PROFINIT

B0M33BDT - Technologie pro velká data

Petr Filas

27.9.2023

Agenda

PROFINIT

- 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

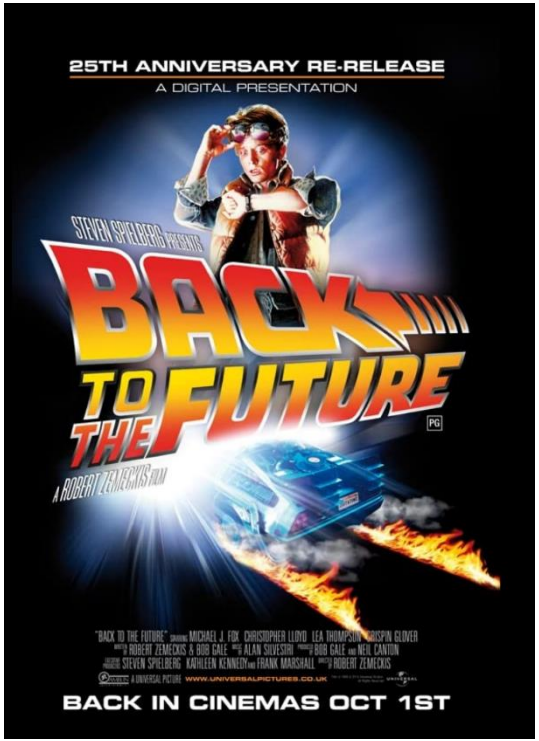
PROFINIT

- **Courseware**
  - https://cw.fel.cvut.cz/wiki/courses/b0m33bdt/start
  - Lessons
- **Description**
  - https://www.fel.cvut.cz/cz/education/bk/predmety/47/73/p4773206.html
- **Excercise storage**
  - GitHub - https://github.com/profinit/BDT
  - Trainings, solutions, links to guides

Lectures I

PROFINIT

- **Intro, organization, motivation**
  - Organizational stuff
  - History of (Big) Data
  - Big Data and Data Science
  - Big Data applications
- **Databricks**
  - Introduction to Databricks
  - Introduction to Spark
- **Architecture (history)**
  - Hadoop, distribtions, resource management YARN, etc.
  - HDFS, fileformats, compression, HIVE, Impala
  - Map-reduce



Lectures II

PROFINIT

- **Data import and manipulation**
  - Apache NiFi, Apache Kafka
- **Data Streaming**
  - Spark Structured Streaming
- **Advanced Spark**
  - Advanced and complex Spark task practicaly
- **Cloud technologies for Big Data**
  - **Azure**, AWS
- **Serverless technologies**
- **(Big) Data Science**

Practices

PROFINIT

- **Introduction to Azure**
  - Azure account creation
  - Databricks setup in Azure
- **Introduction to Databricks**
  - Practical introduction to Databricks in Azure
  - Dbfs, cluster creation
  - SparkUI
- **Processing batch in Databricks**
- **Processing stream in Databricks**
- **Homework assignment**

Timeschedule

PROFINIT

- **Odd week**
  - Lectures Wed: 9:15-10:45 **KN:E-126**
  - Practices: 11:00-12:30 **KN:E-310**
- **Even week**
  - Lectures Wed: 9:15-10:45 **KN:E-126**

Timeschedule

- **1. week (27. 9.2023):** Organization, classification, motivation, overview
  - Practices - Intro to Azure and Databricks
- **2. week (4.10.2023):** Introduction to Databricks
- **3. week (11. 10.2023):** Spark basics in Databricks
  - Practices - Intro to Databricks
- **4. week (18.10.2023):** Hadoop and parallel data processing 1
- **5. week (25.10.2023):** Hadoop and parallel data processing 2
  - Practices - Batch processing in Databricks
- **6. week (1.11.2023):** Import of data
- **7. week (8.11.2023):** Streaming
  - Practices - Batch processing in Databricks
- **8. week (15.11.2023):** Advanced Spark practically
- **9. week (22.11.2023):** Cloud introduction
  - Practices - Stream processing in Databricks + **mid-term test**
- **10. week (29.11.2023):** Cloud - Azure
- **11. week (6.12.2023):** Databricks Advanced
  - Practices - Stream processing in Databricks + **homework assignment**
- **12. week (13.12.2023):** Serverless
- **13. week (20.12.2023):** Big Data Science
  - Practices - Homework consultations
- **14. week (3.1.2024):** Winter holidays - **the lecture is cancelled**
- **15. week (10.1.2024):** Homework consultations + reserve
  - Practices - Homework consultations + **final test**

PROFINIT

General warning

PROFINIT

- This is not the first year of this lecture
- BUT we changed significantly practices (Databricks instead Hadoop) and slightly lectures (more Databricks)
- You can use materials from the past years but keep in mind that it might be not enough or completely different

Cloud – IMPORTANT!

PROFINIT

- For practices and homework it’s necessary to create Microsoft Azure account using your FEL account

- If you’ve used it in the past, let us know, there should be some solution

How to get credits?

PROFINIT

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

A	B	C	D	E
90+	90-80	80-70	70-60	60-50

- Tests and homeworks (get at least 30 points to qualify for the exam)
  - Midterm test, max. **20** points
  - Homework – **20** points
  - Final practice and theory test, max. **20** points
- Exam
  - **20** points - teoretical questions
  - **20** points – oral exam

Mentors/teachers

Lecture teachers I

PROFINIT

- Petr Filas
  - Overall
  - Azure

- Martin Oharek
  - Databricks
  - Spark

- Josef Vonášek
  - Hadoop
  - Kafka & Streaming

Lecture teachers II

PROFINIT

- Tomáš Duda
  - Advanced Spark

- Petr Paščenko
  - Data Science

Practice teachers

PROFINIT

- Adam Nimrichter
  - Databricks & Azure

- Filip Trusina
  - Databricks & Azure

Contact

PROFINIT

- In case of any question/trouble/absence please mail to:

vyukaFEL@profinit.eu

History of (Big) Data

A Very Short History Of (Big) Data

PROFINIT

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

PROFINIT

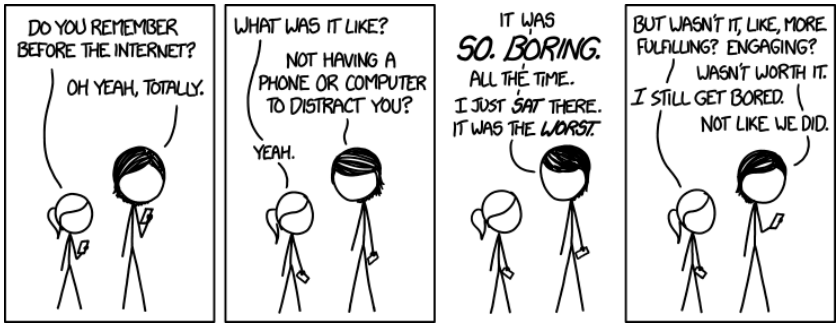
- **1980:** I.A. Tjomsland, Fourth IEEE Symposium on Mass Storage Systems
  - „Parkinson's 1st Law paraphrased: Data expands to fill the space available.“
  - „The penalties for storing obsolete data are less apparent than are the penalties for discarding potentially useful data.“
- **1986:** Hal B. 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 $1.25x10^{11}$ 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

PROFINIT

- **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

PROFINIT

- 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**

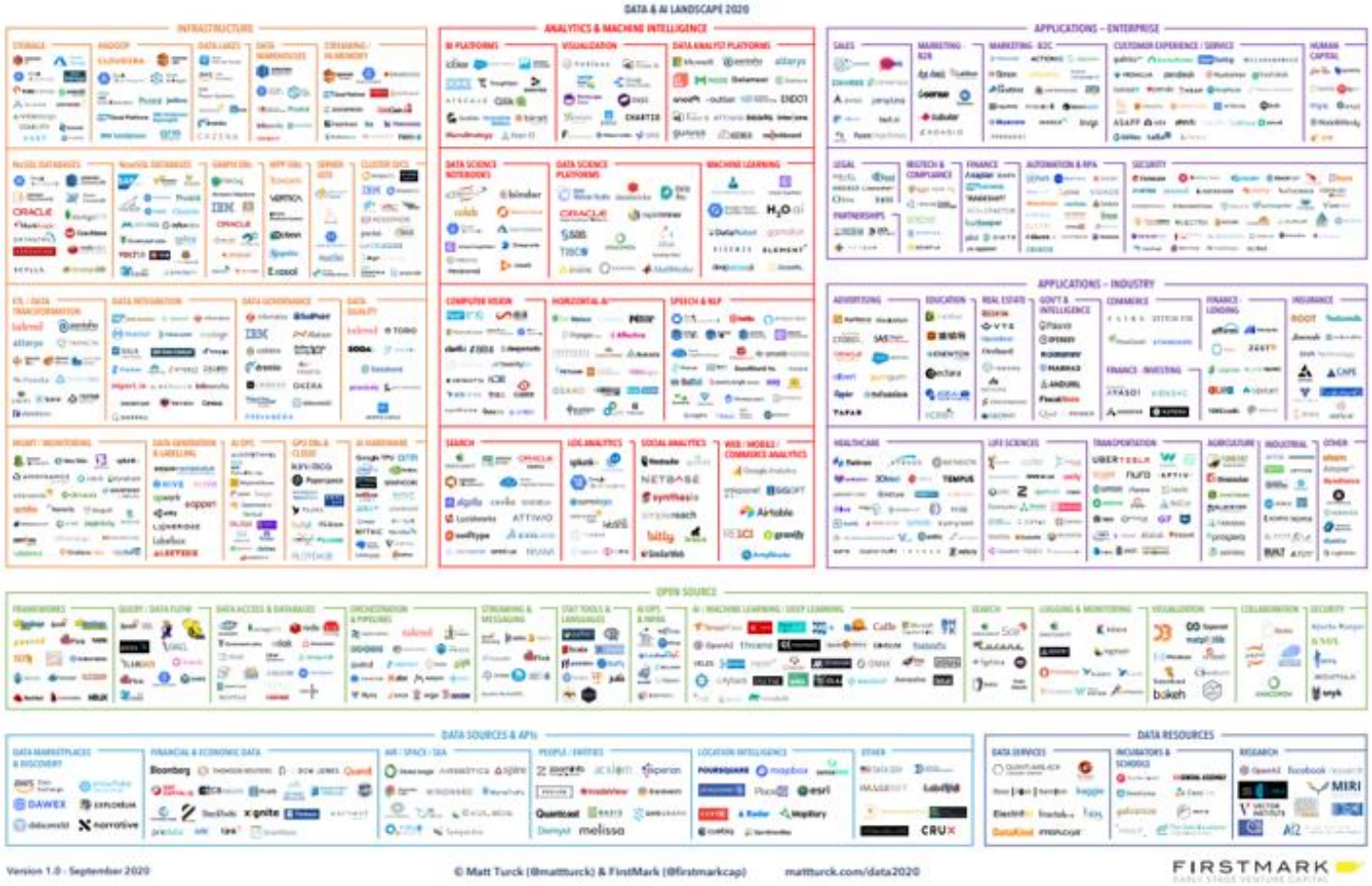
(Big) Data

Big Data?

PROFINIT

Big Data?

PROFINIT



Big Data?

PROFINIT

- 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

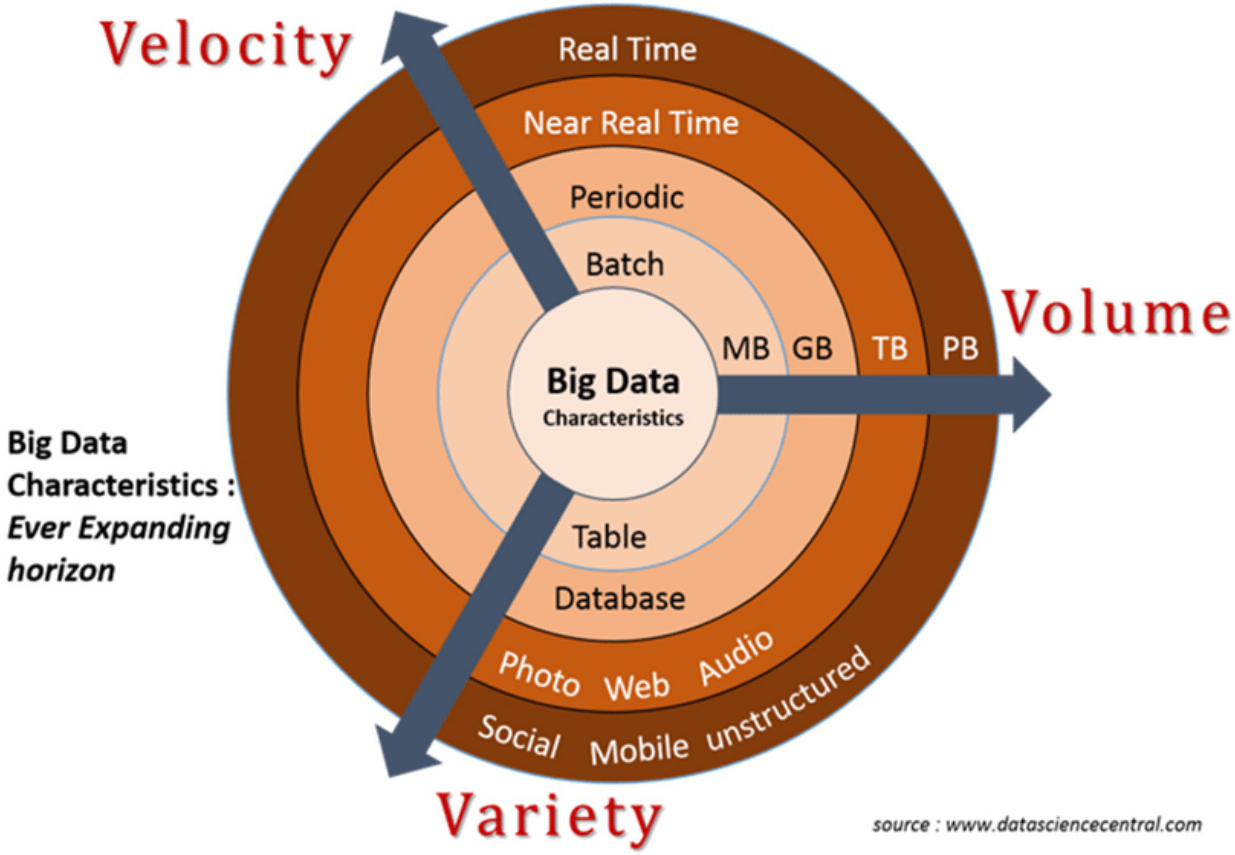
PROFINIT

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

Big Data Definition aka 3V

PROFINIT



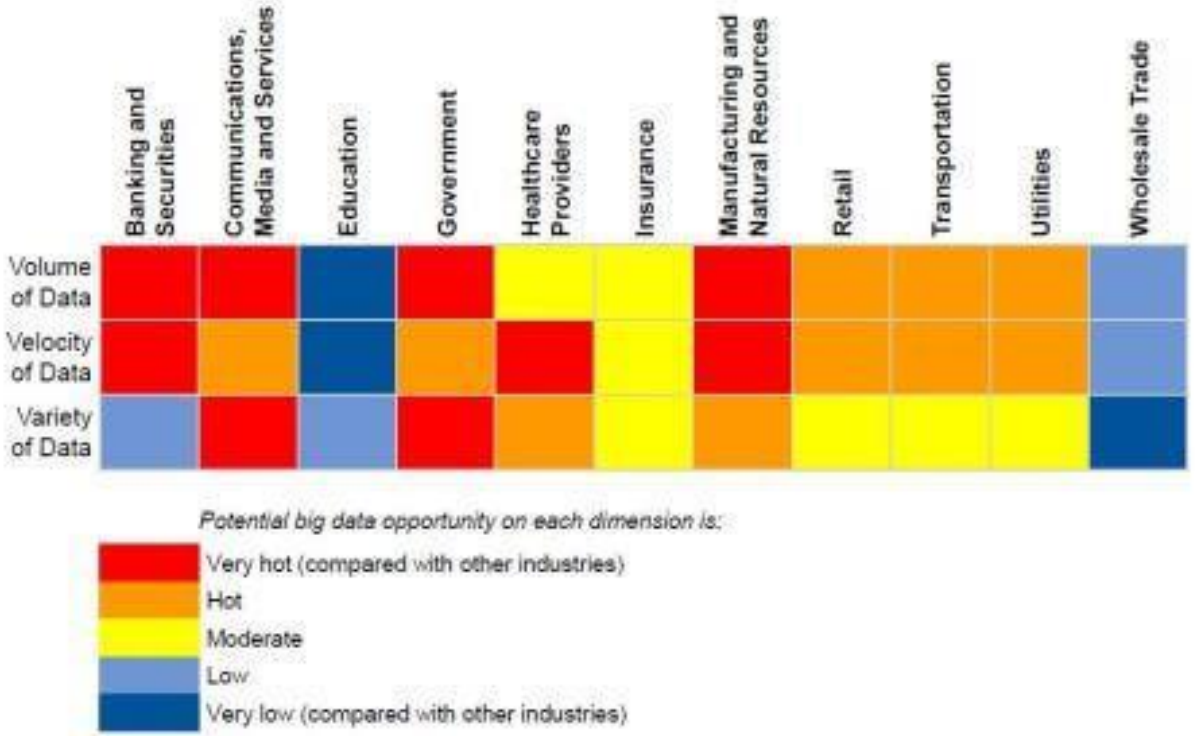
Big Data Characteristics : ***Ever Expanding horizon***

source : www.datasciencecentral.com

Big Data Definition aka 3V

PROFINIT

Comparison of Data Characteristics by Industry



Databases / Technologies

History of Databases I

PROFINIT

- **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 II

PROFINIT

- **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**
- **20' – Cloud**

RDBMS vs Big Data

PROFINIT

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

What technology to choose?

PROFINIT

Magic Quadrant™ for Cloud Database Management Systems

What technology to choose?

PROFINIT

- Cloud based - native
  - Microsoft Synapse
  - Microsoft Cosmos DB
  - AWS Redshift
  - AWS DynamoDB
  - Google BigQuery
  - Cloudera Data Platform

- Cloud based - agnostic
  - Databricks
  - Snowflake

- On-premise
  - Cloudera Data Platform

- Open-source
  - Apache Cassandra
  - MongoDB
  - Handoop

Programming

Programming – history of abstraction I

PROFINIT

- 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’)
  - Isolate programmer from specific OS a HW

Programming – history of abstraction II

PROFINIT

- **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

Paralel Framework I

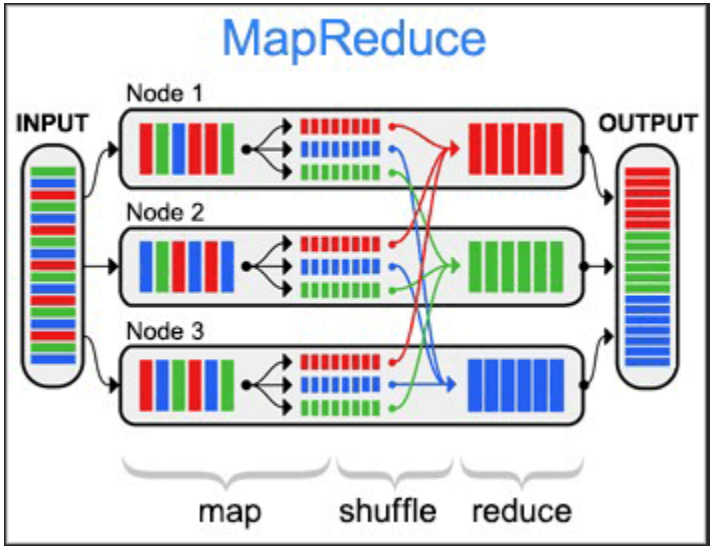
PROFINIT

- Paralell program issue – complexity comes from above
  - It’s 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.

Paralel Framework II

PROFINIT

- **Map-Reduce** paradigm
  - Limited functionality for data processing
  - But the structure is easy to understand



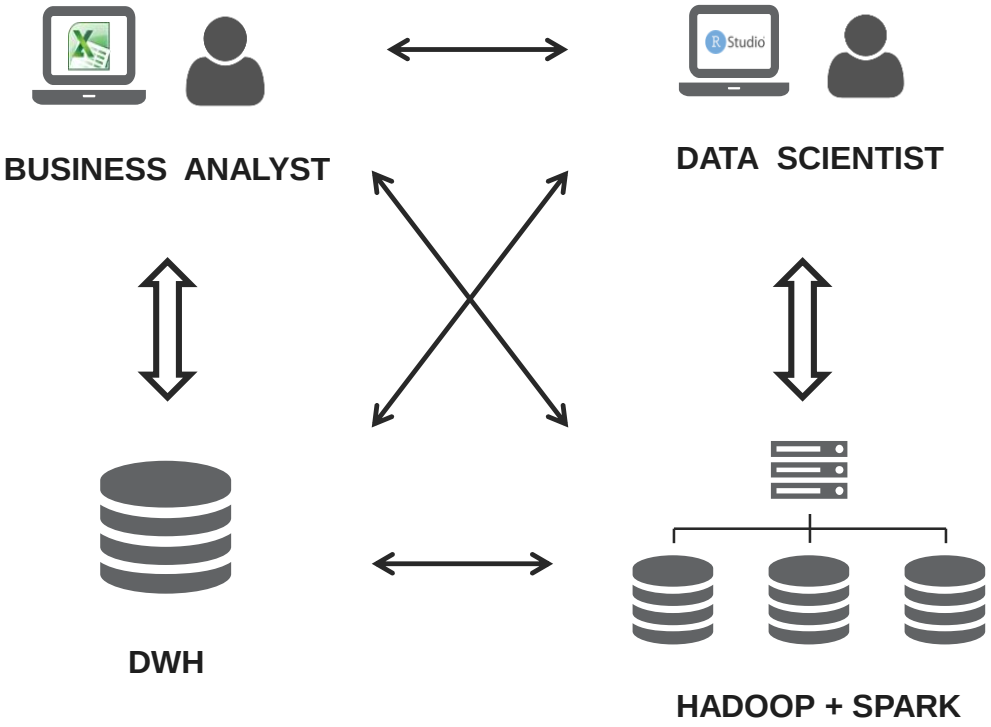
- Map - each node applies the mapping function to its data portion, filtering and sorting it according to parameters.
- Shuffle - mapped data is redistributed to other nodes on the system so that each node contains groups of key-similar data
- Reduce - Data is processed in parallel, per node, per key

Application of Big Data (Data Science)

Data Science

PROFINIT

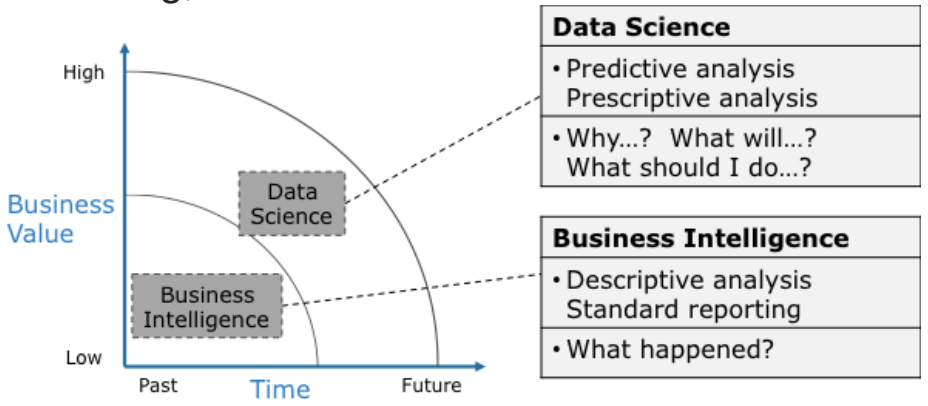
- DWH is to Business Intelligence like Big Data to Data Science



Data Science

PROFINIT

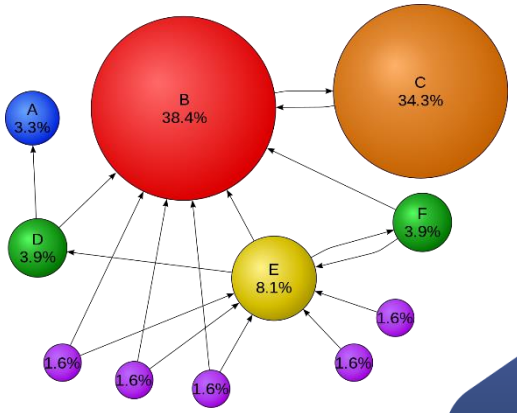
- Data Science versus Business Intelligence
  - BI: How many pens we sold in September?
  - DS: How many will we sell in October?
- Work with uncertainty, probabilistic result
  - Predictive modelling
  - Segmentation, clustering
  - Similarity modelling, collaborative filtering, recommendation
  - Anomaly detection
  - Text-mining
  - Web-mining
  - Image processing



Google use-case

PROFINIT

- 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
  - Webpages are nodes, links are edges (but also a vote)
  - Links/Edges have relevance

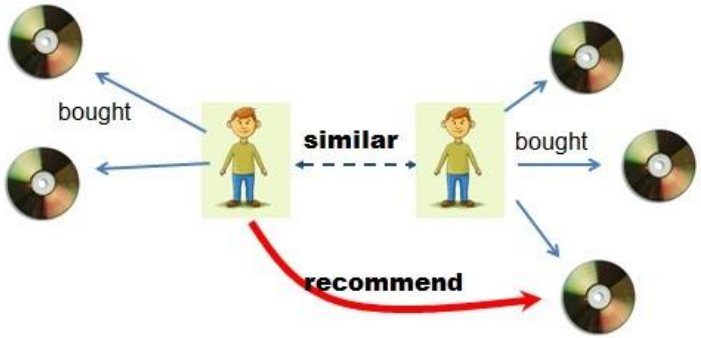


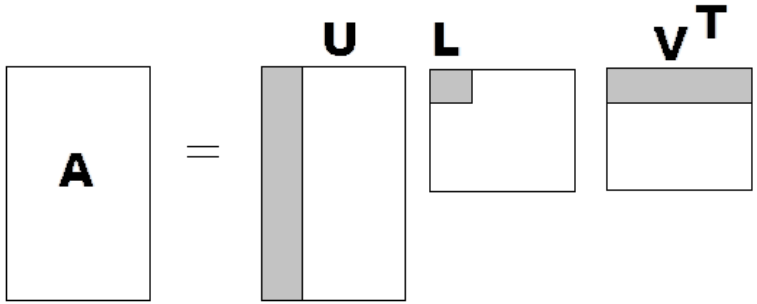
https://en.wikipedia.org/wiki/PageRank

Amazon, Netflix, YouTube use-case

PROFINIT

- Recommendation of next …
  - You saw these 10 movies, watch this one
- Two approaches
  - Goods similarity (Content-Based Filtering)
  - Customer similarity (Collaborative filtering)
- Collaborative filtering
  - Singular Value Decomposition
    - A – matrix client x product (what products clients dis-like)
    - U – realtionship client x factor
    - L – strength of each latent factor
    - V – realtionship factor x product
  - Multiply it again = recommend the next…





Summary

Summary

PROFINIT

- Big Data is a tool for practical applications especially in Data Science
- Big Data starts with **Data Engineering** and ends with **Data Science**
- It’s not possible to cover all technologies
- The most used and perspective technologies will be covered
  - But the world is still changing…

PROFINIT

Questions?

Profinit EU, s.r.o.
Tychonova 2, 160 00 Praha 6

Tel.: + 420 224 316 016, web: www.profinit.eu

LINKEDIN
linkedin.com/company/profinit

TWITTER
@profinit_EU

FACEBOOK
facebook.com/Profinit.EU

YOUTUBE
Profinit EU, s.r.o.