COSC6340: Data Systems: Theory and Infrastructure

Catalog Description

To be updated.

Prerequisites

The course is self-contained, assuming the student has a strong CS background. It is expected students have programming experience and basic knowledge on: optimization algorithms, relational database systems, operating systems and implementing programming languages (i.e. compilers theory). On the programming side, students should be familiar with Unix (Linux) development environment including ssh, vi, g++, python and node.

Course contents

This is a graduate computing systems-level course on implementing data storage, data retrieval and data analysis systems, including structured database-like files (e.g. CSV, JSON), relational database management systems (SQL, DBMSs), and data science Python libraries (e.g. Pandas, JSON). There is no textbook, but these references are helpful: [2], [3] and [1]. The course will require reading research papers and reading from open source development web sites.

Topics include the following. Review of relational database systems theory. Internal subsystems of most data systems: new hardware, secondary storage principles, faster storage architectures, buffer and main memory management, indexing data structures, concurrency and version control, transaction processing, fault tolerance: for transaction processing (recovery) and for long query processing (parallel, incremental). Query and transaction processing: query optimizer, faster transaction processing (distributed, lockfree, relaxing consistency), Advanced SQL (SPJA queries, derived tables/view, pivoting, recursive queries). Reverse data model engineering, security mechanisms against hackers, pushing processing to main memory, blockchain support for distributed transactions, query languages and data manipulation libraries beyond SQL and Datalog.

Grading

- 60%: 1 programming project. Developed in a team of 2 students. Project due in 2 phases, 1 month each. Project assigned on 2nd week of class. Project due one week before last class.

- 40%: 2 written exams: individual, 20% each, in class, closed-everything (no electronics, no book or notes). 1st exam: 3rd week of class. 2nd exam: 8th or 9th week of class.

Exams: based on papers and textbook chapters indicated in class. Exams will not be based on slides.

Project: Programming will be developed with a combination of different programming languages including C++, C, ANSI SQL and Python libraries, depending on assigned project. Other languages like Java, SPARQL, JavaScript may be required. Programming project will be developed in pairs (i.e. a team of 2 students). Teams will be assigned by instructor, but justified exceptions can be made (e.g. two students already
working in a lab). Programs will be carefully tested for correctness and benchmarking, with correctness being the most important requirement, followed by a comparison against competing systems or libraries.

Contact information

- Messages: messaging system chosen on 1st week of class (e.g. discord (most likely), whatsapp, telegram). MS Teams will not be used because it does not scale well to hundreds of messages and it is not efficient for smartphones.

- Preferred contact for personal matters: in person after class. Otherwise, a short email message to firstname@uh.edu. Warning: do not write to the other Ordonez professor, in the Physics department.

- Office hours: right after class, posted on CS department web pages.

References

