B+ Tree and Computational Storage Drives: A Perfect Match

Tong Zhang, ScaleFlux
tong.zhang@scaleflux.com

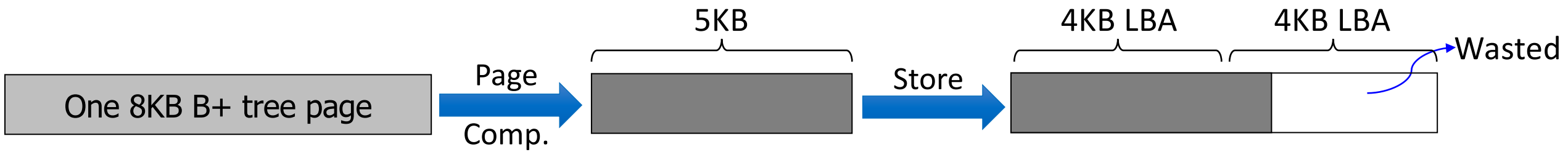
B+ Tree

- Powers (almost) all the relational database systems in the world

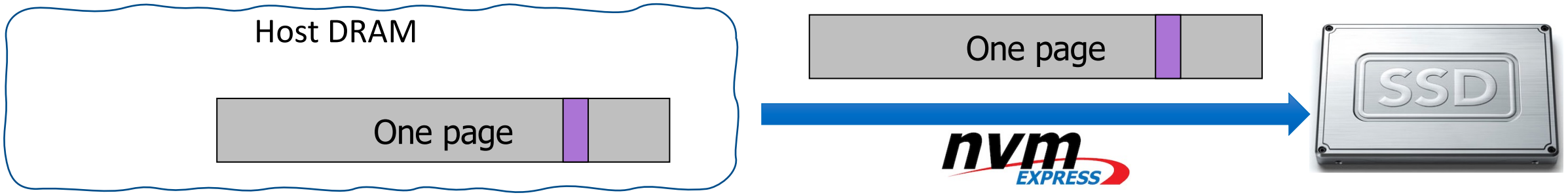
- Competition from log-structured merge tree (LSM-tree)

Lower storage cost **Lower write amplification**

B+ Tree: Storage Cost & Write Amplification

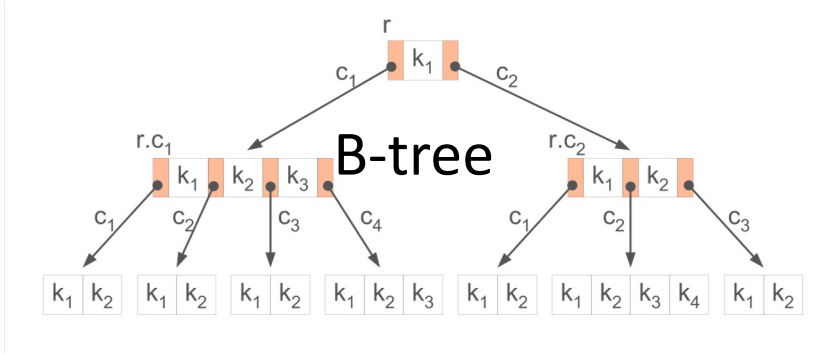


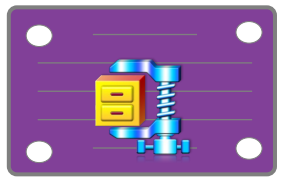
B+ tree stores pages uncompressed despite high data compressibility → **High storage cost**



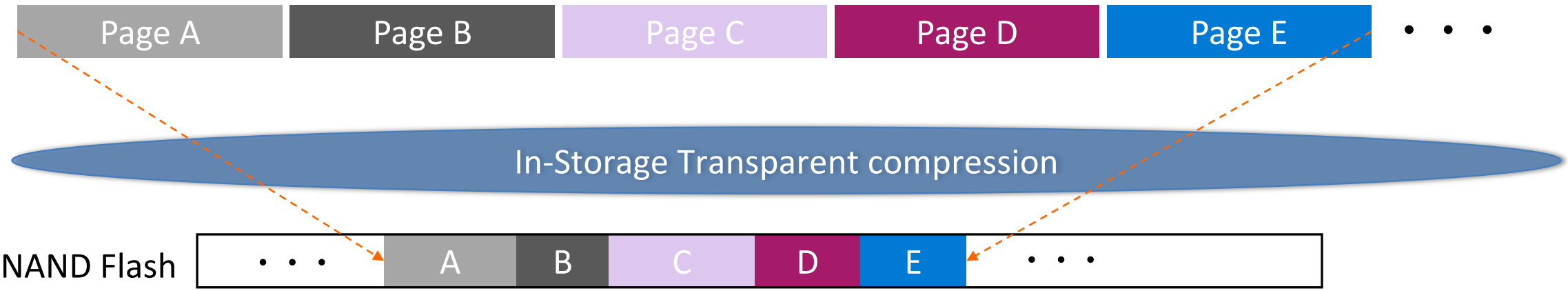
Always flush the entire page to SSD → **High write amplification**

B+ Tree: Storage Cost



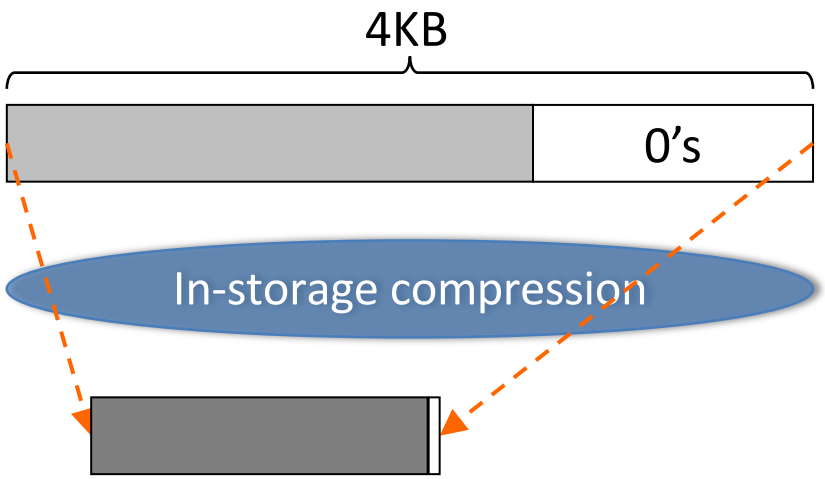


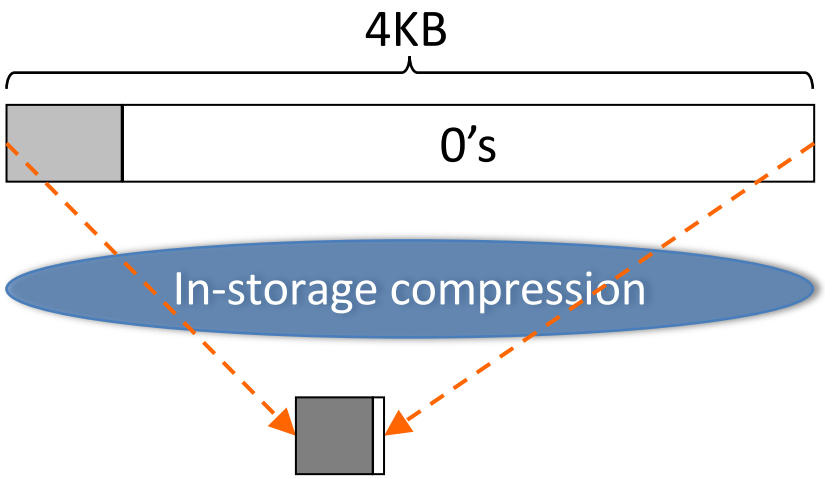
Computational Storage Drive



Close the storage cost gap between B+ tree and LSM-tree

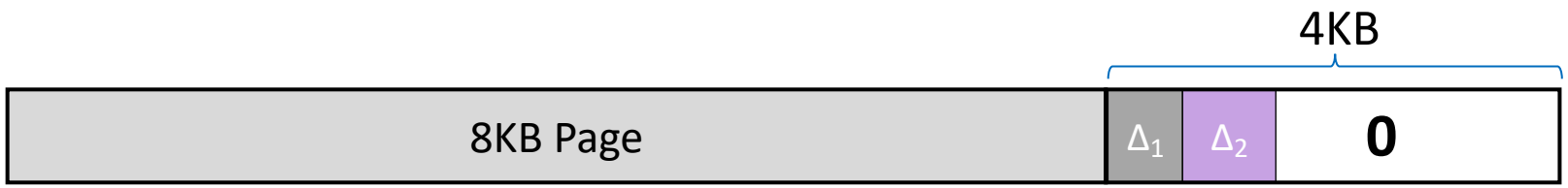
B+ Tree: Write Amplification





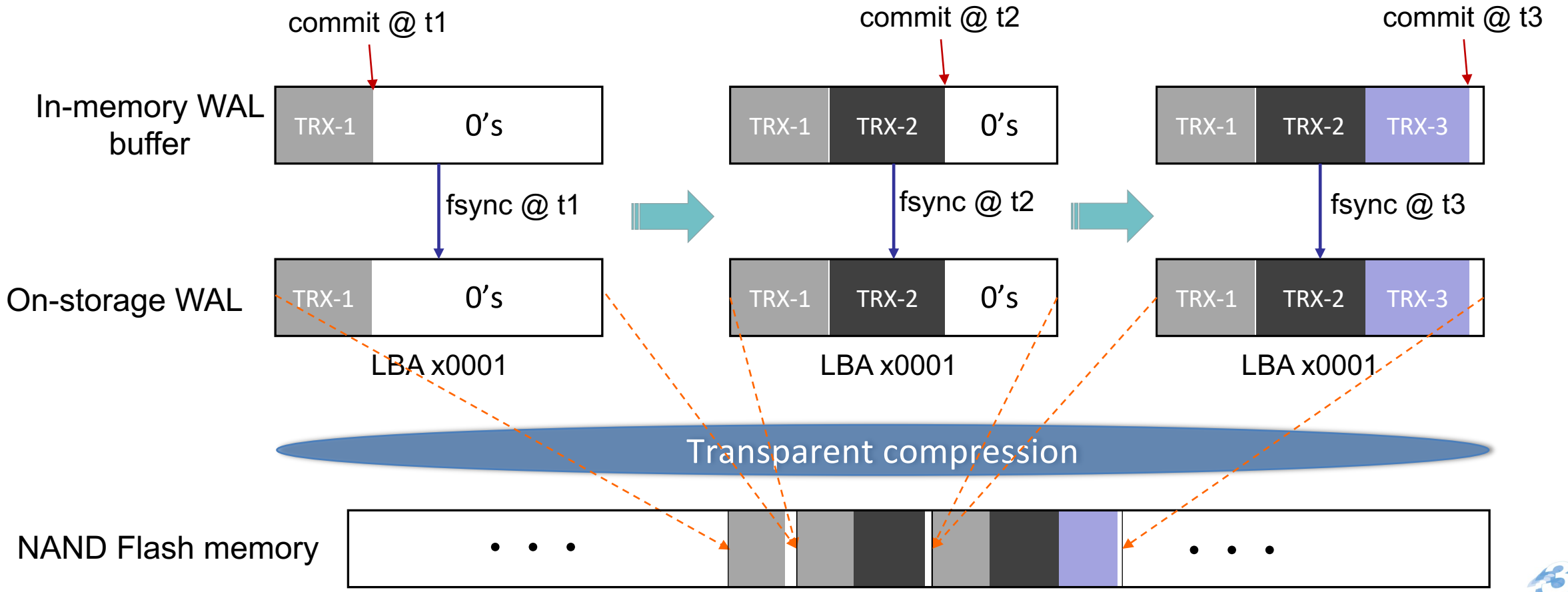
Virtually variable-length block I/O

Per-page modification logging



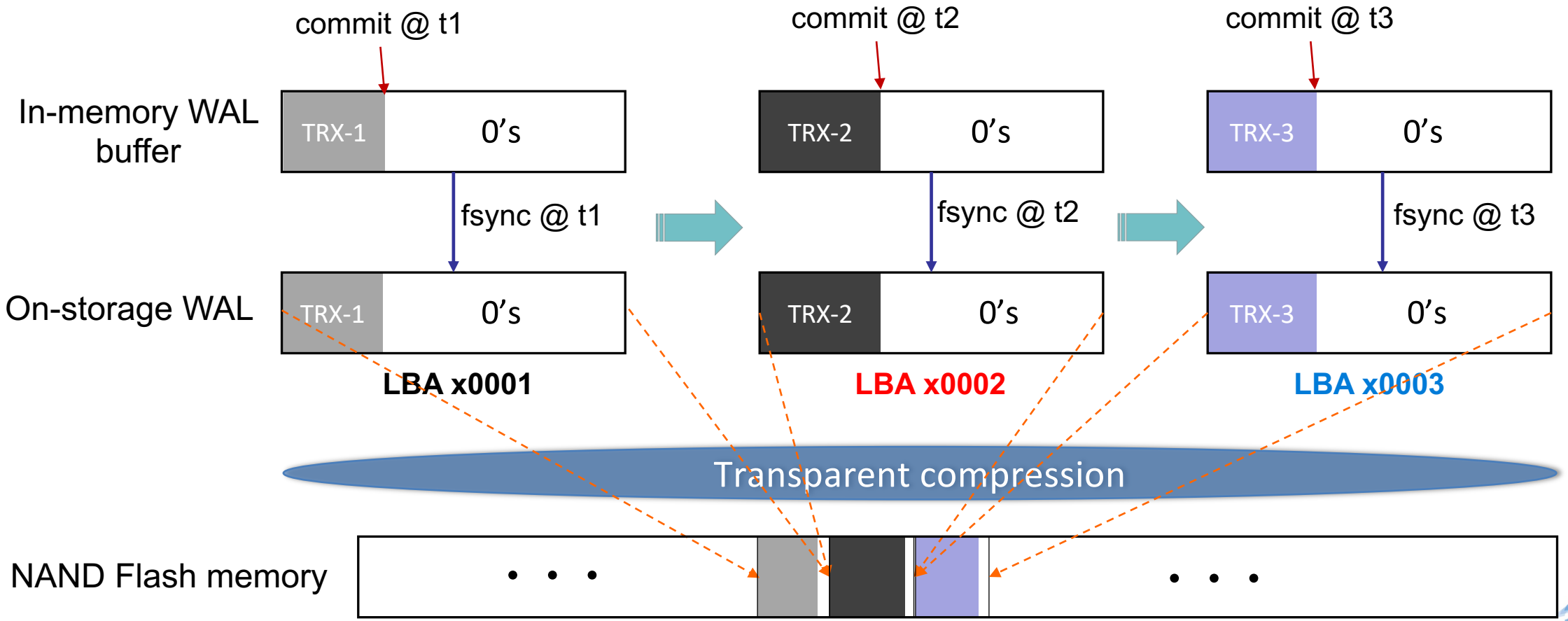
B+ Tree: Write Amplification

- Write-ahead logging (WAL)
  - Universally used by data management systems to achieve *atomicity* and *durability*



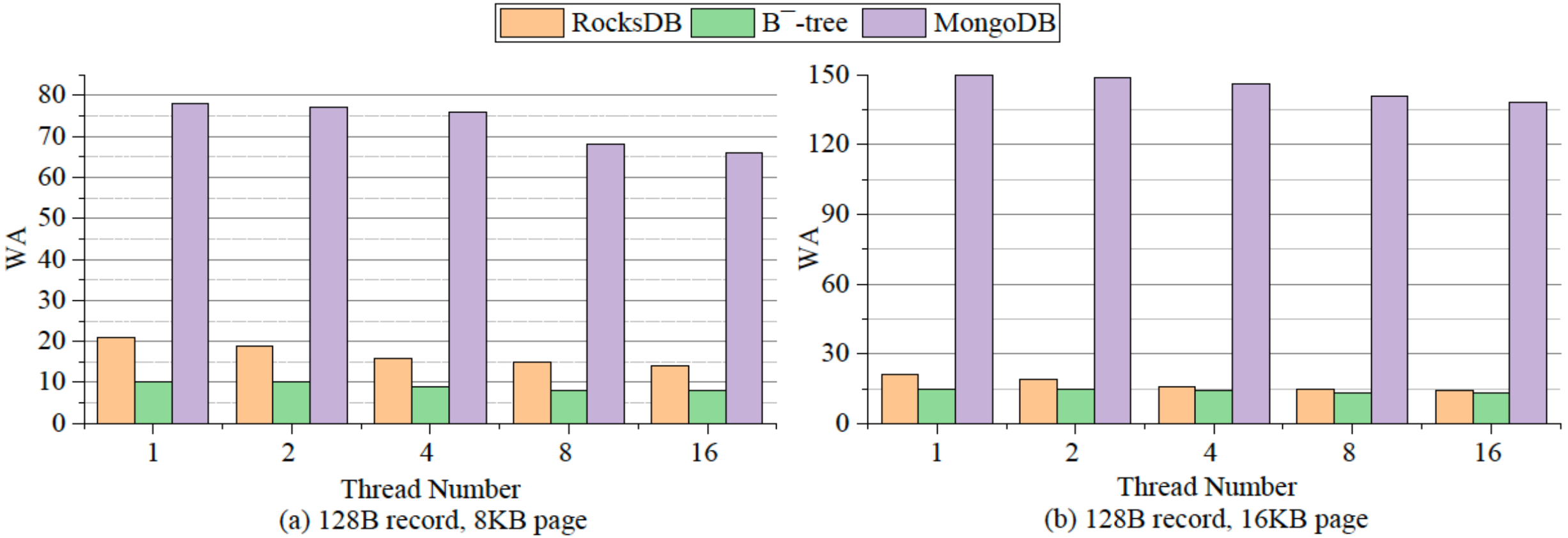
B+ Tree: Write Amplification

- Sparse WAL: Allocate a new 4KB sector per transaction commit
  - ✓ Waste logical storage space ➔ reduce WAL-induced write amplification

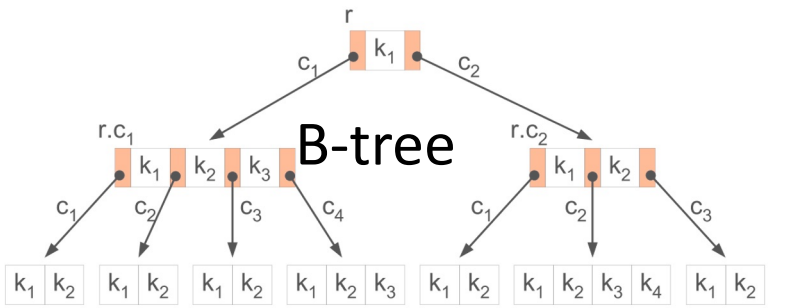


B+ Tree: Storage Cost & Write Amplification

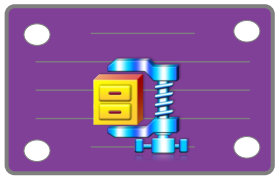
150GB dataset & 1GB cache



Conclusion



B-tree



Computational Storage Drive

Lower storage cost

Lower write amplification

Make B+ tree the **perfect** indexing data structure!

