Scaling Concurrent Log-Structured Data Stores

Edward Bortnikov, Guy Gueta, Eshcar Hillel
Yahoo Labs

Idit Keidar
Technion EE, Yahoo Labs

Log-Structured-Merge (LSM) Data Stores

Goal
Dramatically speedup in-memory operations

How
Non-blocking (lock-free) synchronization

Challenges
Support rich API:
Atomic R/W, Snapshot Scan, RMW

Impl
Extend popular open-source LevelDB

cLSM - Scalable Concurrent LSM

Merge
update pointers before & after merge
*lock exclusive* mode (block writes)

Write
*lock shared* mode
update \(C_m\)

Read
*no locking*
read \(C_m\), then \(C'_m\), then \(C_d\)
may read same data twice

Scan
*lock shared* mode to get a timestamp
iterate over \(C_m\), \(C'_m\), and \(C_d\)
track active operations
beware of races

RMW
*lock shared* mode
read \(C_m\), then \(C'_m\), then \(C_d\)
update \(C_m\)

Evaluation
RocsDB, bLSM, LevelDB, HyperLeveDB, cLSM