LightTx: A Lightweight Transactional Design in Flash-based SSDs to Support Flexible Transactions

Youyou Lu\textsuperscript{1}, Jiwu Shu\textsuperscript{1}, Jia Guo\textsuperscript{1}, Shuai Li\textsuperscript{1}, Onur Mutlu\textsuperscript{2}

\textsuperscript{1}Tsinghua University

\textsuperscript{2}Carnegie Mellon University
Data updated in a single operation should be performed atomically and durably, and this is called a transaction.

Software Transactions

High overhead:
  Duplicated writes
  Synchronization for ordering

Software
Flash pages are update in an **out-of-place** way, but this property is transparent from the software by the Flash Translation Layer (FTL) in SSDs.

**Hardware Supported Transactions**

**Problem:**

- How to support different isolations? (Flexible transaction requirement from software)
- How to cluster pages for each transaction? (Internal Parallelism of SSDs)
Design Issues:

**Flexibility**: support tx with flexible requirements

**Lightweight**: low overhead on the device

Observations and Key Ideas:

Simultaneous updates can be written to different physical pages, and the FTL mapping table determines the ordering

=> (Flexibility) make commit protocol *page-independent*

Transactions have birth and death, and the near-logged update way enables efficient tracking

=> (Lightweight) track recently updated flash blocks, and retire the dead transactions

Results:

20.6% throughput improvement (flexibility)

Stable garbage collection overhead

Fast recovery with negligible persistence overhead
Today 1:15pm
CSA-2: Memory Systems

LightTx: A Lightweight Transactional Design in Flash-based SSDs to Support Flexible Transactions

Youyou Lu¹, Jiwu Shu¹, Jia Guo¹, Shuai Li¹, Onur Mutlu²

¹Tsinghua University
²Carnegie Mellon University