# Practical Course: Cloud-Based Data Processing

Michalis Georgoulakis

Chair for Database Systems
School of Computation, Information and Technology
Technical University of Munich

# Motivation

**Input**: A 1TB click/log file + a small “lookup” table

**Task:** “What are the top 10 sites today?” (enriched with their category)

**Constraints:**

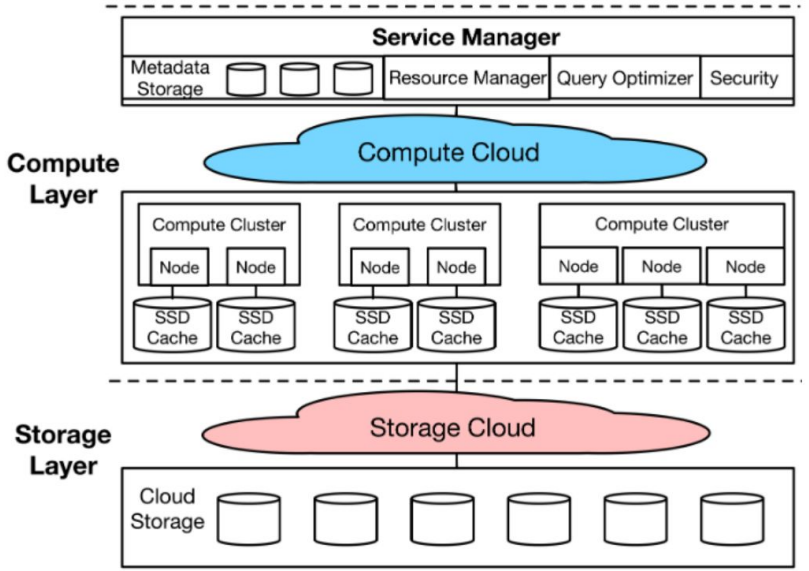
- Fast
- Cheap
- Survives failures

**At small scale:** a few lines of code on your laptop

# What changes when you move to the cloud?

Three new bottlenecks appear

- Coordination & skew
  - System as fast as its slower component
- Data movement
  - Network exchange dominates
- Storage reality
  - Data is no more co-located with compute




Src: Li et al. Cloud-Native Databases, VLDB'22

**The praktikum is about engineering around these bottlenecks.**

# Structure of the course

- **Phase 1**: Weekly/bi-weekly programming tasks (~8-10 weeks)
- **Phase 2**: Project (~4 weeks)

# Structure of the course

- Weekly meetings, Tuesdays at 14:00 (*02.11.018 Seminarraum*).
  - On-site.
  - First Session: 14.04.2026 (location TBD because of room unavailability)

- Mattermost Channel for offline communication
- GitLab server for uploading/submitting assignments.


# Phase 1 Contents

**I. Distributed Data Processing (programming model)**

- Network programming (coordinator ↔ workers)
- Partitioning & shuffle (scaling database operators)
- Fault tolerance basics (retries, stragglers)

**II. Cloud as a Runtime Target (cloud reality)**

- Deploy & operate on VMs/containers
- Compare pricing models (serverless, VMs)
- Communication in the cloud (latency, bandwidth, failures)
- Working with object storage (compute–storage separation)

**III. Reliable cloud-native applications (control plane)**

- Building a service you can trust under failures
- Consensus / replicated state (e.g., Raft)
- Integrate with your system (metadata/job state, scheduling)

# Phase 2: Project

**Individual Project:**

- Try to reproduce a paper, or perform a case study on one aspect (e.g., scheduling algorithms).
- Implement some fancy feature in a distributed system, or
- Work on a topic provided by a **PhD student** of the chair

- More freedom on programming language, might be able to use Chair servers or cloud resources depending on topic and advisor.
- Produce interesting insights and pave the way for future collaborations with us.

# What happens in a weekly session

- **Q&A / troubleshooting** (current + previous assignments)
- **Key Concepts** (theory + techniques for next assignment)
- **Systems Spotlight** (industry + research system example)
- **Hands-on** (cloud setup + live demo + guided work)

# Grading

- Grading based on the assignments and final project.
- Both code and presentation are important.
- Grade thresholds TBD.

# Requirements

- Interest in *databases* and *distributed systems*.
- Basic understanding of *modern computers*
- Programming fluency in one *systems language* (**C/C++**, **Rust**).
- *Linux* programming and access

**Recommended previous courses (or similar)**

- Basic Principles: Operating Systems and Systems Software (IN0009)
- Fundamentals of Databases (IN0008)
- Introduction to Computer Networking and Distributed Systems (IN0010)

**The praktikum is not aimed for students who have taken the CBDP course before.**

- Course and Praktikum are diverging this semester; overlap expected

# Objectives

After the course, the students should be able to

- Design and implement distributed data processing applications
- Understand key cloud and distributed-systems design trade-offs
- Deploy and operate their system on the cloud
- Build reliable cloud-native services, including replicated state and consensus.
- Reliably measure performance and cost, run controlled experiments, and analyze results scientifically.
- Read, critique, and discuss related systems research papers

# Materials

- Old slides of CBDP
- "Designing Data-Intensive Applications" by Martin Kleppmann;
- Related videos from CMU Database group youtube channel

# Important Dates

- **18/02 - 23/02:** Register your priorities via Matching System
- **26/02:** The matching results will be published around 6 pm.

- **14/04 - 14/07:** Teaching Period (13 sessions)

# Questions

Thank you for your attention!