Trie-based Indexes in In-Memory HTAP

**Keywords:** DBMS, Concurrent Programming, Data Structures, Indexes, Trie

**Problem:** DBMS employs indexes for fast and efficient range- and point-queries in the database. Indexes enable DBMS to quickly select data without scanning all records in the table. Indexes, through trie data structures provides favorable properties, that is, sorted tree structure which enables sublinear search for point lookups, and range traversals. However, indexes are mostly studied in the context either in pure online-transaction processing (OLTP) or online-analytical processing (OLAP) but not in the case of the mixed workloads, that is hybrid analytical and transactional processing (HTAP).

**Project:** In this project, the student will implement an adaptive radix tree [1] and Cuckoo trie [2] in the HTAP engine, Proteus. He/she will familiarize himself/herself with trie-based data structures, concurrent data structures, and performance tuning of such data structures for high-performance transactional and analytical workloads. Furthermore, the student will experimentally evaluate, analyze, and compare the different trie-based indexes in HTAP workloads.

**Plan:**
2. Implement Cuckoo Trie index.
3. Analyze performance gains and bottlenecks compared with each other, and to hash-indexes.
4. Analyze performance implications in an HTAP setting.

**Supervisor:** Prof. Anastasia Ailamaki, anastasia.ailamaki@epfl.ch

**Responsible collaborator(s):** Aunn Raza, aunn.raza@epfl.ch

**Duration:** 6 months

**References**
