Project

Getting hands-on experience with state-of-the-art data systems!

Every student should complete a semester-long class project. The students can decide between a systems project and a research project.

Useful links:

- Introduction to Storage and Memory Hierarchy (resources/short-intro-storage-hierarchy.pdf) and some example code (https://github.com/manathan1984/CacheTests)
- Introduction to debugging and performance tools [material developed at Harvard] (resources/DevTools.pdf)
- Performance monitoring [material developed at Harvard] (resources/PerfTools.pdf)
- TPCH Data and Query generator tool (resources/tpc-h-tool.zip)
- A short summary of how to efficiently code multi-cores (https://dl.acm.org/citation.cfm?doid=2588555.2588892) (for a more detailed version drop by my office)

System Projects

A system project sharpens your systems skills and provides background on state-of-the-art systems, data structures and algorithms. For a successful systems project you will design and implement a systems component in C or C++, and you will deal with low-level system

implementation details like memory allocation and management, cache-aware processing, parallel and concurrent processing and a deeper understanding of read/write performance trade-offs, and performance scalability. Systems projects will be carried out by groups of two students.

This year we will have two topics for a systems project.

Project

Implementation of LSM-Trees (documents/CS591-S20-SysProj-LSM-trees.pdf)

Benchmarking RocksDB (documents/CS591-S20-SysProj-Benchmarking-RocksDB.pdf)

Research Projects

A research project, on the other hand, aims at challenging the state-of-the-art. The goal is (i) either to better understand an open research problem through analysis and benchmarking, or (ii) to solve open problems through new designs and proof-of-concept implementations. The ultimate goal of a research project is to give a taste of research to students, and ideally lead to publications. When working on a research project, the student will interact with the instructor and the teaching assistants closely. Students will work in groups of three students.

We have a number of possible research topics below. The students can also propose their own project (subject to instructor's approval).

Subjects

Develop a benchmark for sortedness (documents/CS591-S20-Research-Build-a-sortedness-benchmark.pdf)

Develop sortedness-aware access methods (documents/CS591-S20-Research-Sortedness-aware-Access-Methods.pdf)

Query-driven LSM-Tree Compaction (documents/CS591-S20-Research-Query-driven-LSM-Compaction.pdf)

Implementation of a variable-size bufferpool (contact instructor for more details) ()

Design and Implementation of a Dynamic Range Filter (contact instructor for more details) ()

Optimize memory allocation for LSM-Trees between fence pointers, buffer, and Bloom filters (contact instructor for more details) ()

Build a learned index (contact instructor for more details) ()

Instant Recovery on Non-Volatile Memories (contact instructor for more details) ()