Building Fast Recoverable Persistent Data Structures

Haosen Wen*, Wentao Cai*, Mingzhe Du, Louis Jenkins, Benjamin Valpey, and Michael L. Scott

University of Rochester

*equal contributions

UNIVERSITY of ROCHESTER

Goal

Build recoverable data structures as fast in *online phase* as possible.

Achieve better performance by **relaxing ordering constraints** and **reducing the amount of persistent data**.

Relax Ordering Constraints

(Non-buffered) durable linearizability requires operations to persist before return; Buffered durable linearizability (BDL) only requires operations *not* to persist in the *opposite* of their happens-before order and allows some progress loss in the wake of a crash. Operations can persist periodically in batches.
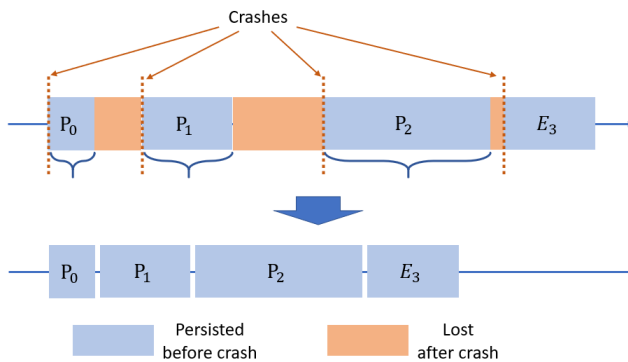

Periodic Persistence

Inspired by Dali[Nawab et al., DISC'17], Montage implements BDL: it breaks execution into epochs and issues fences only at epoch boundaries.
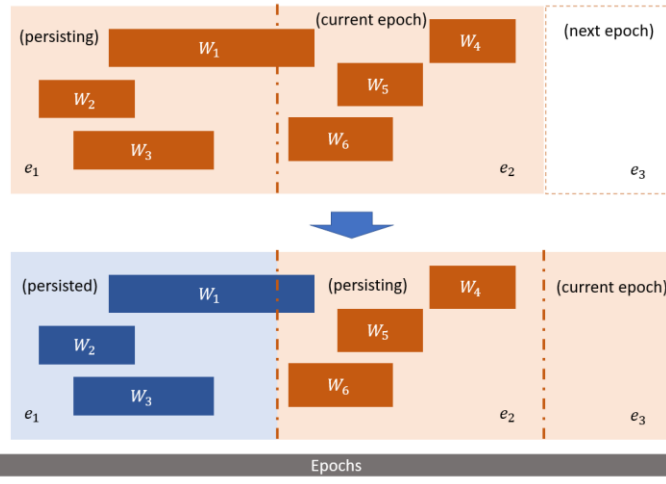
Reduce Persistent Data

Inspired by Pronto[Memaripour et al., ASPLOS'20], among others, Montage reduces the amount of persistent data by persisting only abstraction-related data. All transient indices can be rebuilt after a crash.
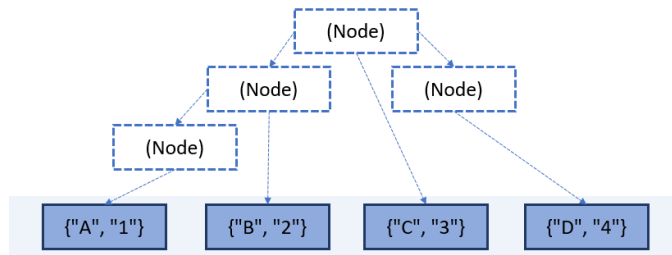
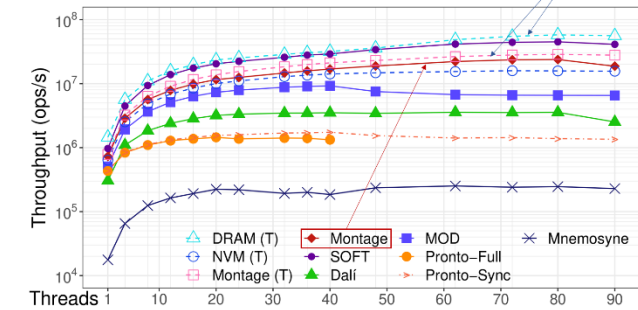
General Buffered Persistence

Montage is the first general buffered persistence system. It suits most abstractions including sets/maps, queues, and graphs.

Montage also works with nonblocking data structures with moderate API restrictions.

Great Performance

Montage shows great performance on multiple data structures:

Hash map (y log scale)

Memcached (linear scale)

Memcached: YCSB-A