Flash Memory Summit

Realizing Instant Database Crash Recovery over SSDs with Transparent Compression

Kecheng Huang*, Zhaoyan Shen**, Zili Shao*, Tong Zhang†, Feng Chen‡

* The Chinese University of Hong Kong ** Shandong University

† ScaleFlux Inc. ‡ Louisiana State University

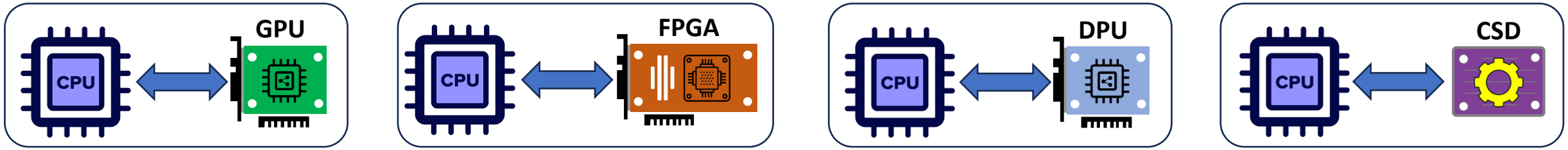
Presenter: Tong Zhang, ScaleFlux Inc.

Innovate Database in Heterogeneous Computing Era

Flash Memory Summit

. . . ORACLE MySQL PostgreSQL MariaDB snowflake . . .

influxdb mongoDB Cassandra Microsoft SQL Server AEROSPIKE



Off-load certain database processing tasks from CPU

❌ Significant changes to databases

❌ Questionable ROI

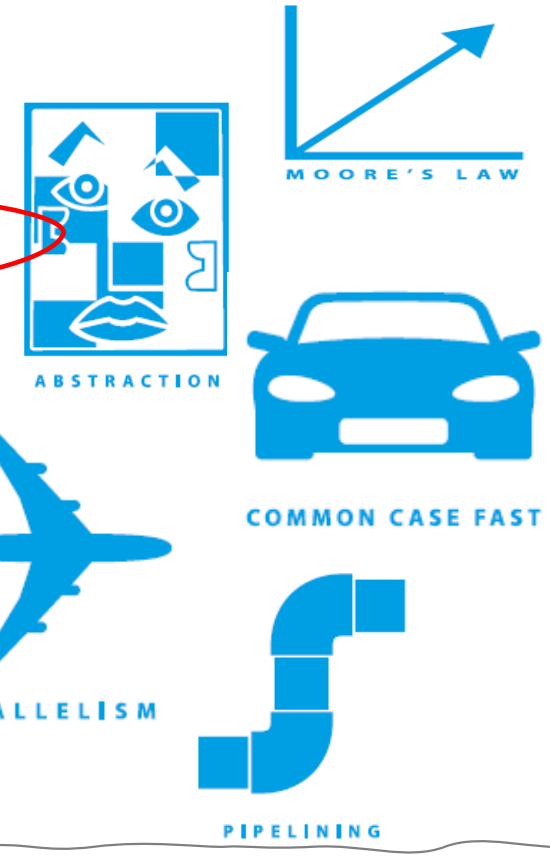
❌ Hardware vendor lock-in

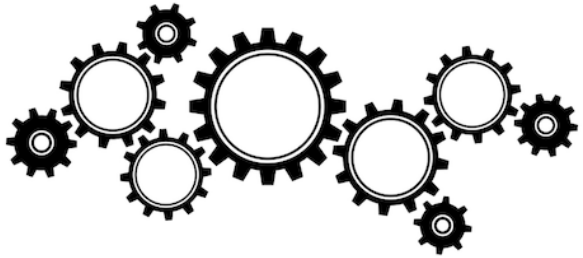
Innovate Database in Heterogeneous Computing Era

Eight Great Ideas in Computer Architecture (Patterson & Hennessy)

1. Design for ***Moore's Law***
2. Use ***abstraction*** to simplify design
3. Make the ***common case fast***
4. Performance *via* ***parallelism***
5. Performance *via* ***pipelining***

...

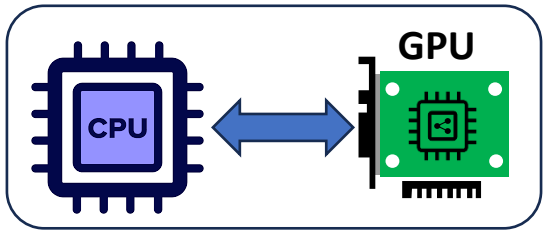


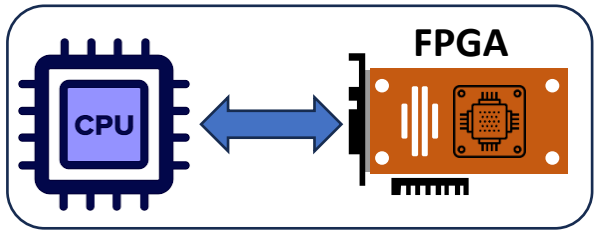


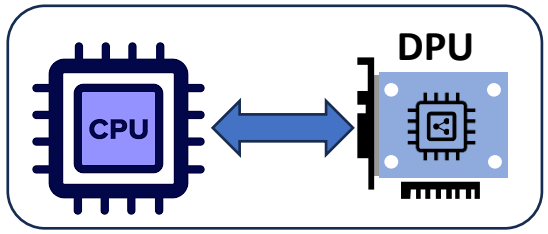
Computation offloading in heterogenous computing

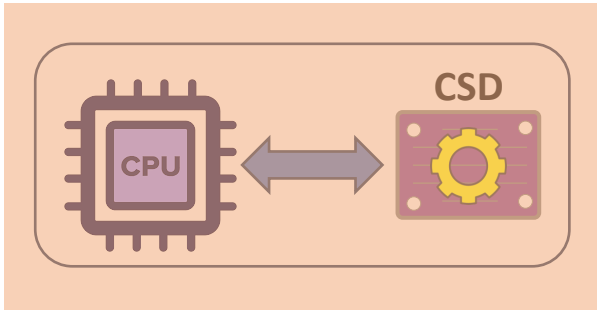
Breaks the principle of abstraction

Innovate Database in Heterogeneous Computing Era





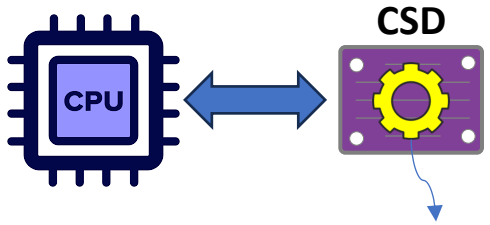




Off-load certain database processing tasks from CPU

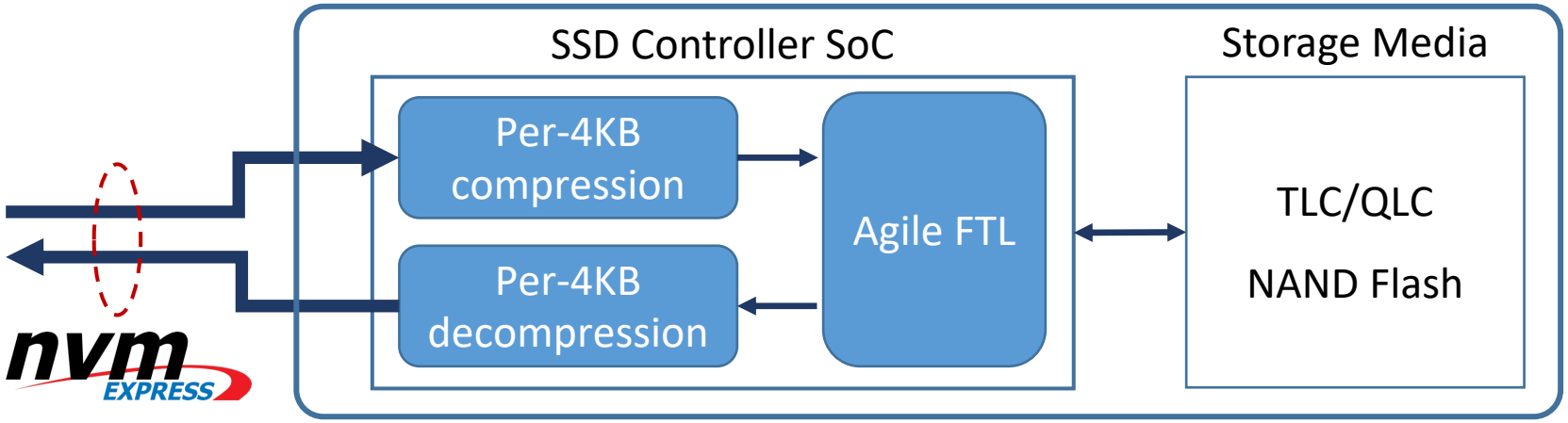
Innovate database in heterogeneous computing era **without** computation off-loading!

Drop the Computation Off-loading Mindset!



Native in-storage functions

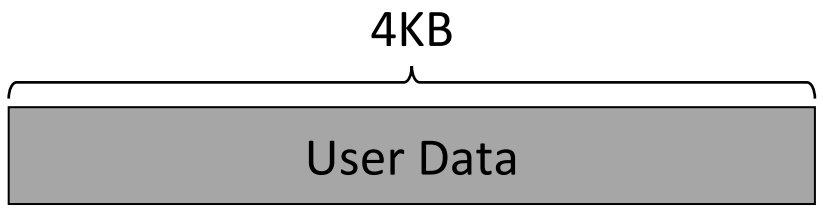
- Transactional memory/storage
- Performance profiling and counting
- Malware detection
- Snapshot
- Security
- Continuous data protection
- Compression
- Data deduplication
- Data transcoding & transformation
- . . .



- ✓ 100% compliant with NVMe
- ✓ Zero changes to I/O stack
- ✓ Transparently reduce cost & improve IOPS

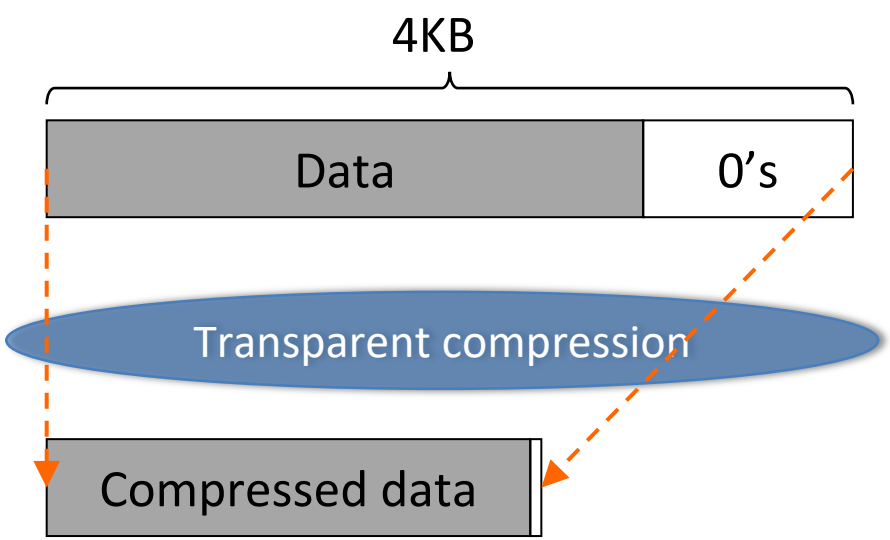
CSD with In-line Transparent Compression

Where the Innovation Potential Come From?

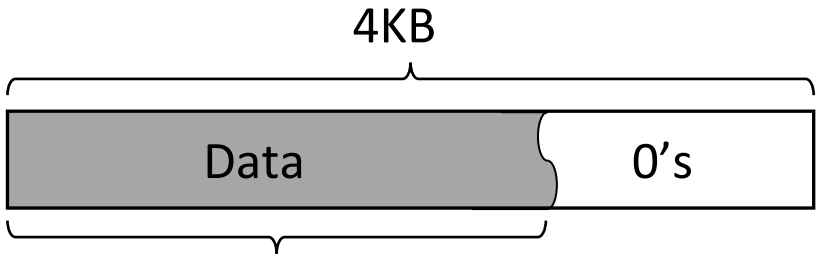


Fixed-size block I/O

Data Management SW



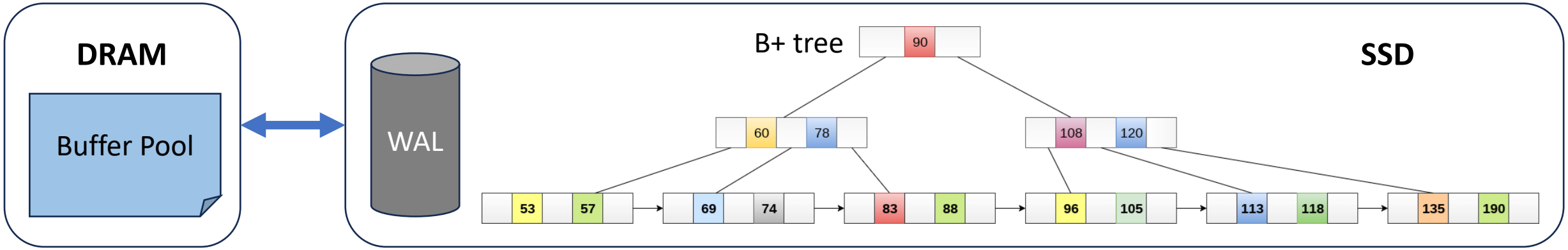
Unnecessary to completely fill each 4KB sector with user data



Arbitrary data size

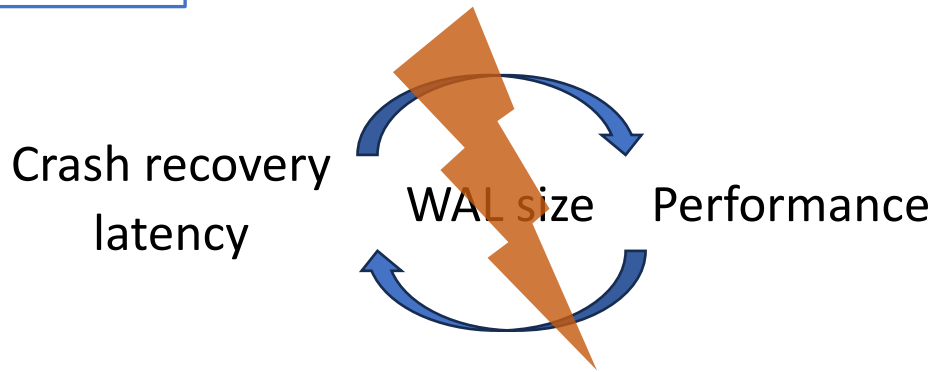
Virtually variable-size block I/O

Innovate Database over CSDs w/ Transparent Compression



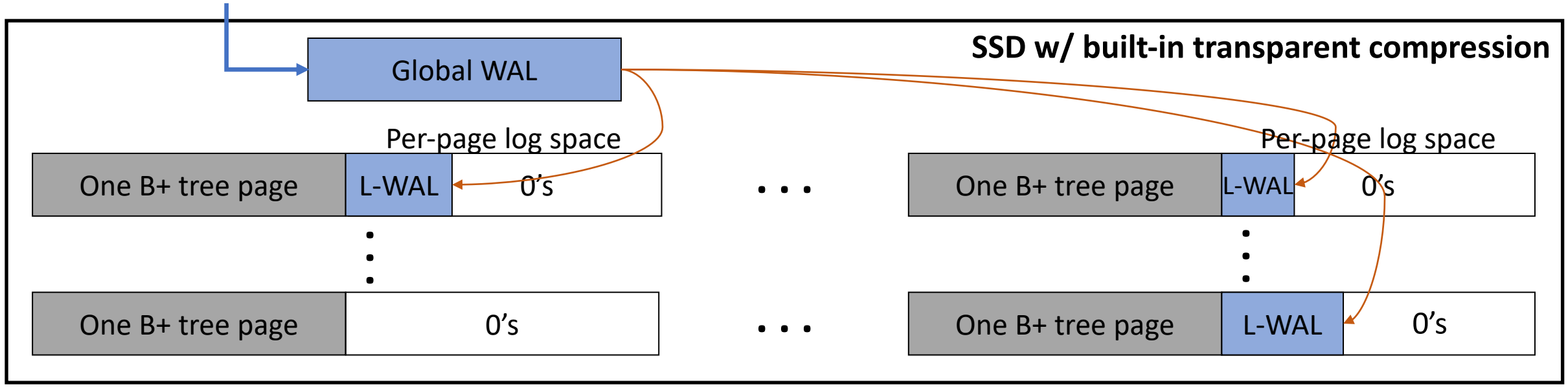
Crash recovery latency ↑ ← WAL size ↑ → Checkpoint frequency ↓ → Write amplification ↓ & performance ↑

Crash recovery latency ↓ ← WAL size ↓ → Checkpoint frequency ↑ → Write amplification ↑ & performance ↓



A Simple Idea: Two-Tier WAL

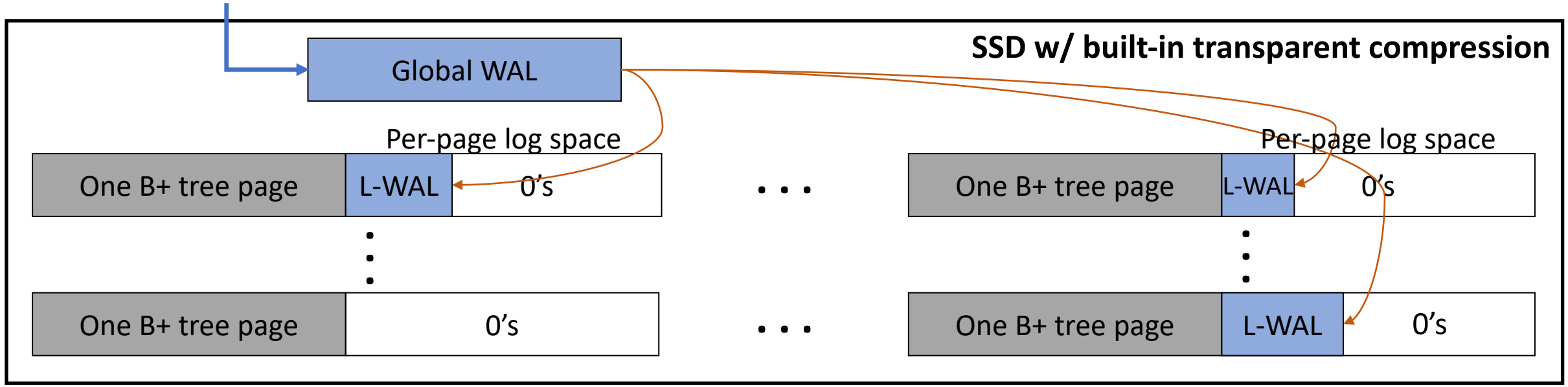
A small global WAL + Distributed per-page WALs



- Pre-allocate a local WAL (L-WAL) block for each B-tree page
- Do not re-play any log records in L-WALs during crash recovery

A Simple Idea: Two-Tier WAL

A small global WAL ➕ Distributed per-page WALs

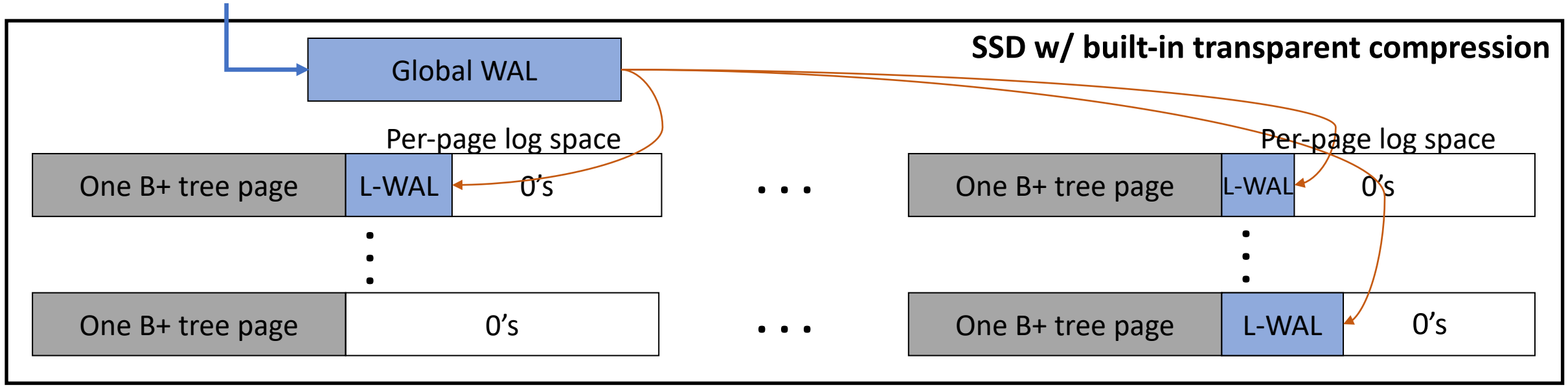


Decouple crash recovery latency from WA & performance

- ✓ Huge total L-WALs ➔ Lower write amplification ➔ Higher performance & longer SSD life
- ✓ Small global WAL ➔ Almost instant crash recovery

A Simple Idea: Two-Tier WAL

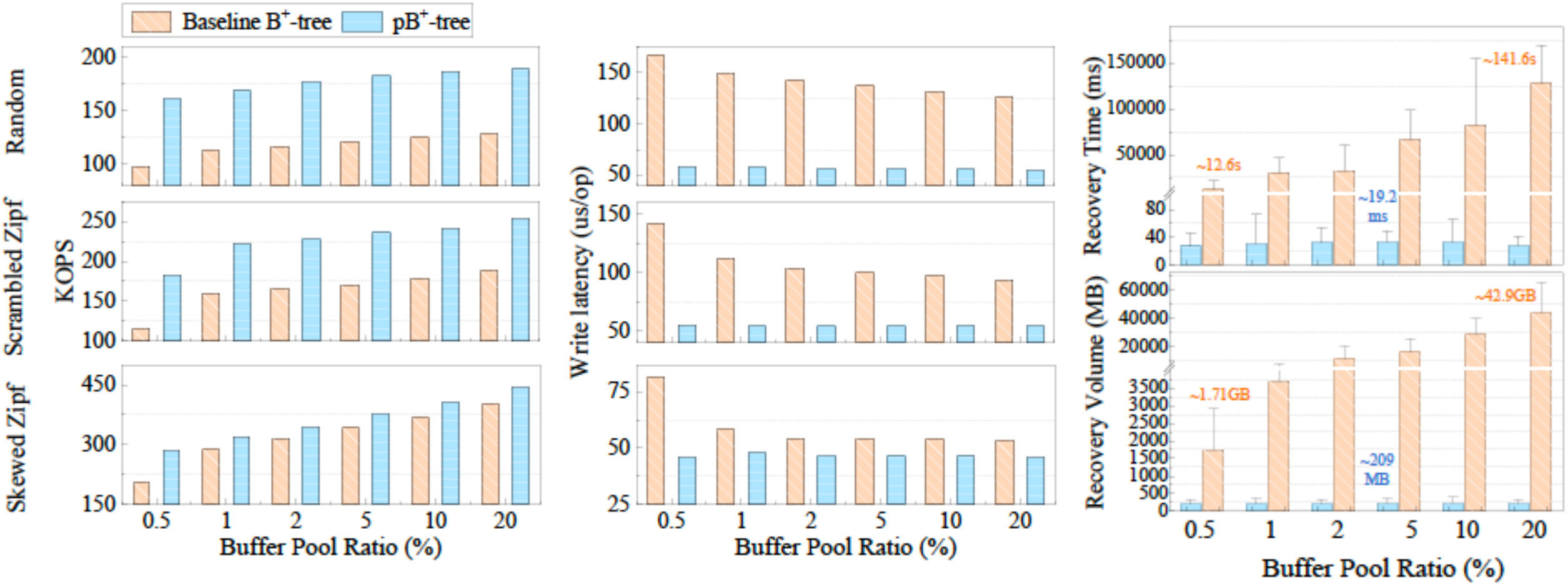
A small global WAL + Distributed per-page WALs



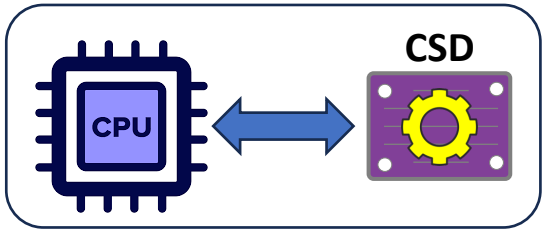
Prototype Implementation

- ✓ Implemented *p*B+ tree to incorporate the proposed two-tier WAL
  - ➢ Operate as a key-value store (PUT, GET, DELETE)
  - ➢ Contains a buffer pool with enhanced management policy
  - ➢ Supports blind key-value pair update
- ✓ Intel(R) Xeon(R) Silver 4310 CPU @ 2.10GHz, 128GB DDR4 DRAM
- ✓ ScaleFlux 4TB CSD 3000 with built-in transparent compression

Experimental Results



Conclusion



Explicit computation off-loading → Innovate database **without** computation off-loading

- ❑ Innovate database over 100% NVMe-compliant CSD with built-in transparent compression
- ❑ A simple two-tier WAL design solution for B+ tree
  - ✓ Fundamentally **decouple** crash recovery latency from WA/performance