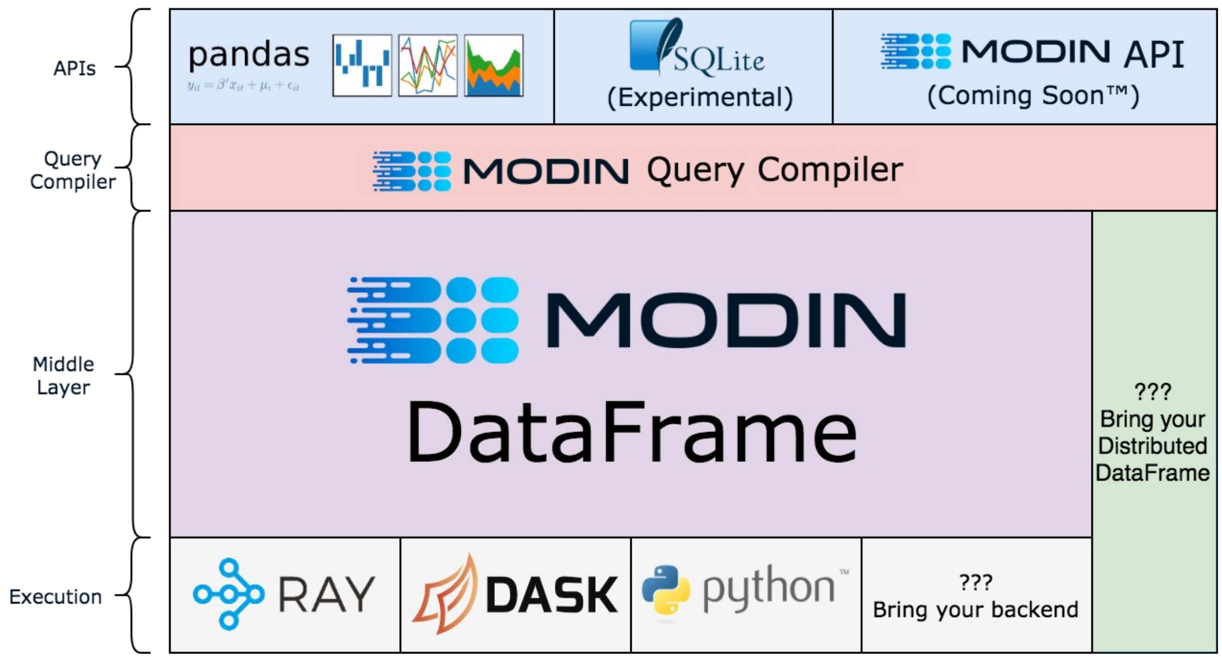
- Map-Reduce-Job external Python module




```python
from mrjob.job import MRJob
import re

WORD_RE = re.compile(r"[\w']+")


class MRWordFreqCount(MRJob):

    def mapper(self, _, line):
        for word in WORD_RE.findall(line):
            yield word.lower(), 1

    def reducer(self, word, counts):
        yield word, sum(counts)

if __name__ == '__main__':
    MRWordFreqCount.run()
```

Programming – Speeding up Pandas

- Vertical vs. horizontal scalability - vertical first: using all processors on machine
- Framework under development, sometimes still disappointing slowdowns



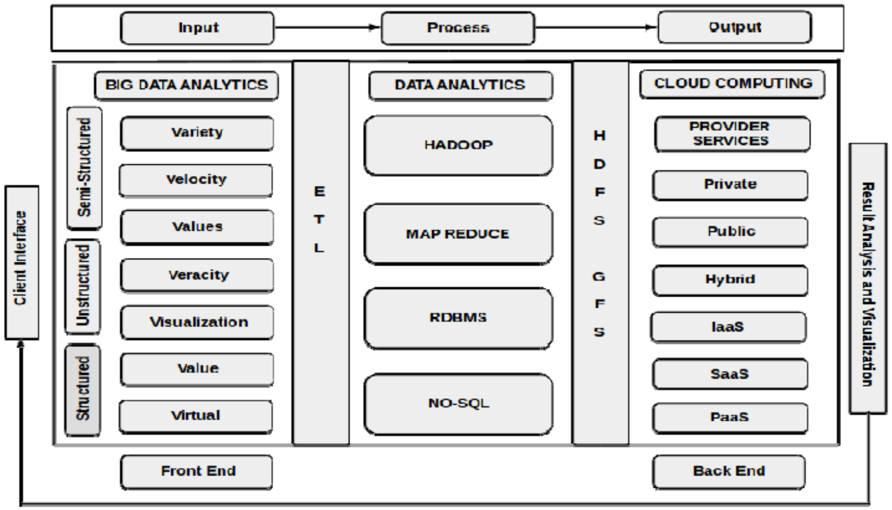
- if horizontal scalability needed: Koalas = Pandas API for Spark

… fully integrated since Spark 3.2 / October 2021

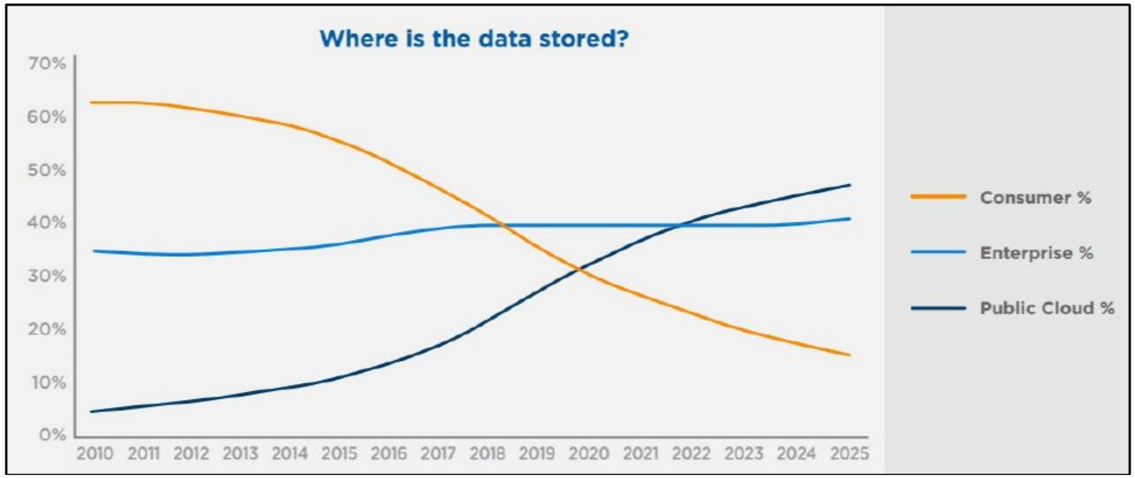
Computing Paradigms Outlook

{Parallel, Distributed, Scalable, Big Data, Cloud, HPC, Continuum, ...} Computing

- convergence, fusion, hybrid approaches...
- evolving acceptance trends and architectural patterns



arXiv:1912.10821



Remote Sens. 2020, 12, 62

- classic (Hadoop/MR) vs. hardware-optimised (Spark/Tensorflow) vs. microbatches (Spark Streaming) vs. stream-based (Flink, Kinesis etc.) vs. discrete events (FaaS/serverless CaaS)

Can we do Big Data without the Big Clouds?

DATABASES

- Bigtable
- Datastore ›
- Database Migration ›
- Firestore ›
- Memorystore ›
- Spanner
- SQL

BIG DATA

- Composer
- Dataproc ›
- Pub/Sub ›
- Dataflow ›
- IoT Core
- BigQuery ›
- Looker
- Data Catalog
- Data Fusion
- Financial Services
- Healthcare ›
- Life Sciences ›

ARTIFICIAL INTELLIGENCE

- AI Platform (Unifi... ›
- AI platform ›
- Data Labelling ›
- Document AI ›
- Natural Language
- Recommendation... ›
- Tables ›
- Talent Solution ›
- Translation ›
- Vision ›
- Video Intelligence

Conclusions

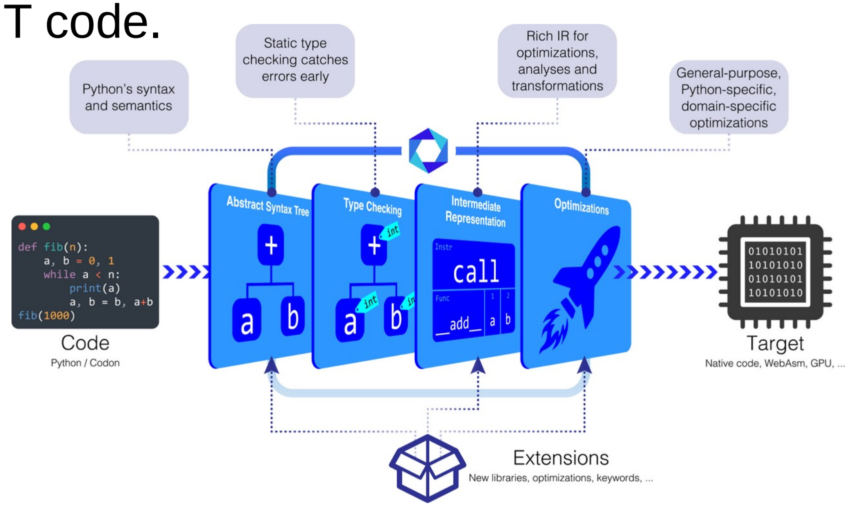
- New waves of data require new approaches
- Hadoop, MapReduce & co: new paradigm for big data processing
- HDFS: distributed, fault tolerant file system - revisited later
- Several ways to approach big data processing from Python
  - consideration of principal scalability models (multi-processor, multi-node, ...)
  - consideration of overhead for not-so-big data
  - limitation: only covering „big“ (voluminous) and „fast“ (velocity) data;

    not solving engineering/integration (variety) or quality/ground truth (veracity) issues, nor business side (value)

Takeaway – Remember Always!

Do you want faster code/cost reduction?

- Do NOT use a big data framework.
- Use modern programming facilities, e.g. Python 3.12 subinterpreters, PyPI, or Codon (CPU)/Numba (GPU) JiT code.



Do you want faster makespan?

- TRY a big data framework.
- Success enablement by horizontal scaling for reaching beyond machine limits.
  - Alternative option for just enablement: pipelining → streaming
- Wall clock time reduced by parallelism, but not lower overall compute time/cost.
- Only faster if data volume is really big, otherwise slowdown.

Takeaway – Small Data Example