Zurich University
of Applied Sciences

Information Engineering 2

Big Data

Prof. Dr. Kurt Stockinger

Semesterplan

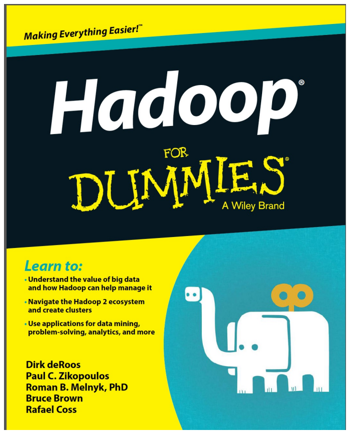
SW	Datum	Vorlesungsthema	Praktikum
1	23.02.2022	Data Warehousing Einführung	Praktikum 1: KNIME Tutorial
2	02.03.2022	Dimensionale Datenmodellierung 1	Praktikum 1: KNIME Tutorial (Vertiefung)
3	09.03.2022	Dimensionale Datenmodellierung 2	Praktikum 2: Datenmodellierung
4	16.03.2022	Datenqualität und Data Matching	Praktikum 3: Star-Schema, Bonus: Praktikum 4: Slowly Changing Dimensions
5	23.03.2022	Big Data Einführung	DWH Projekt - Teil 1
6	30.03.2022	Spark - Data Frames	DWH Projekt - Teil 2 (Abgabe: 4.4.2022 23:59:59)
7	06.04.2022	Data Storage: Hadoop Distributed File System & Parquet	Praktikum 1: Data Frames
8	13.04.2022	Query Optimization	Praktikum 2: Data Storage
9	20.04.2022	Spark Best Practices & Applications	Praktikum 3: Query Optimization & Performance Analysis
10	27.04.2022	Machine Learning mit Spark 1	Praktikum 3: Query Optimization & Performance Analysis (Vertiefung)
11	04.05.2022	Machine Learning mit Spark 2 + Q&A	Praktikum 4: Machine Learning (Regression)
12	11.05.2022	NoSQL Systems	Big Data Projekt - Teil 1
13	18.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 2
14	25.05.2022	Keine Vorlesung (Arbeit am Projekt)	Big Data Projekt - Teil 3 (Abgabe: 30.5.2022 23:59:59)

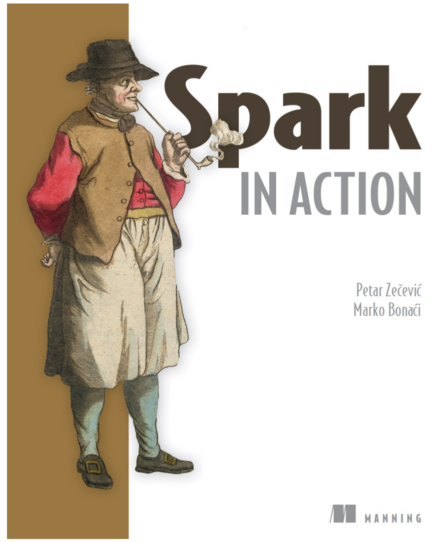
Educational Objectives for Today

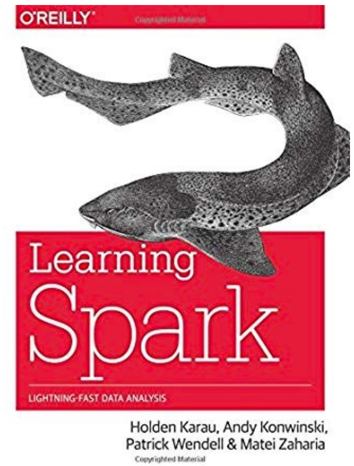
- Learn about the main definitions of big data and its historic background
- Understand the main concepts of map/reduce
- Understand fundamentals of Hadoop

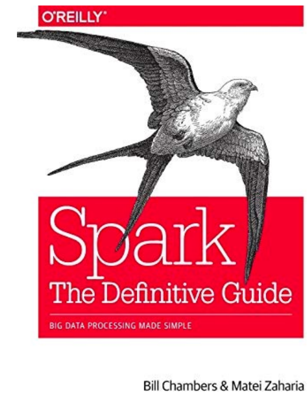
Zurich University
of Applied Sciences

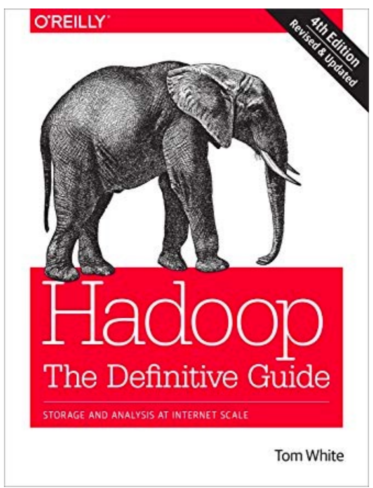
Literature

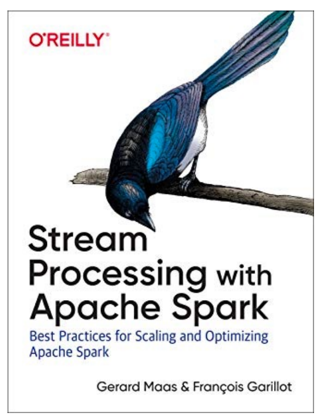


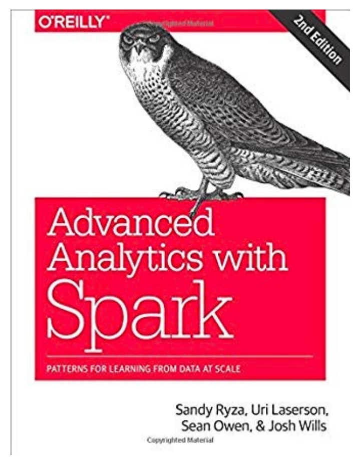










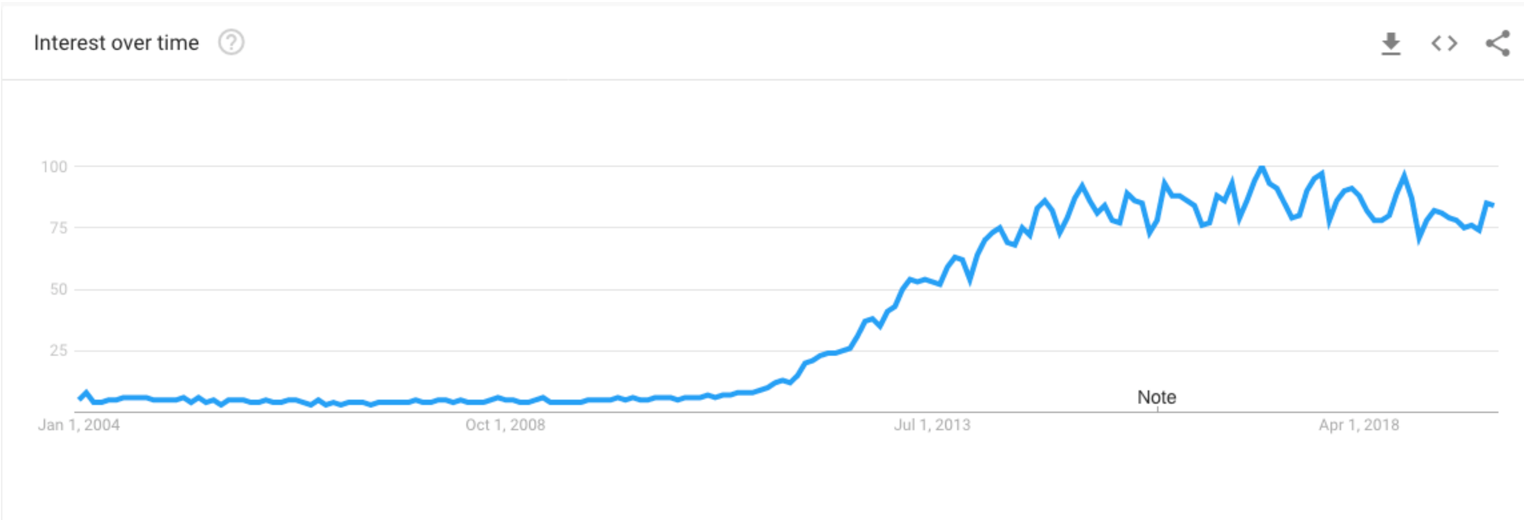


Big Data

- What is big data for you?

“Big Data” in Google Trends Worldwide

- Since 2011 we see a significant rise

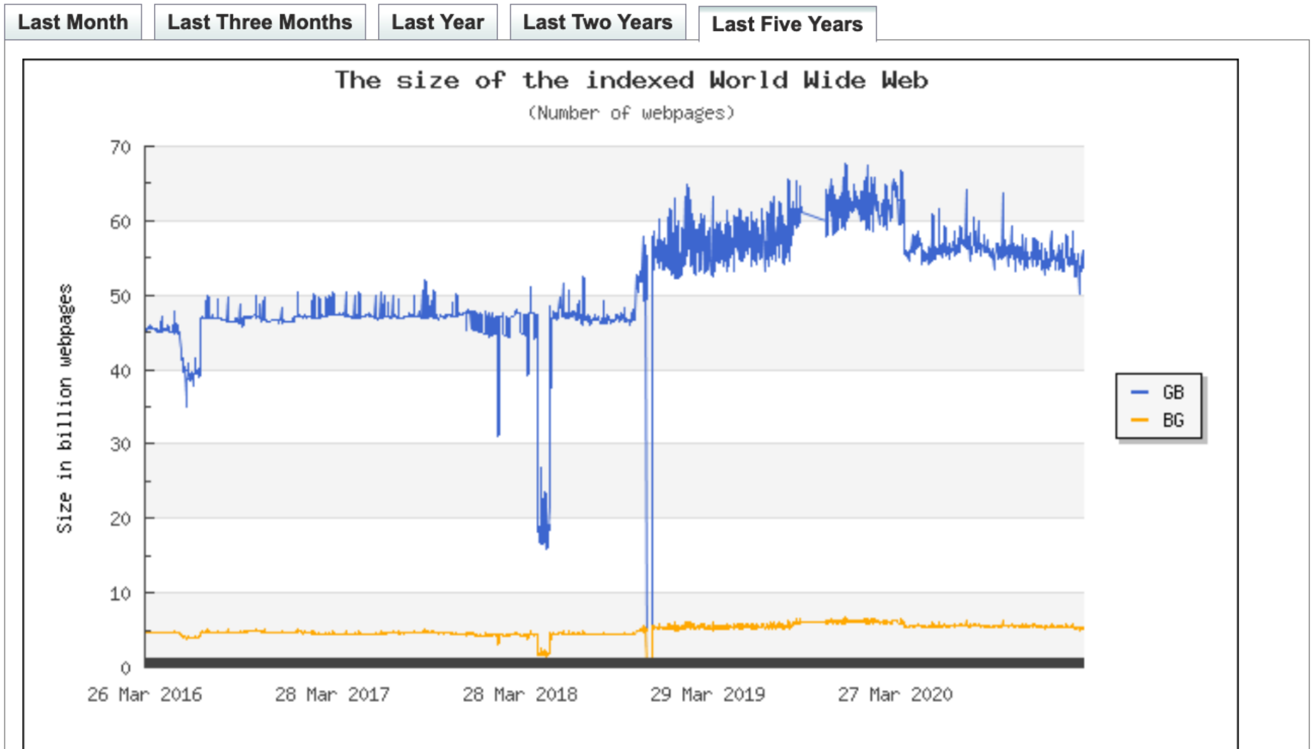


Big Data: Largest Library in the World

- Library of Congress in Washington
- Contains more than 36 million books

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

Big Data: Number of Pages on the Internet



GB = Sorted on Google and Bing

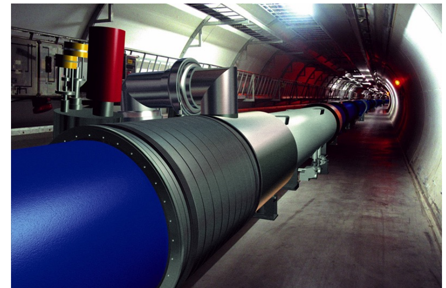
BG = Sorted on Bing and Google

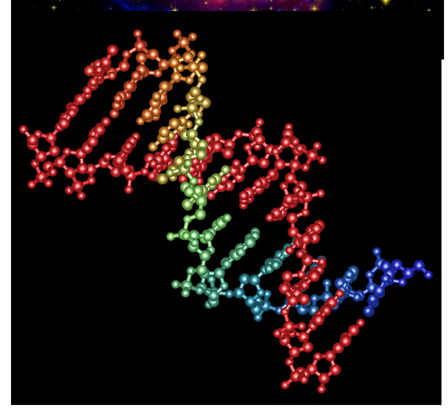
Source: http://www.worldwidewebsize.com/

- How long do you need to count now?

Major Waves of Data Tsunami

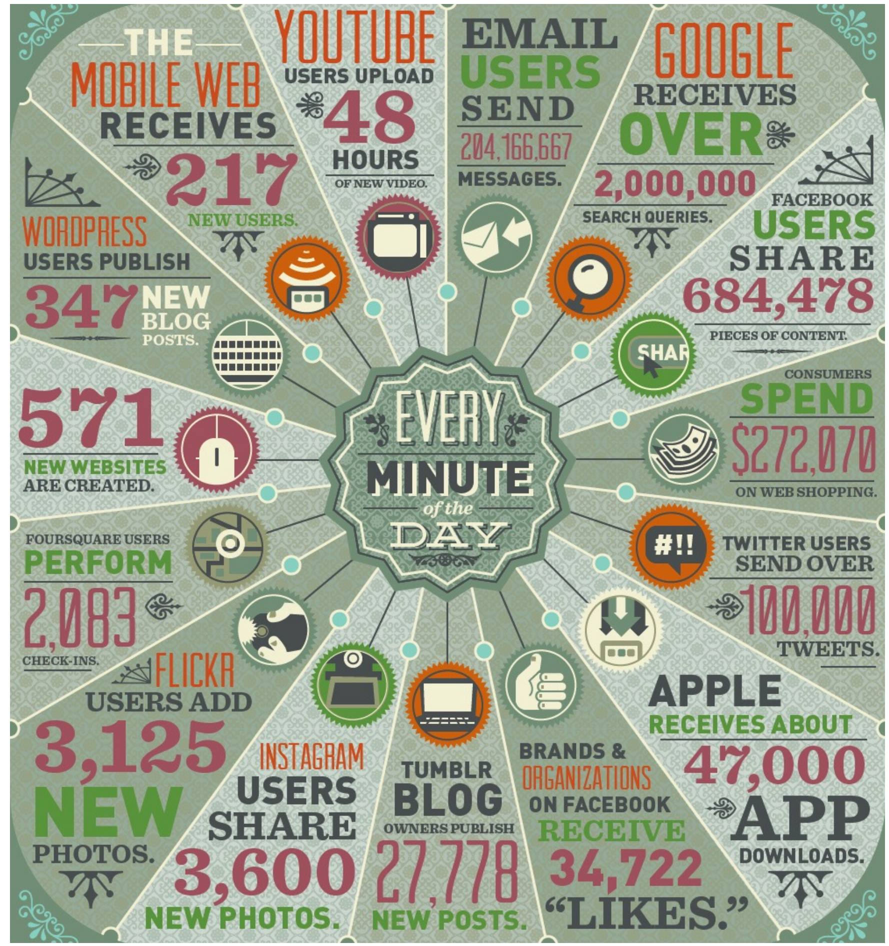
- Wave 1: Big research
  - E.g. CERN
- Wave 2: Companies
  - E.g. Amazon, eBay, Google, Yahoo
- Wave 3: Social networks
  - E.g. Facebook, Twitter, LinkedIn
- Wave 4: Machine-generated data
  - E.g. mobile phone, industry 4.0





How Large is Big Data?

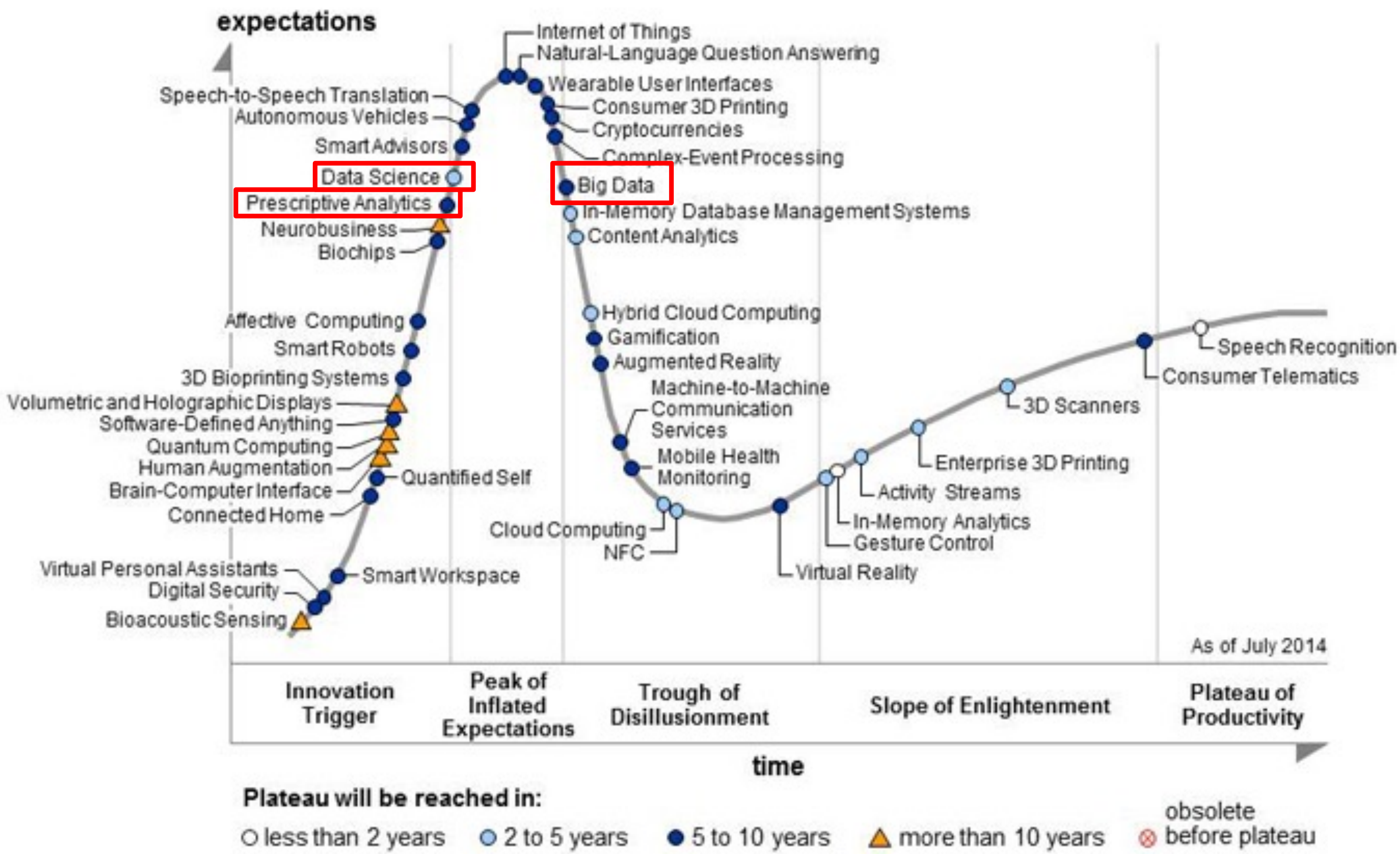
- 1 TB:
  - 1000 Gigabytes
  - 300 hours of high quality videos
  - Encyclopedia Britannica
- 10 TB:
  - All books of the Library of Congress
- 1 PB:
  - 1000 TB
  - Data of a large bank
- 100 PB:
  - Data at CERN



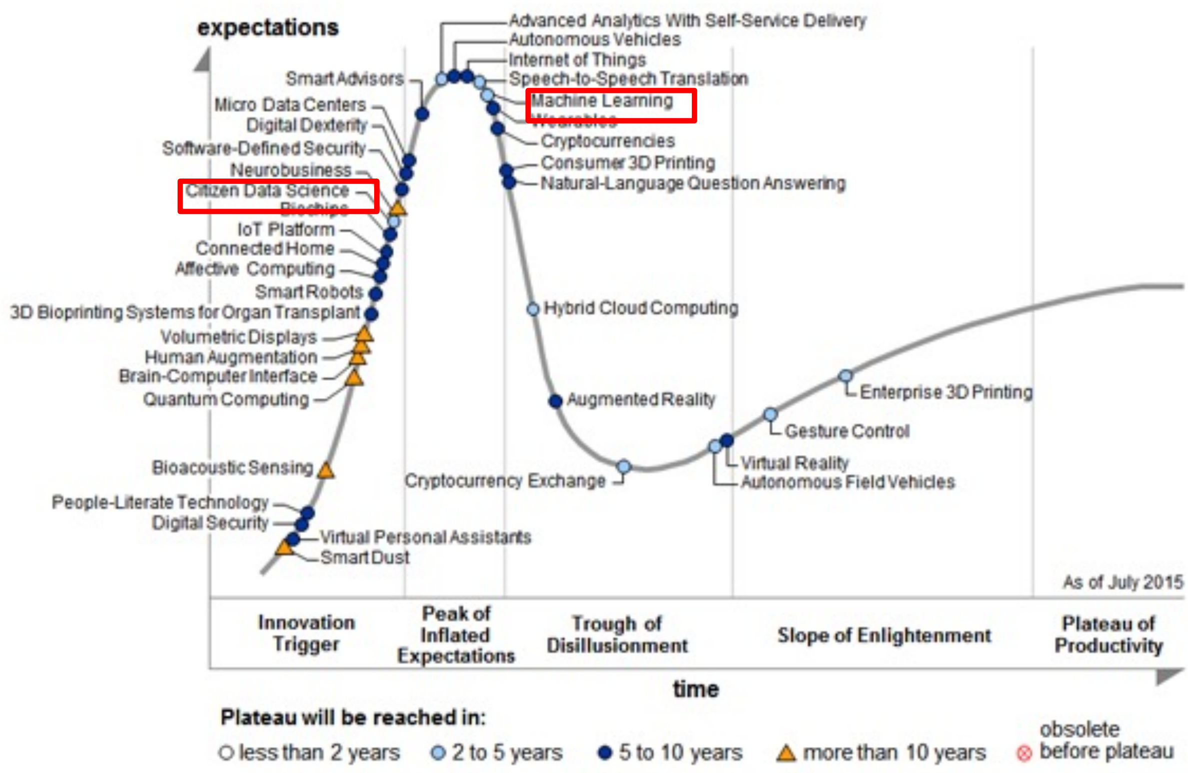
Big Data Definition – the 5Vs

- Volume (big):
  - “Large” amounts of data
- 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)
- Quizz: Which V is the hardest to manage?

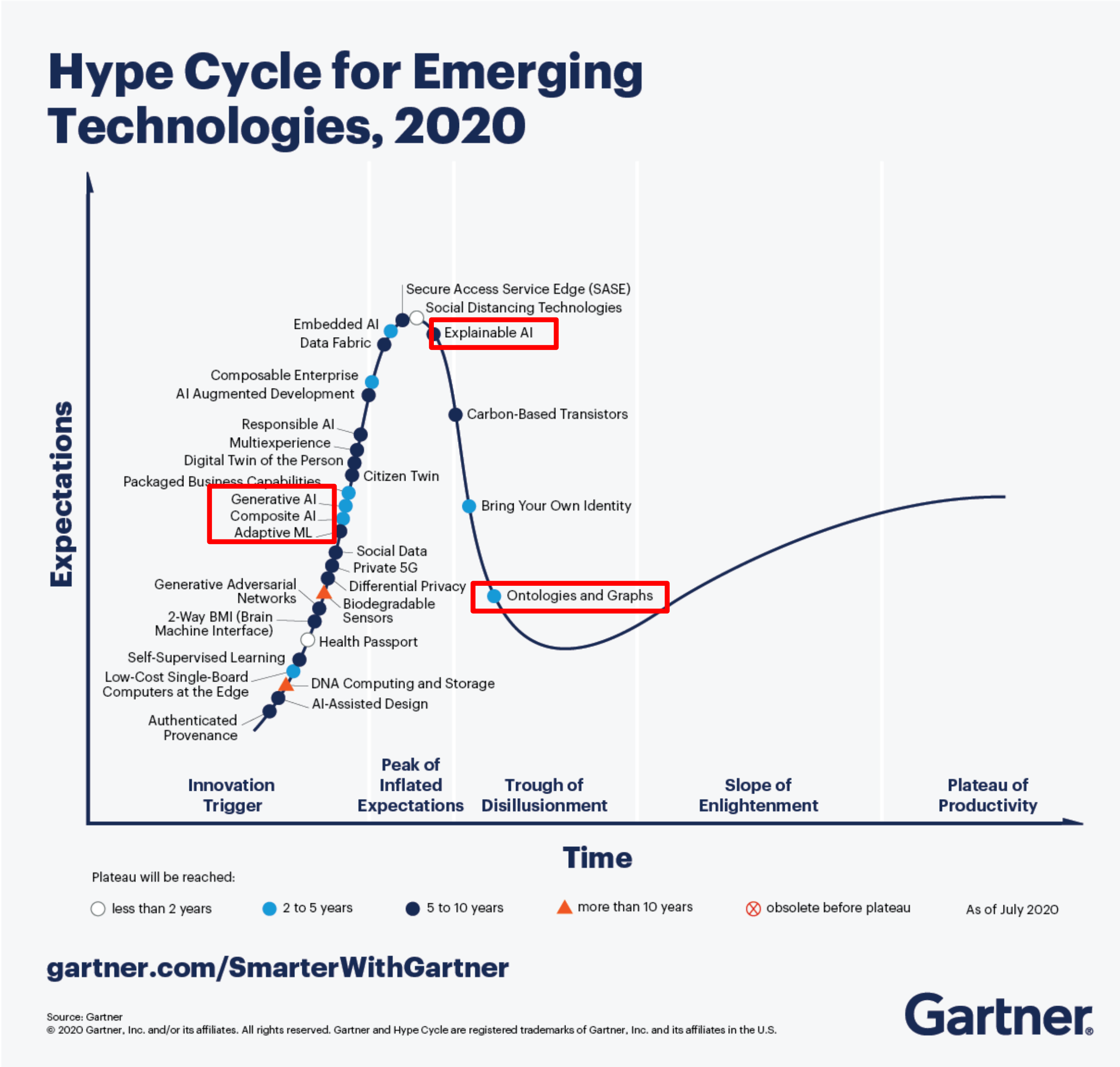
Gartner Hype Cycle for Emerging Technologies 2014



Gartner Hype Cycle for Emerging Technologies 2015



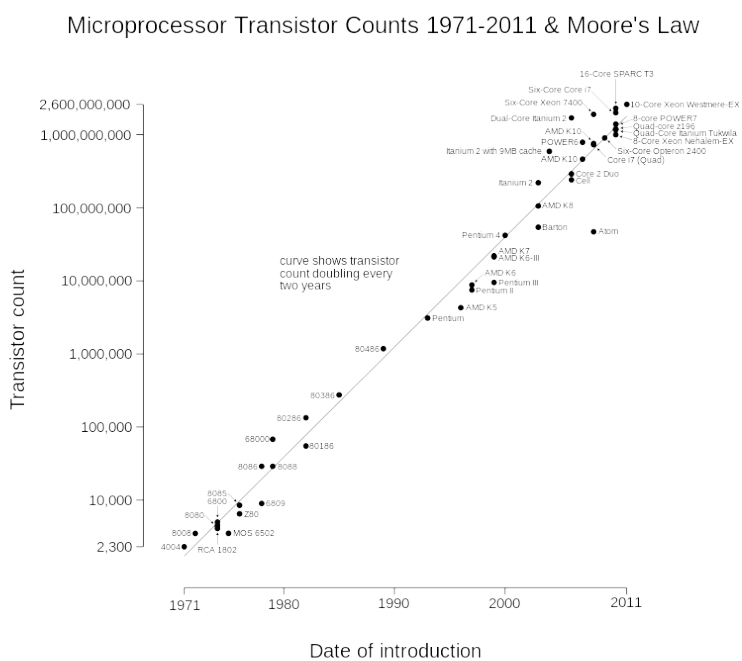
Big Data is not a hype anymore, but it is already in use



It's (mostly) all about AI

Moore's Law

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



The Opinion Pages | OP-ED COLUMNIST

Moore's Law Turns 50

MAY 13, 2015

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.

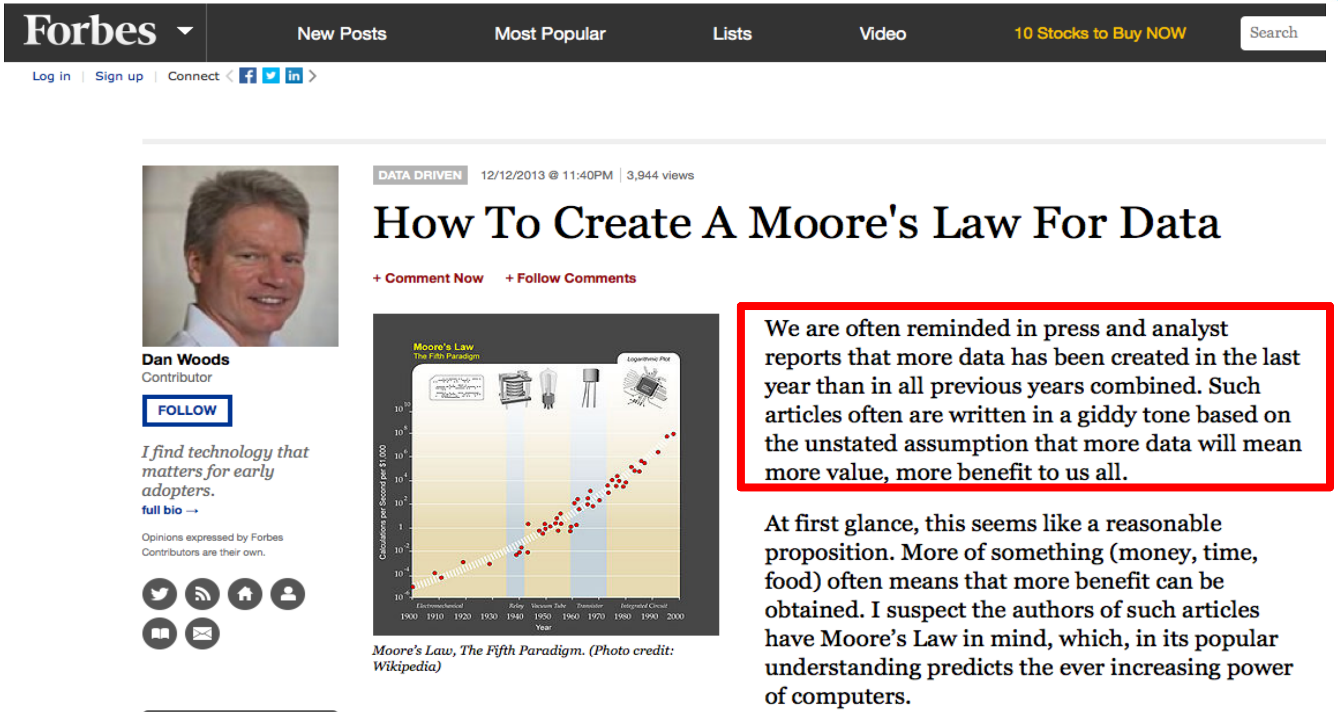
Moore's Law Applied to VW Beatle

- 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
  - Comparison:
    - SpaceX’s Falcon 9 Heavy: 39,600 km/h

Moore‘s Law - Data



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

Log in | Sign up | Connect

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

How To Create A Moore's Law For Data

+ Comment Now + Follow Comments

Dan Woods
Contributor

FOLLOW

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

Opinions expressed by Forbes Contributors are their own.

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

Zurich University
of Applied Sciences

Hardware Trends

	2010
Storage	100 MB/s (HDD)
Network	1Gbps
CPU	~3GHz

Matei Zaharia, Spark Summit East 2018

Zurich University
of Applied Sciences

Hardware Trends #2

Hardware Trends

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

Matei Zaharia, Spark Summit East 2018

Hardware Trends #3

Zurich University
of Applied Sciences

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)

Matei Zaharia, Spark Summit East 2018

Impact of Hardware Trends

In 2005-2010, I/O was the name of the game

- Network locality, compression, in-memory caching

Now, compute efficiency matters even for data-intensive apps

- And harder to obtain with more types of hardware!

Matei Zaharia, Spark Summit East 2018

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

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

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

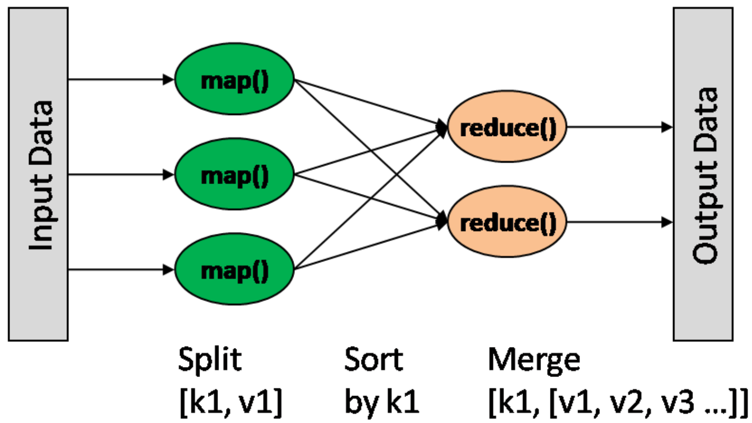
- Enables parallel and fault-tolerant data processing on commodity hardware
- Map() and reduce() functions taken from functional programming

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?

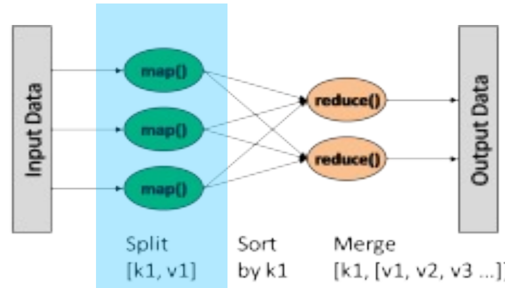
- The problem shall be solved with three worker nodes.

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

(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)

Zurich University
of Applied Sciences

Reduce Input (Sorted)

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)

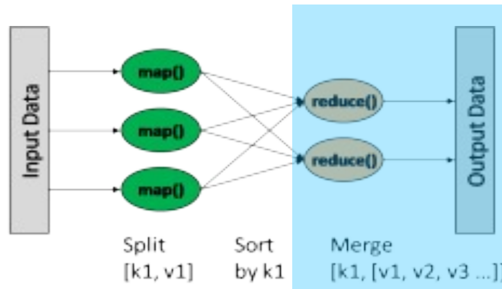
→

Reduce Output (Sorted)

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 RDBMS
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

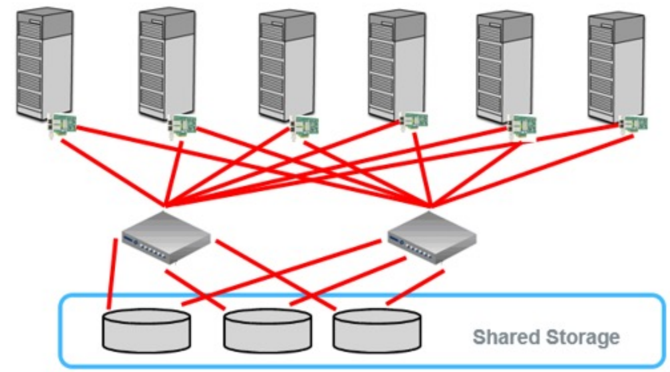
- Implementation of MapReduce paradigm

- Major distributors:

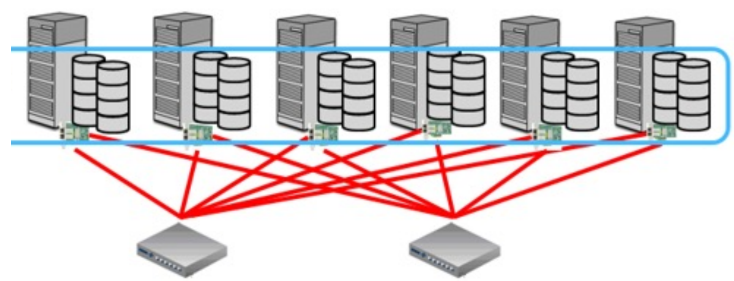
  - Cloudera
  - Horton Works
  - Microsoft
  - MapR

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 (rot) : 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)

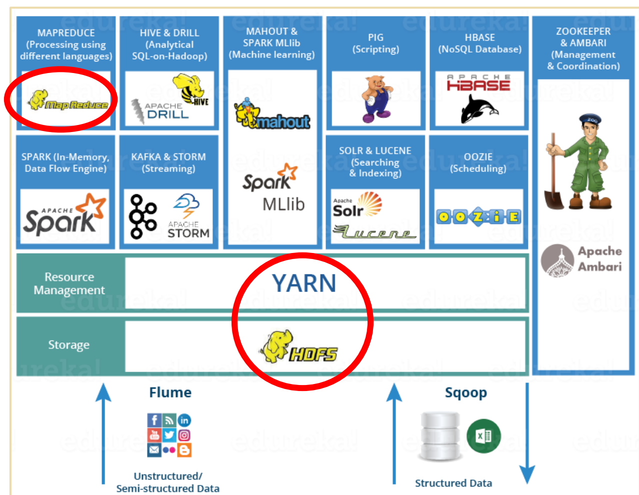
Hadoop Family

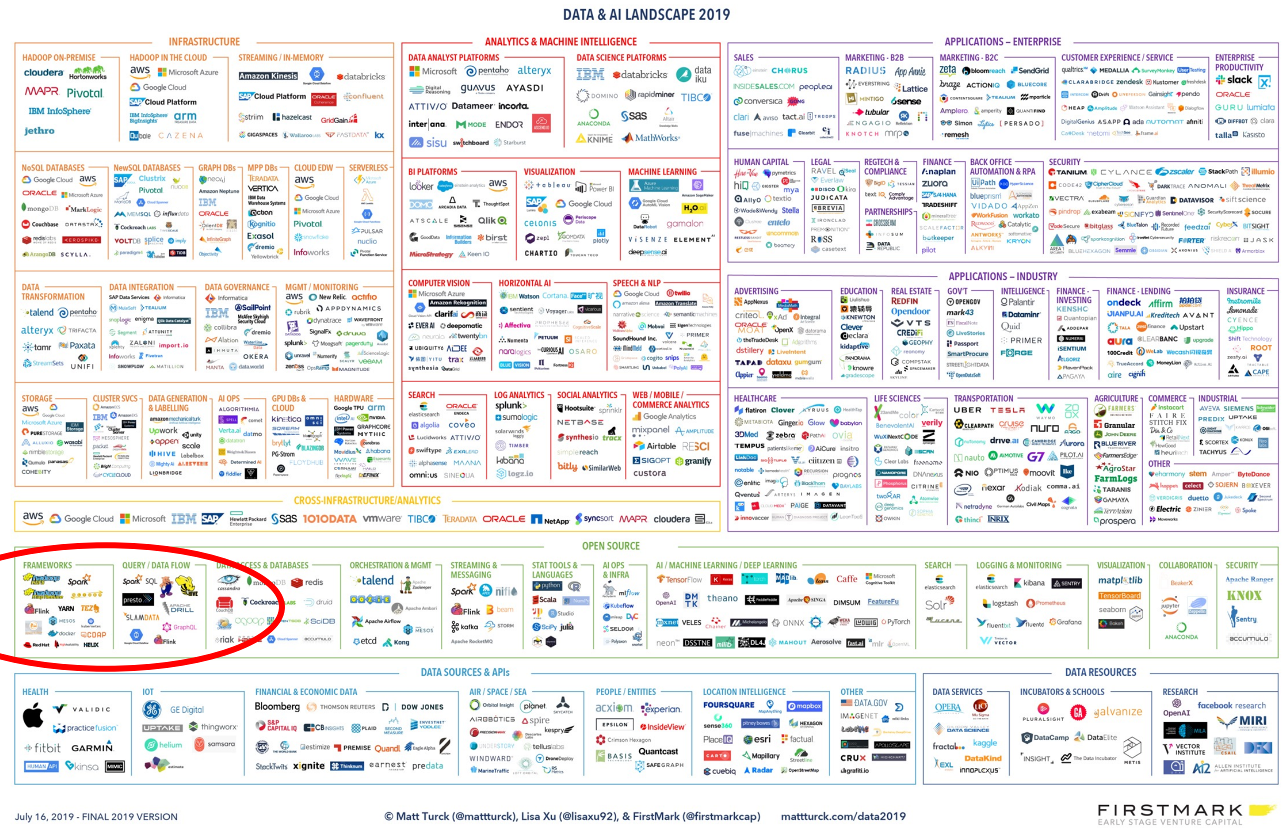
Name	Description
Pig	High-level Data Flow Language and parallelel execution framework
Hive	Distributed data warehouse
HBase	Distributed, column-oriented database
Zookeeper	Distributed coordination service
Scoop	Bulk transfer beteween RDBMS (structured data) und HDFs
Mahout	Machine Learning Library
BigTop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

Zurich University
of Applied Sciences

Hadoop Today

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





Conclusions

- New waves of data require new approaches
- Hadoop and MapReduce: new paradigm for big data processing
- HDFS: Distributed, fault tolerant file system