TreeLine: An Update-In-Place Key-Value Store for Modern Storage

Geoffrey X. Yu*, **Markos Markakis***, Andreas Kipf*,
Per-Åke Larson, Umar Farooq Minhas, Tim Kraska

Code: github.com/mitdbg/treeline
Paper: tinyurl.com/treeline-paper

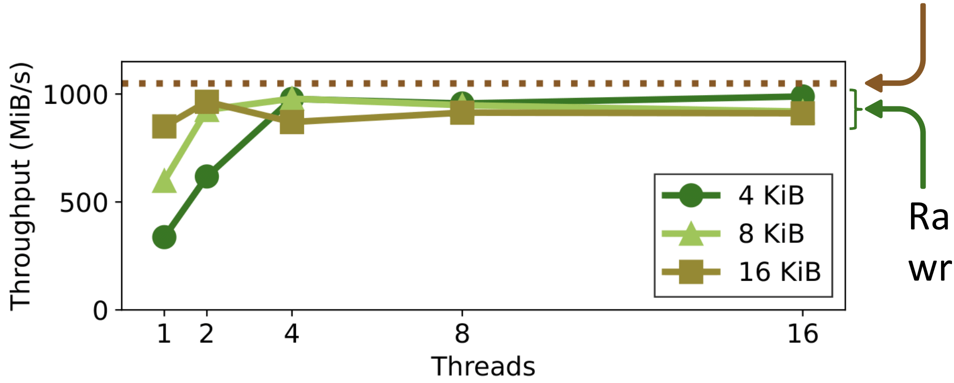
Photo by Richard Main on Unsplash

Modern storage unlocks new opportunities

Persistent key-value stores favor Log-Structured Merge (LSM) trees

- Absorb writes in memory, write them out to disk sequentially whenever out of space.
- Periodically compact on-disk files into logarithmically larger ones.
- Good when writing sequentially to storage is crucial.

Random parallel writes are also fast on modern SSDs



Sequential writes

Random writes

Update-in-place designs can be made efficient

3 Key Ideas to balance reads/writes

- **Cache records** to maximize utilization.
- Leverage linear models to **group pages** and accelerate scans.
- **Forecast inserts** to allocate appropriate space.

TreeLine shines across the standard YCSB workloads

- Point workloads: **2.20x** and **2.07x** over RocksDB, LeanStore on average
- Uniform scan-heavy (16 threads): **2.50x** and **2.80x** over RocksDB, LeanStore
- Up to **10.95x** and **7.52x** over RocksDB, LeanStore overall.