Data at Scale:

Course Description: Principles of data modeling, storage, transport, and processing at large scales. Failure tolerance and coordination at large scale.

What you will learn:

This class explains how very large data sets are stored, maintained and processed. In the last decade, we have seen a redefinition of what “large” means. We have seen the advent of virtual machines and of Infrastructure as a Service.

The question of how to organize commercial data for automatic processing has been with us since the advent of computing for business. Some of the solutions considered then have now reappeared in disguise as new solutions. Our journey through the world of cloud-based services therefore starts with traditional database systems.

If your current computer does not support your application, the easiest way is vertical scaling, just getting a new, better computer. However, this opportunity exhausts itself. The next step is to distribute the work over several systems. But this is still a far cry from processing the oceans of data that some application are processing. As we scale up to thousands and millions of machines and storage devices, new phenomena emerge. Failure of individually reliable components becomes common and consensus and cooperation become unachievable in the basic interpretation of these terms. New technologies such as families of no-sql databases and processing paradigms have and are still emerging.

You will gain / reinforce practical skills:

- How to determine requirements of data processing at large
- How to interface with traditional database systems — SQL
- How to use PIG in order to avoid programming in the Map-Reduce paradigm
- How to use Amazon web services.

Text Books:

- Martin Kleppmann: Designing Data-Intensive Applications, O’Reilly 2017 (strongly recommended)
- Michael Wittig & Andreas Wittig: Amazon Web Services, Manning, 2019 (recommended)

Contents:

1. Introduction: Reliability, Scalability, and Maintainability
2. Data Models and Query Languages:
    1. Overview of traditional data base systems: What is a database, what does it provide, and what does it cost
        1. Hierarchical Databases Codasyl
        2. Network model
        3. Relational databases
        4. Object Oriented and Object Relational Databases
        5. Transactions: ACID vs BASE
        6. Data modeling

        7. SQL
    2. Documentary and Graph-Like Data Models
    3. Storage, Transmission, and Retrieval
        1. Language specific formats
        2. JSON, XML, AVRO
    4. Dataflow
3. Distributed Systems Principles
    1. Failure Tolerance
    2. Replication and Partitioning (Sharding)
    3. Distributed Consensus
    4. CAP “Theorem”
    5. Distributed relational databases
    6. Scalable Distributed Data Structures (SDDS)
        1. LH*, HBase, BigTable
    7. Coordination Services: Zookeeper
4. Storing Data at Scale
    1. No SQL Databases
        1. Column Databases
        2. Document and XML Databases
        3. Key-value
        4. Graph-based databases
        5. Examples: AllegroGraph, Cassandra, Couchbase, CouchDB, Dynamo, Giraph, MongoDB, Neo4j, Zookeeper
    2. Distributed File Systems: GFS, HDFS
5. Processing Data at Scale
    1. Map-Reduce Paradigm
        1. Hadoop
        2. Pig
    2. Data flow engines: Spark, Tez, Flink
6. Web Services
    1. Virtualization
    2. Infrastructure as a Service

Grading

Biweekly homework (in printed form, no electronic submission)	30 %
Midterm Examination	30 %
Final Examination	40 %

Instructor

Thomas Schwarz, CU 320B

Course Web Page:

tschwarz/mscs.mu.edu/~Classes