

DĚLAT
DOBRÝ SOFTWARE
NÁS BAVÍ

PROFINIT

B0M33BDT

Big data technologies

Agenda

- › Class intro - Important information
- › Introduction of technologies
- › Architectural overview
 - Global overview + insights
- › User view
 - How to use the technologies
- › Programmer view
 - How to develop big data applications
- › Business view
 - Data Science as a motivation for Big Data
- › Real world view



Class intro

Important links

› Courseware

- <https://cw.fel.cvut.cz/b191/courses/b0m33bdt/start>
- Lessons

› Description

- <https://www.fel.cvut.cz/cz/education/bk/predmety/47/73/p4773206.html>

› Excercise storage

- GitHub
- Trainings, solutions, links to guides

› Metacentrum – computational cluster

- <https://www.metacentrum.cz/cs/Sluzby/Hadoop/>
- Ask for access !



*"So what am I doing here—knighting,
beheading, or what?"*

CN

English x Czech ?

Lectures

- › **Intro, organization, motivation**
 - Organizational stuff
 - Big Data
 - Big Data and Data Science
 - Big Data applications
- › **Architecture**
 - Hadoop, distribute, resource management YARN, etc.
- › **Storage**
 - HDFS, fileformats, compression, HIVE, Impala
- › **Map-reduce**
 - SQL relations, paradigm
- › **Apache Spark**
 - Distributed processing



Lectures

- › **Big Data Science a Data architectures**
 - page rank, collaborative filtering, SNA
 - Big Data solutions
- › **Cloud Big Data technologies**
 - Azure, AWS
 - Serverless principles
- › **Elastic**



"So what am I doing here—knighting, beheading, or what?"

CN

Practices

- › **First steps**
 - Connection, tests, work with filesystems, HDFS
- › **Refresh of necessary prerequisites**
 - Linux, shell utility, SQL, Python
- › **Hive**
 - Tables, int./ext., create, manage, queries, partitions
- › **Spark**
 - Spark intro, transformations and actions
 - Two approaches: RDD (map-reduce) a SQL (data frame)
- › **Cloud**
 - Azure first steps
- › **Test**
 - Real tasks processing with help of selected technologies

Timeschedule – change from the last year

- › **Classical even – odd weeks**
- › **Odd week**
 - Lectures Wed: 9:15-10:45 [KN:E-127](#)
- › **Even week**
 - Lectures Wed: 9:15-10:45 [KN:E-127](#)
 - Practices: 2 11:00-12:30 and 12:45-14:15 [KN:E-307](#)
- › Possible switch to online if COVID19 ...
 - Your opinion?

Timeschedule – draft plan

Week	S even – L odd	Name	Mentor	Date
1	S	Organizace přednášek, klasifikační požadavky Motivace, přehled, aplikace	Sušický Marek	22.09.2021
2	L	Hadoop, architektura clusteru	Sušický Marek	29.09.2021
3	S	Storage (HDFS, Hive, Impala), Map+reduce – paradigma a implementace	Stamenov Sergii	06.10.2021
4	L	Import dat - NiFi, (Kafka minimalne), Big Datové Architektury a security	Sušický Marek	13.10.2021
5	S	Spark basic	Vonášek Josef-Duda Tomáš	20.10.2021
6	L	Spark - advanced	Vonášek Josef Duda Tomáš	27.10.2021
7	S	Kafka a její ekosystem	Stamenov Sergii	03.11.2011
8	L	Streaming	Duda Tomáš	10.11.2021
9	S	Státní svátek		17.11.2021
10	L	Cloudové technologie - obecně	Benešová Alisa	24.11.2021
11	S	Azure		01.12.2021
12	L	Serverless architektury - AWS etc	Sušický Marek	08.12.2021
13	S	Elastic - včetně OS varianty	Ulitin Valerii	15.12.2021
14	L	Big Data Science	Paščenko Petr	05.01.2022

General warning

- › This is not the first year of this lecture
- › We have changed almost a half of the agenda
- › You can use materials from the past but it keep in mind that it is not enough

Cloud – IMPORTANT!

- › We have some lectures about cloud/Azure
- › It is necessary to have Microsoft credits
- › If you've used it in the past, let us know, there should be some solution

1st practice - IMPORTANT!

- › Will not happen today
- › We've missed the fact that 1st week is even
- › We've decided to do it via Teams next week in the same time and it will be recorded
- › It is necessary to have metacentrum account !
- › If it will be not possible for you, let me know

How to get credits?

- › **Get at least 50 points from 100**

- 1. Tests and homeworks
 - Small theory test, max. 20 points
 - Final practice and theory test, max. 40 points

- › Exam
 - 30 points - teoretical questions
 - 10 points – oral exam
 - Some basic questions can be

Mentors

Marek Sušický



› Skills

- Databases – Oracle, PostgreSQL, graph dbs
- Security, fraud, networks
- Big data

› Experience

- 11 years in Profinit
- Work for large banks, telco operators, Prague airport
- Experience from international teams

Stamenov Sergii

› Skills

- Data Science, Machine learning, Big Data
- .NET development

› Experience

- 5 years in Profinit
- Data science for T-Mobile, UK bank, Large Czech bank
- Spark trainings

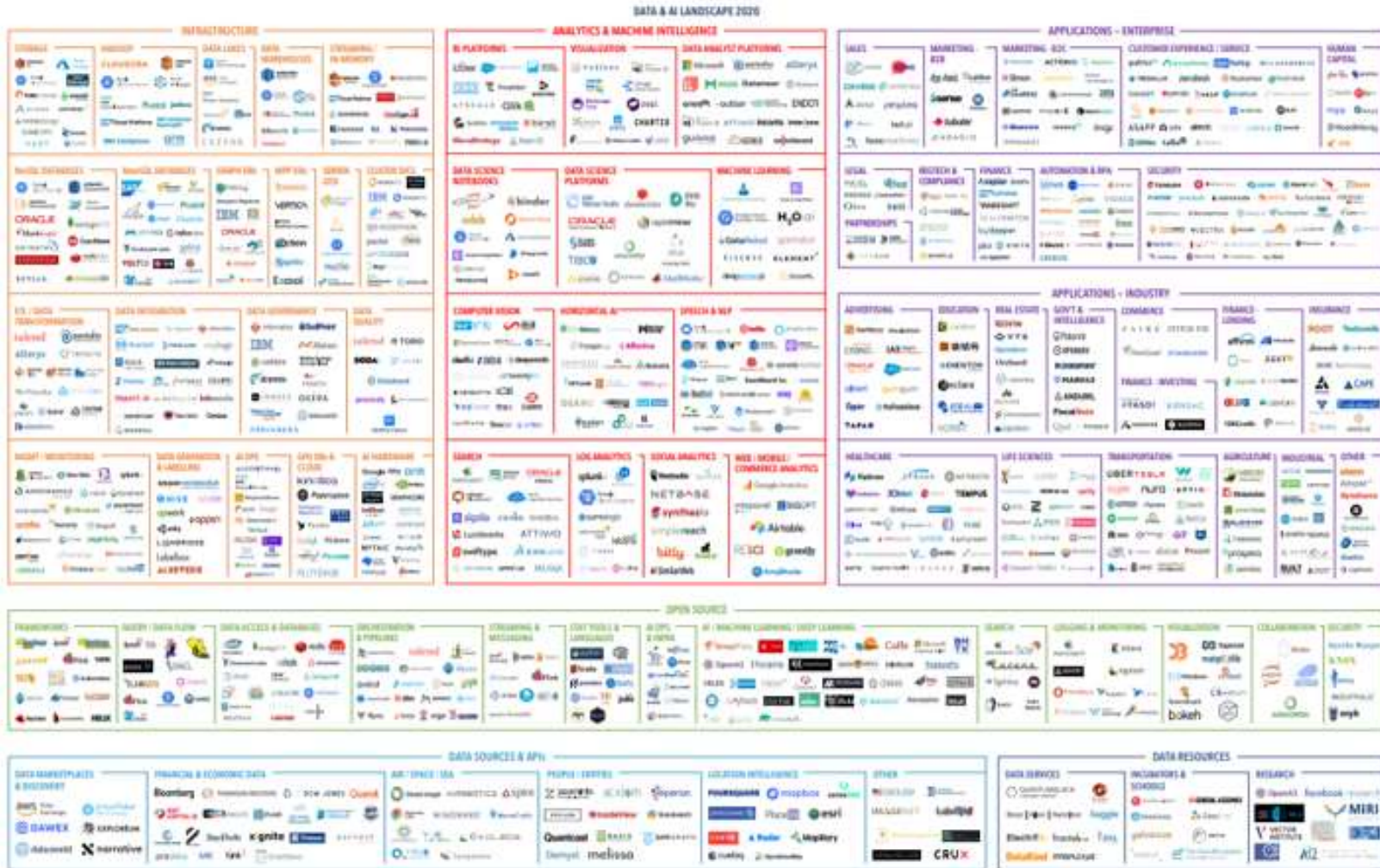


Other colleagues (but not less important :)

- › Alisa Benešová
- › Tomáš Duda
- › Josef Vonášek
- › Petr Paščenko

Big Data

Big Data?



Big Data?

- › That's the reason why to have multiple mentors
- › We'd like to cover a typical technologies that you can meet during your work
- › After this class you should have rough overview and not detailed knowledge

Big Data – driving factors

Trend movements:

- › Data
 - From small to large
 - From simple to complex
- › Databases
 - From files to distributed clusters
- › Programming
 - From procedural to functional frameworks
- › Data Science
 - From chosen statistics to detailed contextual analysis



Data

A Very Short History Of (Big) Data

Gil Press for [forbes](#)



- › **1944:** Fremont Rider, Wesleyan University Librarian
 - „American university libraries were doubling in size every sixteen years.“
 - „Yale Library in 2040 will have approximately 200,000,000 volumes, which will occupy over 6,000 miles of shelves“
- › **1961:** Derek Price, Science Since Babylon
 - „the number of new journals has grown exponentially rather than linearly, doubling every fifteen years and increasing by a factor of ten during every half-century.“
 - „Each scientific advance generates a new series of advances at a reasonably constant birth rate, so that the number of births is strictly proportional to the size of the population of discoveries at any given time. “
- › **1975:** The Ministry of Posts and Telecommunications in Japan
 - „information supply is increasing much faster than information consumption“
 - „the demand for information provided by mass media, which are one-way communication, has become stagnant, and the demand for information provided by personal telecommunications media, which are characterized by two-way communications, has drastically increased.“

A Very Short History Of Big Data

Gil Press for [forbes](#)



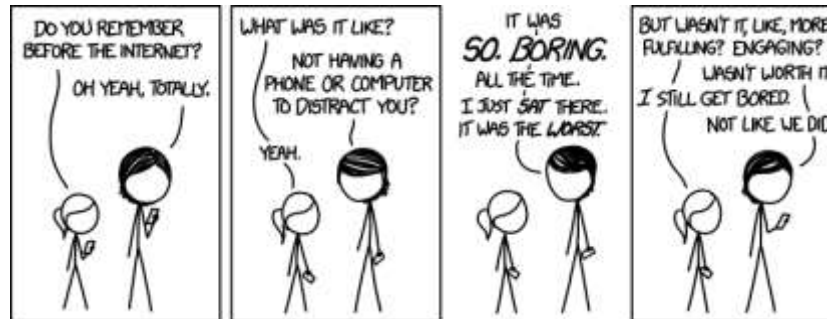
© Scott Adams, Inc./Dist. by UFS, Inc.

- › **1980:** I.A. Tjomsland, Fourth IEEE Symposium on Mass Storage Systems
 - „Parkinson’s 1st Law paraphrased: Data expands to fill the space available.“
 - „The penalties for storing obsolete data are less apparent than are the penalties for discarding potentially useful data.“
- › **1986:** Hal B. Becker, Can users really absorb data at today’s rates?
 - „The recoding density achieved by Gutenberg was approximately 500 symbols per cubic inch – 500 times the density of [4,000 B.C. Sumerian] clay tablets. By the year 2000, semiconductor random access memory should be storing 1.25×10^{11} bytes per cubic inch.“ – po pravdě o řád přestřelené v roce 2017
- › **1996:** B.J. Truskowski, The Evolution of Storage Systems
 - Digital storage becomes more cost-effective for storing data than paper.
- › **1997** Michael Cox and David Ellsworth
 - „Data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk. We call this the problem of *big data*. When data sets do not fit in main memory (*in core*), or when they do not fit even on local disk, the most common solution is to acquire more resources. “

A Very Short History Of Big Data

Gil Press for [forbes](#)

- › **1997** Michael Lesk, How much information is there in the world?
 - „In only a few years, (a) we will be able [to] save everything—no information will have to be thrown out, and (b) the typical piece of information will never be looked at by a human being.“
- › **1998:** K.G. Coffman and Andrew Odlyzko
 - „ the growth rate of traffic on the public Internet, while lower than is often cited, is still about 100% per year, much higher than for traffic on other networks.“



› 2000: Big Data Era

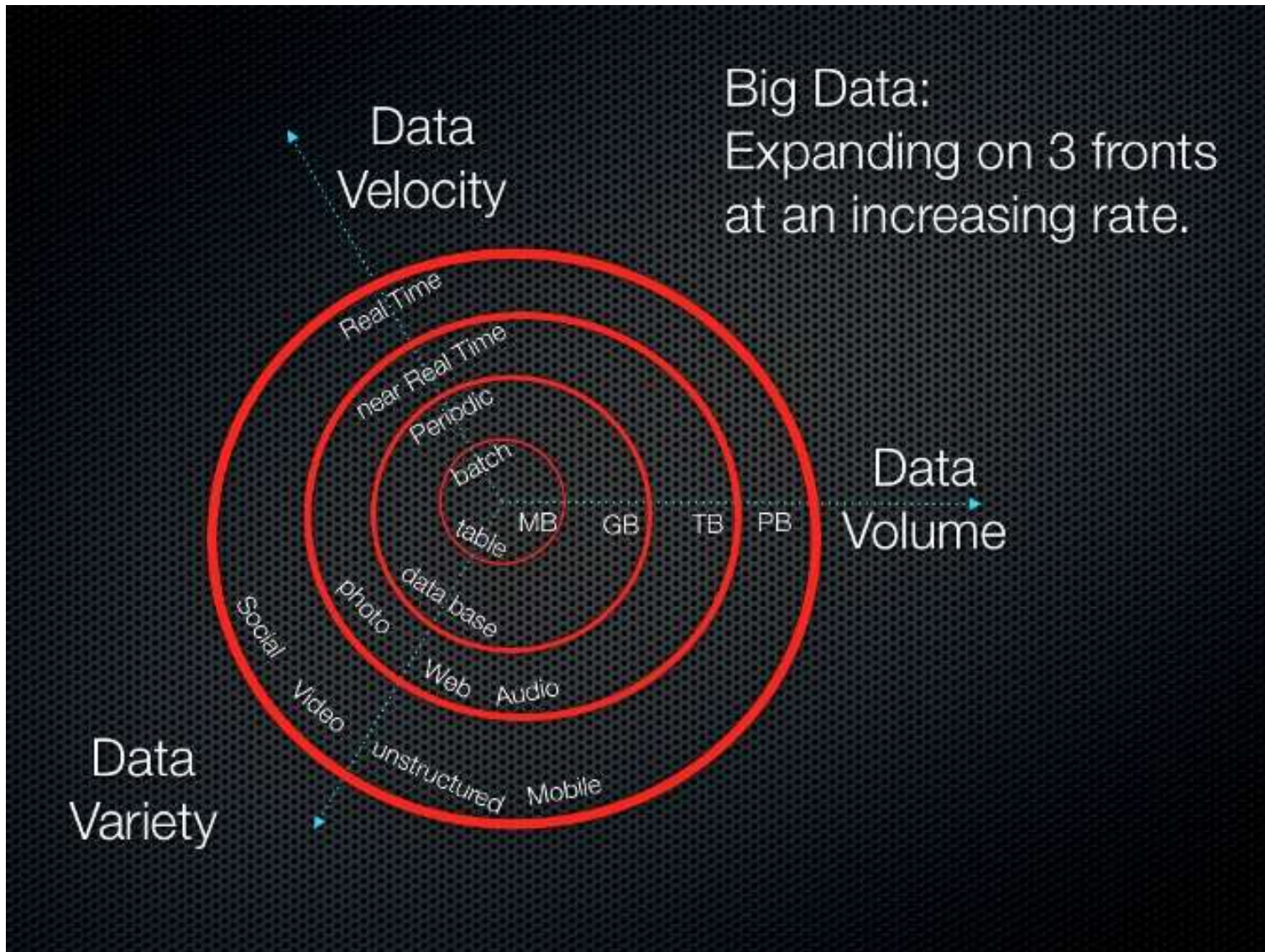
- „the world produced about 1.5 exabytes of unique information, or about 250 megabytes for every man, woman, and child on earth. It also finds that “a vast amount of unique information is created and stored by individuals” (what it calls the “democratization of data”) and that “not only is digital information production the largest in total, it is also the most rapidly growing.“

A Very Short History Of Big Data – summary

- › The amount of data is rising exponentially, the same with transfer capacity
- › The data supply is rising faster than the demand
- › In communication prevails bidirectionality and active role of users
- › It is easier to store data than sort them and delete them

- › Storage capacity grows and data just fill it – we can store almost everything
- › Data files can be larger than one data device / medium
- › **Average information is not read by human**

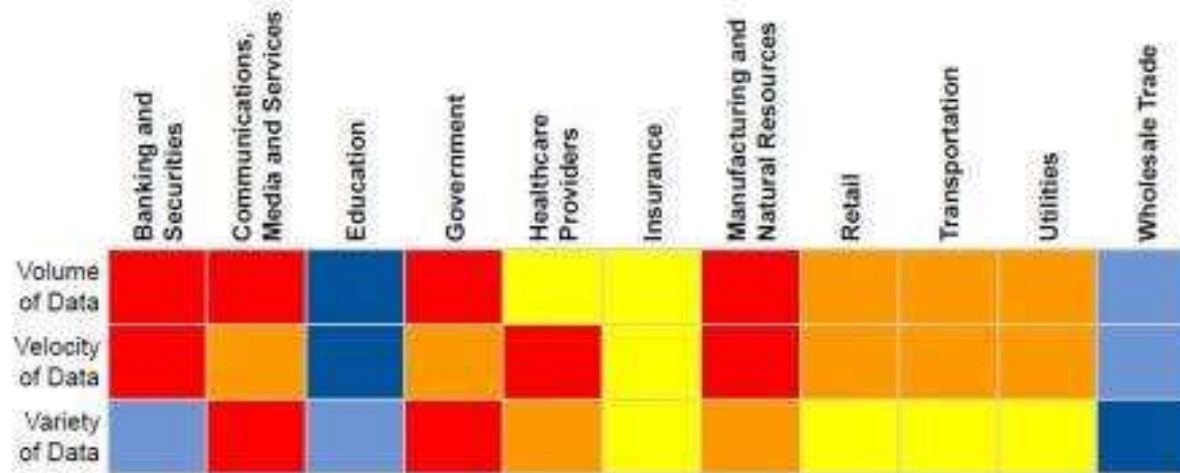
Big Data Definition – 3V



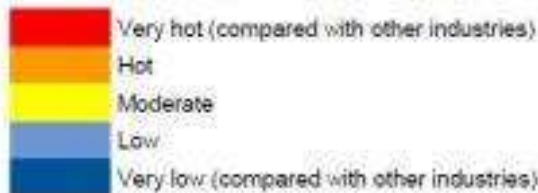
Big Data Definition – 3V

- › Volume, Velocity, Variety

Comparison of Data Characteristics by Industry

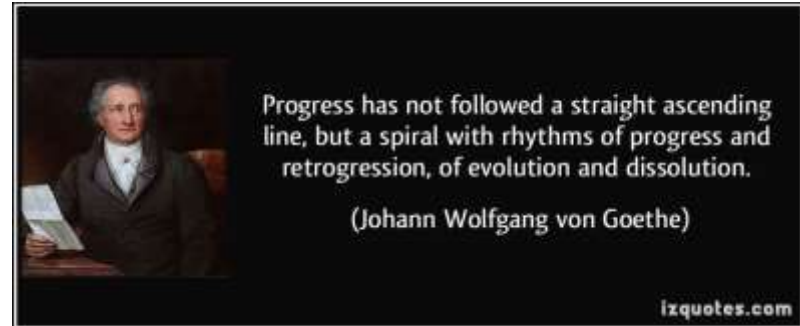


Potential big data opportunity on each dimension is:



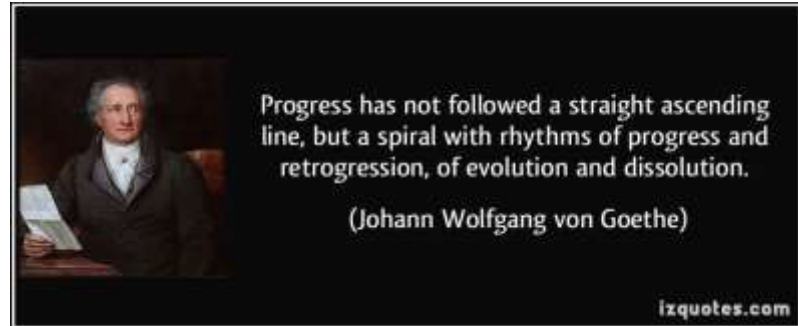
Databases

History of databases



- › **40' – Ad hoc data streams**
 - Punch cards and tapes
- › **50' – File System**
 - Files and folders with hierarchical structure and unique path
 - Media independence (tape, disk, ...)
- › **60' – DBMS**
 - Tables and indexing: hashing, B-trees. Client/server architecture
- › **70' – R-DBMS**
 - Relational paradigm: normal forms, relational algebra, selection, projection, etc.

History of databases



- › **80' – SQL**
 - Unified query language, proprietary databases
- › **90' – Data warehouses**
 - Data integration, one thruth, analytical reporting
- › **0' – NoSQL**
 - Web programming, graph databases, first clouds
- › **10' – Big Data and cloud**

RDMS vs Big Data

	RDBMS	Hadoop
Size	GB, TB	TB, EB, PB
Access	Interactive and batch	Batch only
Queries	SQL + addons	Map-Reduce a SQL emulation
Changes	Repeatable rw	Write once, repeatable read
Structure	static database schema	dynamic schema
Integrity	ACID	no
Performance	Limited, optimization needed	„Linear“
Latency	minimal (ms)	high (dozens of seconds)
HW	Well tuned expensive	Commodity HW
License	Commercial, expensive	open source + support
Paralelization	Limited and expensive per core	Yes

Programming

Programming – history of abstraction



- › Procedural programming
 - Specify the order of commands
- › **Unstructured paradigm** (<= 80')
 - Asembler instructions and conditional jumps – goto era
- › **Structured paradigm** (circa 60'-80')
 - Functions grouped to libraries
- › **Object paradigm** (circa 90')
 - Connect functions and data to objects, inheritance, polymorphismus
- › **Virtualization** (circa 0')
 - Split programmer from specific OS a HW
- › **The main idea**
 - Improve the level of abstraction (compiler, linker, vm)
 - Remove the complexity from the bottom – libraries can cover lowlevel tasks
- › Problem: the complexity is everywhere, not only bottom

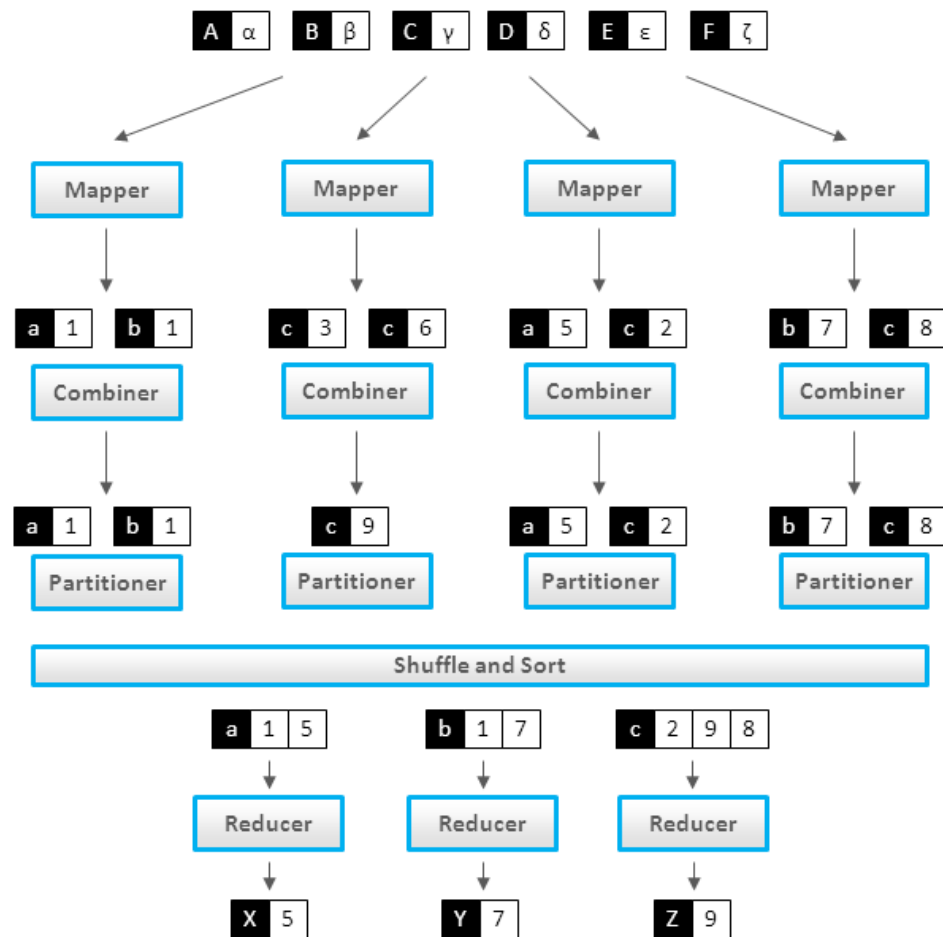
Parallel framework



- › Parallel program issue – complexity comes from above
 - It is necessary (recommended) to define the algorithm as a parallel from the beginning
 - It is hard to maintain the program, debug it, optimize it and tune it
- › **Framework improvements** (present)
 - Framework is a skeleton that describes the structure of the algorithm
 - For example quick-sort with user comparator
 - Programmer has to put his code to defined places – inheritance or templating can be used
 - Programmer can partially omit the complexity
 - Event driven programming, web frameworks, Tensorflow for neural networks, Map-Reduce paradigm etc.
- › Parallel libraries (MPI, BSP, OpenMP ...)
 - Universal frameworks have full power but don't solve algorithm complexity
- › **Map-Reduce** paradigm
 - Limited functionality for data processing
 - But the structure is easy to understand

Map-Reduce

- › Algorithm is known
- › Add method implementation
- › Map
 - Build pairs <key,val>
- › Combine*
 - It is possible to merge pairs with the same key to save network transfer
- › Shuffle and Sort
 - Transfer between nodes
- › Reduce
 - Result aggregation
- › Functions can be complex
- › Algorithm = chain Map and Reduce



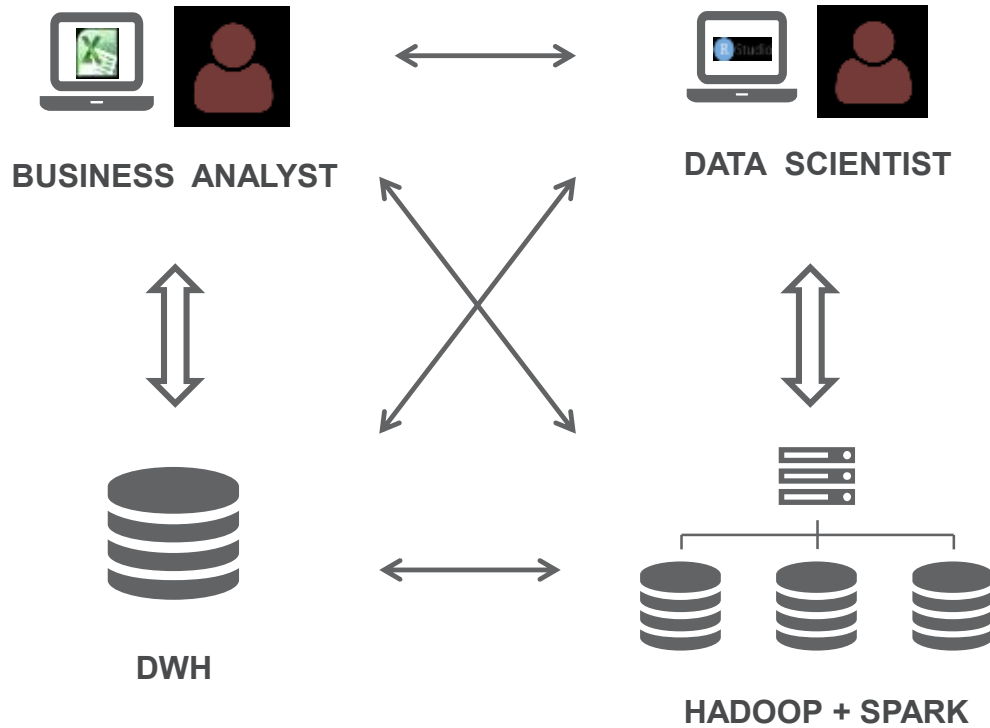
Data Science

Big Data and Data Science

DWH is to Business Intelligence

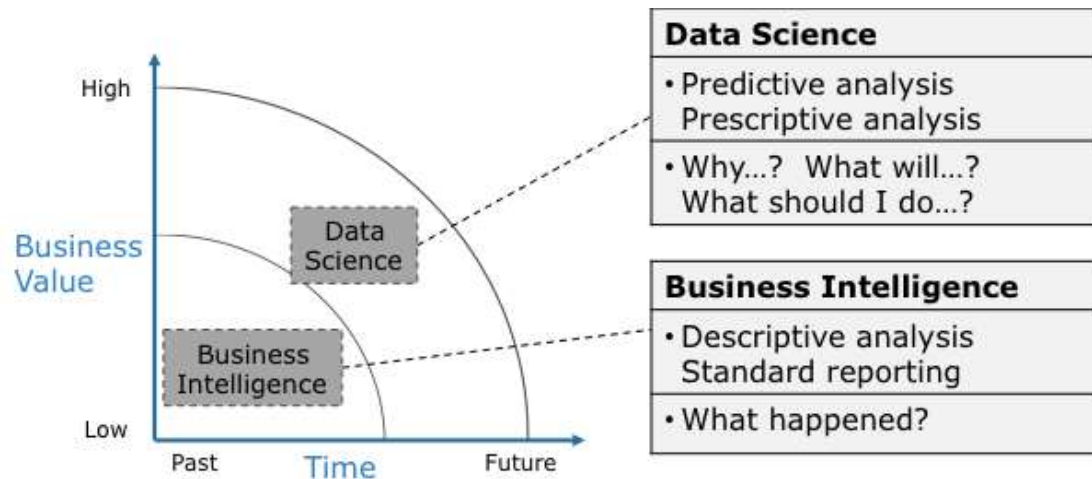
like

Big Data to Data Science



Data Science

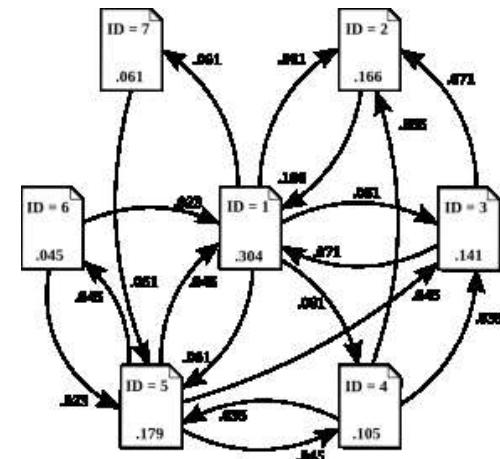
- › Specialization merge
 - Statistics, informatics, Data mining, machine learning, AI
- › Data Science versus Business Intelligence
 - BI: How many pens we sold in September?
 - DS: How many will we sell in October?
- › Work with uncertainty, probabilistic result
 - Predictive modelling
 - Segmentation, clustering
 - Similarity modelling, collaborative filtering, recommendation
 - Anomaly detection
 - Text-mining,
 - Web-mining,
 - Image processing,
 - SNA,
 - etc.



Google

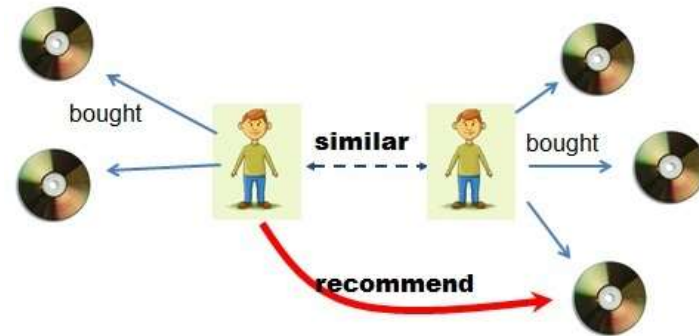


- › Google is not searching but sorting
- › 100T+ pages with much more links
- › Fulltext search
 - Crawlers crawling and indexing – engineering task
 - User enters a query (word, phrase) and the result is list of pages
- › In which order should they appear?
 - What about the relevancy?
- › Google Pagerank
 - Interactive method for page ranking
 - Pages are nodes, links are edges
 - Markov chains with conditions
 - Sparse matrix multiplication
 - Spark implementation



Amazon, Netflix, YouTube

- › Recommendation
 - You saw these 10 movies, watch this one
 - When you buy Babička, offer Broučci
- › Two approaches
 - Goods similarity
 - Customer similarity
- › Collaborative filtering
 - Singular Value Decomposition
 - Alternating Least Squares (Spark)
- › A – matrix client x product
- › U – matrix client x factor
- › L – matrix factors
- › V – matrix factor x product
- › Multiply it again
 - Recommend the missing

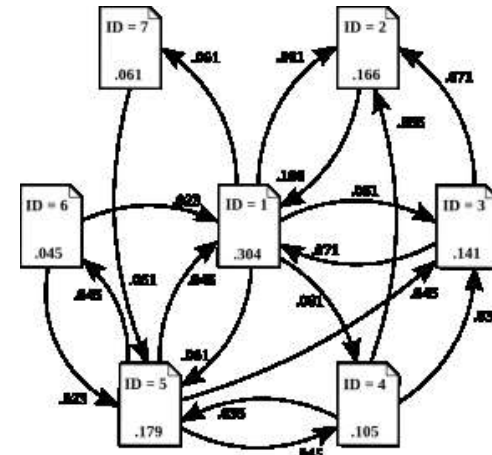


$$\begin{array}{c} \mathbf{A} \end{array} = \begin{array}{c} \mathbf{U} \\ \begin{array}{|c|} \hline \text{shaded} \\ \hline \end{array} \end{array} \begin{array}{c} \mathbf{L} \\ \begin{array}{|c|} \hline \text{shaded} \\ \hline \end{array} \end{array} \begin{array}{c} \mathbf{V}^T \\ \begin{array}{|c|} \hline \text{shaded} \\ \hline \end{array} \end{array}$$

Big Data Science - summary

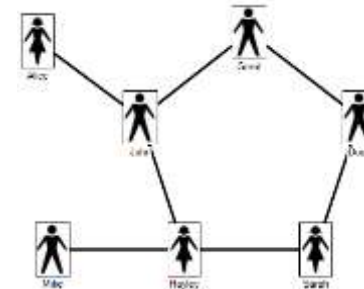
> Google

- Google is not searching but sorting
- 1G pages with much more links
- How to sort them?
- Big Data algorithm PageRank
 - Find own vector of the large matrix



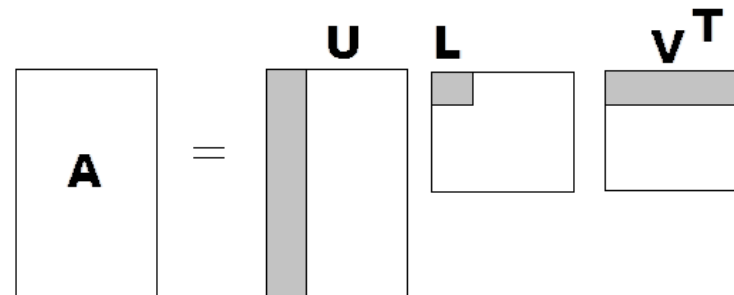
> Amazon, Netflix, YouTube

- How to recommend?
- When you buy Babička, offer Broučci
- Big Data Algorithm Collaborative filtering
 - Singular Value Decomposition



> Facebook

- Which content to show?
- Combination
 - SNA – friends
 - Collaborative filtering – like

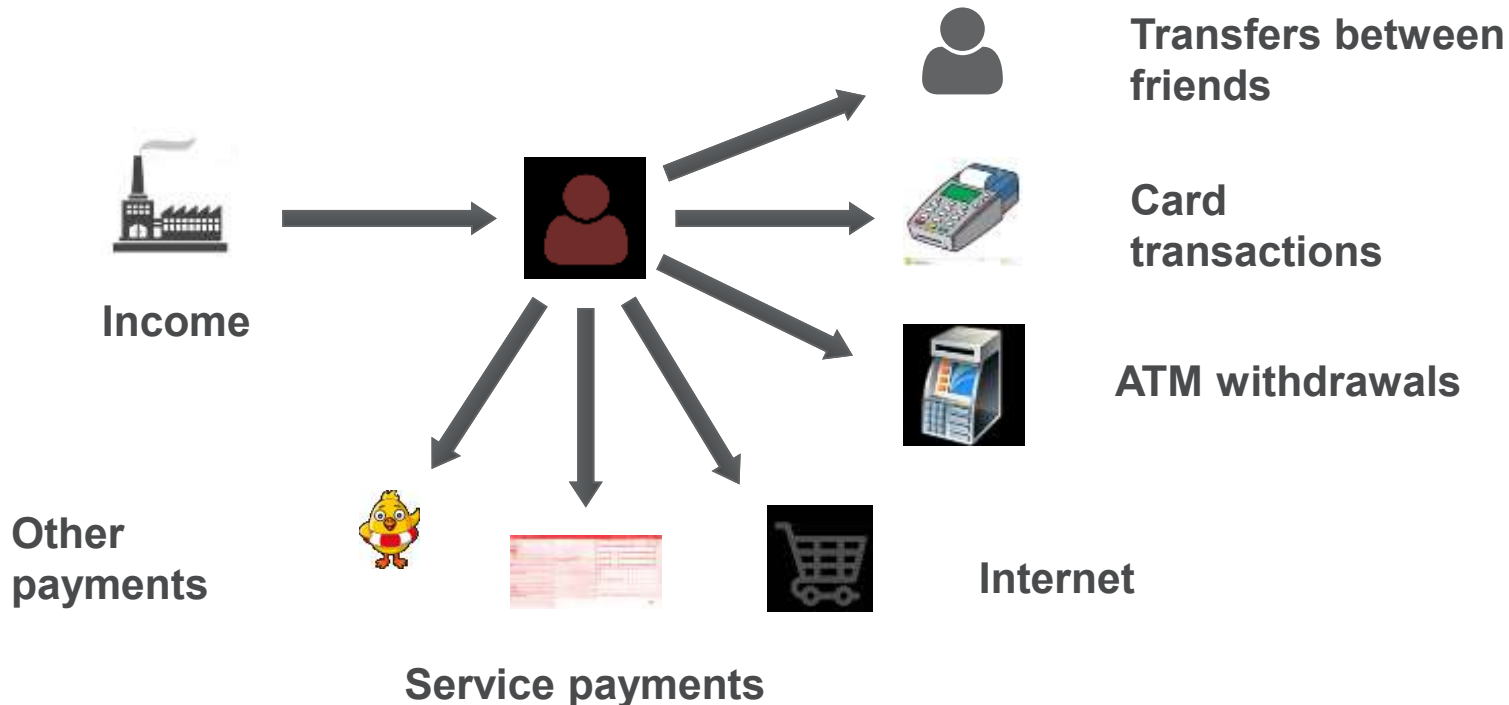


> What about banks?

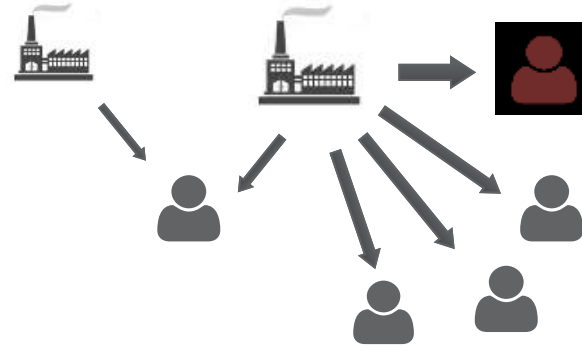
„Czech“ reality

Big Data transactional analysis

- › We create models for retail banks
- › Input – financial transactions
- › Output – valuable information, events, labels
- › The main goal is to enrich the standard process with new knowledge



Salary detector



- › Input
 - Financial transactions – company- client
- › Output
 - Relations – employer - employee
- › Business case
 - Risk score, event detection, similarities (c2c/b2b),...
- › Principles
 - Detect transactional patterns, text mining, advanced statistics
- › High accuracy not only for large corporations, but for
 - Short contracts – lenght under 3 months
 - Notstandard contracts (part-time jobs, contractors etc.)
 - Small businesses

Household detection – Bank/Telco

› Input

- Client transactions – bank (c2c, card transactions,...)
- Network information – telco (cdr, location, billing)
- Basic demography (age, sex, address, surname,...)

› Output

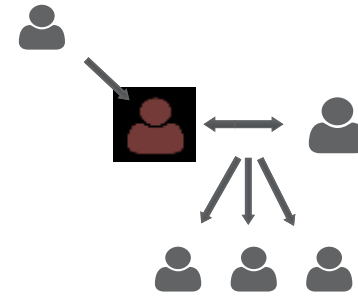
- Identified household members and family relations

› Business opportunities

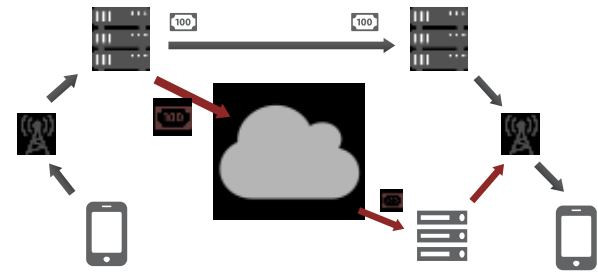
- Family marketing, robust risk score,...

› Principles

- Detect the patterns, transaction analysis, text mining



Telco Big Data SimBox Fraud



- › Scenario
 - Foreign operator is cheating and using VoIP to save money for the interconnection fees
- › Input
 - Telco, network data (cdr, location, billing)
- › Output
 - Suspected SIM cards detected
- › Principles
 - Detect unusual behavior of SIM cards
 - Identify groups with the same behavior
 - Automatic detection with roaming data

Data masking - UMT

- › Universal masking tool
 - Pseudonymization
 - Anonymization
 - Spark is used
 - “Back to school” – math (bijection, encryption, etc.)

Goal is to create good testing data that are safe and fully compliant with regulations

Summary

Summary

- › Check the links
- › Ask for Metacentrum access
- › Check if you have still Microsoft credits
- › Try to attend first practice next week

- › It is not possible to cover all technologies
- › Most used will be covered



Dotazy