Motivation

- Modern persistent key-value stores, such as RocksDB [1] and LevelDB [2], typically use log-structured merge trees (LSMs) [3].
- Stellar write performance: large sequential writes exploit the high sequential throughput.
- Slow read performance: need caches, Bloom filters [4, 5], compaction strategies [6, 7, 8]—complex and hard-to-tune. [9]

Random Writes = Sequential Writes on NVMe SSDs

- Random write throughput across (i) request sizes, and (ii) number of writing threads.
- With high parallelism, can reach advertised peak sequential write throughput [10].

LSMs Leave Read Performance on the Table

- Zipfian-distributed (θ = 0.79) workload of reads, updates, and scans on 64 B records.
- Of the requests that are not updates, 10% are range scans and the rest are point reads.
- For read-heavy, the disk-based B-tree outperform/is competitive against RocksDB.
- TreeLine can outperform both systems all the way up to 80% updates.

Design Overview

Key Idea A: Record Caching

- Key-to-page mapping is expensive to change in update-in-place design.
- But variable hotness among records on the same page.
- Solution: Only cache records, to increase memory efficiency.

Key Idea B: Page Grouping

- Unlike writes, sequential reads still faster vs. random reads on modern SSDs.
- Pages must be sequential to make scans fast.
- Solution: Group pages into contiguous segments. Use linear models to index records within segments.

Key Idea C: Insert Forecasting

- To avoid constant reorganization, pages should have some empty space.
- But too much empty space increases I/O amplification.
- Solution: Leave empty space based on epoch-based insert forecast.

Evaluation

Point Workloads

- TreeLine outperforms RocksDB (LeanStore) by 1.62× (2.81×) and 2.99× (1.53×) on average for 64 B/204 B records.

Scans

- With 16 request threads, for uniform scans, TreeLine outperforms RocksDB (LeanStore) by 2.21× (1.58×) and 2.50× (2.80×) on average for 64 B/204 B records.
- With 16 request threads, for Zipfian-distributed scans, TreeLine outperforms RocksDB (LeanStore) by 1.74× (0.91×) and 1.88× (1.86×) on average for 64 B/204 B records.