MSST '20
October 29-30, 2020

LightKV: A Cross Media Key Value Store with Persistent Memory to Cut Long Tail Latency

Shukai Han, Dejun Jiang, Jin Xiong

Institute of Computing Technology, Chinese Academy of Sciences

University of Chinese Academy of Sciences

Outline

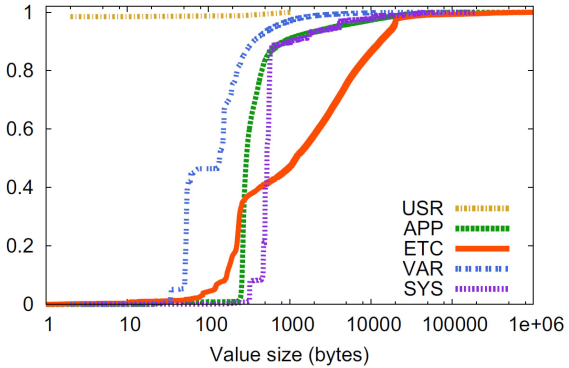
- ✓ **Background & Motivation**
- Design
- Evaluation
- Conclusion

Key-Value Store

- Key-Value (KV) stores are widely deployed in data centers.

MEMCACHED redis LEVELDB RocksDB

- KV stores are latency-critical applications.



Workloads with a high percentage of small KV items[1]

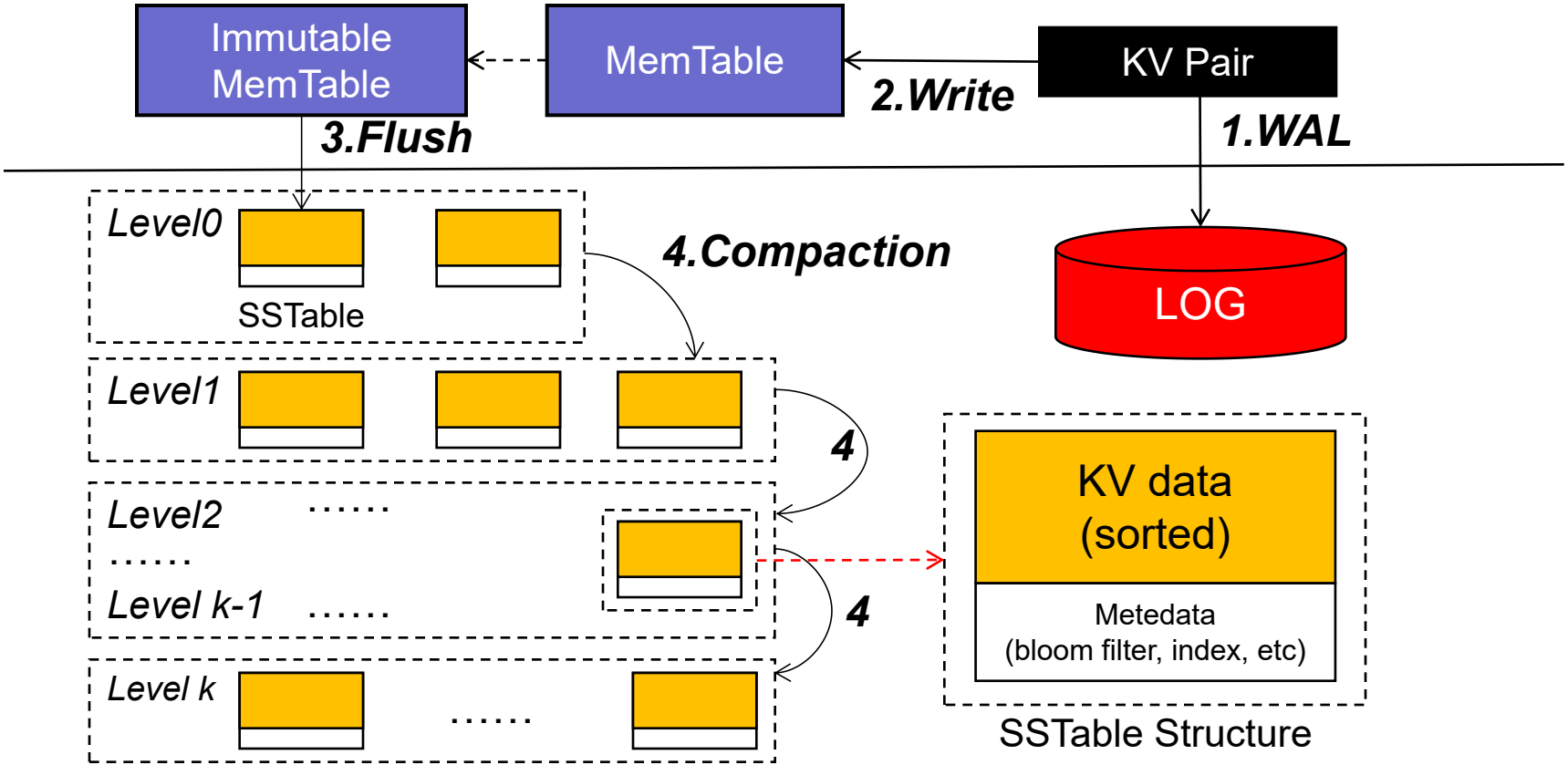
Applications with low latency requirements

[1] Berk, SIGMETRICS '2012

LEVELDB

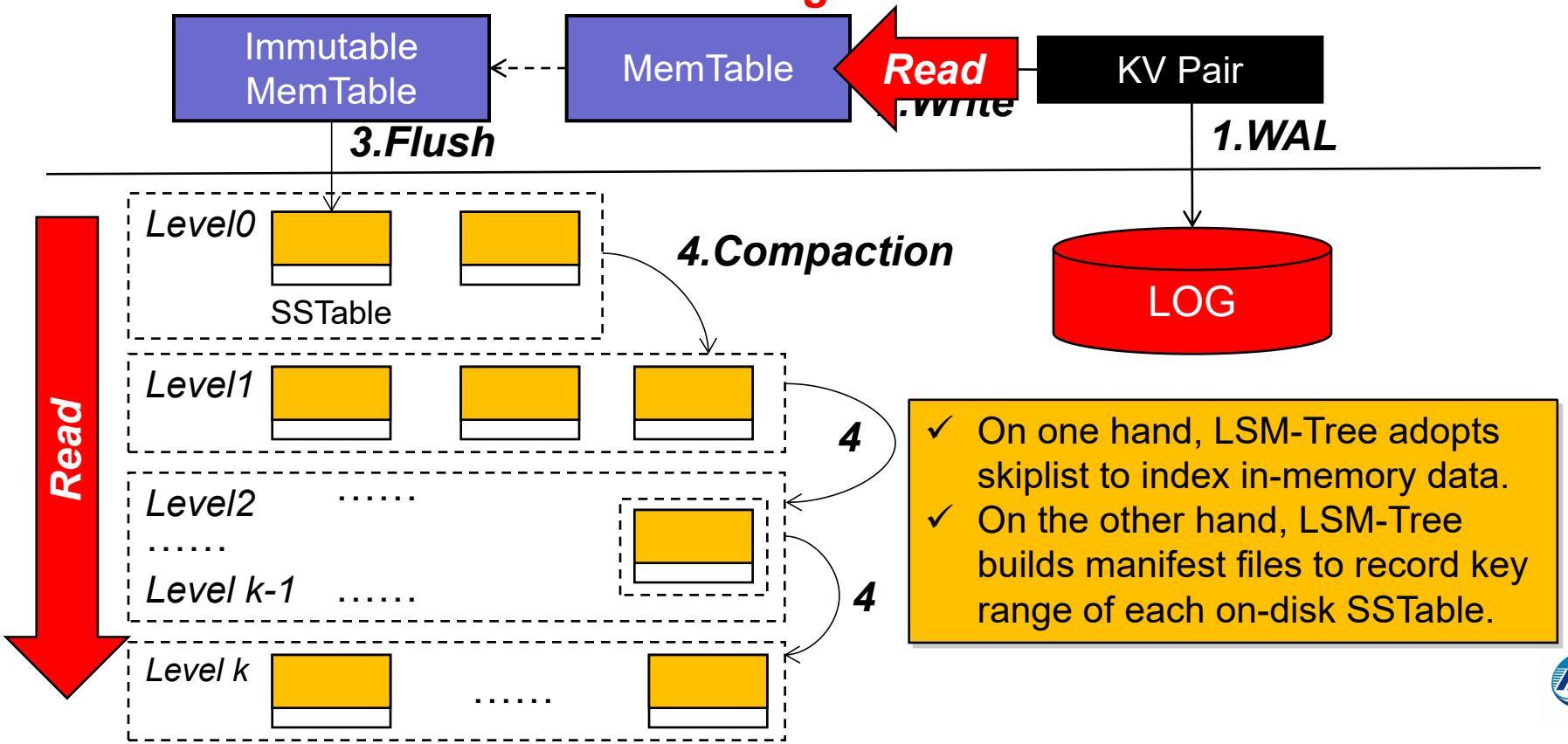
RocksDB

Log-Structured Merge Tree (LSM-Tree)



Limitations of Persistent KV Store

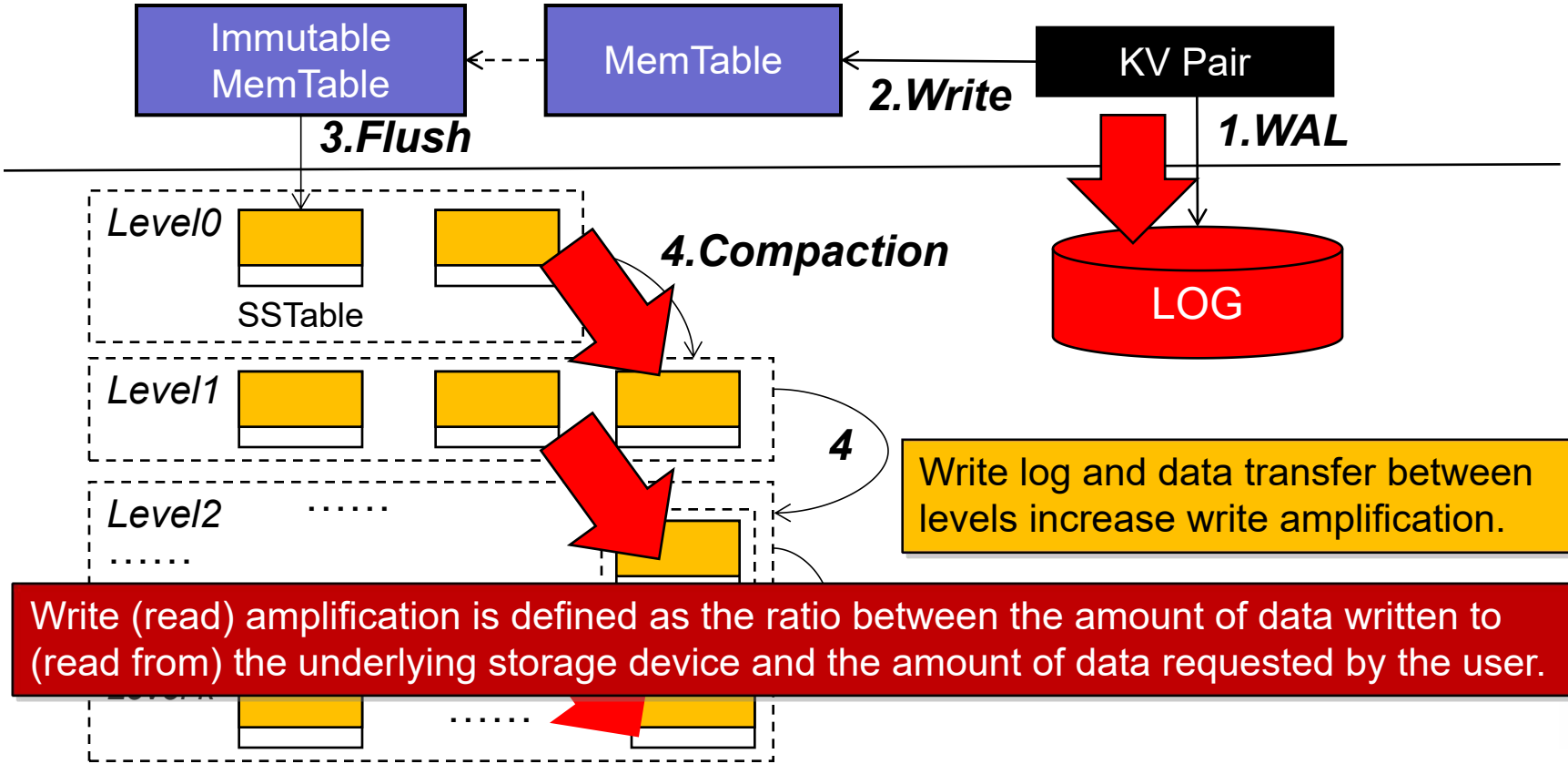
Inefficient indexing for cross-media



- ✓ On one hand, LSM-Tree adopts skiplist to index in-memory data.
- ✓ On the other hand, LSM-Tree builds manifest files to record key range of each on-disk SSTable.

Limitations of Persistent KV Store

High write amplification

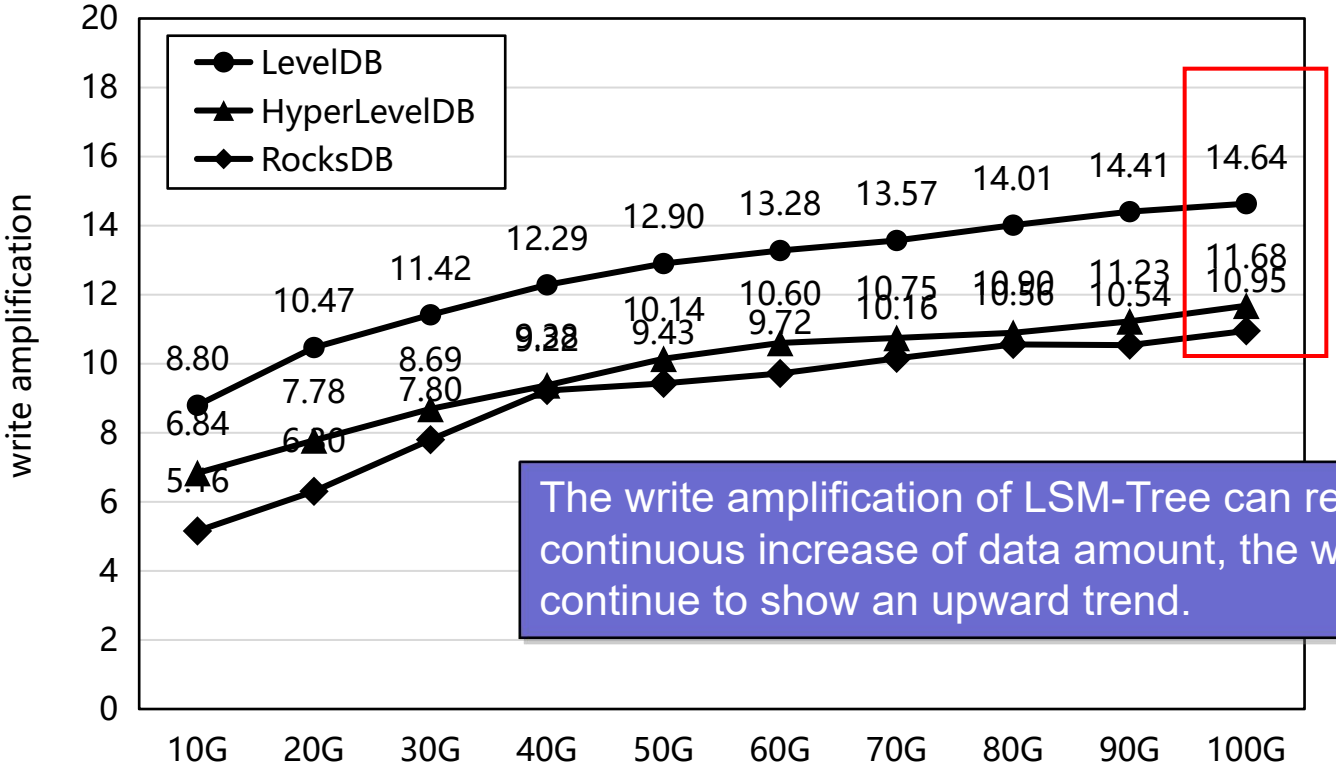


Write log and data transfer between levels increase write amplification.

Write (read) amplification is defined as the ratio between the amount of data written to (read from) the underlying storage device and the amount of data requested by the user.

Limitations of Persistent KV Store

High write amplification

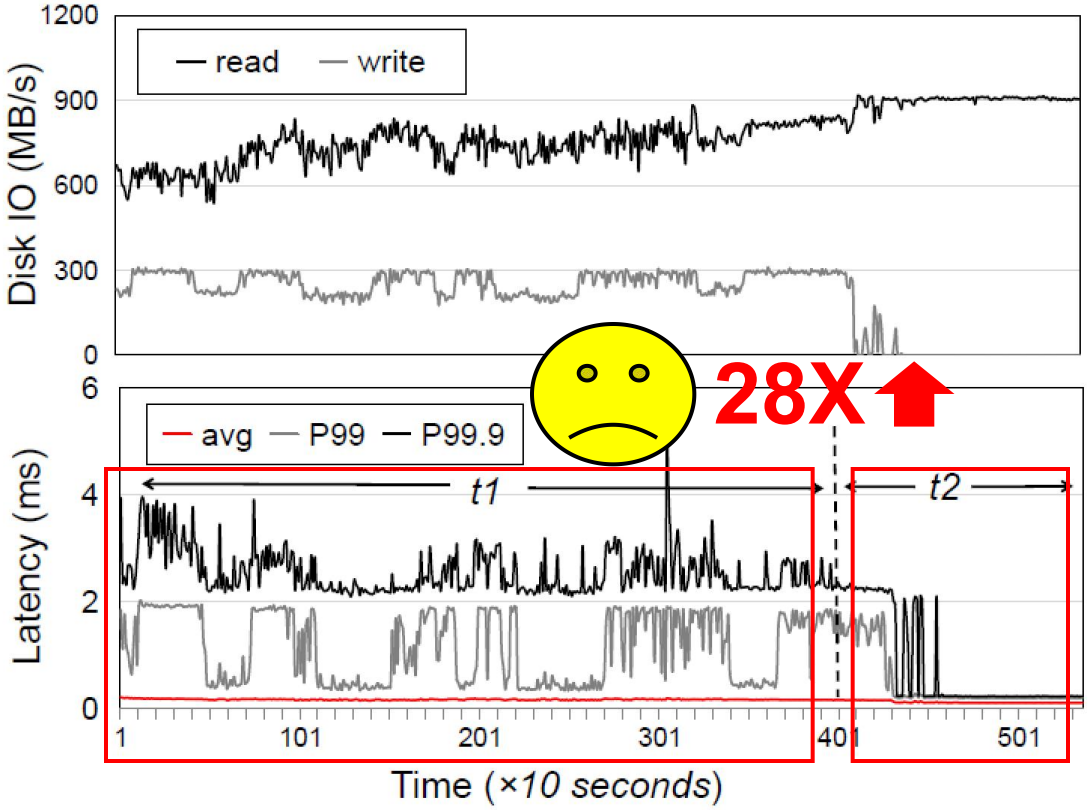


10X+

The write amplification of LSM-Tree can reach 10x, and with the continuous increase of data amount, the write amplification continue to show an upward trend.

Limitations of Persistent KV Store

Heavy tailed read latency under mixed workload



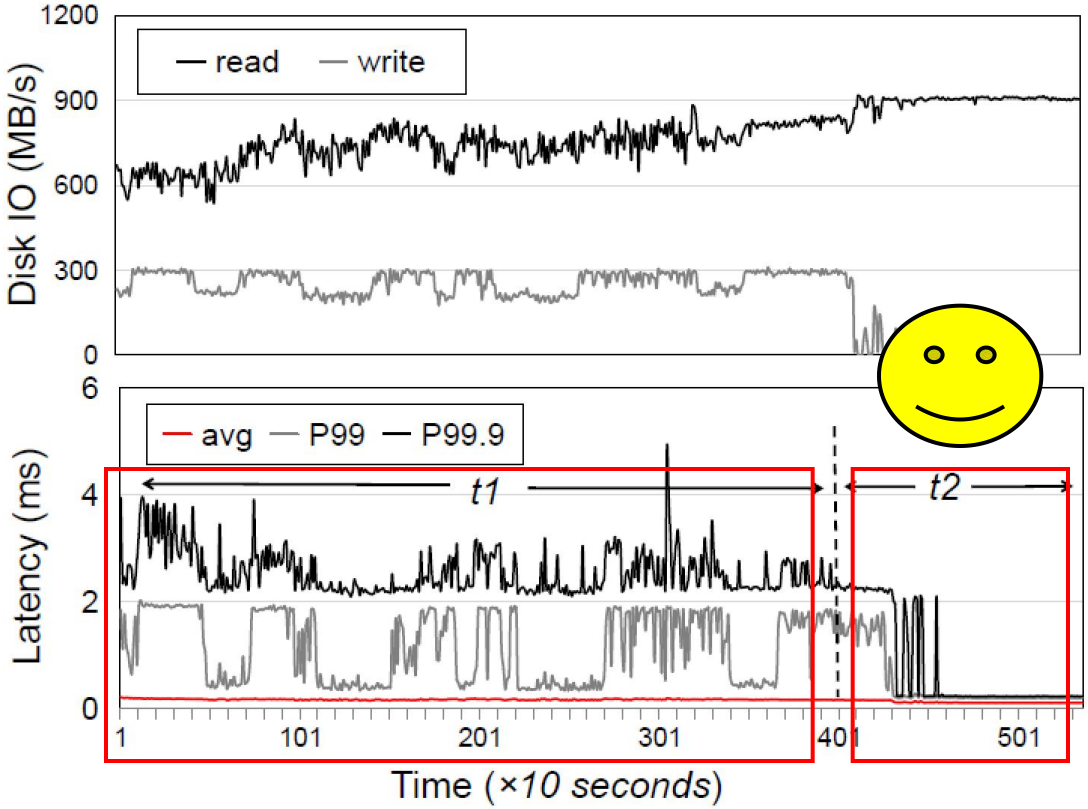
- ✓ We first warm up LevelDB with 100 GB data.
- ✓ We measure the average latency as well as 99th and 99.9th percentile read latencies every 10 seconds.

t1: Run a mixed workload of randomly reading 50 GB existing data and randomly inserting another 50 GB data.

The maximum 99th and 99.9th percentile read latencies can reach **13 and 28 times** than the average read latency.

Limitations of Persistent KV Store

Heavy tailed read latency under mixed workload



t2: Run read-only workload.

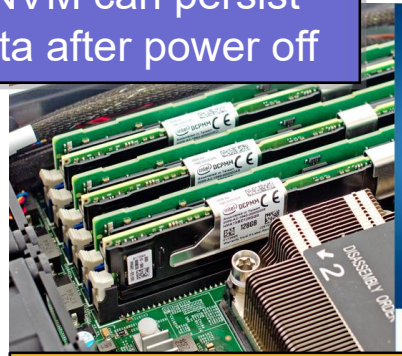
After the compaction finishes, the read tail latency is significantly reduced.

Reducing write amplification is not only helpful for reducing the total write amount of the disk, increasing system throughput, but also helping to reduce the read tail latency under mixed read and write loads.

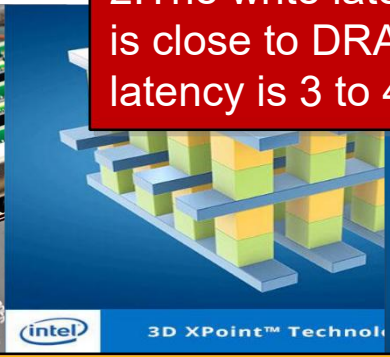
Non-Volatile Memory

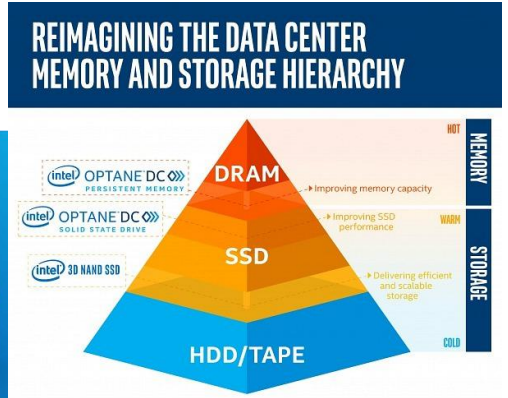
- Non-Volatile Memories (NVMs) provide low latency and byte addressable features.
  - 3D XPoint, Phase Change Memory (PCM), and Resistive Memory (ReRAM)
- The first PM product, Intel Optane DC Persistent Memory (PM), was announced [19] in April 2019.

1.NVM can persist data after power off



2.The write latency of Optane DC PM is close to DRAM, while its read latency is 3 to 4 times that of DRAM.



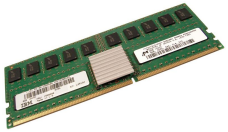


3.The write and read bandwidths of Optane DC PM are around 2GB/s and 6.5GB/s, which is about 1/8 and 1/4 that of DRAM separately.

Outline

- Background & Motivation
- ✓ **Design**
- Evaluation
- Conclusion

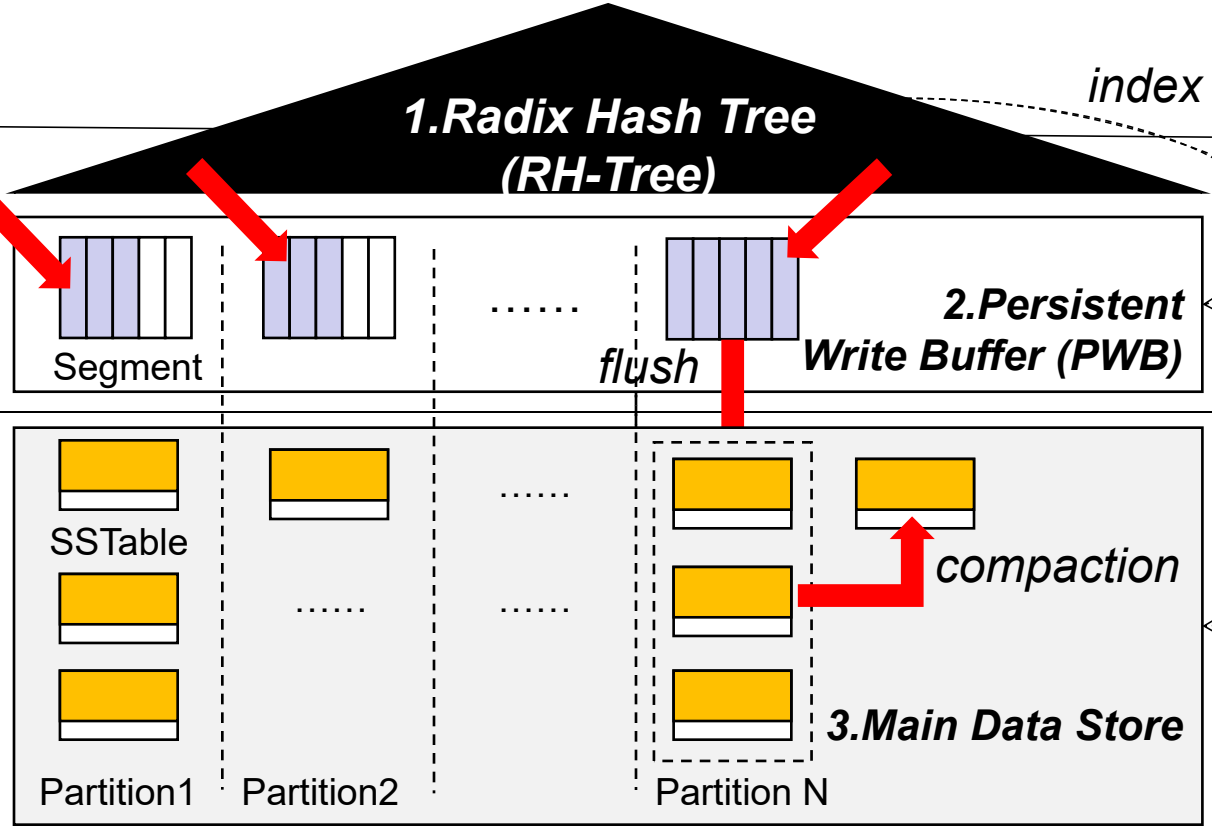
LightKV System Overview



DRAM

Persistent Memory

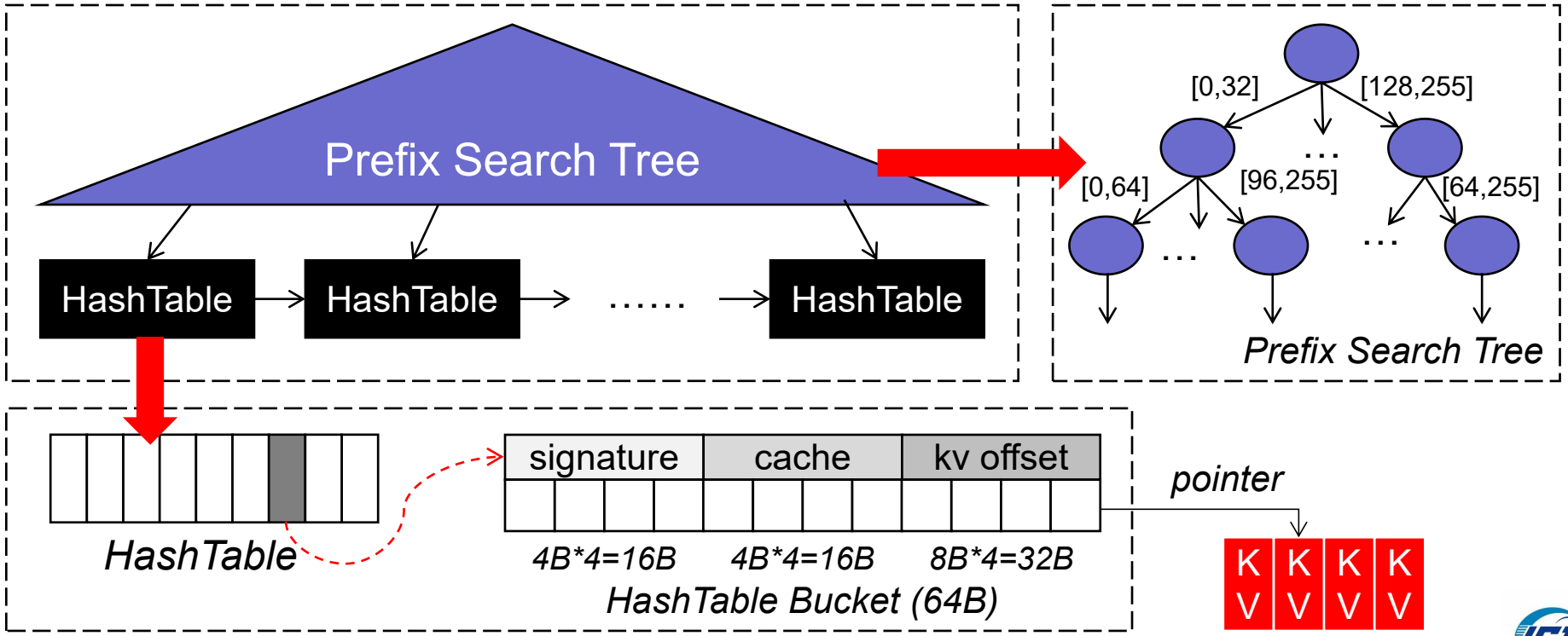
SSD



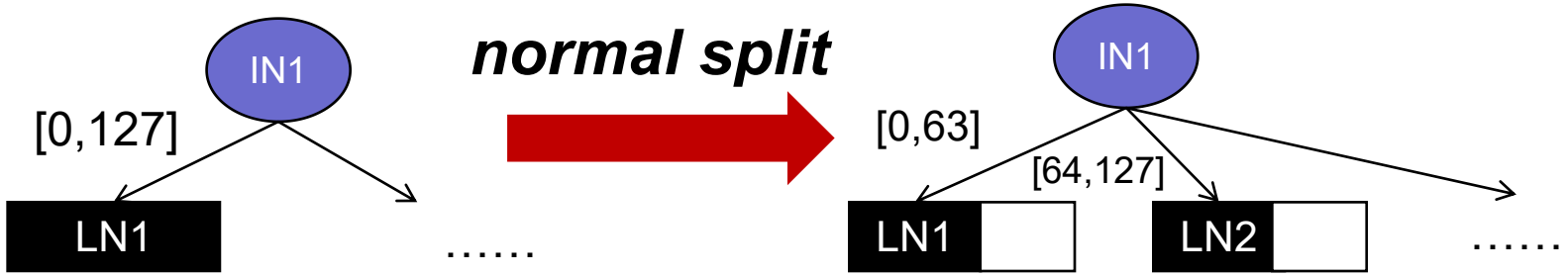
Challenges

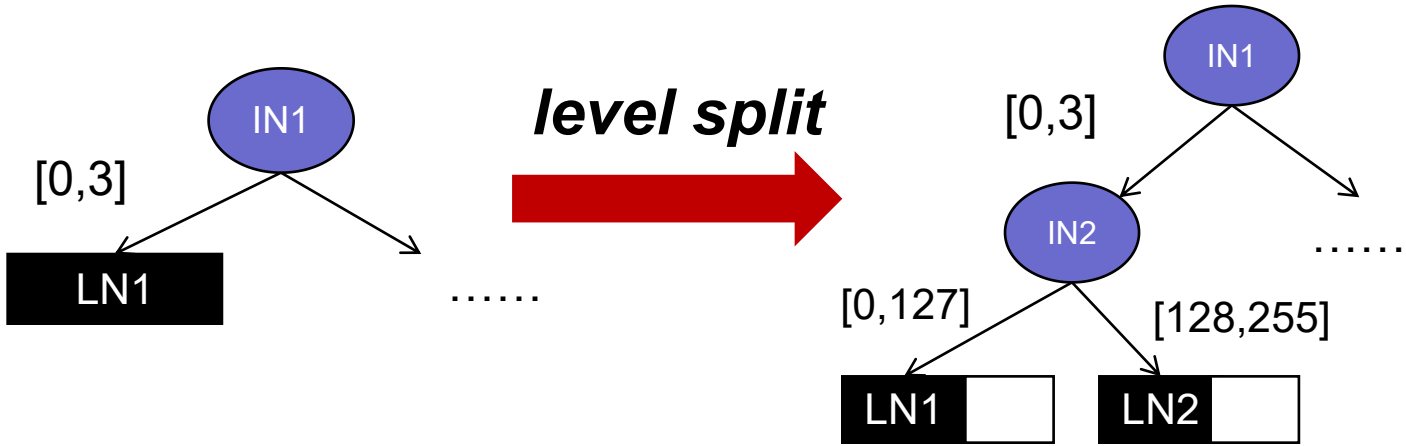
- How does Radix Hash tree index KV items across media?
- How does Radix Hash tree balance performance and data growth?
- How does Radix Hash tree conduct well-controlled data compaction to reduce write amplification?

Radix Hash Tree Structure

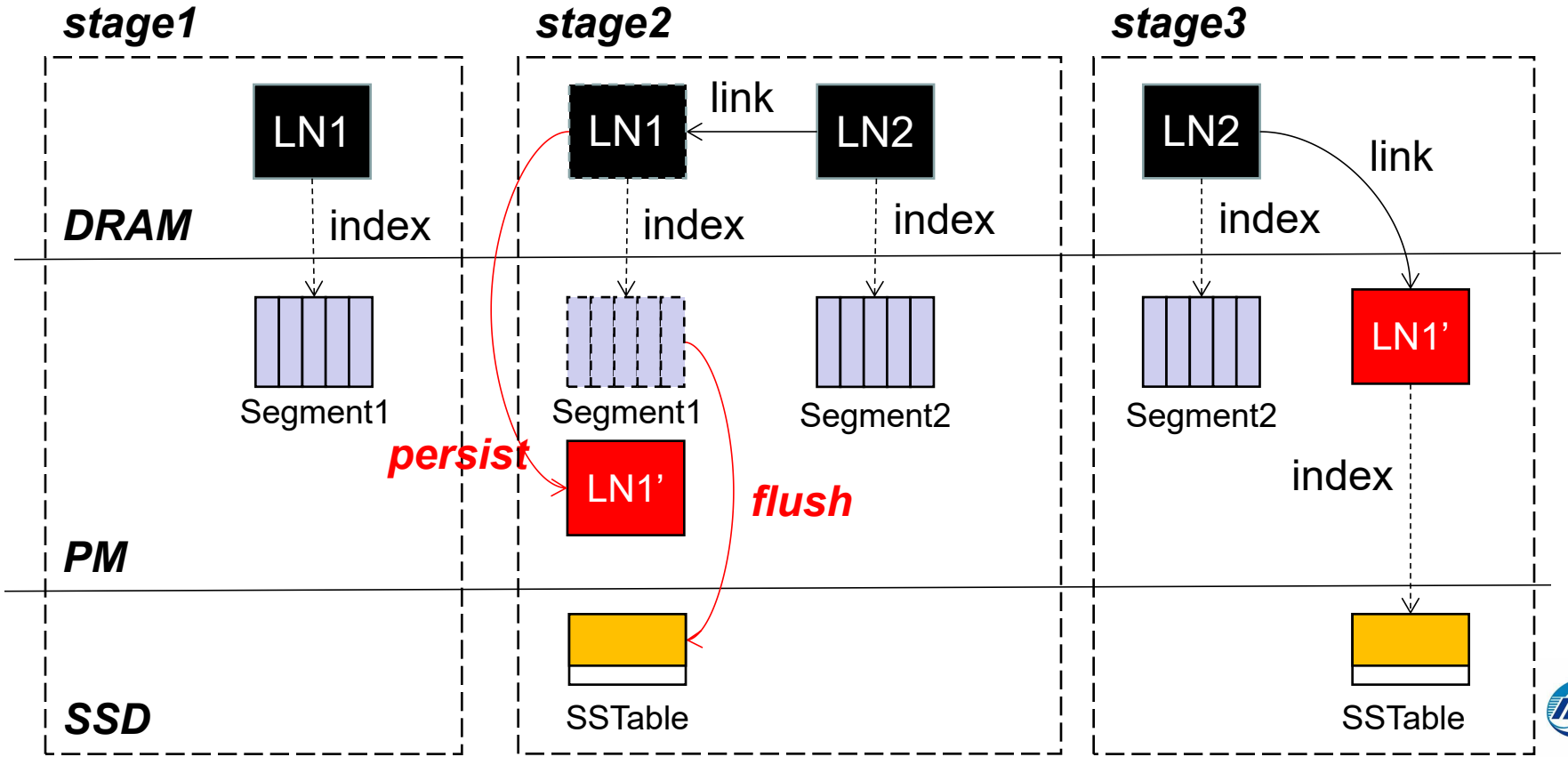


RH-Tree split

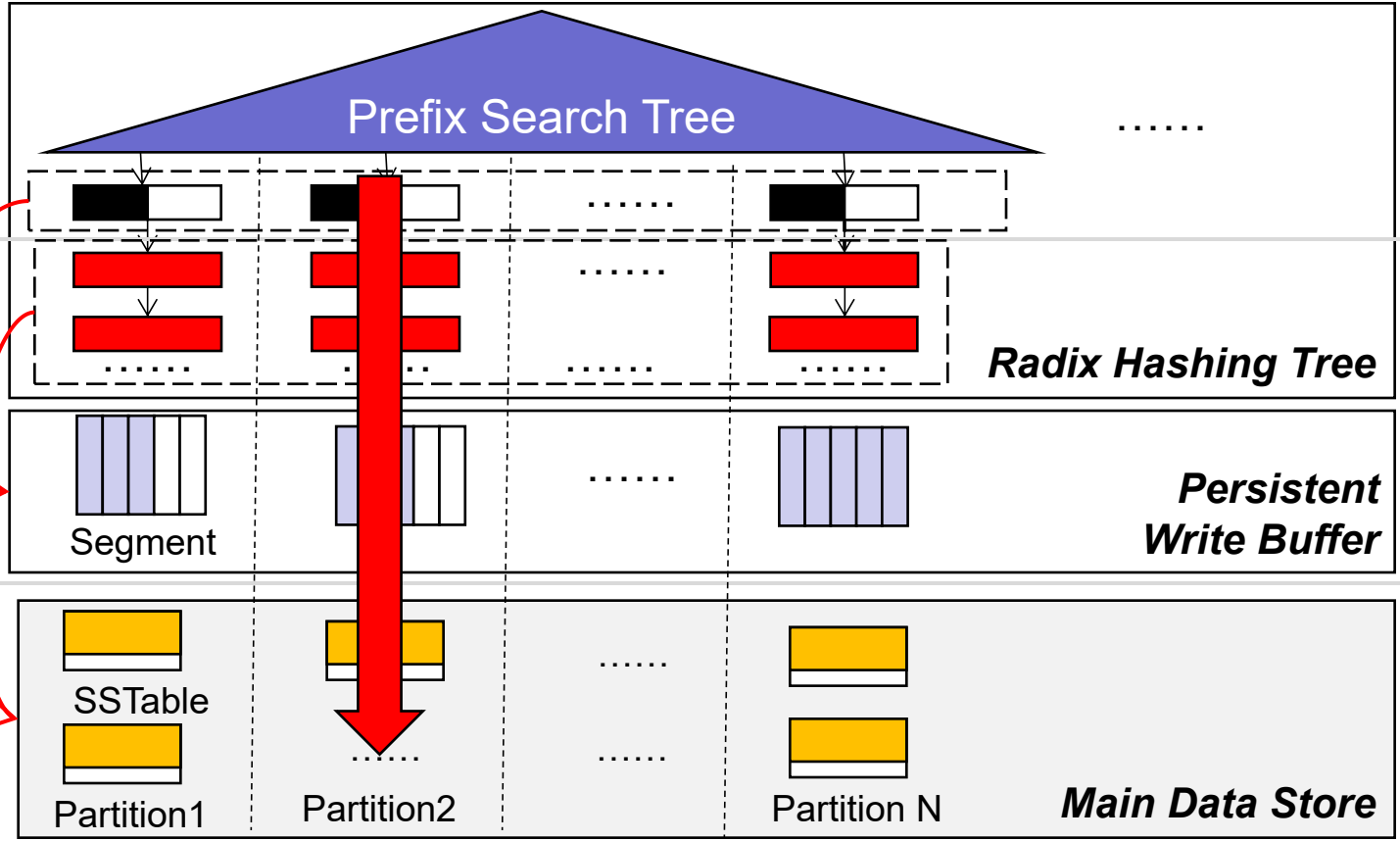


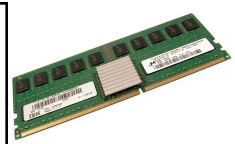


Linked hash leaf node



RH-Tree placement



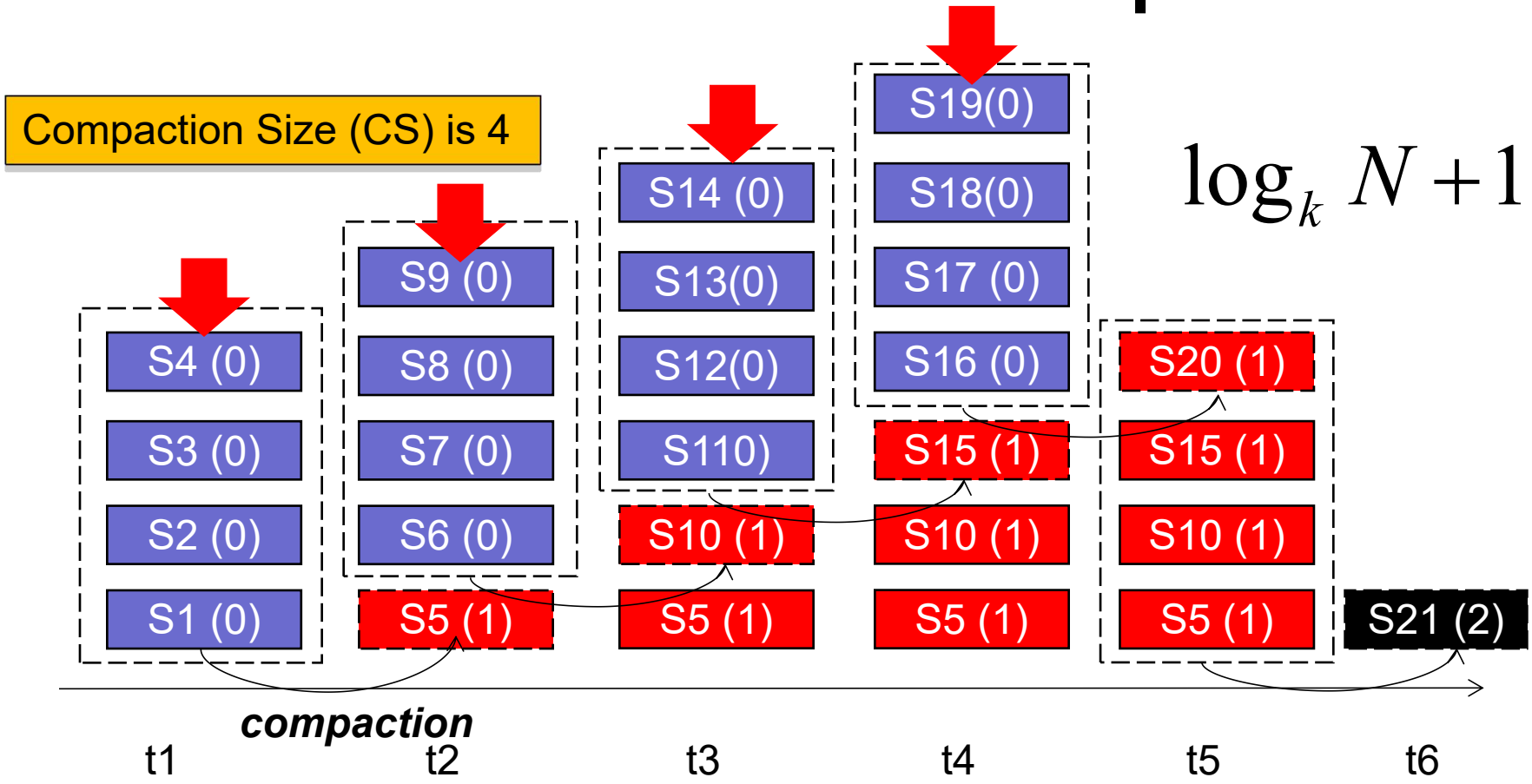


DRAM

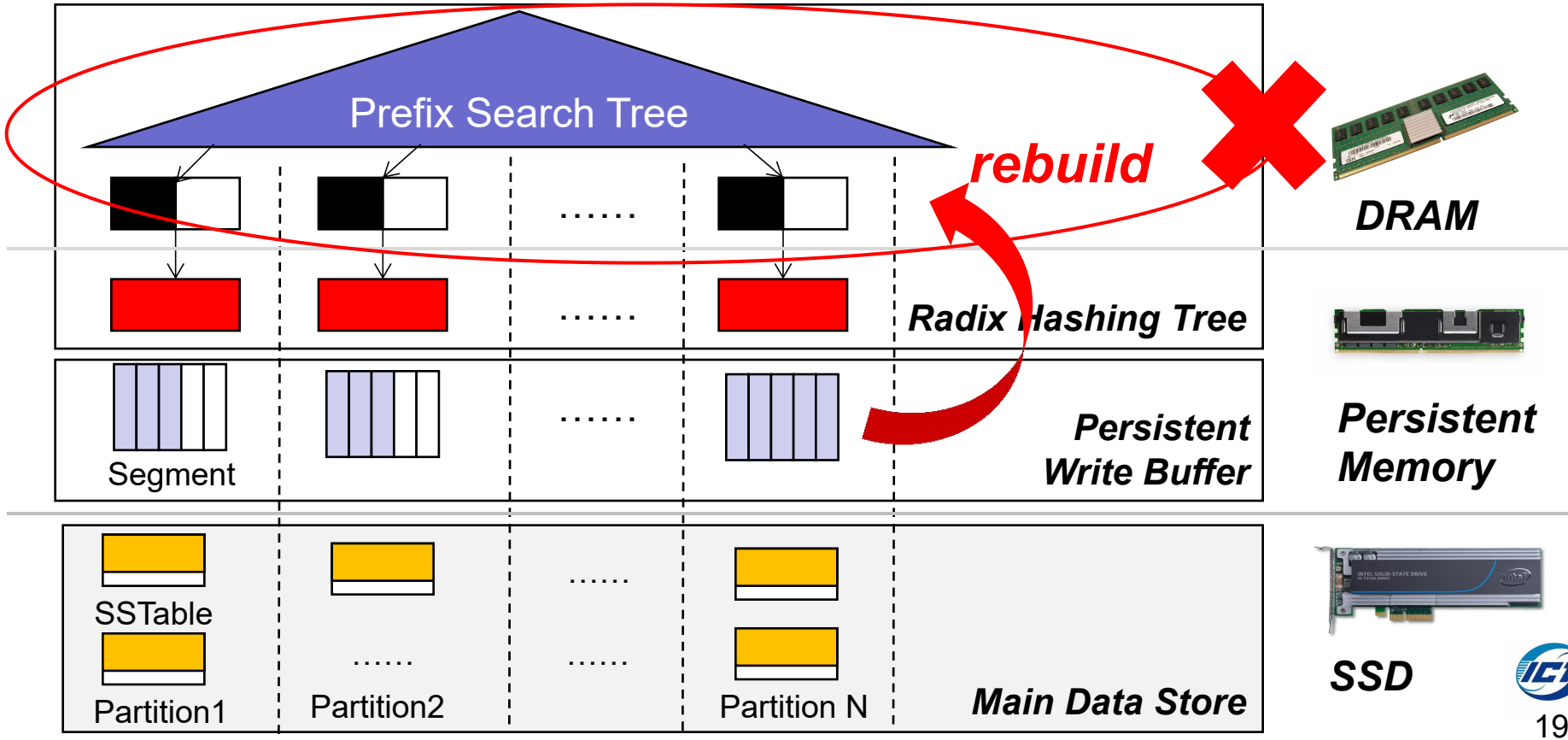
Persistent Memory

SSD

Partition-based data compaction



Recovery


Prefix Search Tree
rebuild
DRAM
Radix Hashing Tree
Persistent Memory
Segment
Persistent Write Buffer
SSTable
Partition1
Partition2
Partition N
Main Data Store
SSD

Outline

- Background & Motivation
- Design
- ✓ **Evaluation**
- Conclusion

Experiment Setup

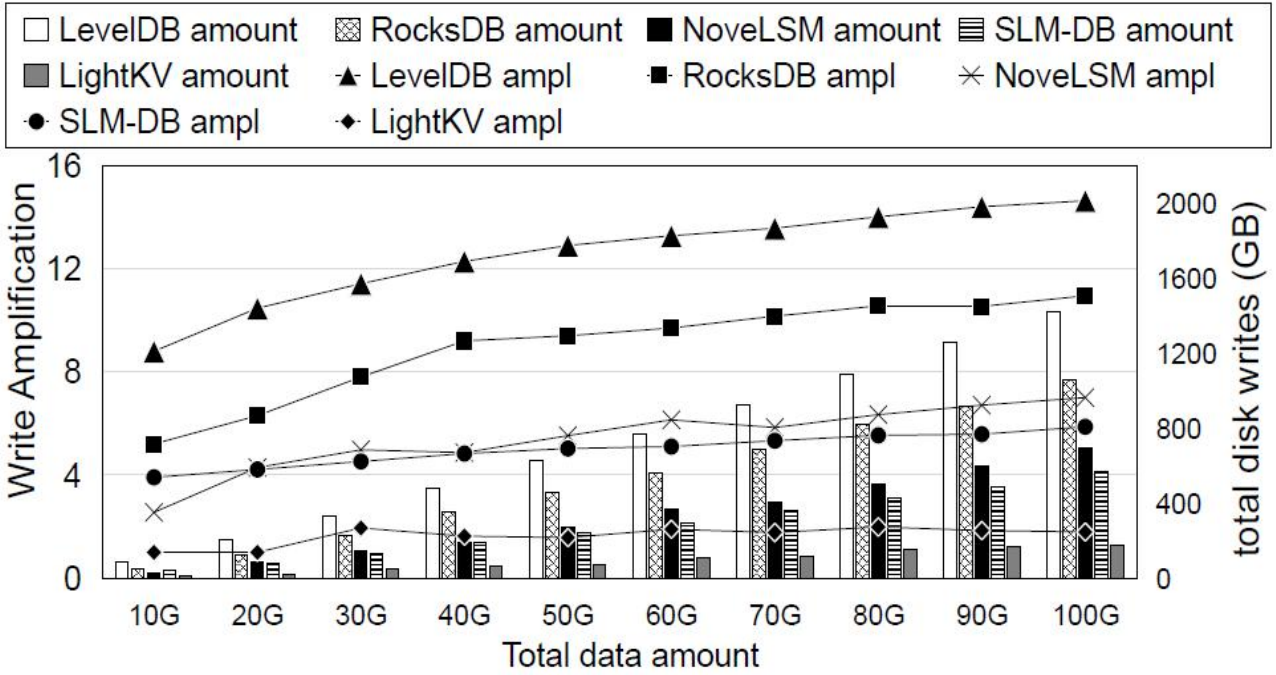
- **System and hardware configuration**
  - Two Intel Xeon Gold 5215 CPU (2.5GHZ), 64GB memory and one Intel DC P3700 SSD of 400GB.
  - CentOS Linux release 7.6.1810 with 4.18.8 kernel and use ext4 file system.

- **Compared systems**
  - LevelDB、RocksDB
  - NoveLSM、SLM-DB

- **Workloads**
  - db_bench as microbenchmark
  - YCSB as the actual workload

Workload	YCSB Workload Description
A	50% reads and 50% updates
B	95% reads and 5% updates
C	100% reads
D	95% reads for latest keys and 5% inserts
E	95% scan and 5% inserts
F	50% reads and 50% read-modify-writes

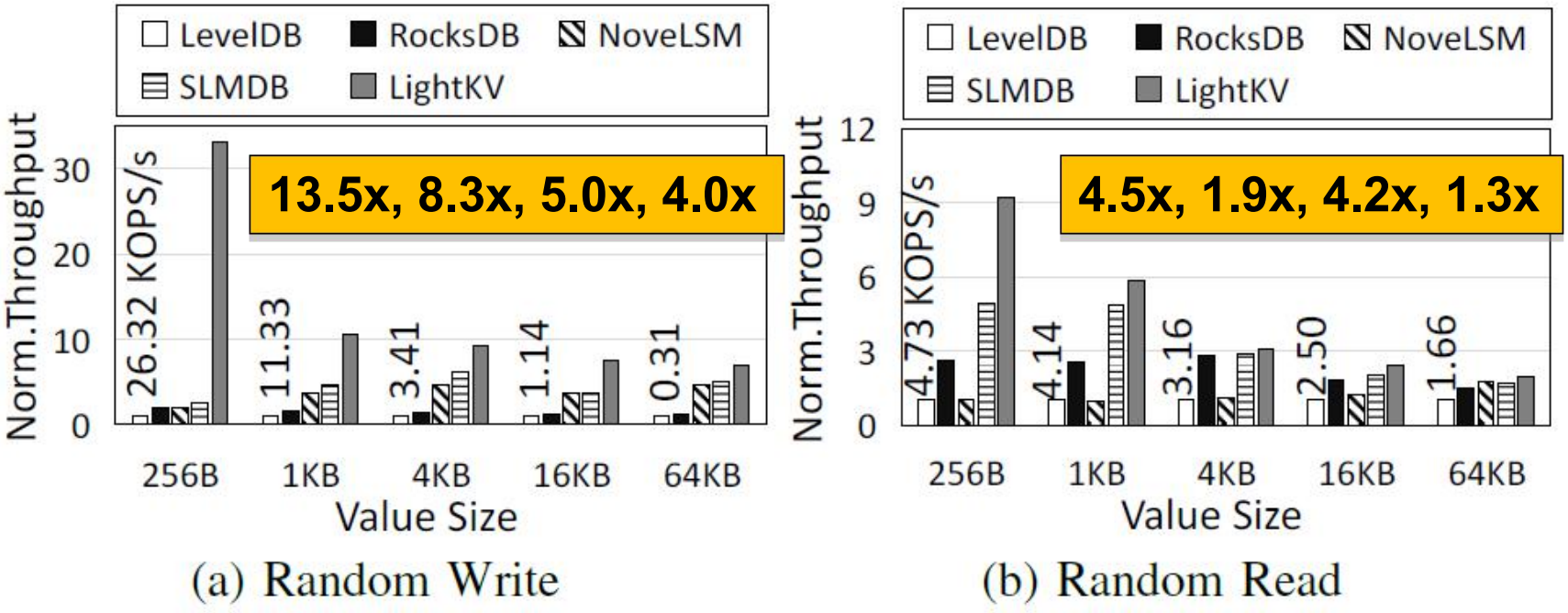
Reducing write amplification



LightKV are reduced by **7.1x, 5.1x, 2.9x and 2.3x** compared to that of LevelDB, RocksDB, NoveLSM, and SLM-DB respectively.

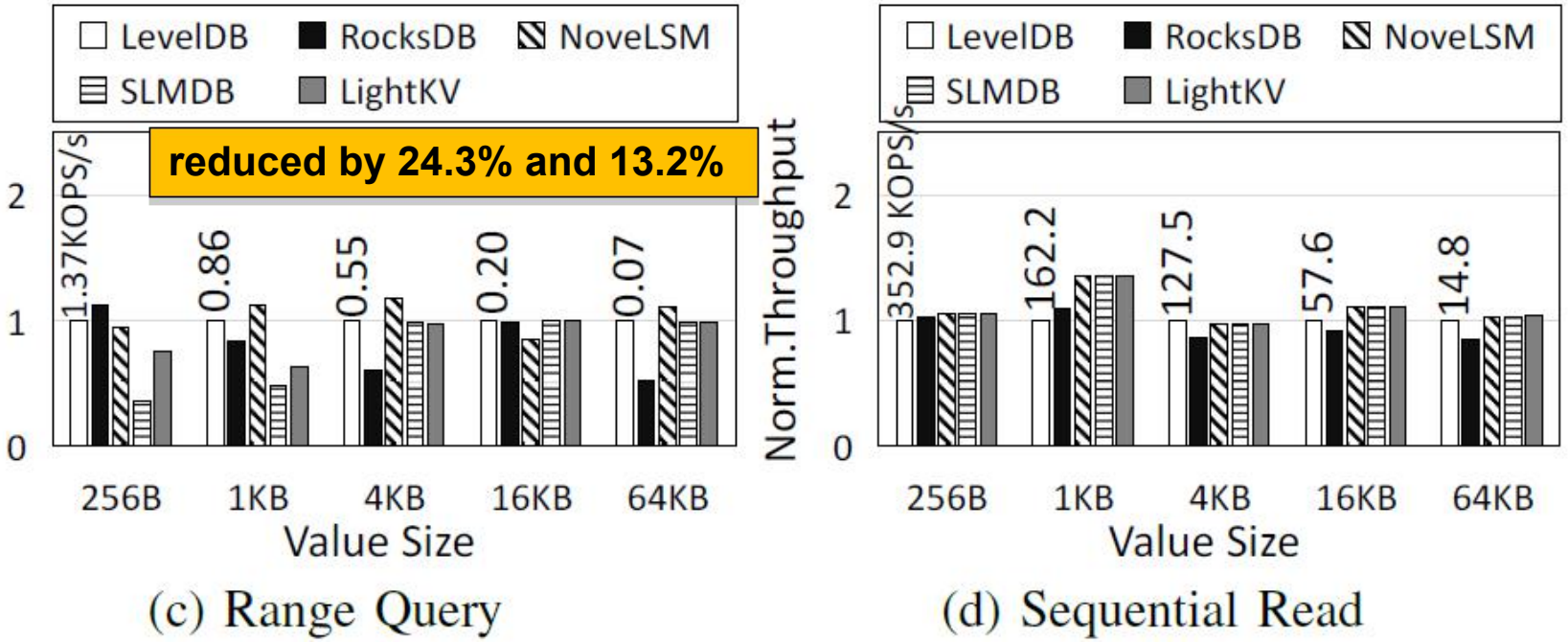
When the total amount of written data increases, the write amplification of LightKV remains stable (e.g. from 1.6 to 1.8 when the data amount increases from 50 GB to 100 GB).

Basic Operations



Thanks to the global index and partition compaction, LightKV can effectively reduce read-write amplification and improve read and write performance.

Basic Operations



The performance of LightKV in short range query is low. This is because it needs to search all SSTables in one or more partitions when performing a short range query.

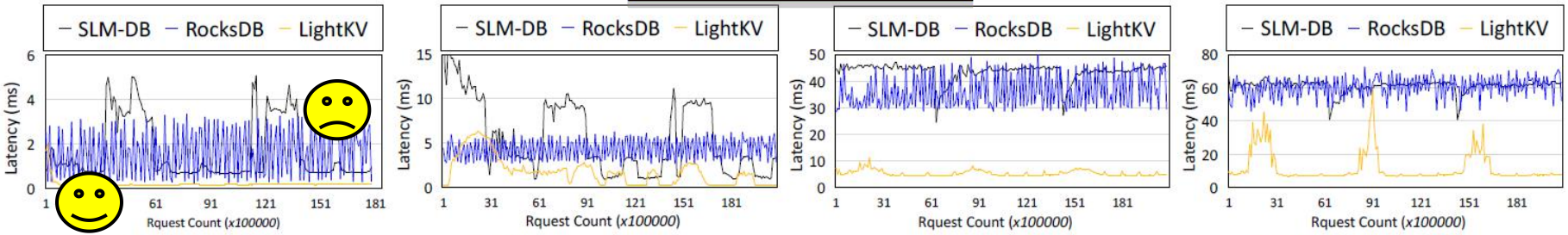
Tail latency under read-write workload

99th:17.9x, 10.5x, 6.4x, 3.5x

99.9th:15.7x, 9.2x, 8.8x, 3.4x

KV Store	256B			1KB			4KB			16KB			64KB		
%	avg	99	99.9	avg	99	99.9	avg	99	99.9	avg	99	99.9	avg	99	99.9
LevelDB	294	2.7	5.8	370	3.6	6.5	429	4.3	7.0	653	5.7	8.7	829	6.1	10.2
RocksDB	106	0.3	3.6	152	2.2	5.1	274	4.1	6.9	671	5.9	9.1	1383	7.9	14.9
NoveLSM	243	0.7	3.5	321	1.4	8.0	399	4.1	9.8	524	6.2	11.1	891	8.7	18.6
SLM-DB	81	0.3	1.5	101	0.8	4.0	201	2.5	4.7	414	7.5	13.8	641	6.5	8.1
LightKV	58	0.2	0.4	74	0.2					258	3.0	5.5	409	4.2	6.5

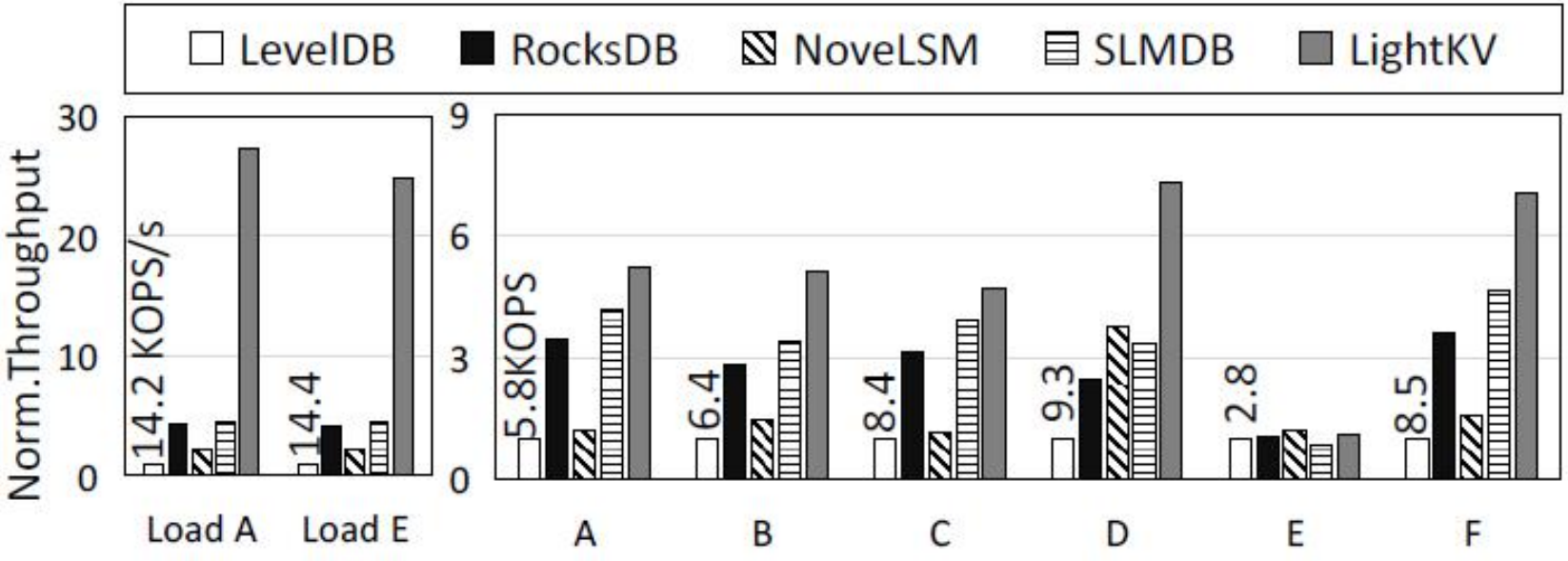
lower and stable



(a) 99^{th} read latency (b) 99.9^{th} read latency (c) 99^{th} write latency (d) 99.9^{th} write latency

Thanks to lower write amplification and global indexing, LightKV provides a lower and stable read and write tail latency.

Results with YCSB



LightKV provides better throughput in simulating actual workloads.

Outline

- Background & Motivation
- Design
- Evaluation
- ✓ **Conclusion**

Conclusion

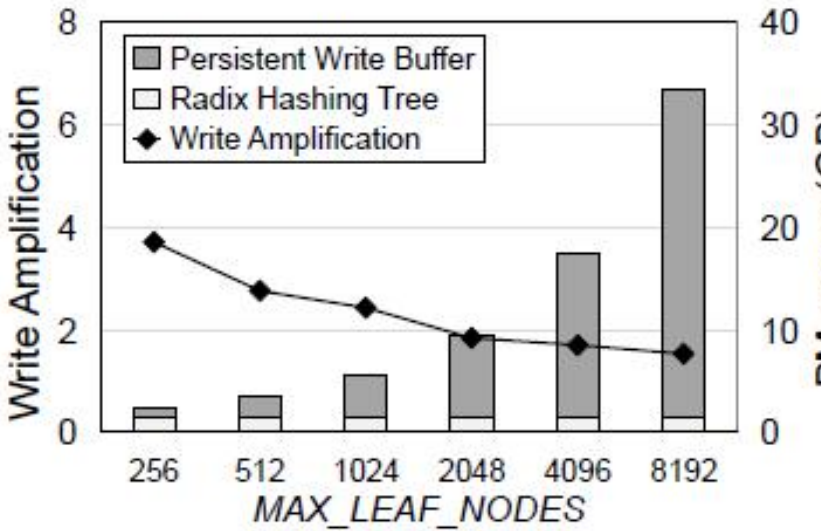
- LSM-Tree based on traditional storage devices faces problems such as read-write amplification
- At the same time, the emergence of non-volatile memory provides opportunities and challenges for building efficient key-value storage systems
- In this paper, we propose LightKV a cross media key-value store with persistent memory. LightKV effectively reduces the read-write amplification of the system by establishing a RH-Tree and adopting a column-based partition compaction.
- The experiment results show that LightKV reduces write amplification by up to 8.1x and improves read performance by up to 9.2x. It also reduces read tail latency by up to 18.8x under read-write mixed workload.

THANK YOU !

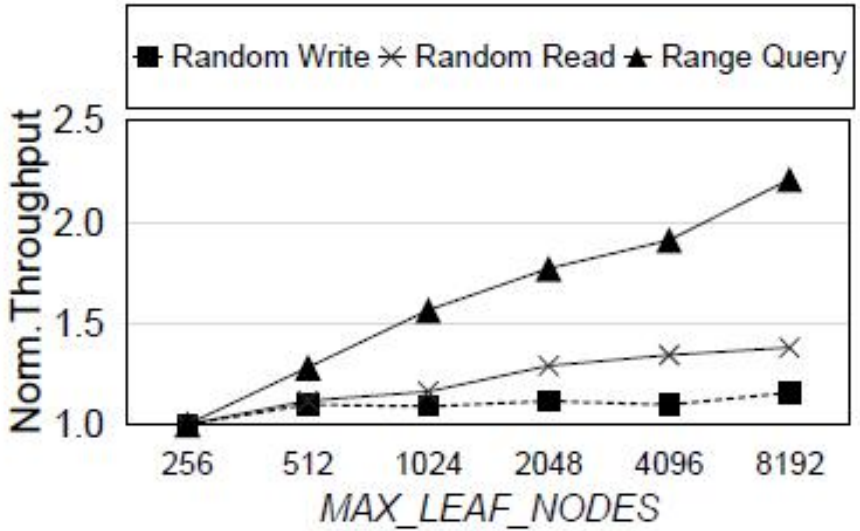
Q & A

Author Email: hanshukai@ict.ac.cn

Sensitivity analysis



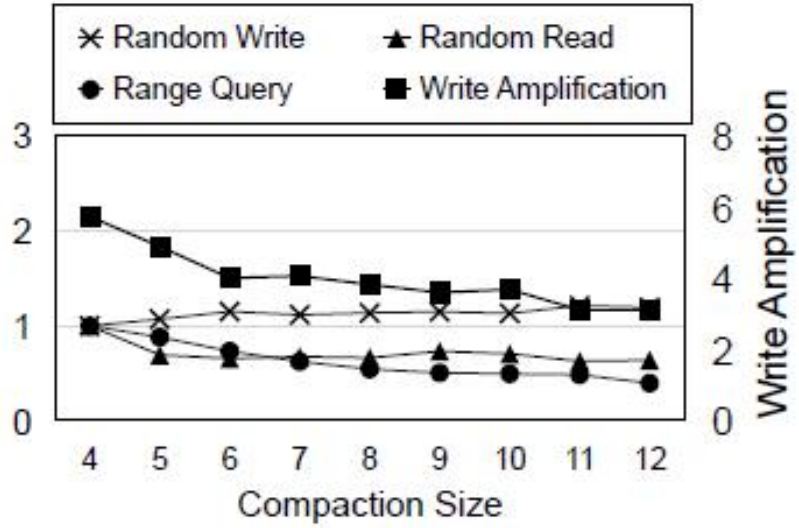
(a) *MAX_LEAF_NODES* sensitivity



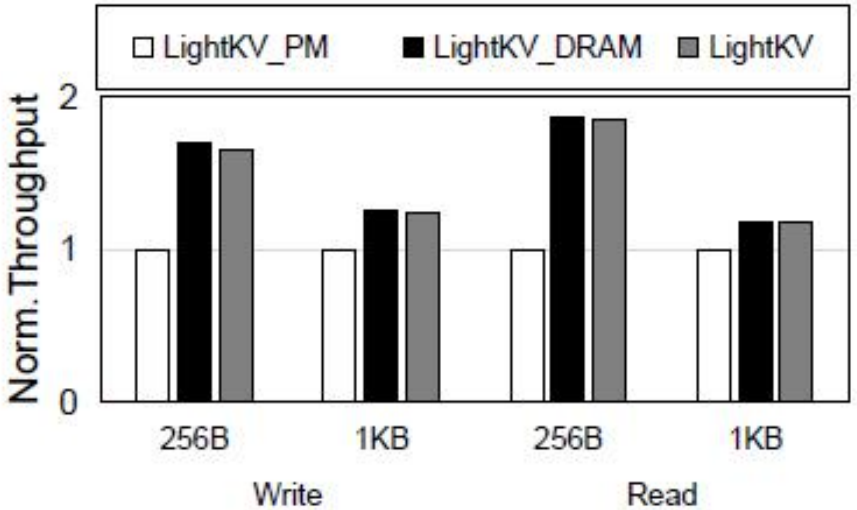
(b) *MAX_LEAF_NODES* sensitivity

As the maximum number of partitions increases, the read and write performance of LightKV increases, but the NVM capacity consumption also increase.

Sensitivity analysis



(c) Compaction size sensitivity



(d) Index placement sensitivity

As the compaction size increases, the merging frequency is reduced, and the write amplification is reduced, which is beneficial to improve the write performance, but is not conducive to reading.