CAS CS 591 A1: Data Systems Architectures
Massive Data Algorithms and Systems Group
Department of Computer Science
College of Arts and Sciences, Boston University
https://midas.bu.edu/classes/CS591A1

CS591 A1 Spring 2020 - Research Project

Title: *Query-driven compaction in LSM-trees*

Background: Log-structured merge-trees (LSM-trees) [1, 2] are one of the most commonly used data structures for persistent storage of key-value entries. LSM-trees store data in the disk as immutable logs (also known as sorted sequence tables (SSTs)), which are maintained in hierarchical levels of increasing capacity. To bound the number of logs that a lookup has to probe, LSM-trees periodically sort-merge logs of similar sizes. Sort-merge in LSM-trees (also referred to as *compaction*) is principally driven by the ingestion of entries in the tree. Range queries, on the other hand, merge all qualifying runs with an overlapping key range to identify and ignore older entries. However, state-of-the LSM-tree designs do not take advantage of the sort-merge performed for range queries by writing the result to the tree.

Objective: The objective of the project is enable (range) query-driven compaction in LSM-trees. The workflow for the project is as follows.

(a) Get familiarized with the underlying mechanism for range queries in the vanilla implementation of the LSM-tree.

(b) Design query-driven compaction algorithms, which exploit the sort-merge performed for range queries by persisting the latest results in the tree while discarding the older ones.

(c) Implement the proposed solutions on top of RocksDB, and analyze their performance with respect to the state-of-the-art.

[1] Patrick E. O'Neil, Edward Cheng, Dieter Gawlick, Elizabeth J. O'Neil. **The Log-Structured Merge-Tree (LSM-Tree)**. Acta Inf. 33(4): 351-385 (1996)

[2] Niv Dayan, Manos Athanassoulis, Stratos Idreos. **Monkey: Optimal Navigable Key-Value Store**. SIGMOD Conference 2017: 79-94

[3] Facebook. **RocksDB.** *http://github.com/facebook/rocksdb.*