CS591A1 Spring 2019 - Systems Project

Title: *Benchmarking RocksDB*

Background: Log-structured merge-tree (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 merge logs of similar sizes. RocksDB [3] is a high-performance, persistent LSM-tree-based key-value store, which is widely used today across Facebook. The primary objective with RocksDB is to efficiently meet the service level requirements (particularly, read and update latencies) while making full utilization of the underlying hardware.

Objective: The objective of the project is to build a benchmarking suite that takes as input arbitrary workload specifications and executes them on top of RocksDB. The workflow for the project is as the following.

(a) Get familiarized with the RocksDB infrastructure (open-source in github)
(b) Build a command-line interface, which can populate a database with data, given the workload details (i.e., proportion of point queries, range queries, point inserts/updates/deletes, and range inserts/updates/deletes), the distribution of access patterns, and the data size (and range).
(c) Execute different workloads on top of the database and benchmark the performance of RocksDB against the workloads.

[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.*