Introduction and Administrative Matters

Big Data Engineering (formerly Informationssysteme)

Prof. Dr. Jens Dittrich

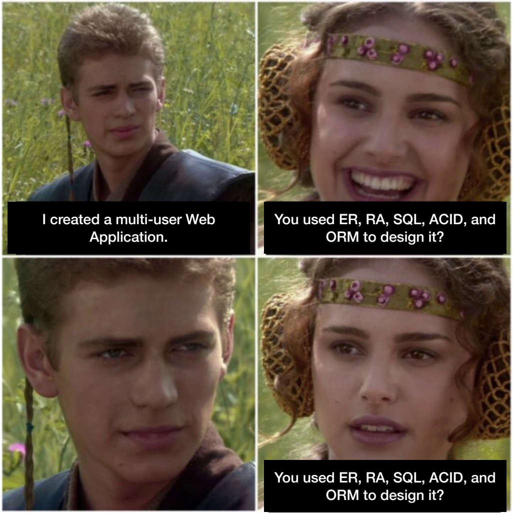
bigdata.uni-saarland.de

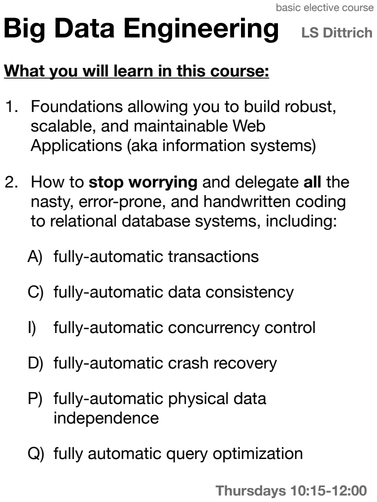
April 14, 2023

Overview of the lecture

- learning objectives
- content
- concept
- assignment sheets
- ChatGPT et al
- tutorials
- exams
- office hours
- Python tools

Learning Objectives


I created a multi-user Web Application.
You used ER, RA, SQL, ACID, and ORM to design it?
You used ER, RA, SQL, ACID, and ORM to design it?



basic elective course

Big Data Engineering **LS Dittrich**

What you will learn in this course:

1. Foundations allowing you to build robust, scalable, and maintainable Web Applications (aka information systems)
2. How to **stop worrying** and delegate **all** the nasty, error-prone, and handwritten coding to relational database systems, including:

   A) fully-automatic transactions

   C) fully-automatic data consistency

   I) fully-automatic concurrency control

   D) fully-automatic crash recovery

   P) fully-automatic physical data independence

   Q) fully automatic query optimization

Thursdays 10:15-12:00

The Laziness Principles in Computer Science

The Laziness Principle

Whenever possible try to map (sub)problems to an existing problem. Then use *existing solutions* to solve that (sub)problem rather than reinventing everything from scratch.

In the context of this lecture *existing solutions* means: use a database system rather than coding the data management stuff yourself! But in other contexts it may also mean any other suitable software (sub-)system and/or library.

The Missed Opportunity for Laziness Principle

If you do not know that a (sub)problem could be mapped to an existing problem, you miss the chance to apply The Laziness Principle.

In other words: if you do not know that certain problems can effectively be solved in certain ways, you will not be able to be lazy! For instance, assume you are simply not aware of a technique X that is always suitable when there is a problem of type Y.

The High Prize of Missed Laziness

Costs for Reinventing the Wheel

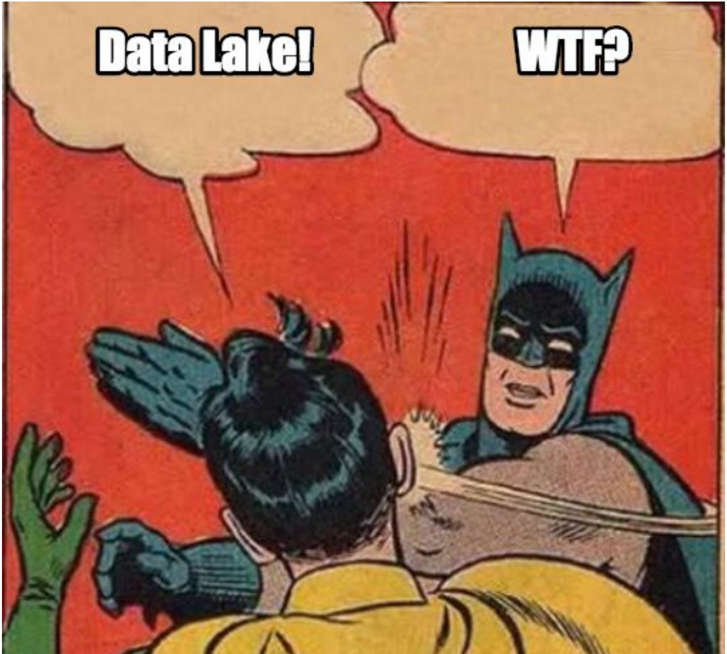
- development costs for features
- testing costs
- deployment costs
- bug-fixing costs
- development costs to add new features (which we might get for free when using an existing lib and/or system)
- maintainability issues

Summary

typically homebrewn solutions fall far behind existing ones in overall quality, features, performance, maintainability, documentation, support, etc.

Example: In the context of this lecture: 50 years of research of thousands of researchers that got condensed into hundreds of different database systems and libraries and best practices

Industry vs University



Buzzwords and other Keywords from the field of Data Analysis/Big Data Engineering

Information Systems
Big Data
Artificial Intelligence
Machine Learning
Deep Learning
Data Mining
Cognitive Computing
NoSQL
SQL
DBMS
RDBMS
ODBMS
Databases
Statistics
Lambda architecture
Cloud Computing
Data Warehousing
Data Science
Data Lake
Data Engineering
Data Cleaning
Data Curation
Spark
Hadoop
MapReduce
Data Streaming
IoT
Realtime Analytics
Big Data Analytics
Key/Value-Stores
Column Stores
Blockchain

The Data Science Cake

Ingredients:
50g statistics
120g linear algebra
200g programming
1kg visualisation
300g software engineering

Additional skills:
creativity
out of the box thinking
grit
team spirit

© istock.com sasilsolutions

The term “Big Data”: The view of a “Databaseologist”(1)

- our top conference has been called VLDB (Very Large Databases) since 1975!
- technically, managing and requesting large amounts of data has long been solved (**if** you know what you are doing)
- performance problems with large amounts of data: It is seldom due to hardware/software, in 99.99% of the cases it is a education problem (of the developer/computer scientist)
- the combination of database technology with other subfields of data science is highly exciting
- example: applied machine learning and databases, various projects
- important database topics for Data Science:
  Data modeling, relational model (RM), relational algebra (RA), SQL, ETL, ELT, data cleaning, data curation, data warehousing, scalability, distributed databases, ...

The term “Big Data”: The view of a “Databaseologist” (2)

Impact of the storage/database system is often underestimated

“*How the data is organized does not matter. Only the complexity and efficiency of the algorithms is important!*”
No! This is wrong for many systems. On modern hardware, it is no longer the CPU that is the bottleneck, but the slow “Delivery” of the data

Example:

- initial situation: Hadoop cluster with Spark and software in Scala
- change from us: different storage layout and a few DB tricks
- prediction of our cost model: factor 10,000 faster
- instead of an expensive Hadoop cluster:
  Laptop or smartphone
- KIWI (Kill It With Iron)
  vs KIWI (Kill It With Intelligence)

The term “Big Data”: The view of a “Databaseologist” (3)

The performance of the database technology plays **no** role.

When?

When the data is small and the hardware is so fast that it makes no difference

Hint: this is the 95% case

versus

The performance of the database technology plays **a major** role.

When?

When the data is “larger” and the hardware does not solve the deficiencies of the software

Hint: this is the 5% case

However: in both cases many other design decisions and techniques play a major role

Learning objectives of this lecture

1. understand fundamental techniques in “Big Data Engineering” conceptually (the 95%-case): slides, exercises
2. learn to apply basic techniques in “Big Data Engineering”: Python, SQL, Jupyter
3. help you avoid reinventing the wheel, learn to apply the *Laziness Principles* whenever possible:
   Learn to map new problems to existing problems and solve them using existing techniques
4. raise awareness for possible problems:
   privacy concerns and deanonymization, ethical issues
5. raise awareness for possible solutions:
   effort, performance, robustness, extensibility, maintainability

Concept of this lecture: Learning by Application

Planned structure for each two-week lecture:

1. Concrete application: XY
2. What are the data management and analysis issues behind this?
3. Basics to be able to solve these problems
   (a) Slides
   (b) Jupyter/Python/SQL Hands-on
4. Transfer of the basics to the concrete application

Planned structure of each assignment sheet:

1. 2 tasks with reference to basics: Slides
2. 1 task with reference to basics: Jupyter/Python/SQL Hands-on
3. 1 task with reference to transfer of the basics to the application

Weekly schedule: Planned Applications and Topics

Topic	Learning objectives
Python (tools, videos)	basics, functions, functional programming, object orientation, and automatic testing
IMDb (Part 1)	data modelling, relational model
IMDb (Part 2)	relational algebra
NSA (Part 1)	introduction to SQL
NSA (Part 2)	analytical SQL, big data arithmetic, big data vs. privacy, countermeasures
Query Optimisation (Part 1)	automatic query optimisation, physical operators, heuristic optimisation
Query Optimisation (Part 2)	cost-based optimisation, join order, plan variants, pipelining, physical optimisations
Trading, banking, and ticket systems (Part 1)	database management systems (DBMS), transactions, serializability theory
Trading, banking, and ticket systems (Part 2)	two-phase Locking (2PL), isolation levels
Data Journalism	pivot tables, graph data, SQL vs graph databases vs hybrids, WITH RECURSIVE, Cypher
IMDb (Part 3)	Front-ends, Web Dev, SQL injection (SQLi), object-relational mapping (ORM), basic safety measures
Data in the Wild	uni vs reality
Recap	

Distinction to the core lecture Database Systems

This (basic) lecture “Big Data Engineering”

Focus on principles, design patterns and application of Big Data technologies

Core lecture “Database Systems”

Deeper dive into the underlying techniques

Lecturer: Prof. Dr. Jens Dittrich

Research:

- Big Data Analytics, Scalable Data Management, Databases, Data Science
- ACM SIGMOD, (P)VLDB, CIDR, ..

Teaching:

- Busy beaver teaching awards 2011, 2013, 2018, 2021
- Busy beaver teaching awards honorable mention 2022
- YouTube: https://www.youtube.com/user/jensdit
- Programme coordinator BSc and MSc Data Science and Artificial Intelligence (since WS 19/20)

https://bigdata.uni-saarland.de/people/dittrich.php

Tutors

Joris Nix (Chief tutor, PhD student)

Despina Constantinidou

Elina Celik

Naya Rudolph

Nicolas Regel

Robert Rabbe

Simon Rink

https://cms.sic.saarland/bde23/tutors/

Lecture

Lecture info

- every Thursday 10:15 – 12:00 in building E2.2, Hörsaal 0.01 (Günter-Hotz-Hörsaal) (Calendar)
- recording of the lecture on YouTube afterwards

Materials from the lecture

- slides: CMS
- code: GitHub

Office Hours

Regular Office Hour

- every Wednesday at 14:15 – 16:00 (Calendar)
- office hours take place in E1.1, R 3.06
- questions on assignment sheets and concepts from the lecture

Office Hour Prof

- directly after each lecture
- questions about the lecture

Vagrant Support

- Wednesday 19.04. at 14:15 – 16:00 in E1.1, R 3.06
- support in setting up the Vagrant VM

Tutorials

Tutorials take place on site on Mondays and Tuesdays (seminar room E1.1, R 3.06). (Overview)

Principle: designed as a LAB

1. 15 minutes discuss solution of the handed in assignment sheet
2. 75 minutes teamwork: solving simple exercises

Each tutorial deals thematically with the material of a 90-minute lecture

Participation

- choose your preferred tutorial time slots until 19.04. 23:59 on your personal status page in the CMS
- we will then assign you to the tutorial
- if you are unable to attend on your assigned date at some point, you are welcome to attend another tutorial

Assignment Sheets

Assignment Sheet info

- release: on Thursday evening after the lecture in the CMS
- submission: until the start of the next lecture in the CMS
- submission in groups of 2 to 3 students
- more detailed information about the submission on the respective assignment sheet
- sample solutions are provided in the CMS
- plagiarism leads to expulsion from lecture

Exam Admission

- a total of at least 50% of the points must be achieved in order to be admitted to the final and re-exam
- a maximum of two assignment sheets with 0 points or not submitted

ChatGPT et al

Rules

- You are fine to use it, these tools are a reality and here to stay.
- The rules of good scientific practice have to be followed, e.g. marking ChatGPT as source and documentation of the used prompts.
- If the use of ChatGPT is not indicated as the source, this may be considered an attempt to cheat.
- You won't be able to use ChatGPT in the exams.

Be Careful

- The quality of ChatGPT's answers ranges from complete rubbish to brilliant
- The problem is to distinguish the former from the latter

ChatGPT et al

Recommended Use

- Try to solve the exercises yourself first.
- In the end, the goal of the exercises is to prepare you to be able to apply the material form the lecture yourself.
- If you leave all of that to the machine, you won't learn from solving the exercises yourself and will have a hard time in the exam.
- use ChatGPT to explore topics, e.g. start with prompts like: "explain database technology as if I were a five year old".
- Then gradually ask for more complex explanations.
- Make sure to check answers against other sources, in particular database textbooks.

Exams

Exam dates

- exam: Thursday, 27.07.2023, 10–12
- re-exam: Friday, 15.09.2023, 14–16

Passing the lecture and grade

- to pass, at least 50% of the points must be achieved in the written exam or the re-exam
- the better result of the written exam and the re-exam determines 100

Python

We use Python and in particular Jupyter Notebooks to explain concepts in this lecture

Python Basics

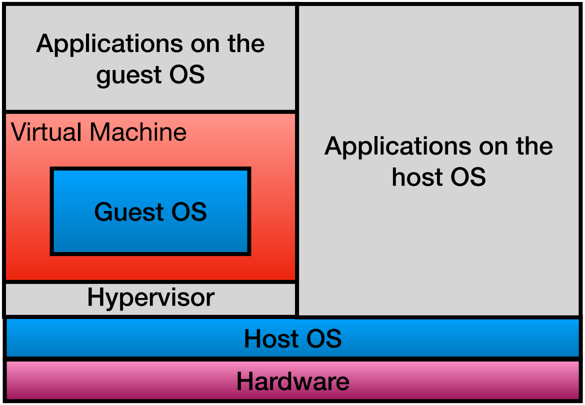
- we provide Notebooks and Videos explaining the basics
- we assume that you passed Prog 1 and Prog 2

Vagrant VM

- since we also use different systems besides Python (e.g. PostgreSQL) we provide a virtual machine
- instructions on how to set up the VM can be found here
- in the first tutorial and in the special office hour we help you with problems
- please try it on your own first!

Virtualisation

- abstraction layer between application and hardware
- virtualisation software, e.g. VirtualBox


Applications on the guest OS
Virtual Machine
Guest OS
Hypervisor
Applications on the host OS
Host OS
Hardware