DENKV: Addressing Design Trade-offs of Key-value Stores for Scientific Applications

Safdar Jamil, Awais Khan, Kihyun Kim, Jae-Kook Lee, Dosik An, Taeyoung Hong, Sarp Oral, Youngjae Kim

7th International Parallel Data Systems Workshop (PDSW'22)

Sogang University
DISCOS Laboratory
KISTI www.kisti.re.kr Korea Institute of Science and Technology Information
Oak Ridge National Laboratory

SC22 Dallas, TX | hpc accelerates.

DENKV: Deduplication-extended Node-local LSM-tree-based Key-value Store

- HPC applications generate huge amount of redundant data
- Distributed key-value stores gained attention for HPC systems
- A node-local LSM-tree-based key-value store for HPC systems
- Integrate data deduplication to overcome write and space amplification problems
- Introduced asynchronous partly inline deduplication (APID)
  - Leverages background thread pool
- Maintained performance while reducing **4x** write and **8x** space amplification

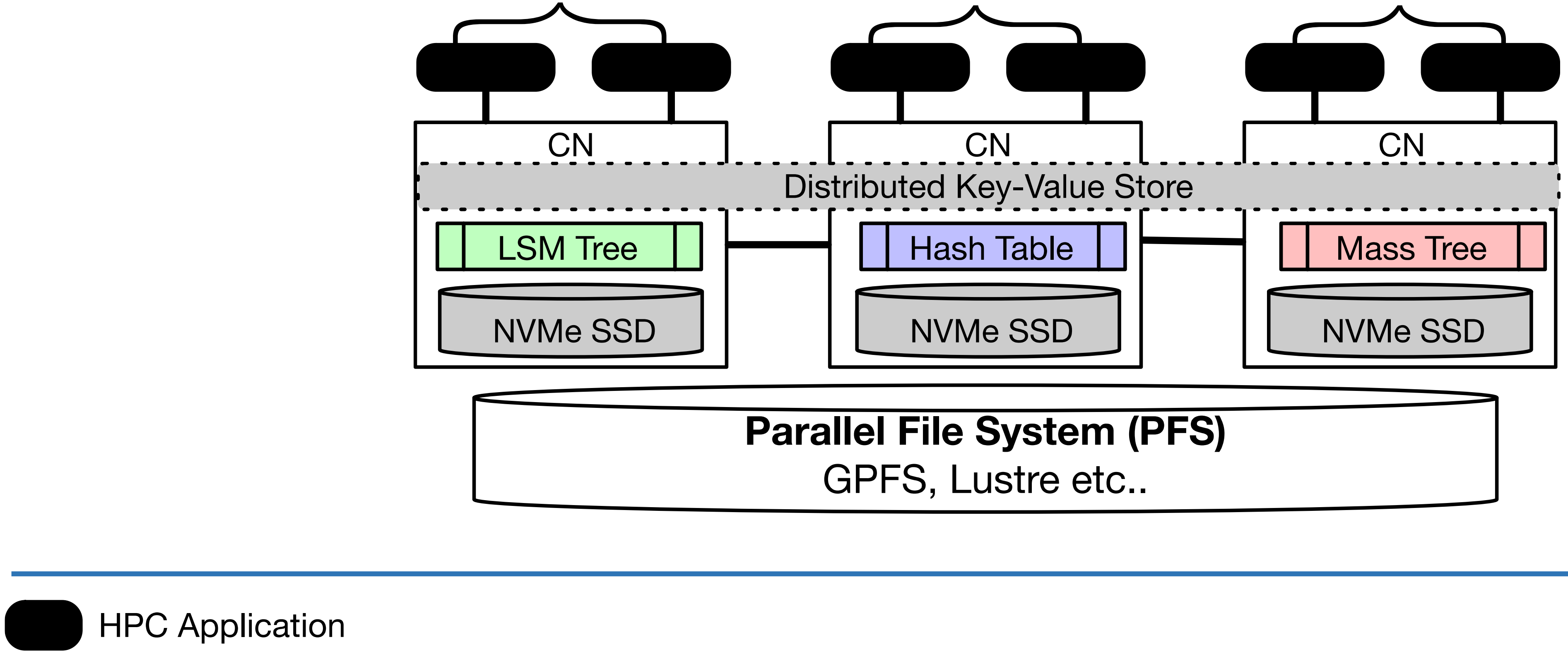
Outline

- ❑ Background
  - ❑ Distributed Key-Value stores in HPC
  - ❑ Log-Structured Merge (LSM) Tree-based KV stores
  - ❑ Deduplication 101
- ❑ Deduplication in HPC
- ❑ Proposed Architecture
  - ❑ DENKV: Design goals
  - ❑ Write and Read operation flow
- ❑ Evaluation
- ❑ Conclusion and Q&A

Background

Distributed Key-Value Stores in HPC

- Emerging storage technologies have opened new opportunities for the use of KV stores in HPC
  - The use-case includes storing intermediate results

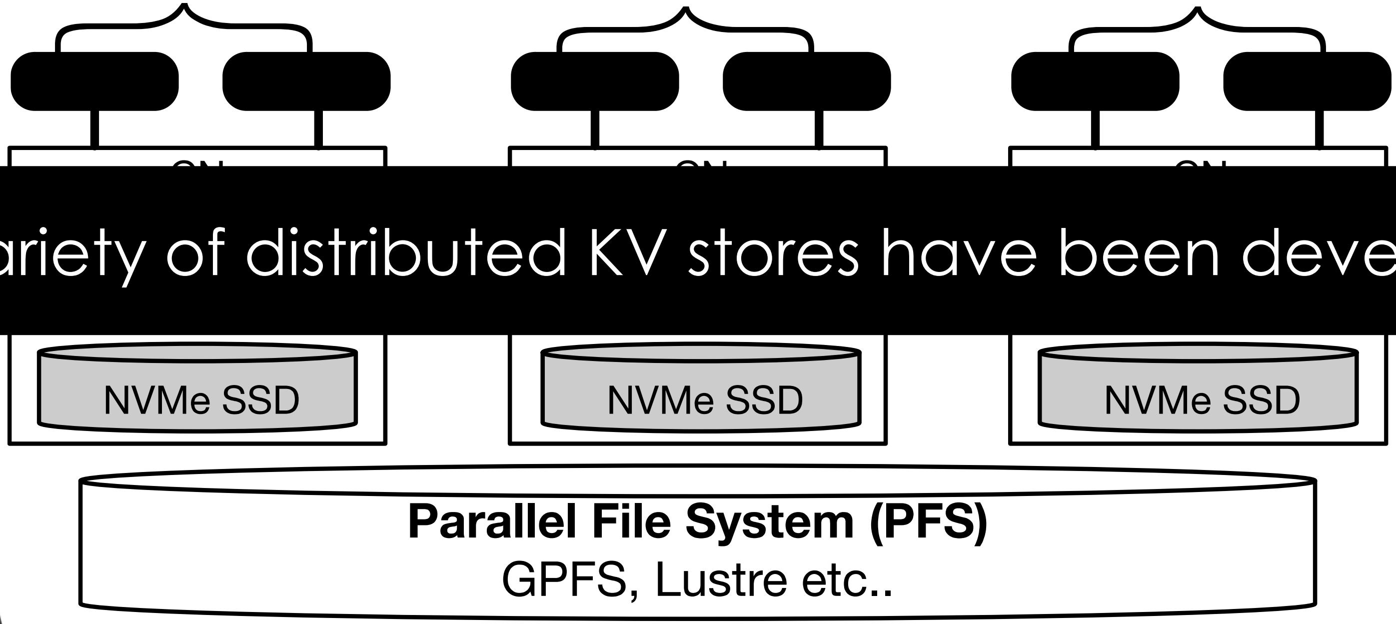


Distributed Key-Value Stores in HPC

- ❑ Emerging storage technologies have opened new opportunities for the use of KV stores in HPC
  - ❑ The use-case includes storing intermediate results

twitter/twemproxy

A fast, light-weight proxy for memcached and redis



❑ A variety of distributed KV stores have been developed.

HPC Application

HPC applications

- ❑ Compute and data intensive → Solve complex problems
- ❑ Execution time in weeks → Simulate world-class scenarios
- ❑ Generate huge amount of data
  - ❑ In terms of terabytes to petabytes
  - ❑ 4 petabytes of data generated for single image
- ❑ High IO bandwidth demand

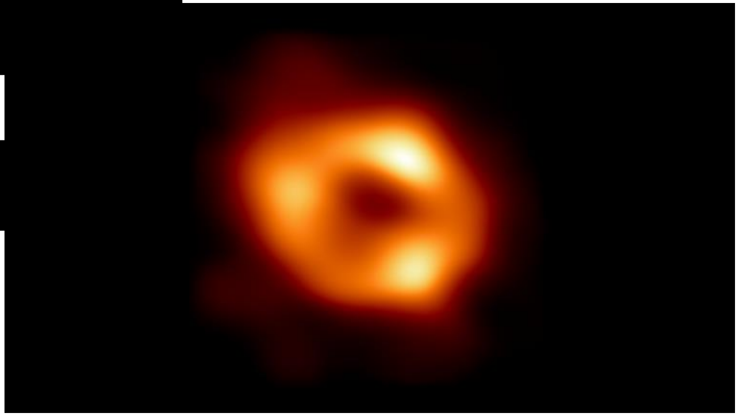
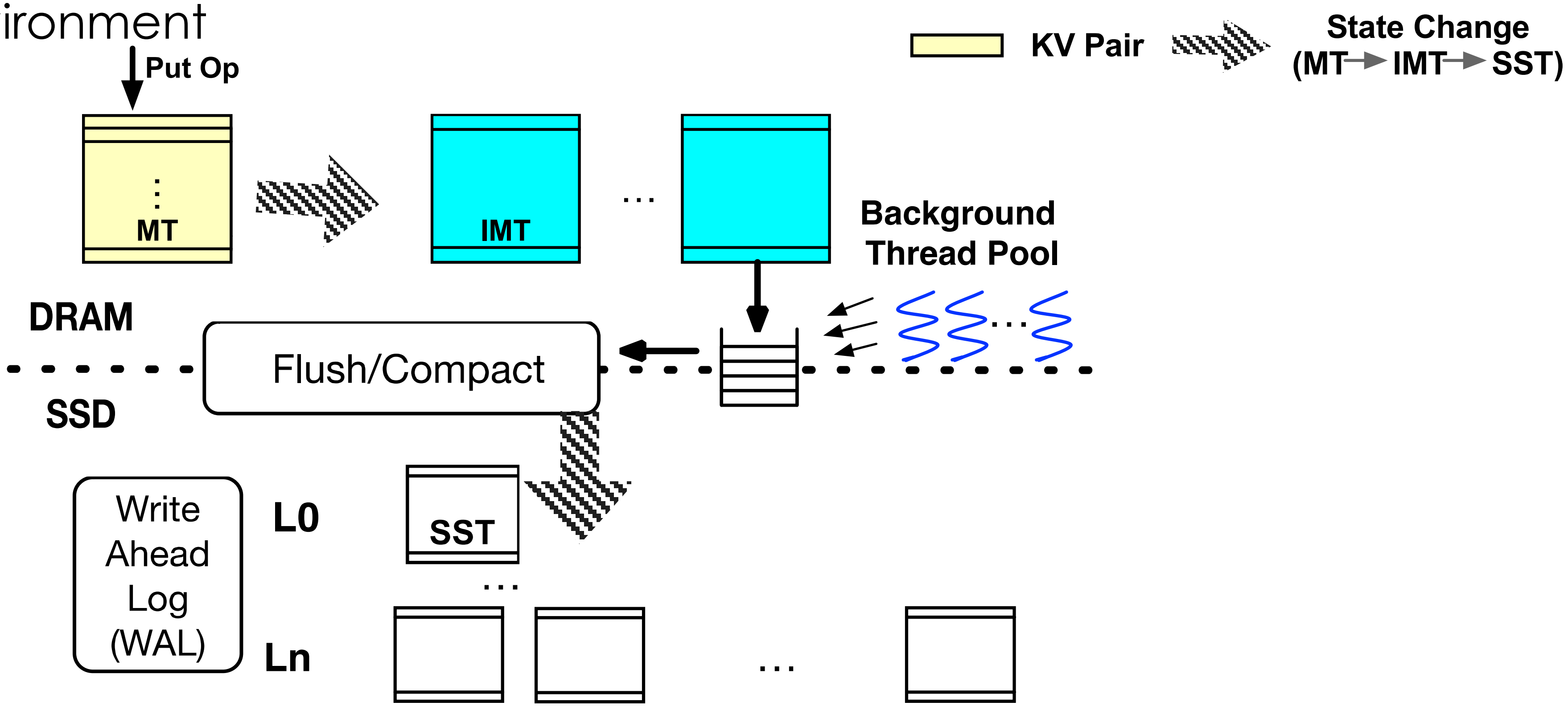


Photo credit: https://eventhorizontelescope.org/blog/astronomers-reveal-first-image-black-hole-heart-our-galaxy

Log-Structured Merge Tree-based Key-Value Stores

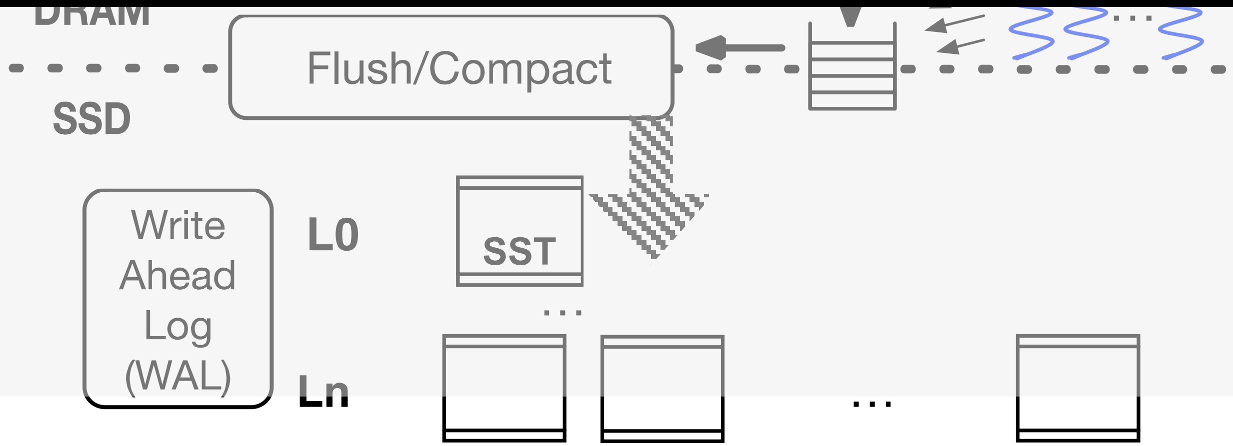
- Log-Structed merge (LSM) tree-based KV stores
  - Highly write-optimized
  - Suitable candidates for node-local NVMe SSDs or burst buffers in HPC environment



Log-Structured Merge Tree-based Key-Value Stores

- Log-Structucted merge (LSM) tree-based KV stores
  - Highly write-optimized
  - Suitable candidates for node-local NVMe SSDs or burst buffers in HPC environment

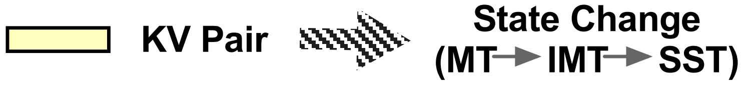
- Limitations of LSM-tree
  - High write amplification (WA) – more writes than application intended
  - High space amplification (SA) – more space utilization than application required

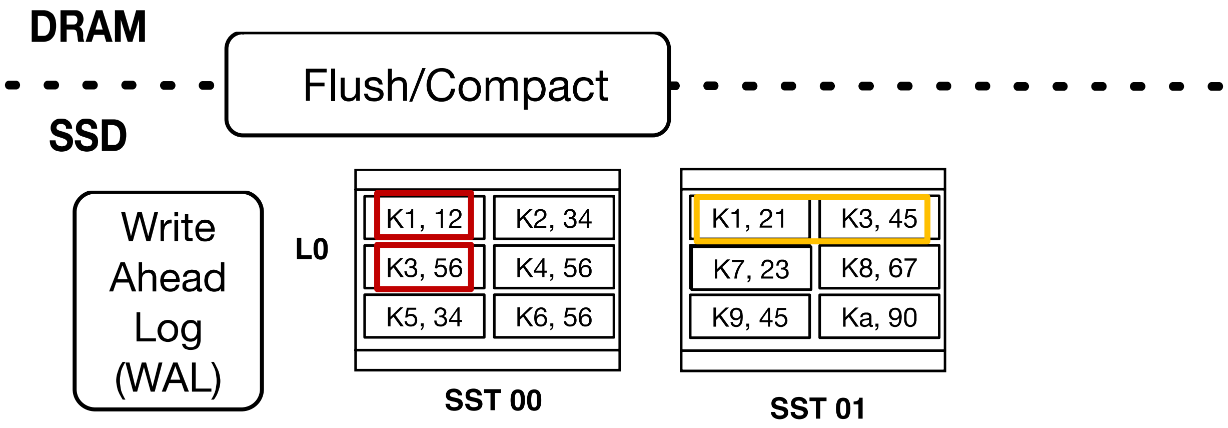


서강대학교 SOGANG UNIVERSITY

Log-Structured Merge Tree-based Key-Value Stores

- Log-Structed merge (LSM) tree-based KV stores
  - Write and Space amplification problems

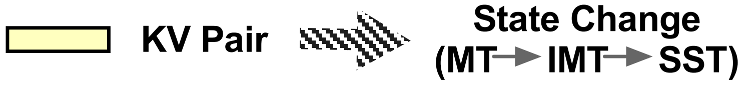




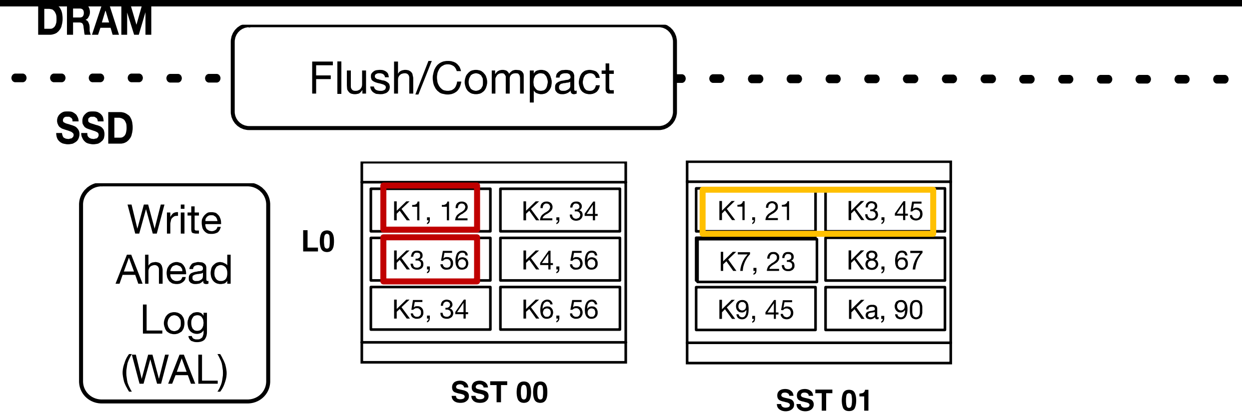
Updated key-value pair   Invalid key-value pair

Log-Structured Merge Tree-based Key-Value Stores

- Log-Structed merge (LSM) tree-based KV stores
  - Write and Space amplification problems



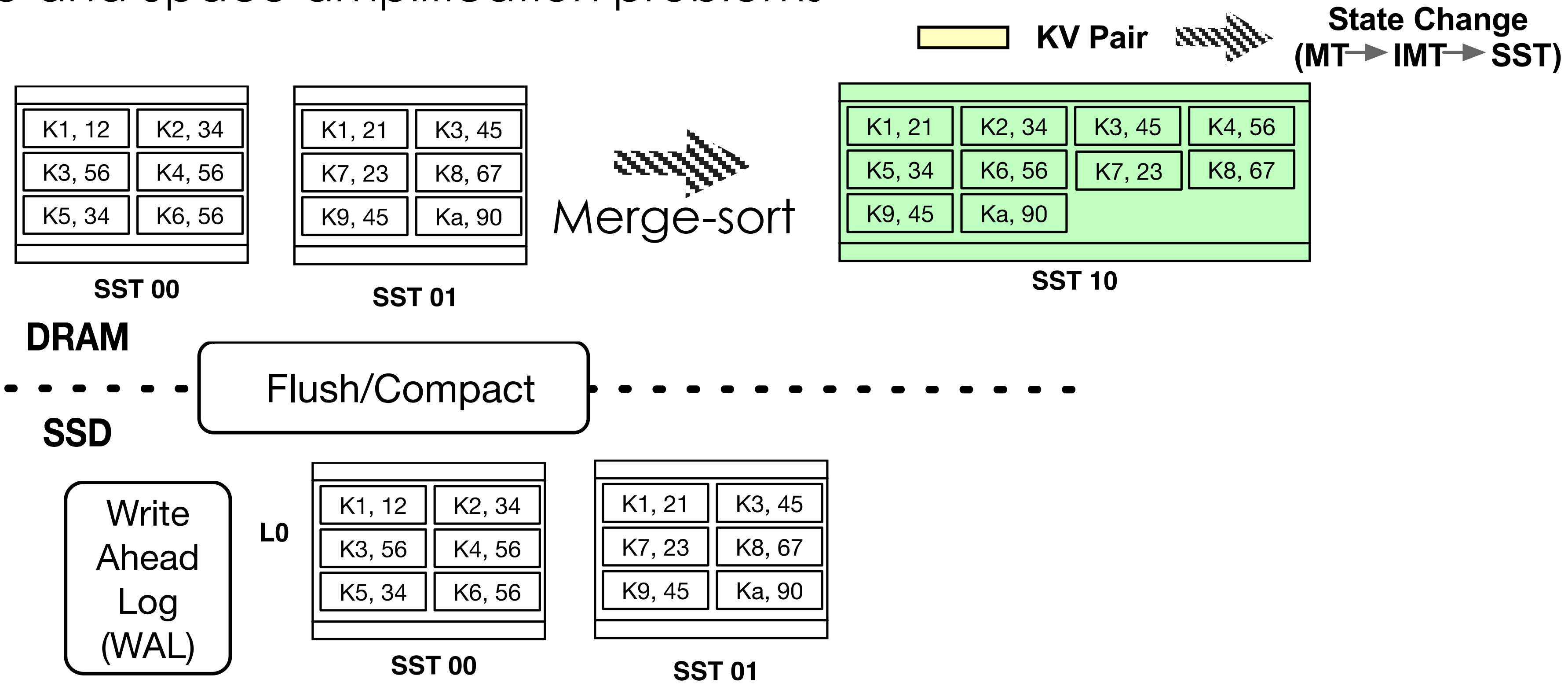
- Unclaimed invalid key-value pairs lead to space amplification



Updated key-value pair   Invalid key-value pair

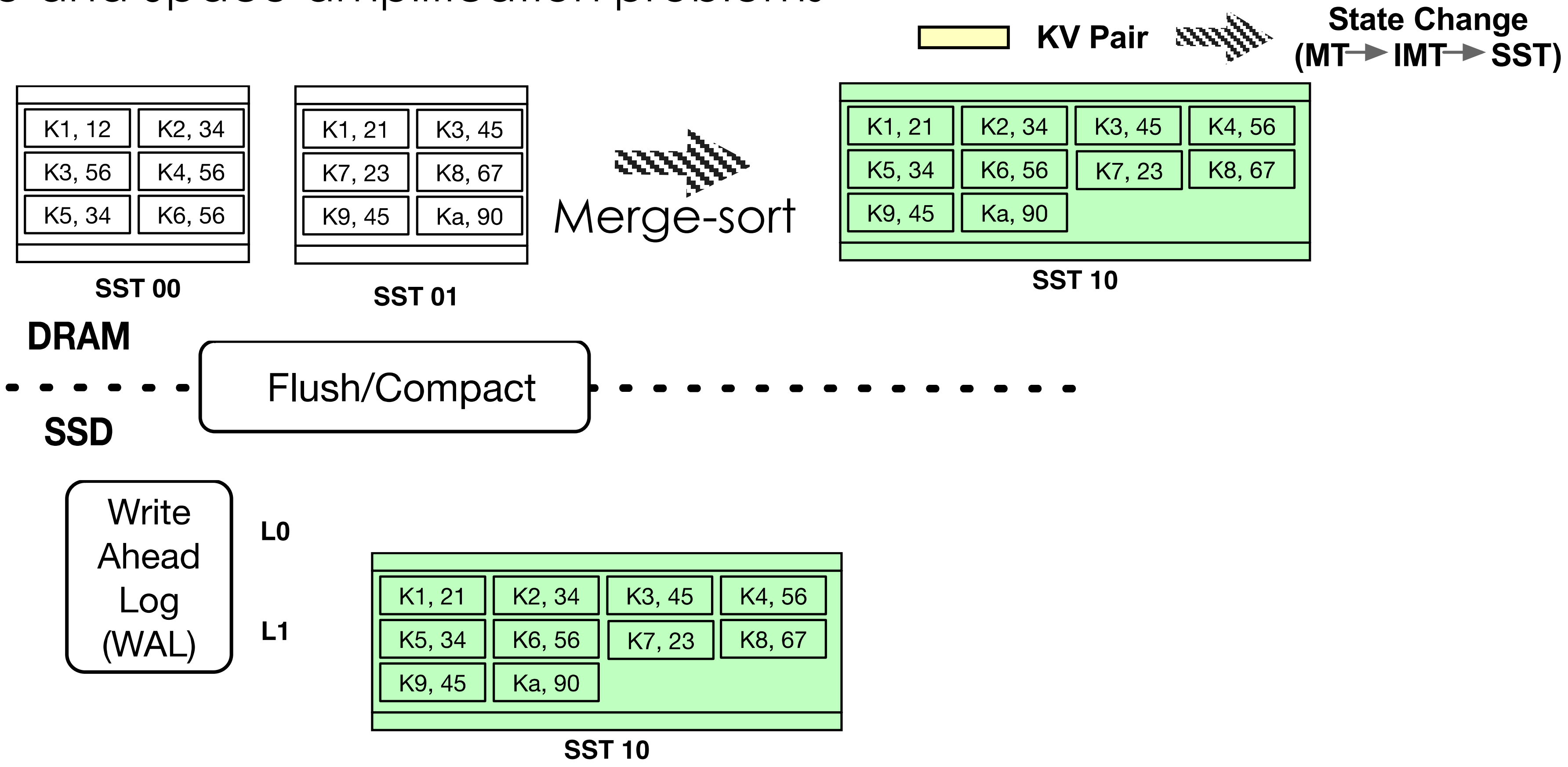
Log-Structured Merge Tree-based Key-Value Stores

- Log-Structed merge (LSM) tree-based KV stores
  - Write and Space amplification problems



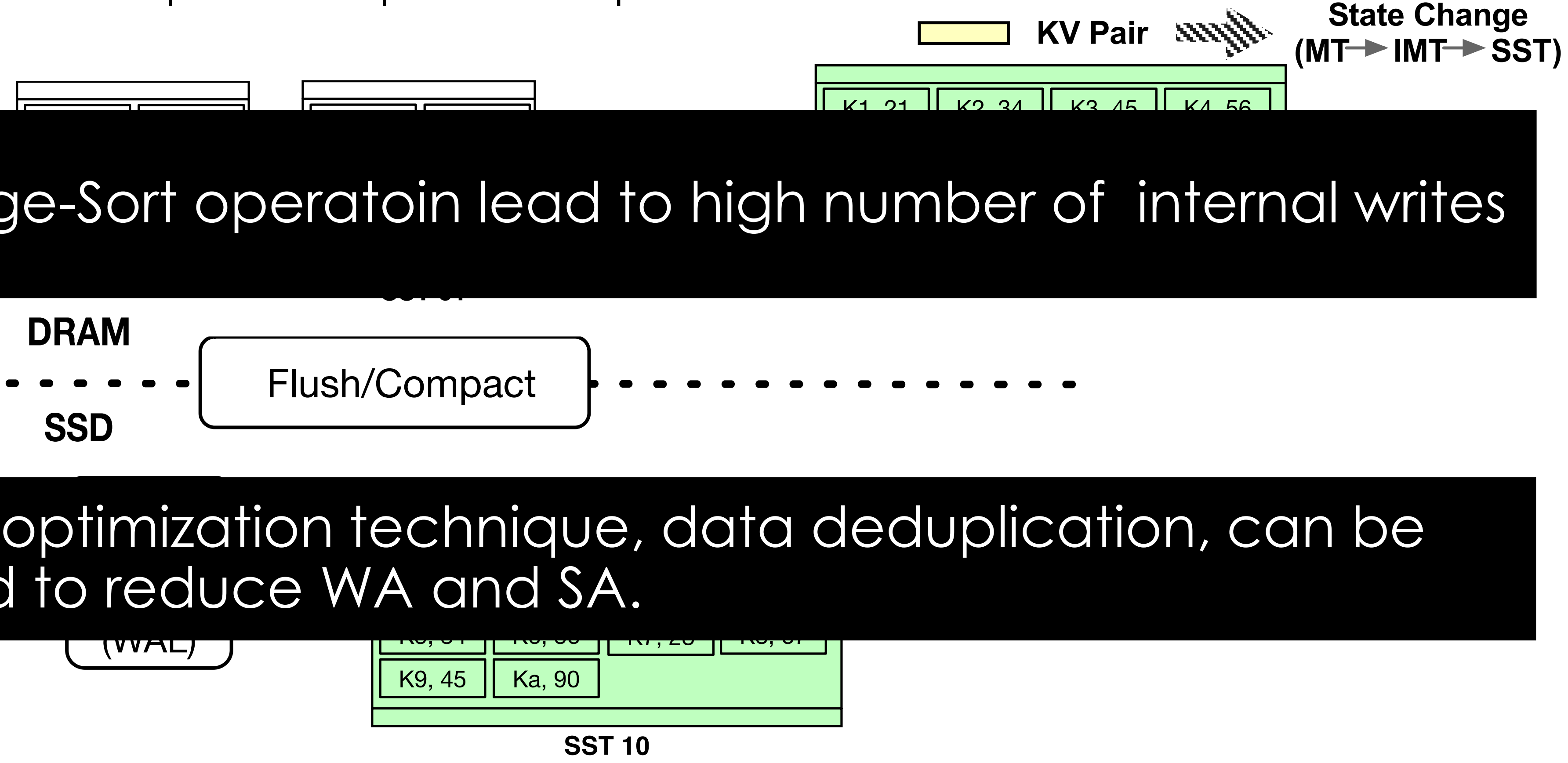
Log-Structured Merge Tree-based Key-Value Stores

- Log-Structed merge (LSM) tree-based KV stores
  - Write and Space amplification problems



Log-Structured Merge Tree-based Key-Value Stores

- ❑ Log-Structed merge (LSM) tree-based KV stores
  - ❑ Write and Space amplification problems

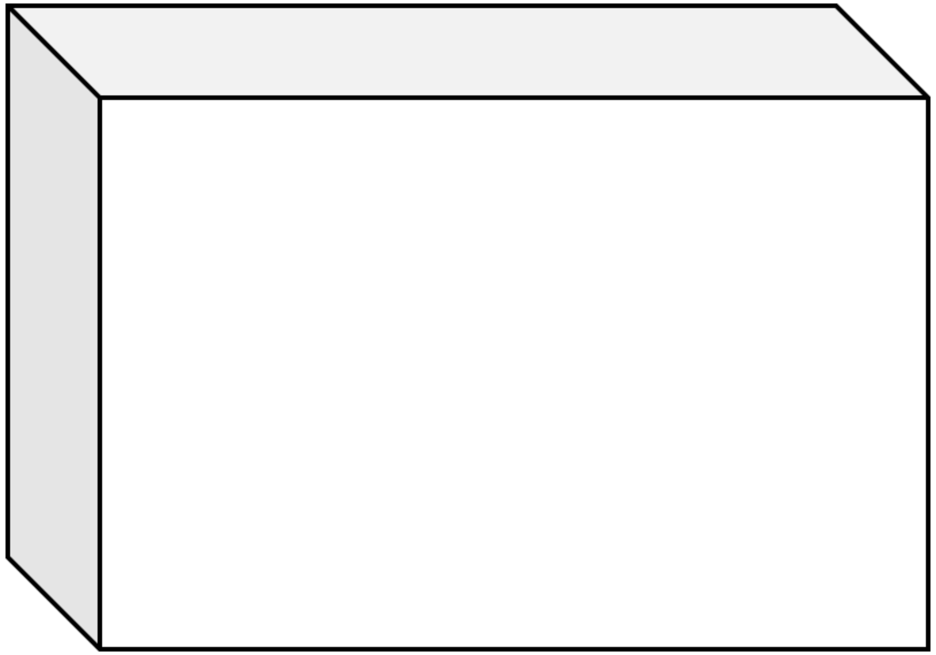


❑ This Merge-Sort operatoin lead to high number of internal writes

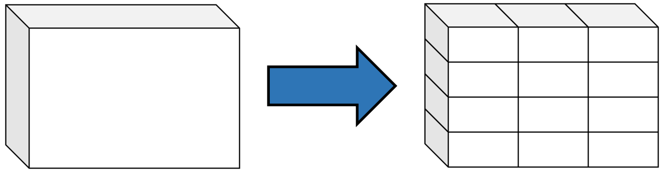
❑ Storage optimization technique, data deduplication, can be adopted to reduce WA and SA.

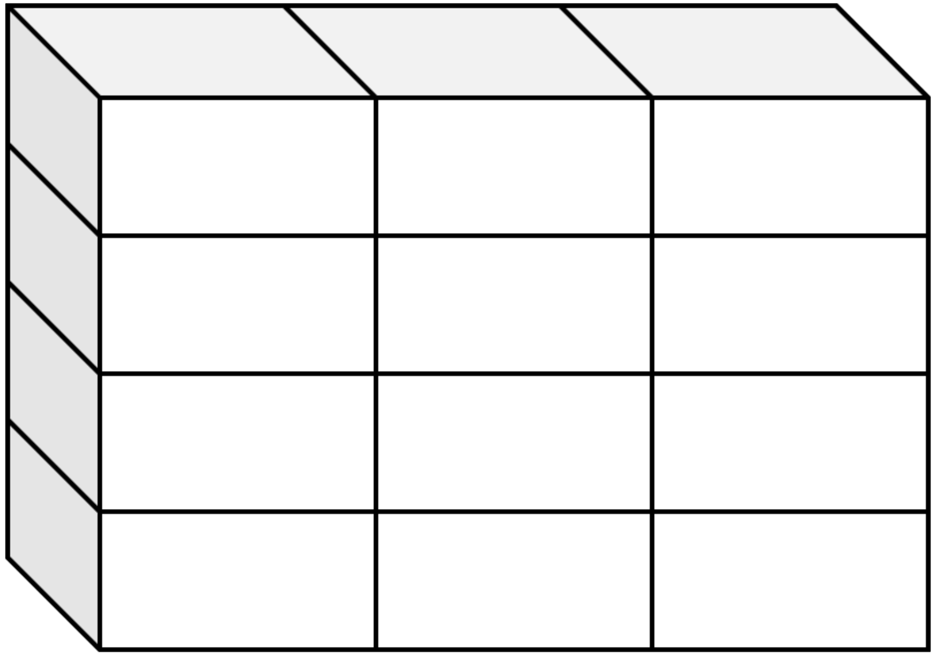
Deduplication 101

0. User Data



Deduplication 101

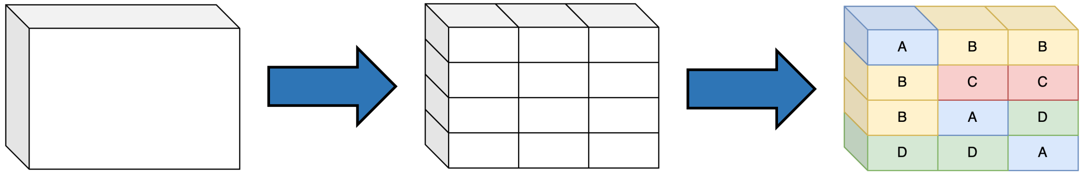


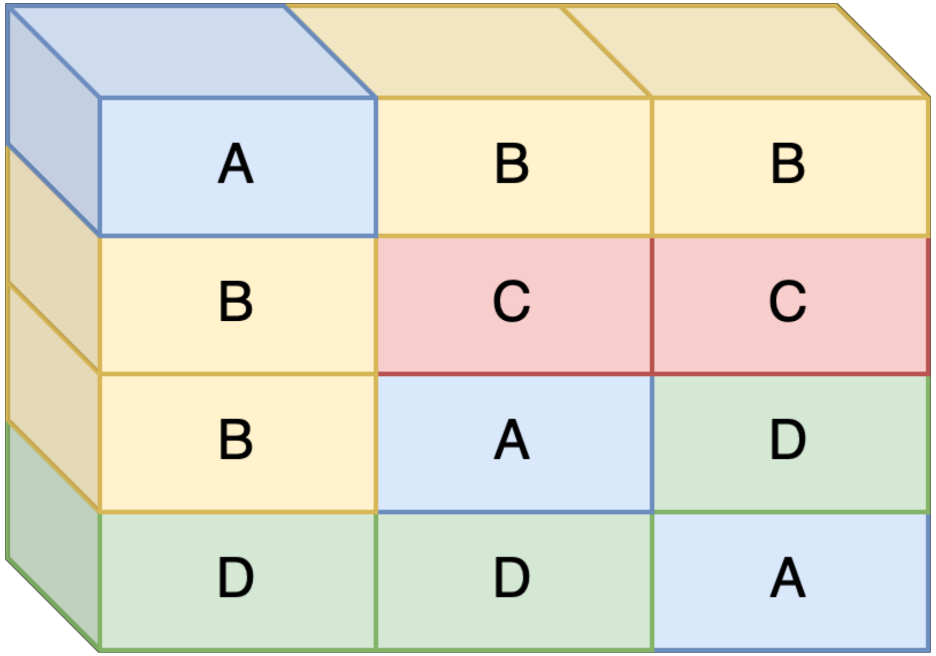


0. User Data

1. Chunking

Deduplication 101





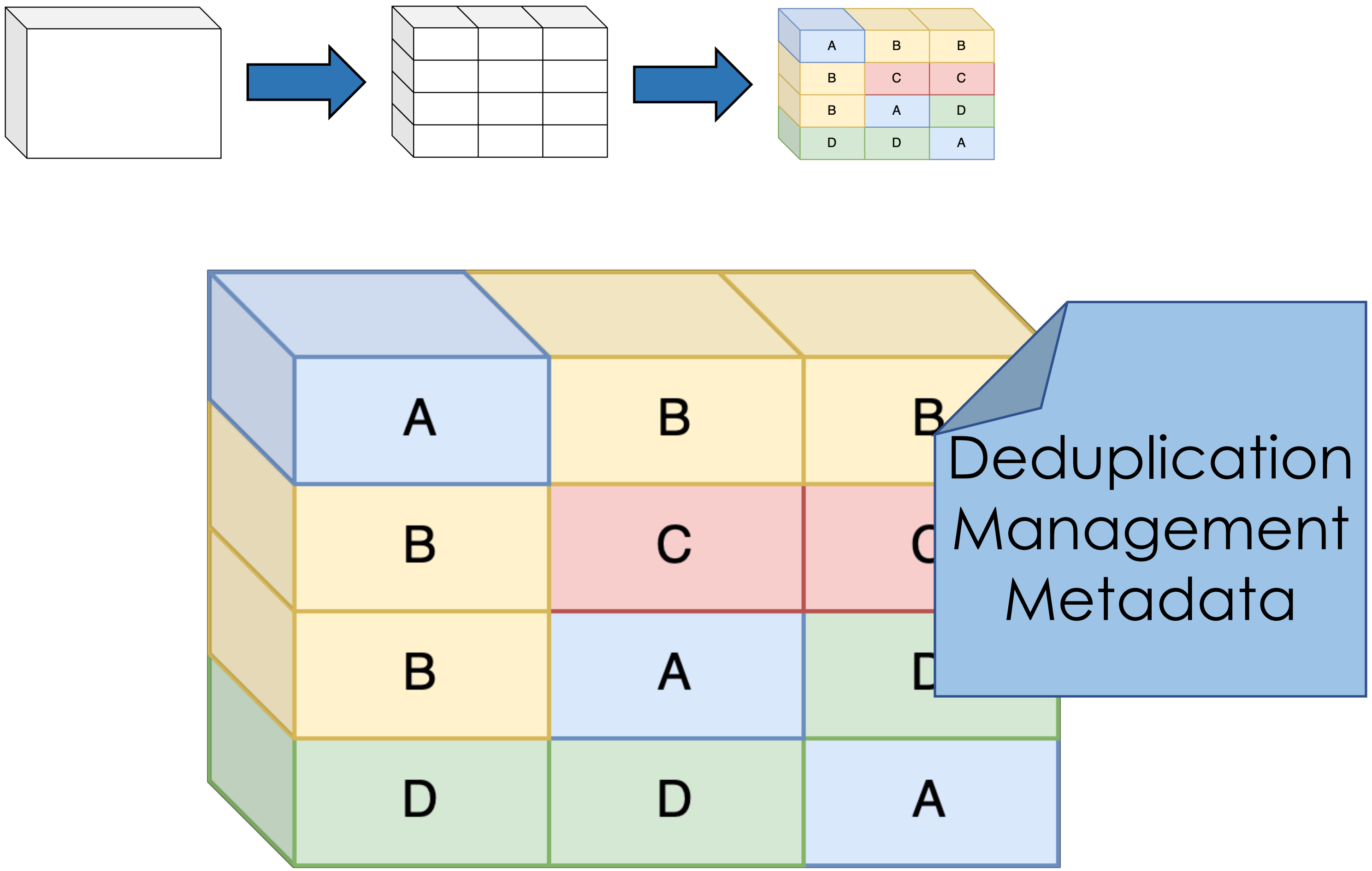
0. User Data

1. Chunking

2. Fingerprinting

Deduplication in HPC

Deduplication 101



0. User Data

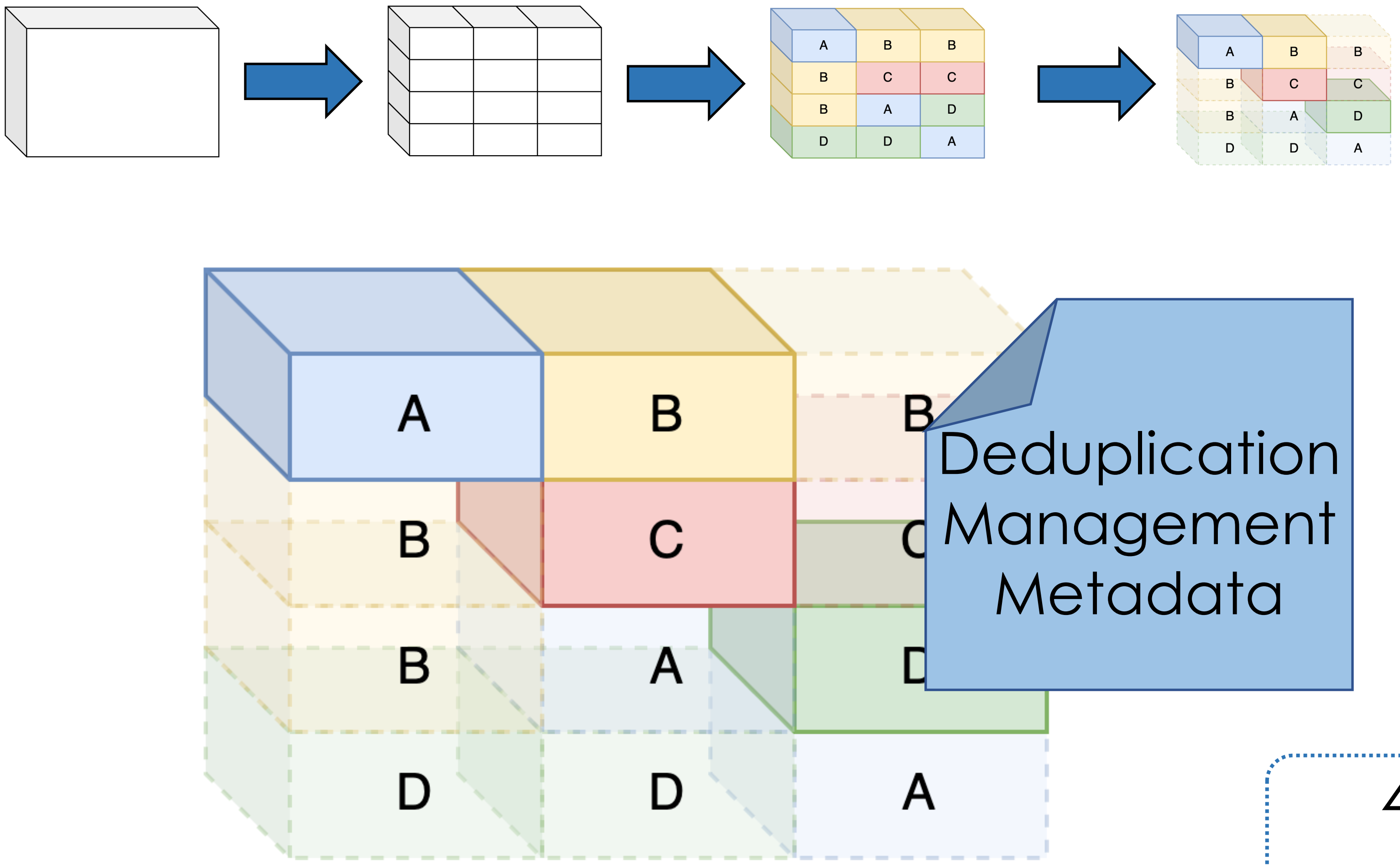
1. Chunking

2. Fingerprinting

3. Duplicate Lookup

Deduplication in HPC

Deduplication 101



0. User Data
1. Chunking
2. Fingerprinting
3. Duplicate Lookup
4. Update Deduplication Metadata

Classification of Deduplication

- Inline Deduplication
  - Performs deduplication during the write process (within critical section)
  - Normally increased write latency
  - Helps improve write endurance problem
  - Immediate improvement of storage

- Offline Deduplication
  - Performs deduplication after the write process finishes (outside of critical section)
  - Lowers write latency compared to inline deduplication
  - Requires temporal storage space to acquire the duplicate data

Deduplication in HPC

Deduplication in HPC applications datasets

- Korean Institute of Science and Technology Information (KISTI) host 5th Supercomputer, Nurion
- A petaflop machine ranked 11th in 2018 by Top500
- Peak performance of 25.3 petaflops
- Cray CS500 with 8,305 compute nodes
- 21 Petabytes of Storage
- Lustre File system

Deduplication in HPC applications datasets

- Collected Top 10 applications dataset at Nurion supercomputer[*]
- Sample of data is collected for only 10 minutes copying
- Implemented in-house deduplication analysis tool
- Analyzed the deduplication ratio
  - Deduplication ratio – amount of data that can be removed

Application	Total Size	Dedup. Ratio	Application	Total Size	Dedup. Ratio
Abacus	386 GB	41.8 %	CESM	273 GB	25.7 %
Charmm	382 GB	23.1 %	Gaussian	293 GB	20.4 %
Lammps	24 GB	42.5 %	MOM	323 GB	53.9 %
MPAS	197 GB	81.7 %	Siesta	566 GB	52.1 %
VASP	1 TB	27.3 %	ANSYS	544 GB	23.8 %

[*]. https://www.ksc.re.kr/eng/resource/nurion

Deduplication in HPC applications datasets

- Collected Top 10 applications dataset at Nurion supercomputer[*]
- Sample of data is collected for only 10 minutes copying
- Implemented in-house deduplication analysis tool
- Analyzed the deduplication ratio

- HPC applications generate highly redundant data [SC'12].

Abacus	386 GB	41.8 %	CESM	273 GB	25.7 %
Charmm	382 GB	23.1 %	Gaussian	293 GB	20.4 %
Lammps	24 GB	42.5 %	MOM	323 GB	53.9 %
MPAS	197 GB	81.7 %	Siesta	566 GB	52.1 %
VASP	1 TB	27.3 %	ANSYS	544 GB	23.8 %

[*]. https://www.ksc.re.kr/eng/resource/nurion

Deduplication in LSM-tree

- Novel way to minimize WA and SA
- Incorporating value-based deduplication
  - Can help reduce the actual size of KV store
- Adopting deduplication at tradition LSM-tree

Performance overhead of inline dedup at MemTable

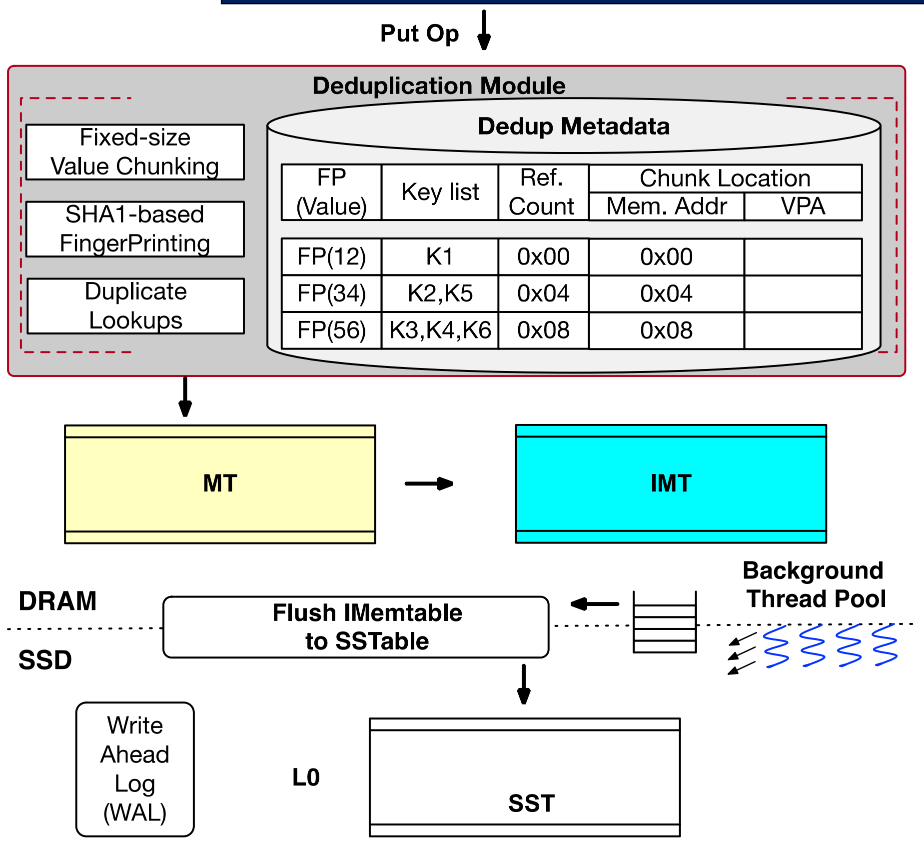
Breaks structural constraints at SSTables
(Single instance of valid KV Pairs)

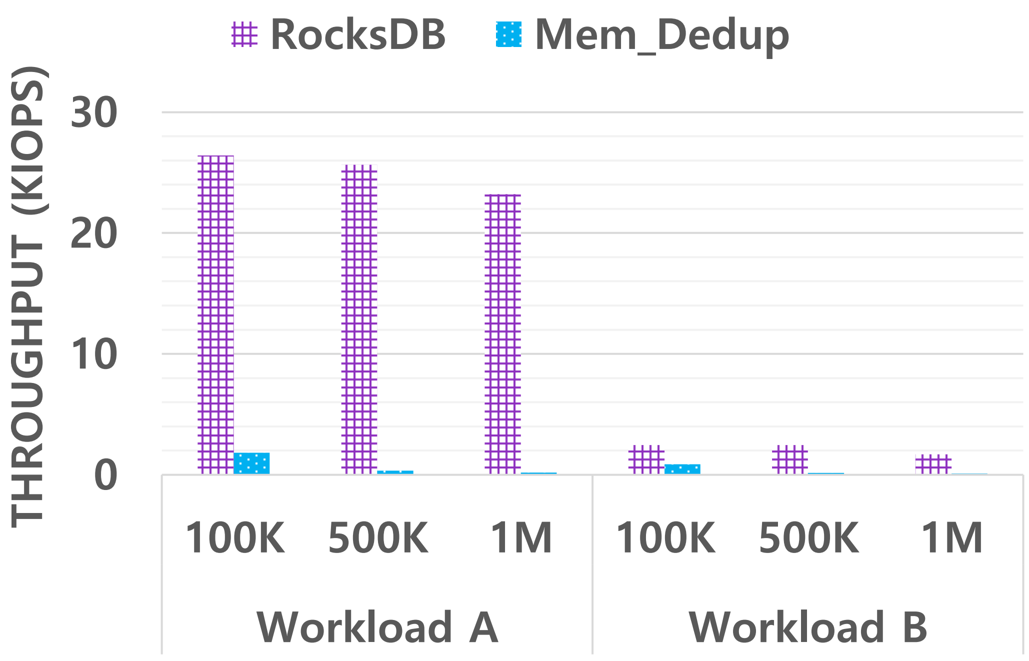
Complex compaction operation

Deduplication in LSM-tree

- ❑ Adopting deduplication at tradition LSM-tree

Performance overhead of inline dedup at MemTable

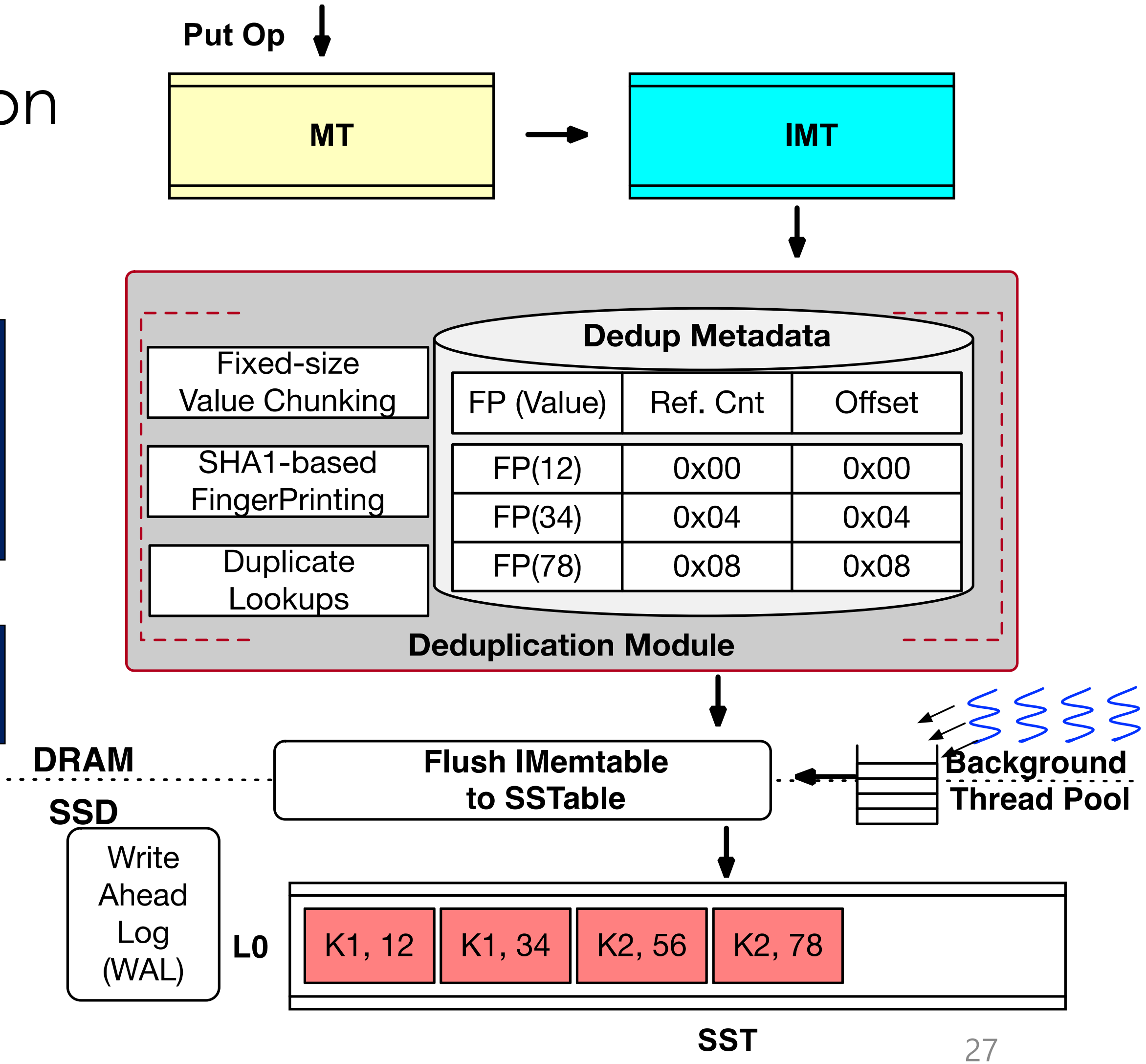




YCSB Benchmark | Workload A: 100% Write | Workload B: 50% write & 50% read

Deduplication in LSM-tree

- ❑ Adopting deduplication at tradition LSM-tree



Breaks structural constraints at SSTables (Single instance of valid KV Pairs)

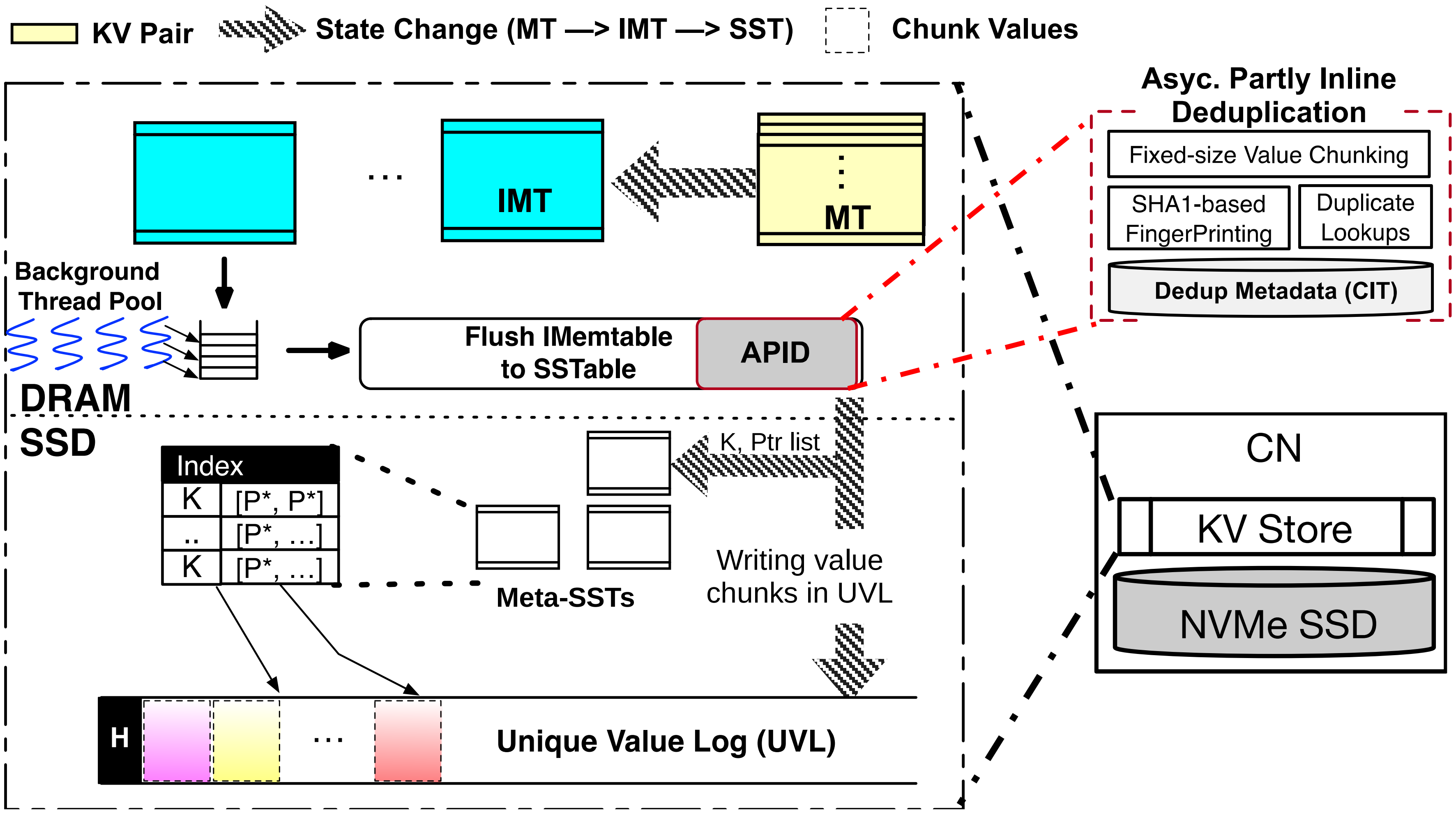
Complex compaction operation

Proposed Architecture

DENKV: Deduplication-extended Node-local LSM-tree-based Key-value Store

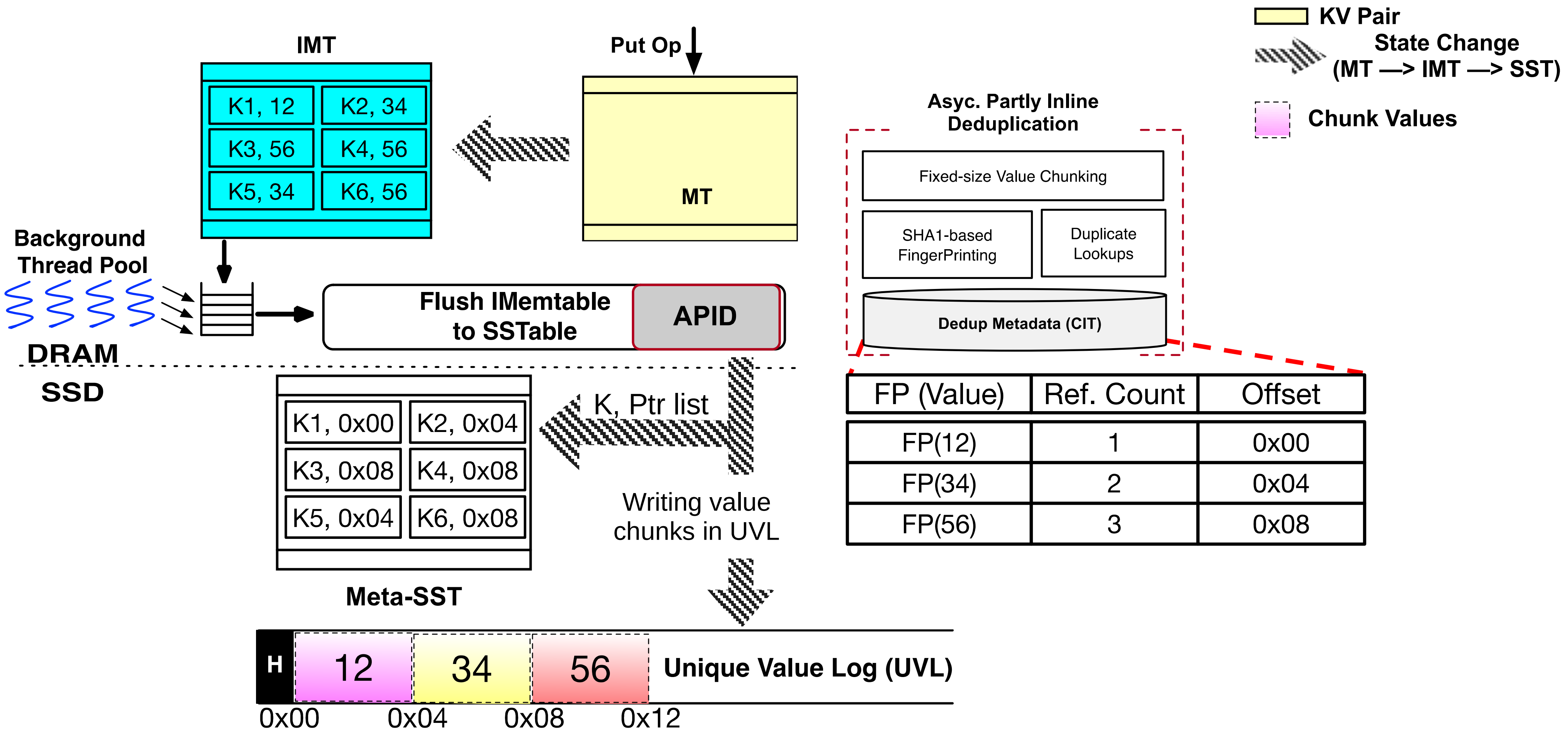
- Design Goals
  - Maintain performance characteristics of LSM-tree
  - Minimum deduplication overhead for client operations
  - Reduce write and space amplification
  - Maintain the structural constraint of LSM-tree

DENKV: Design Overview

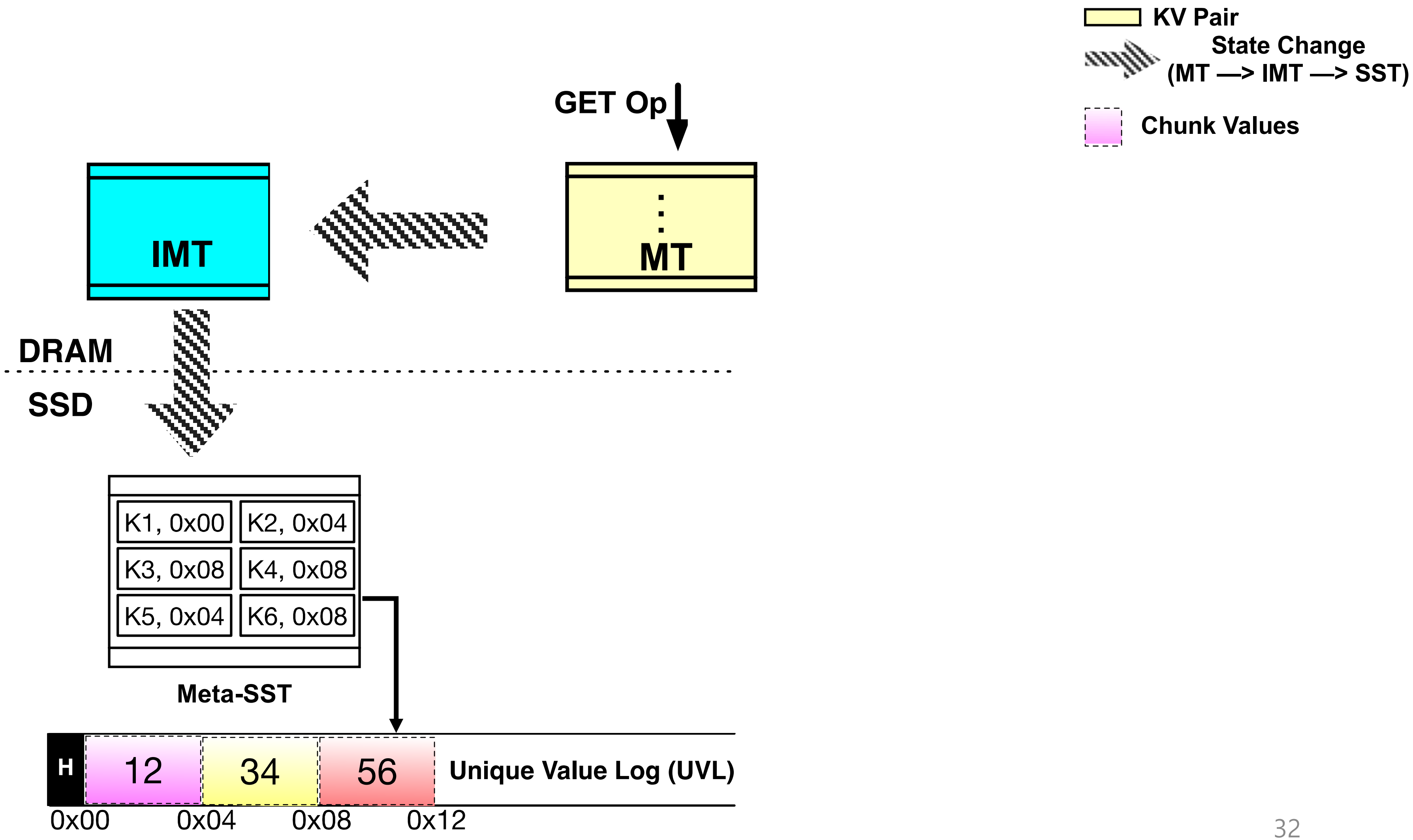


- ➢ Asynchronous
  - Background thread pool
- ➢ Partly inline
  - Out of critical section

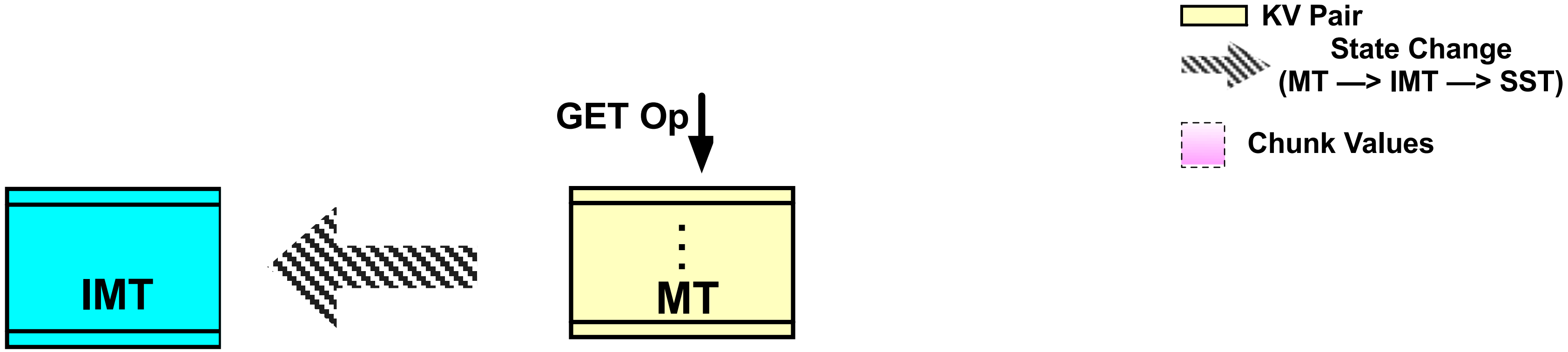
DENKV: Write Operation Flow



DENKV: Read Operation Flow

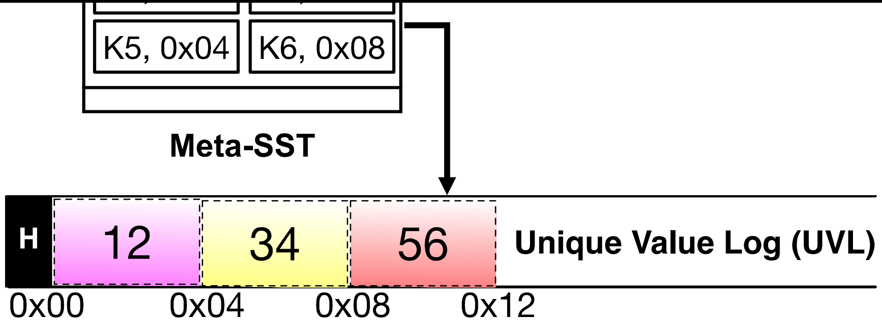


DENKV: Read Operation Flow



Refer Manuscript

- ❑ Garbage Collection
- ❑ Crash Consistency of Chunk Information Table



Evaluation

System configuration

System Setup

CPU	Intel(R) Xeon(R) CPU E5-4640 v2 @ 2.20GHz 4 CPU nodes (10 cores per node)
DRAM	256 GB DDR3 DRAM
Storage	Samsung SSD 970 EVO 1TB

Benchmark

- In-house simulation of dedup patterns of HPC application
- Varying value sizes: 4KB and 1MB
- Fixed size keys 16 bytes
- 1 Million KV pairs for 4KB
- 100 thousand KV pairs for 1MB

Compared systems

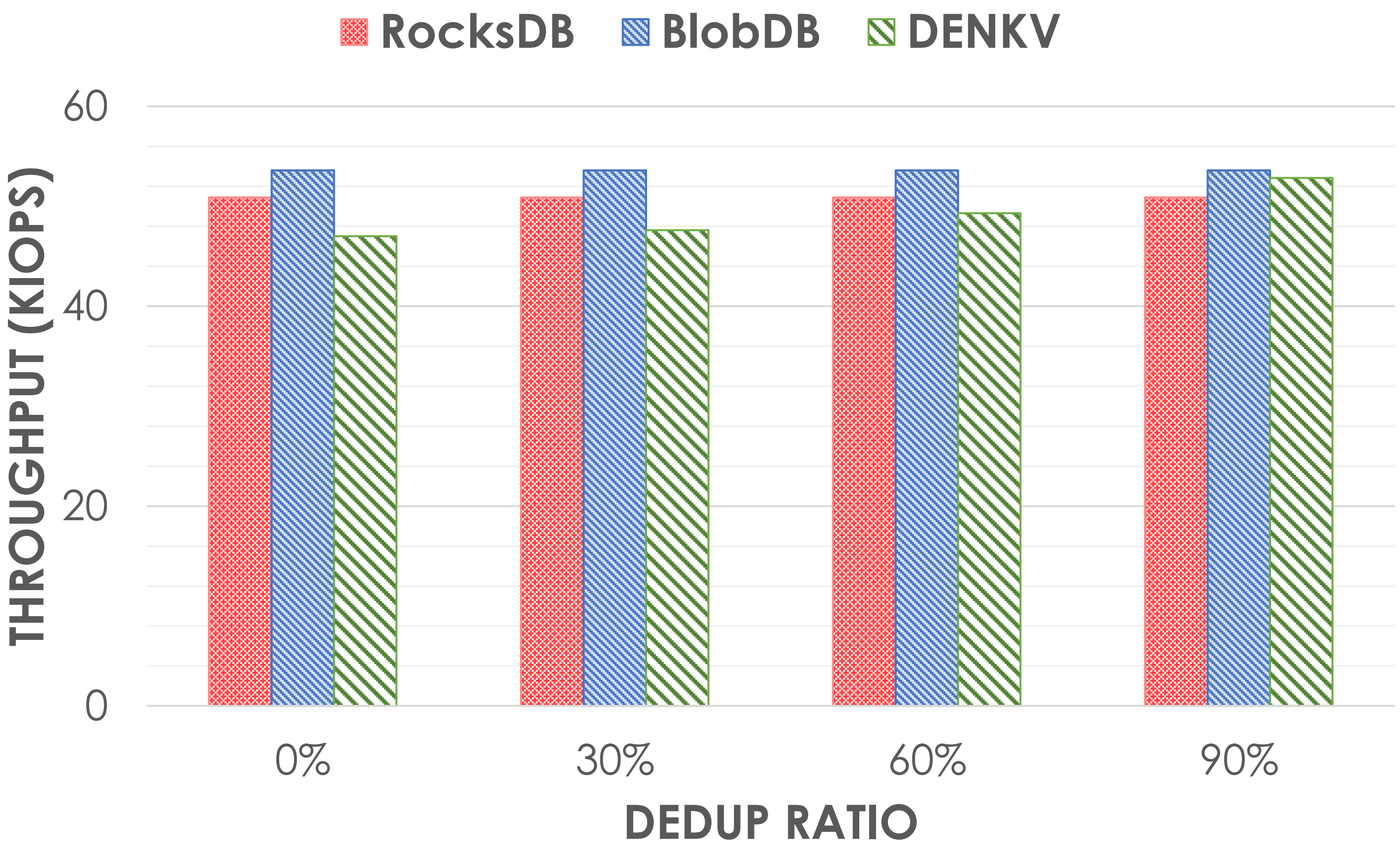
- RocksDB
  - Vanilla LSM-Tree based KV Store
  - Follows the traditional LSM-Tree structure
- BlobDB
  - KV separation design atop of RocksDB
  - Optimized for write and read operations
- DENKV
  - Our proposed deduplication incorporated KV Store

Questions to be answered

- How much deduplication influence the performance in general?
- How much write amplification is reduced?
- How much space amplification is reduced?
- What are the bottlenecks?

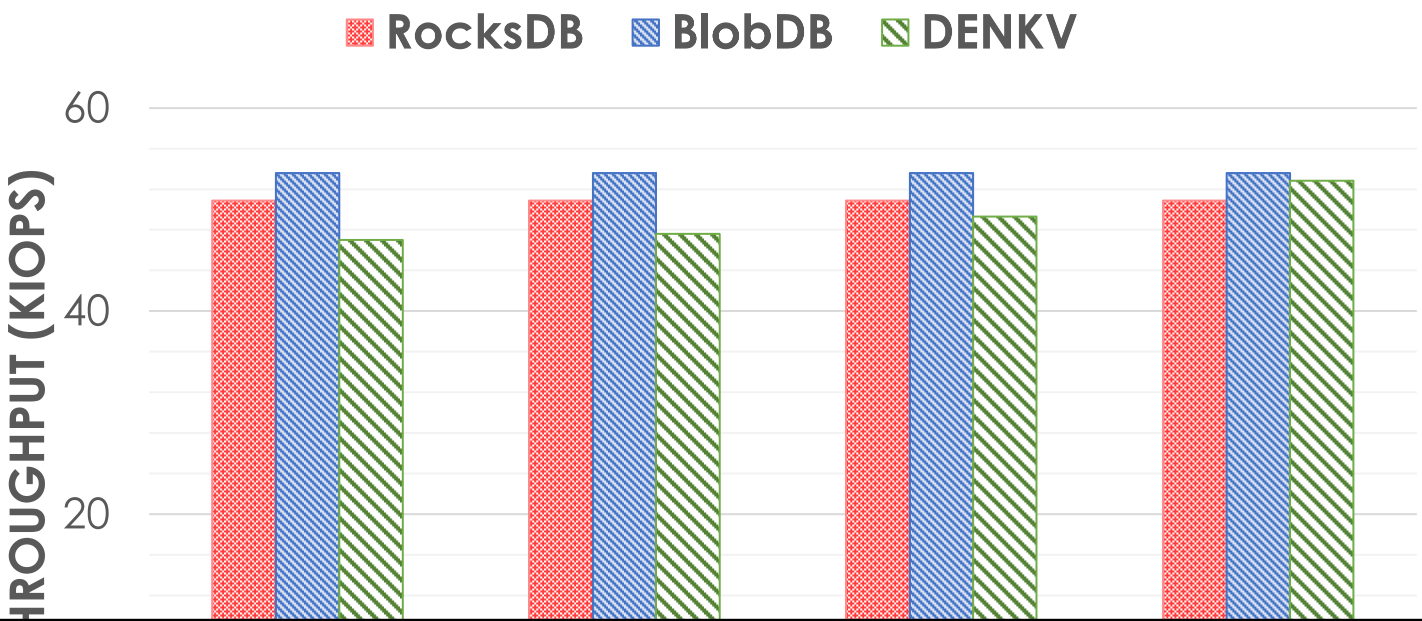
Performance analysis

- 4 KB KV Pairs



Performance analysis

❑ 4 KB KV Pairs

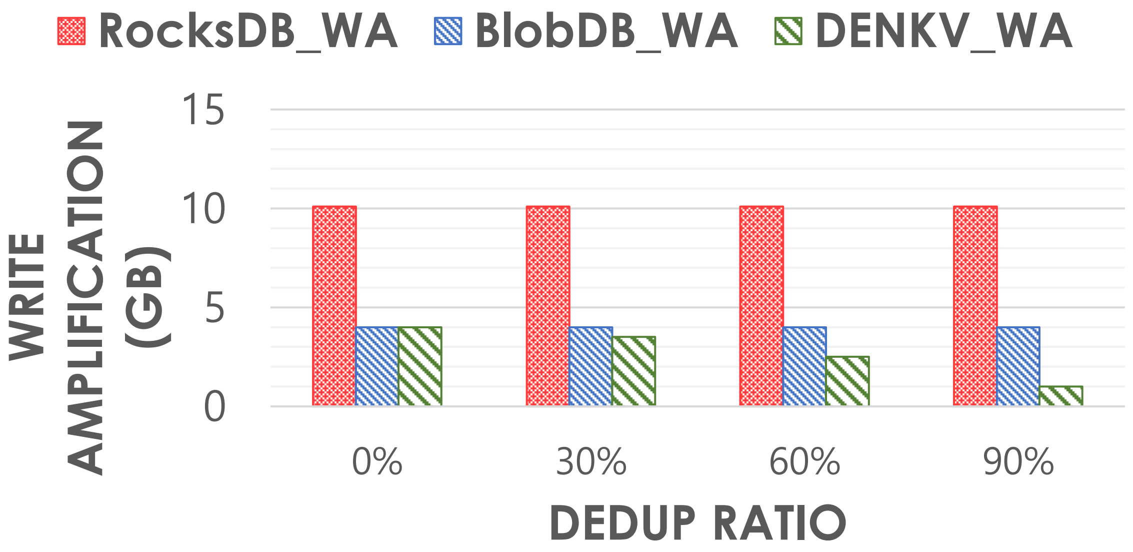


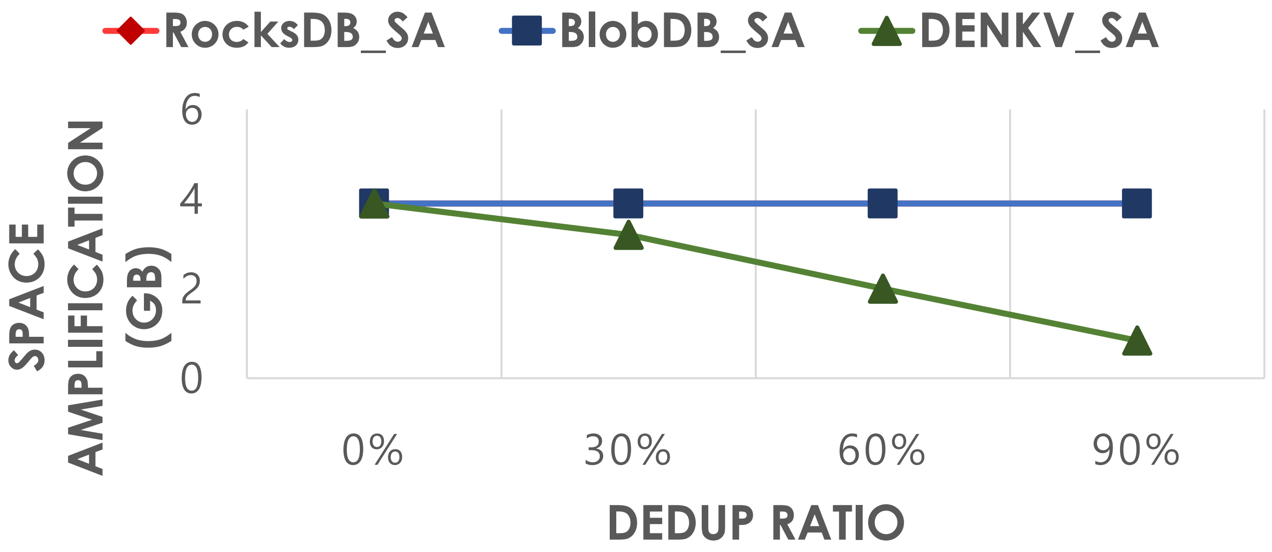
❑ Performance drops due to extra deduplication steps

❑ With increasing dedup ratio, performance improves

Write and space amplification analysis

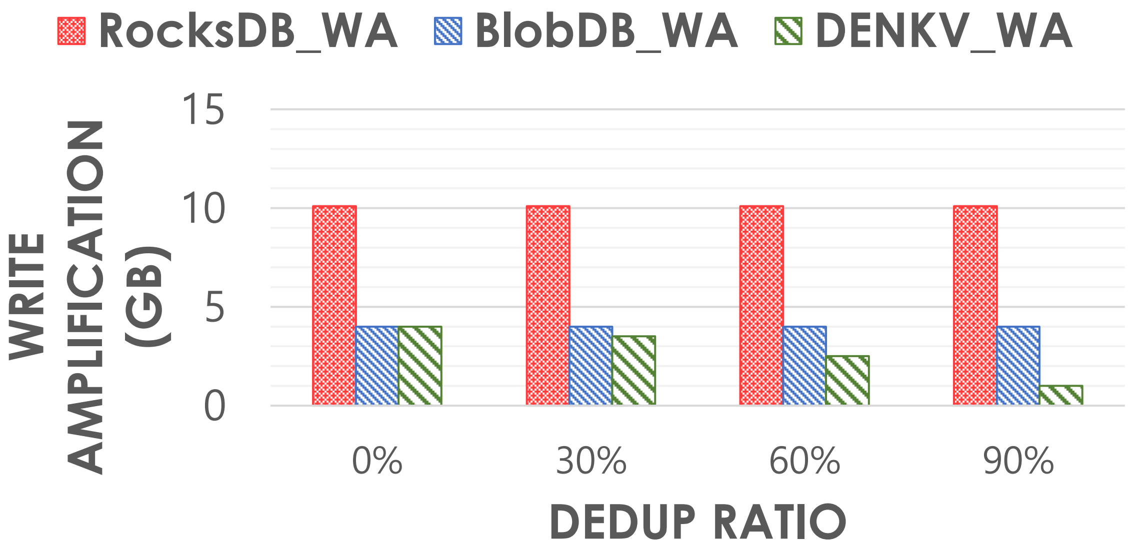
- 4 KB KV Pairs

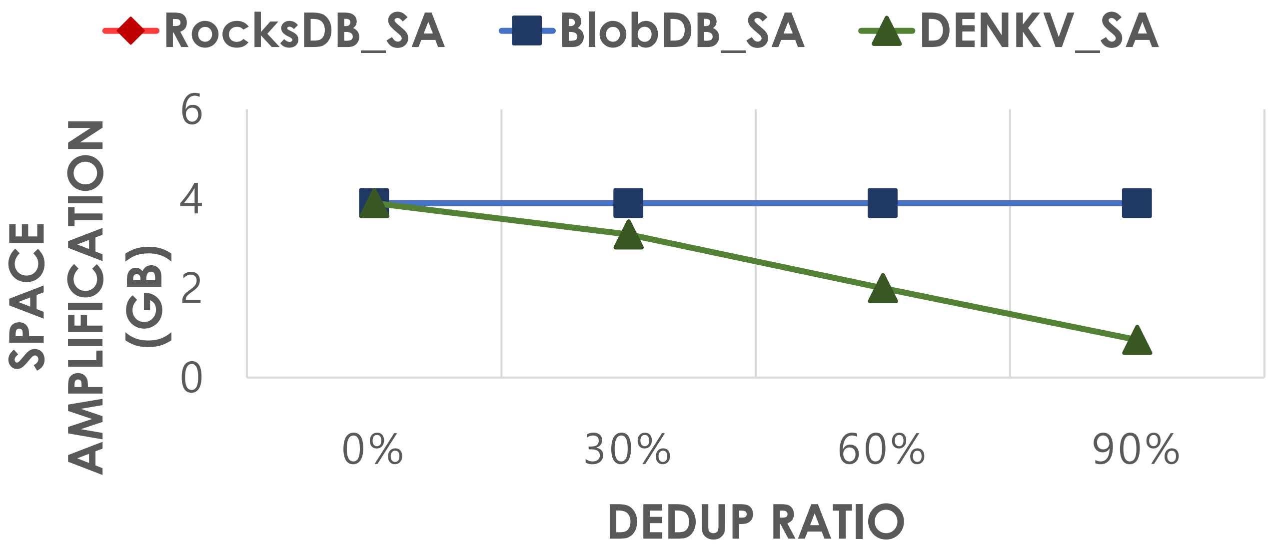




Write and space amplification analysis

- ❑ 4 KB KV Pairs

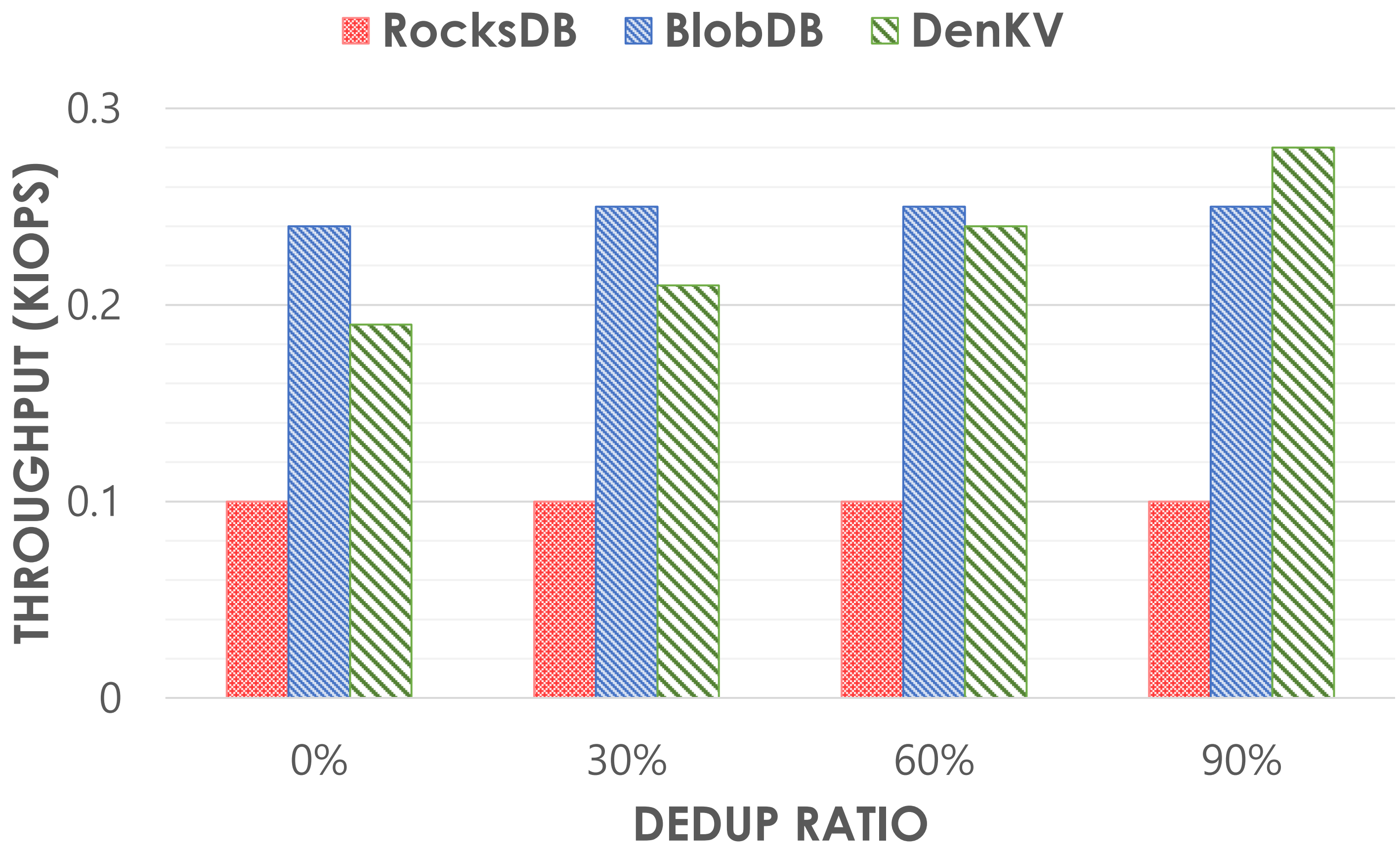




- ❑ 4x WA reduced with small KV pairs
- ❑ 4.6x SA reduced with small KV pairs

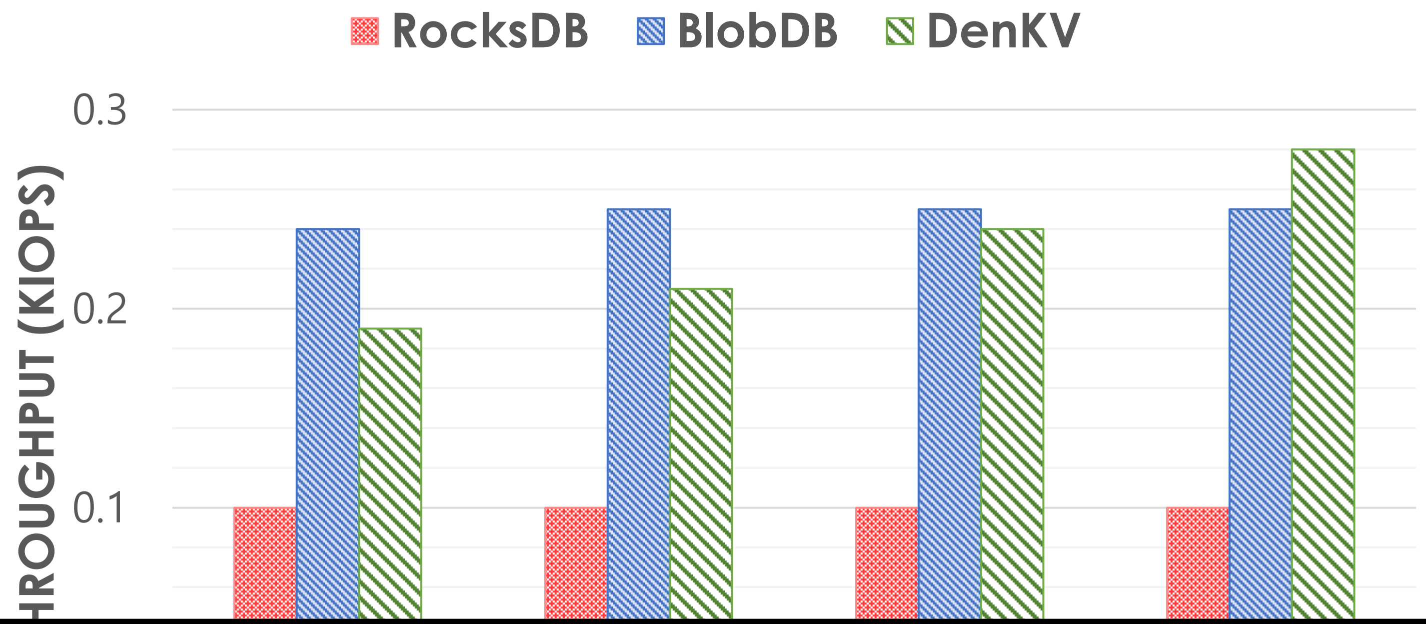
Performance analysis

- 1 MB KV Pairs



Performance analysis

❑ 1 MB KV Pairs

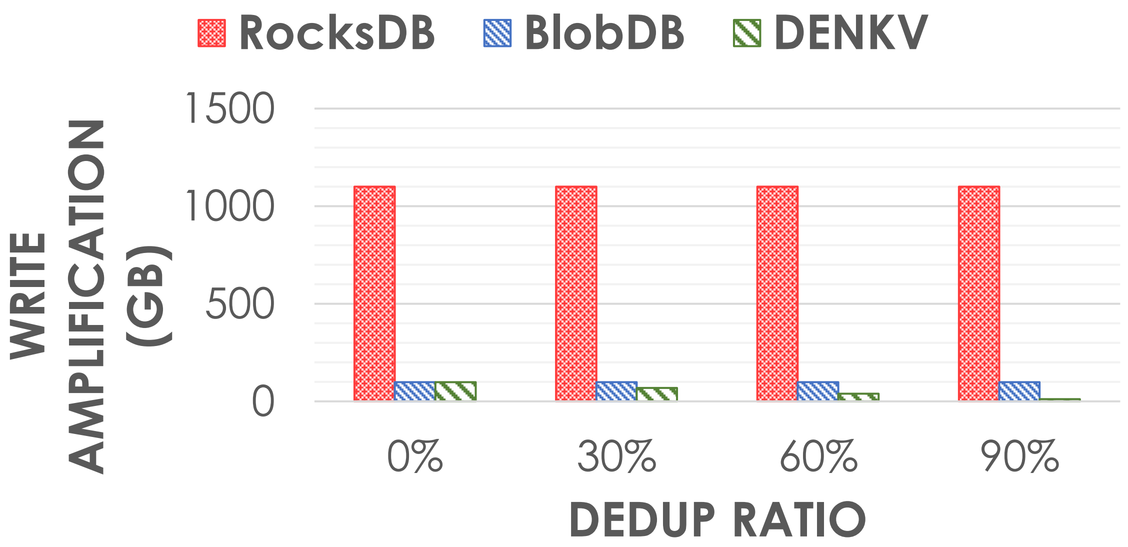


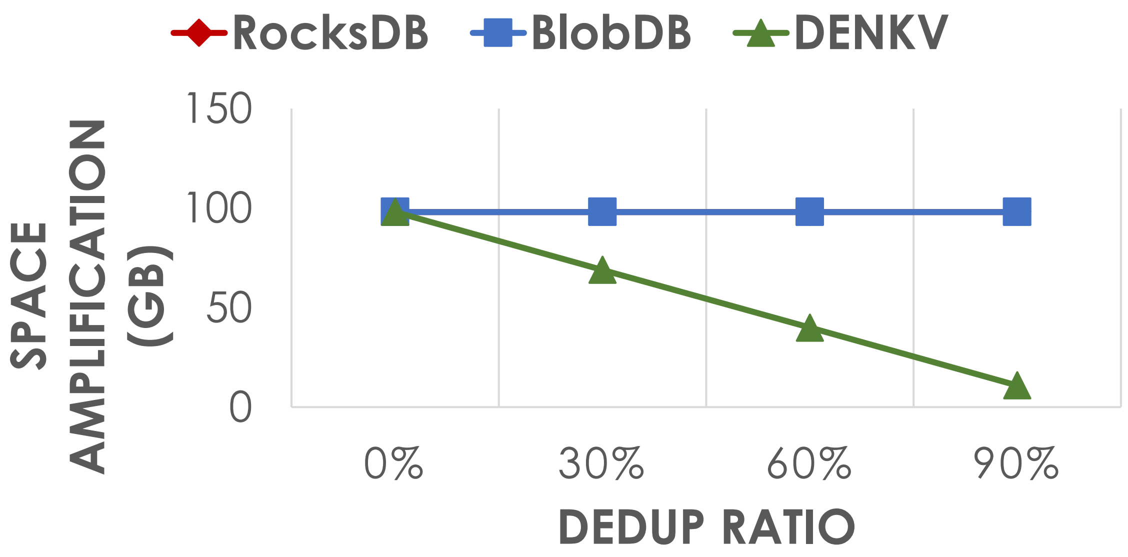
❑ Performance drops due to extra deduplication steps

❑ Outperforms all with highest dedup ratio

Write and space amplification analysis

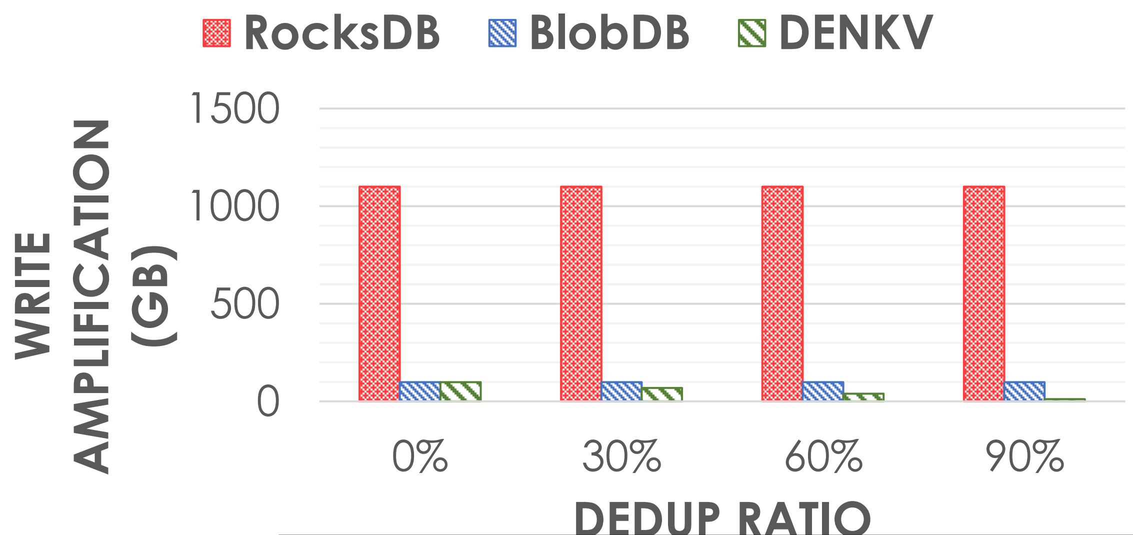
- 1 MB KV Pairs

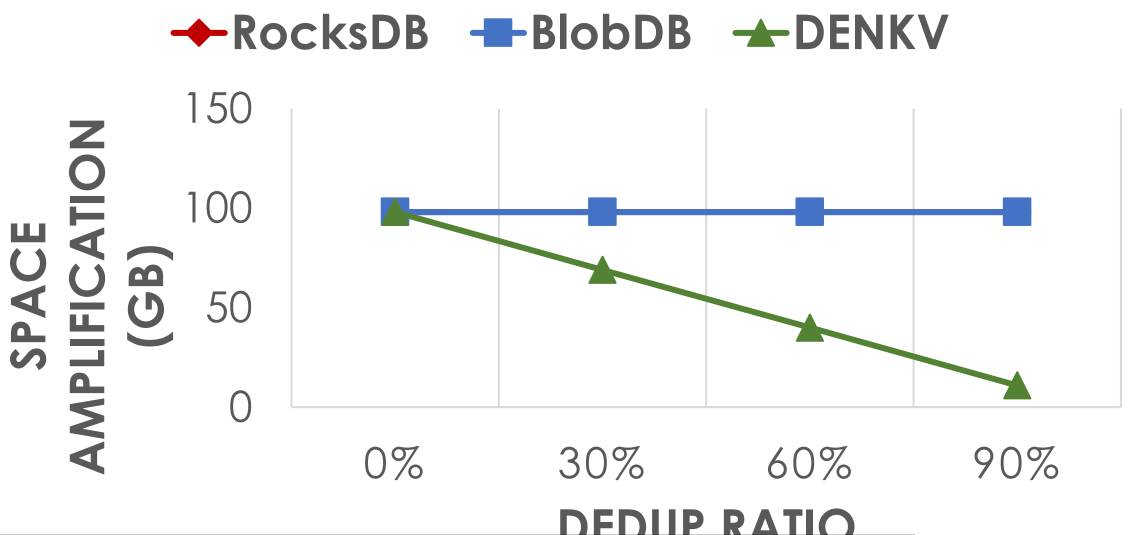




Write and space amplification analysis

- ❑ 1 MB KV Pairs





- ❑ 8x WA reduced with small KV pairs
- ❑ 8.9x SA reduced with large KV pairs

Questions to be answered

- How much deduplication influence the performance in general?
  - With small keys, performance is comparable.
  - There is a performance drop with large KV pairs.
- How much write amplification is reduced?
  - With 50% deduplication ratio, around 43% write amplification is reduced on average
- How much space amplification is reduced?
  - With 50% deduplication ratio, on average 45% less amount of space is utilized
- What are the bottlenecks?
  - Deduplication operation interfere the foreground IOs results in write stalls.

Conclusion

Conclusion

- HPC applications generate significant amount of redundant data
- Distributed KV stores are gaining significant attention in HPC
  - Distributed KV stores rely on monolithic KV stores
  - LSM-tree-based KV stores suffer from high WA and SA
- DENKV introduced APID (asynchronous partly inline deduplication) module
  - Reduces WA and SA while maintaining the performance

Thank you

safdar@sogang.ac.kr
https://sites.google.com/view/safdarjamil95