Share Interface in Flash Storage for Relational and NoSQL Databases

Gihwan Oh, Chiyoung Seo, Ravi Maruyam,
Yang-Seok Ki, Sang-Won Lee

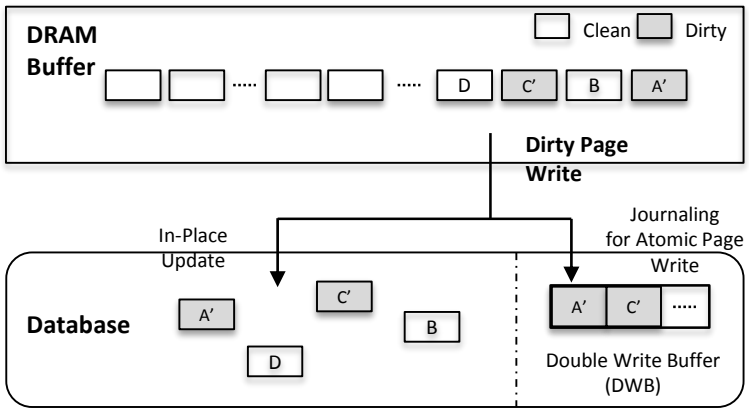
Overview

- Motivation
- Share Interface
- Application Extension with Share
- Performance Evaluation

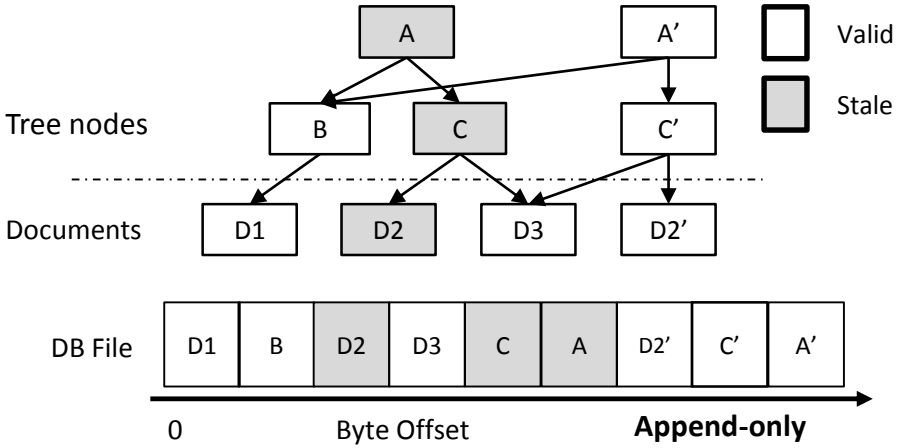
Motivation

- To guarantee write atomicity is critical for database
  - Journaling: MySQL/InnoDB, SQLite, and Sybase SQL Anywhere
  - Copy-on-Write: Append-only B+tree NoSQL engine (e.g. Couchbase)
- But, write amplification
  - Double-write journaling
  - Tree-wandering problem in B+tree-structured NoSQL engines

DWB in MySQL/InnoDB



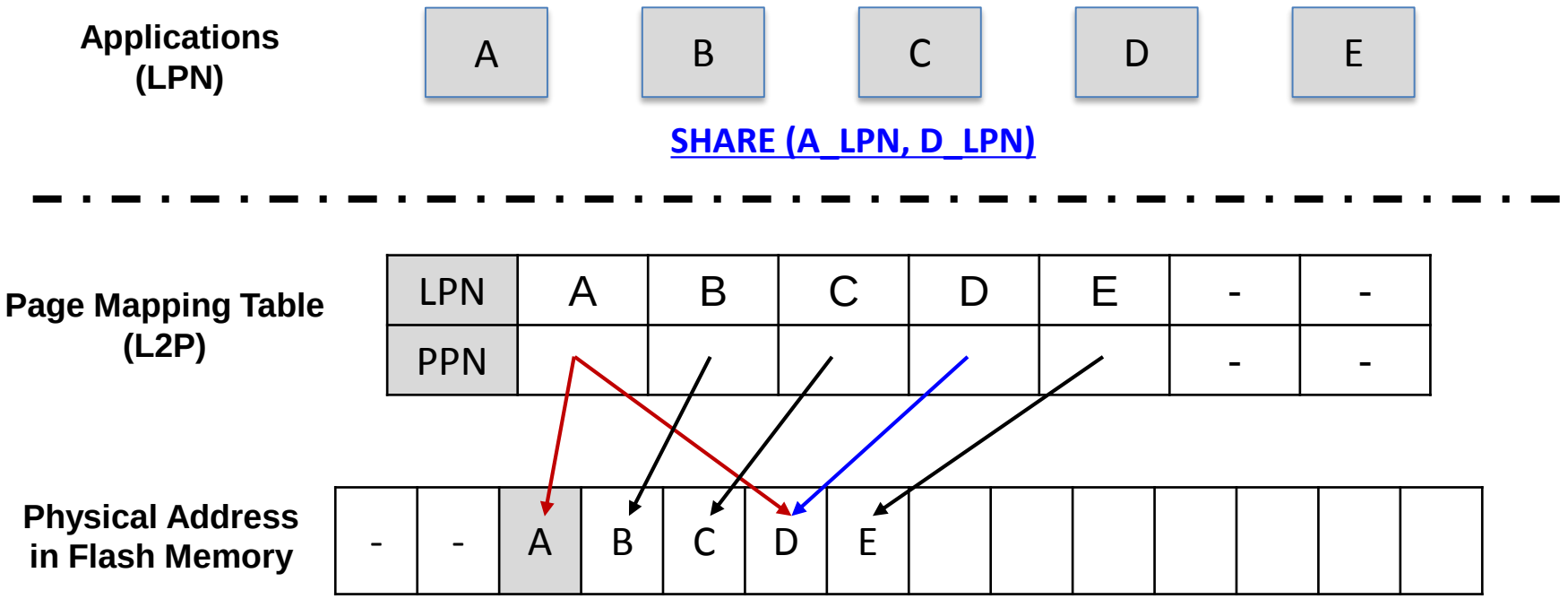
Copy-On-Write in ForestDB



Share Interface

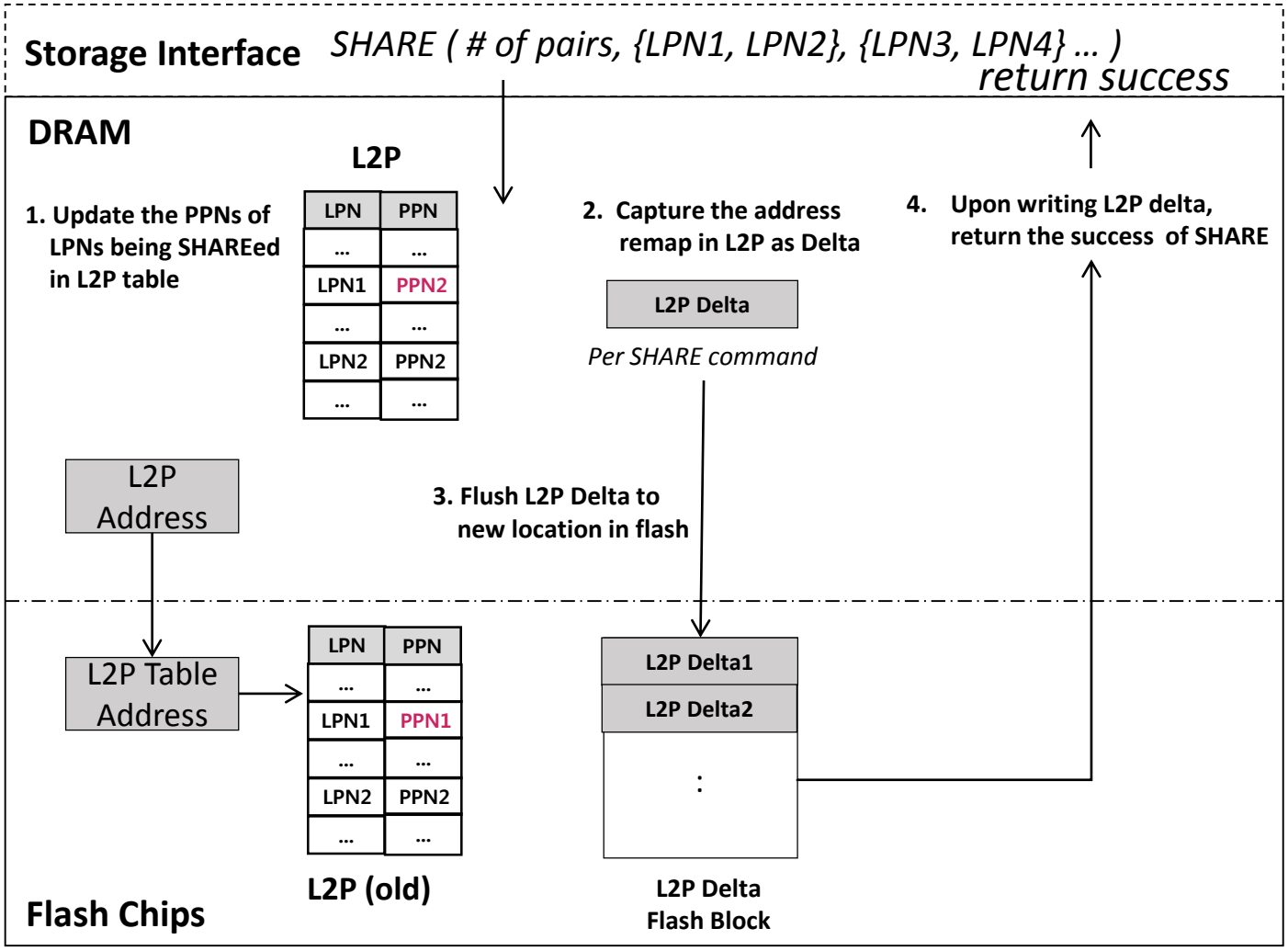
- Flash storage
  - Flash memory does not allow in-place update
  - Thus, flash storages are equipped with Flash Translation Layer (FTL)
  - Page mapping FTL: LPN → PPN
- Share interface
  - Explicit semantic interface beyond read/write operations
  - c.f. X-FTL

Share Interface (2)



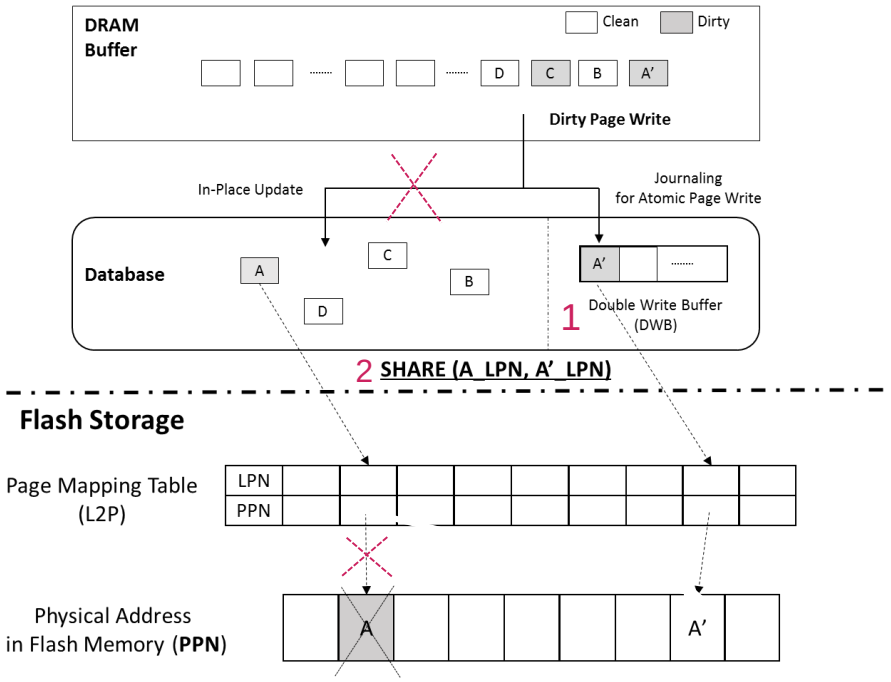
- Optional len. parameter, multiple pairs of LBAs
- Upon receiving share, FTL should remap L2P mapping table atomically and durably

Share Interface (3)

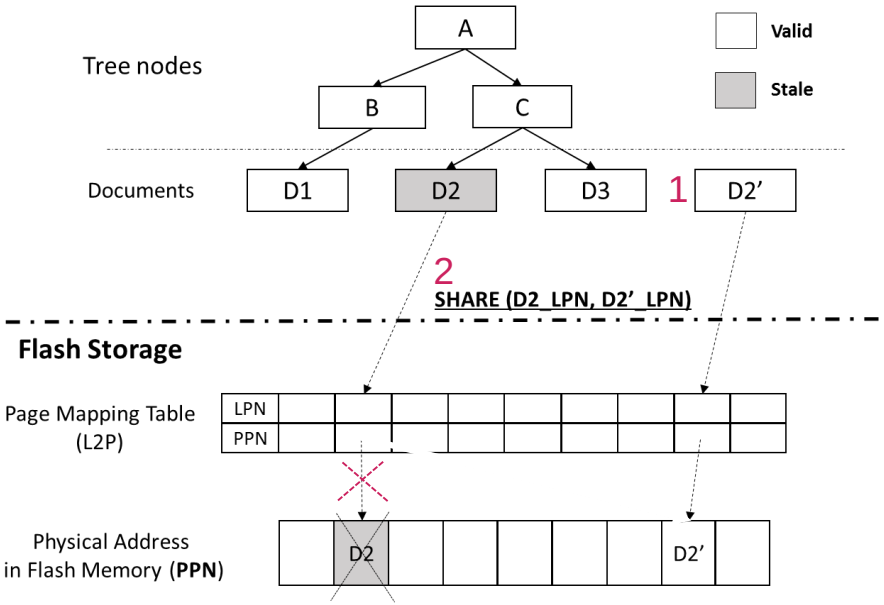


Share: Use Cases

DWB with SHARE

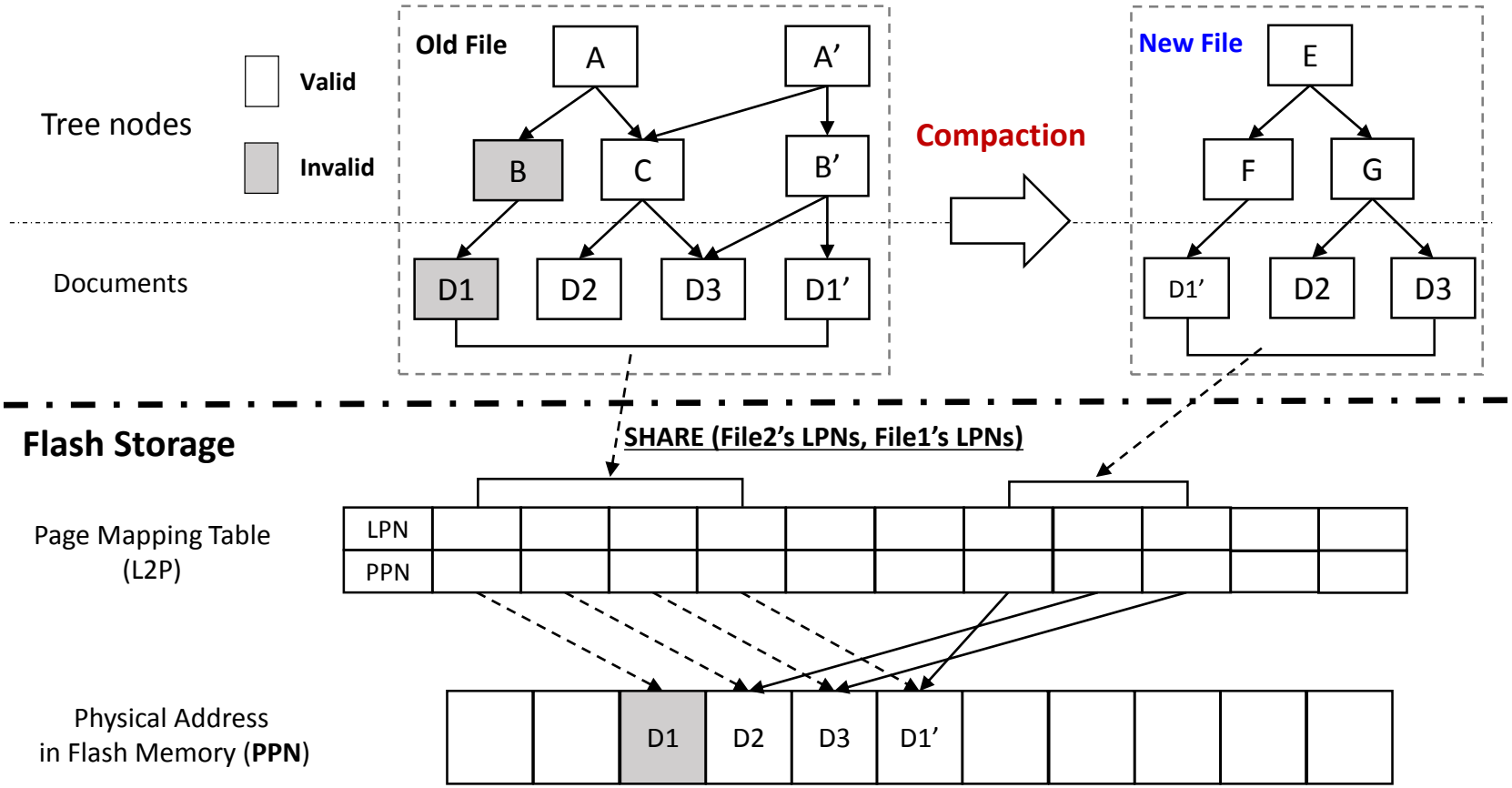


ForestDB with SHARE



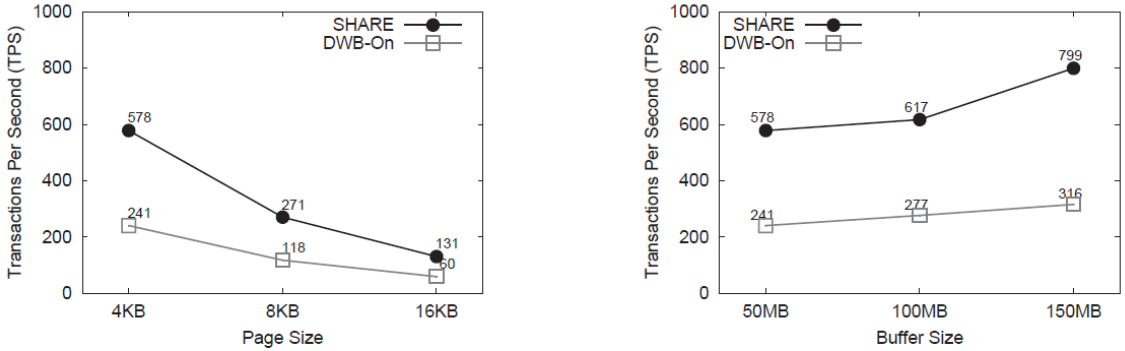
Share: Use Cases

- ForestDB's Compaction with SHARE



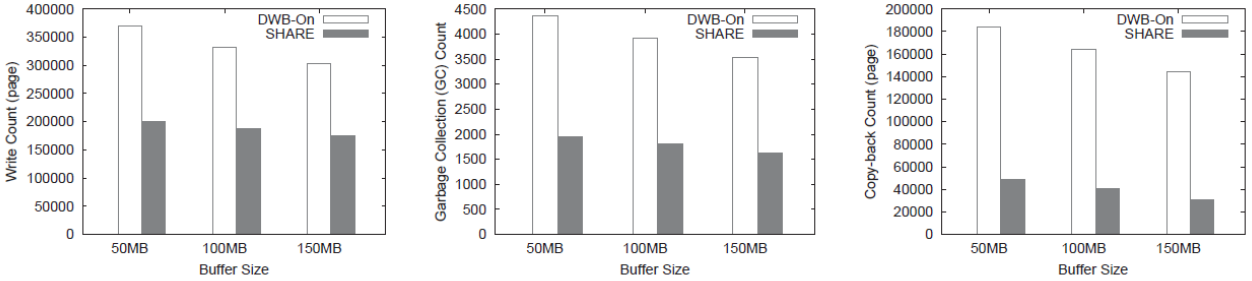
Performance Evaluation

- LinkBench on MySQL/InnoDB
  - MySQL 5.7.5-m15 with Linkbench workloads
  - Varying page size in InnoDB: of MySQL/InnoDB



(a) Varying page size (b) Varying buffer size

Figure 5: LinkBench throughput on MySQL/InnoDB

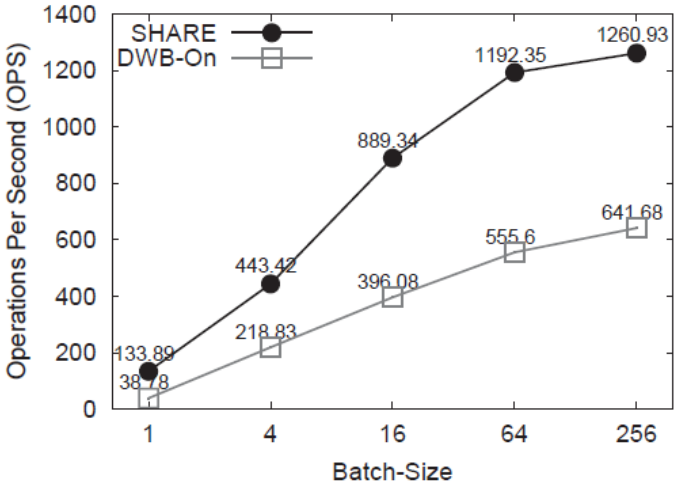


(a) OS page write count (b) Garbage collection count (c) Copyback page count

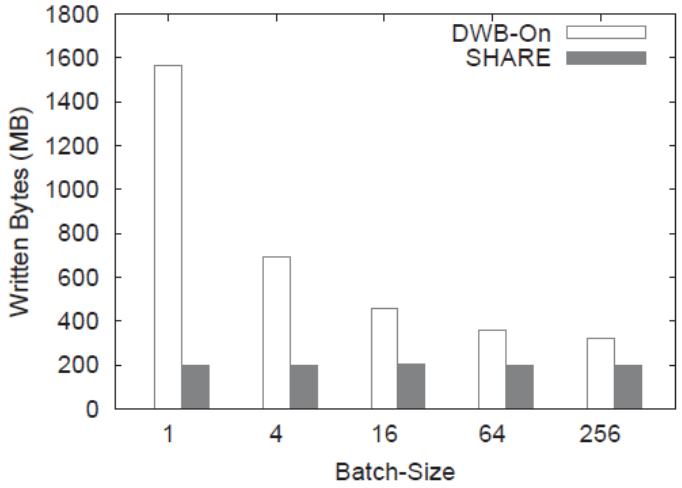
Figure 6: IO activities inside OpenSSD (50MB buffer cache, 4KB page)

Performance Evaluation

- YCSB Workload-F on ForestDB



(a) Throughput



(b) Total amount of written data

- Compaction

	Elapsed Time (sec)	Written Bytes (MB)
Original	277.52	1126.4
SHARE	88.38	150.6

Related Work

- Transactional FTL: vs. X-FTL
  - In-place update vs. out-of-place
  - Transaction concept: explicit vs. implicit
- JFTL
  - Inspired X-FTL and Share
  - Specific to FS journaling and metadata (implicit)
- ANViL (FAST 2015)
  - Remapping at block layer
  - CoW-style mapping info. / background GC for map. Info. / Large-batch (O) vs. Small-random (X)

Candidate Appl.s for Share

- SQLite RBJ / WAL mode
- Postgres full page write
- File systems
  - Ext4 journaling, F2F2, BtrFS
- Else?
  - Ubiquitous double-write journaling
  - CoW > In-place update in flash storage

Q & A