Flash Memory Summit

Architecture of an Ordered KVCSD in a HPC System

Woosuk Chung,

Director, Memory Systems Research, SK hynix

35 YEARS OF NAND FLASH MEMORY INNOVATION

Legal Disclaimer

The information contained in this document is claimed as property of SK hynix. It is provided with the understanding that SK hynix assumes no liability, and the contents are provided under strict confidentiality.

This document is for general guidance on matters of interest only. Accordingly, the information herein should not be used as a substitute for consultation or any other professional advice and services.

SK hynix may have copyrights and intellectual property right. The furnishing of document and information disclosure should be strictly prohibited.

SK hynix has right to make changes to dates, product descriptions, figures, and plans referenced in this document at any time. Therefore the information herein is subject to change without notice.

Background: Scientific Simulation & Analysis

- HPC application performs scientific simulation and then performs analysis.
  - Simulation produces petabytes of data and trillion records.
  - Analysis finds meaningful data in a specific range.

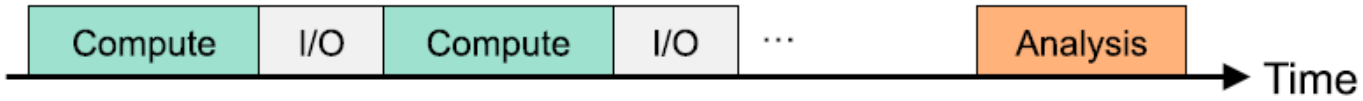
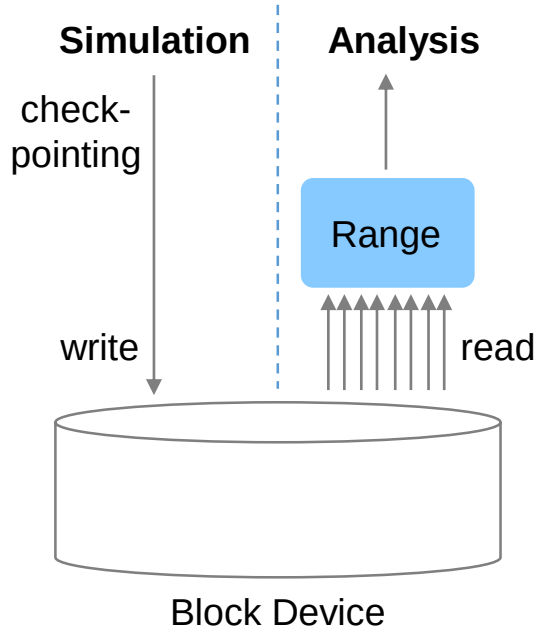


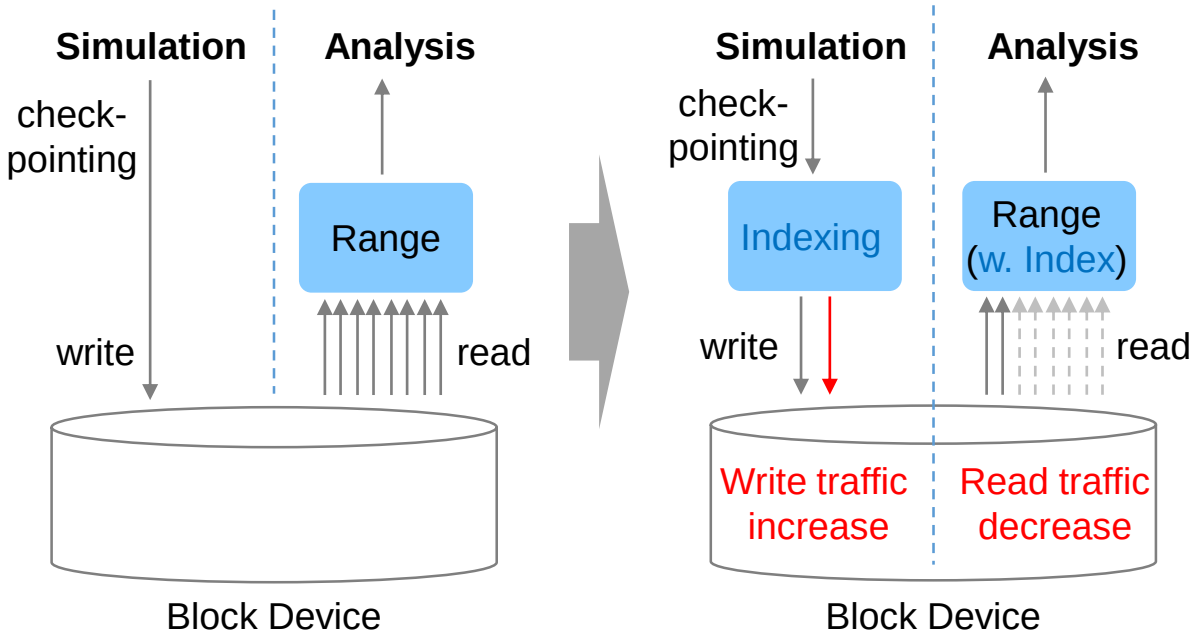
Illustration of a typical scientific workflow consisting of a bulk-synchronous parallel simulation application acting as a data writer and a subsequent data analysis program acting as a data reader with the execution of the writer further divided into iterations of non-overlapping compute and I/O phases.

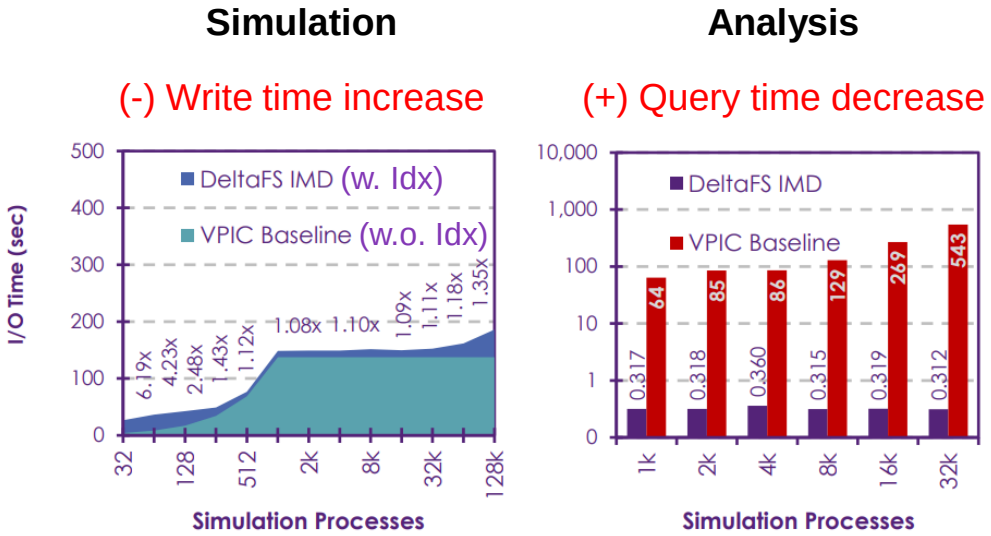
* Source: “Streaming Data Reorganization at Scale with DeltaFS Indexed Massive Directories”, ACM Trans. Storage, 2020



Background: Range Query with Index

- Filtering through a per-record index improves a range query for analysis.
- However, additional write traffic by indexing increases I/O time.
  - Simulation performance degradation.

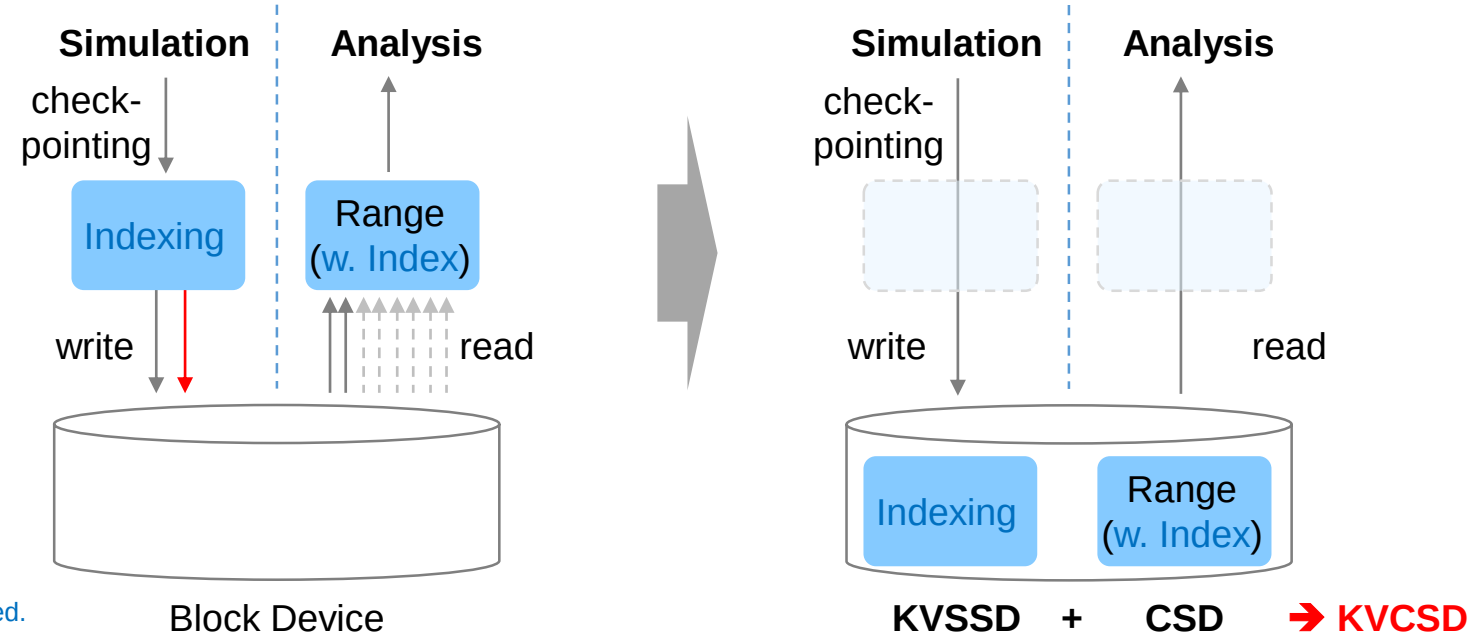




* Source: “Practical Computational Storage: Performance, Value, and Limitations”, SNIA PM+CS Summit, 2021

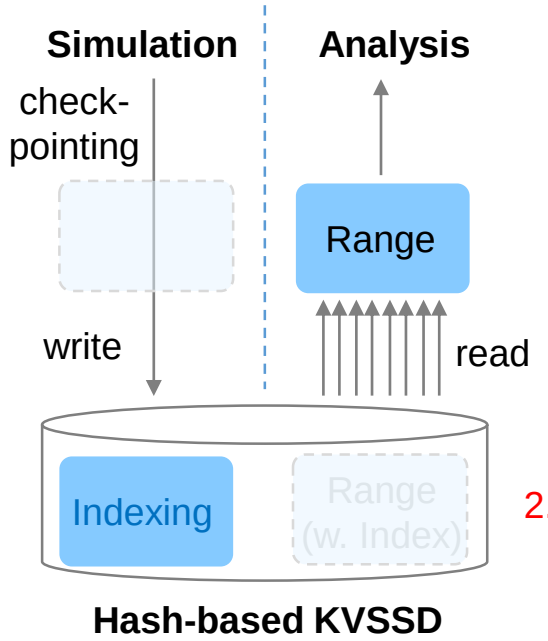
Our Goal: High Performance KVCSD for Analytics

- Supporting indexing and range query processing within the device.
  - Save host resource and increase parallelism.
  - Provide high range query performance while eliminating indexing overhead.
- We call the device Key Value Computational Storage Drive (***KVCSD)***.
  - We offload KV-granularity indexing to the device (KVSSD).
  - We also offload range query computation to the device (CSD).



Existing Approach: Hash-based Key Value SSD

- Key-Value interface is proper for offloading per-record indexing to the device.
  - File interface is an abstraction for storing a bunch of data. It does not provide an explicit way to find specific data within the file.
  - On the other hand, Key-Value interface is an effective way to expose data structure directly. In scientific simulation & analysis, particle record consists of key and value.
- With KV interface, KVSSD can support indexing, but hash-based KVSSD has two problems.

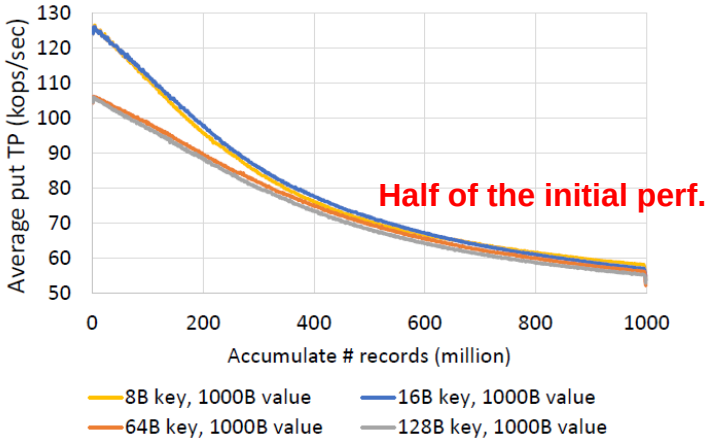


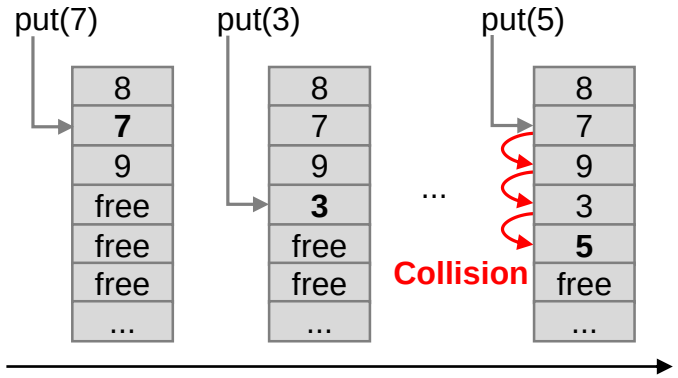
1. Aging index problem

2. Cannot support range computation

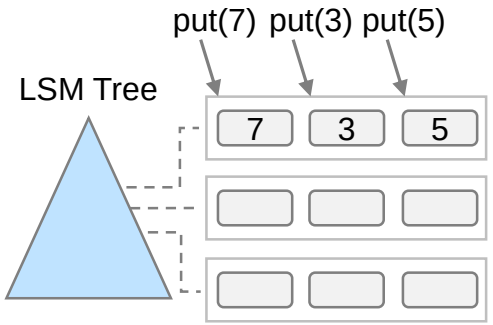
Motivation: LSM for Large Data

- Scientific simulation such as kinetic plasmas simulator (VPIC) produces petabytes of data and trillion records.
- Hash-based KVSSD cannot guarantee steady put performance.
  - The more records are inserted, the more hash collisions occur and the put performance drops.
- We adopt LSM for KVCSD, which does not incur the hash collision.





Hash-based KVSSD

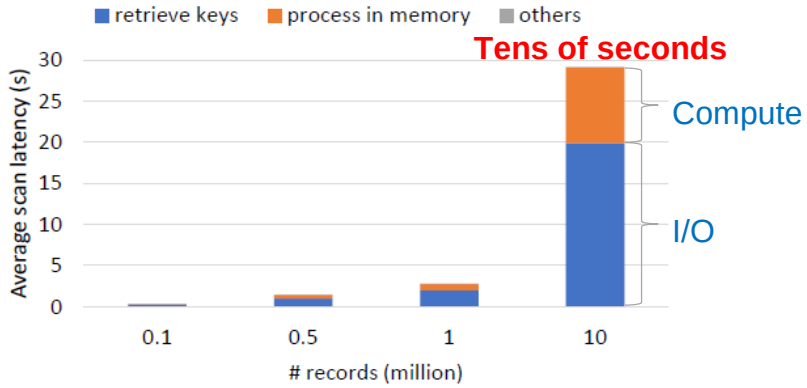


LSM-based KVCSD

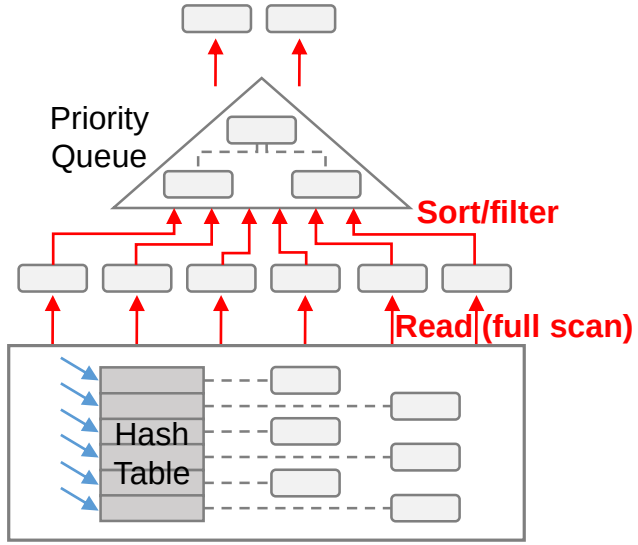
* Source: "ACCELERATING STORAGE APPLICATIONS WITH EMERGING KEY VALUE STORAGE DEVICES," Doctoral dissertation, Texas A&M University, 2021

Motivation: Ordering for Range Query

- Scientific analysis selects highly energetic particles such as “energy > 1.5”.
- Hash-based KVSSD cannot process a range query.
  - Host has to process the query. It incurs a lot of unnecessary I/O (read) and requires host computing resource (sort/filter).
- We fully exploit ordered data structure (LSM), which does not rely on the host for range query.

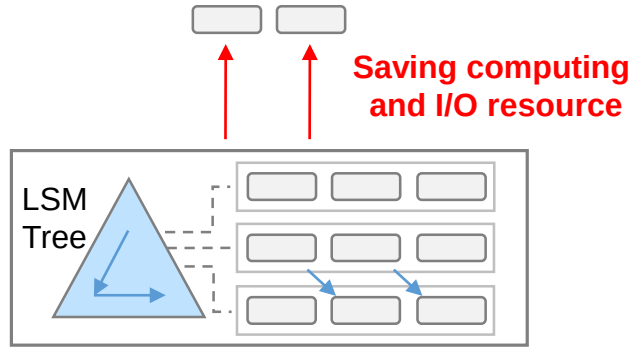


* Source: "ACCELERATING STORAGE APPLICATIONS WITH EMERGING KEY VALUE STORAGE DEVICES," Doctoral dissertation, Texas A&M University, 2021



Hash-based KVSSD

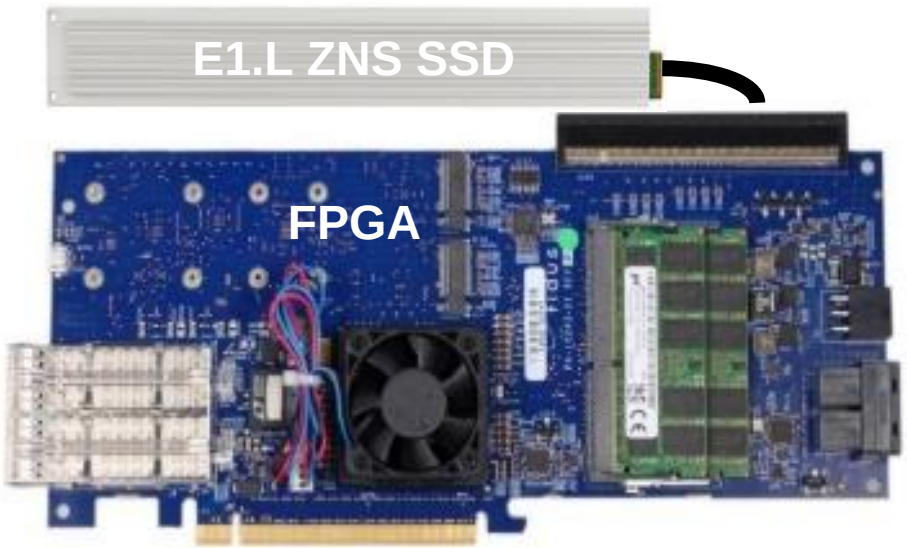
With keys ordered, LSM-based KVCSD can quickly retrieve keys in a range.



LSM-based KVCSD

Overview: SKH KVCSD Features

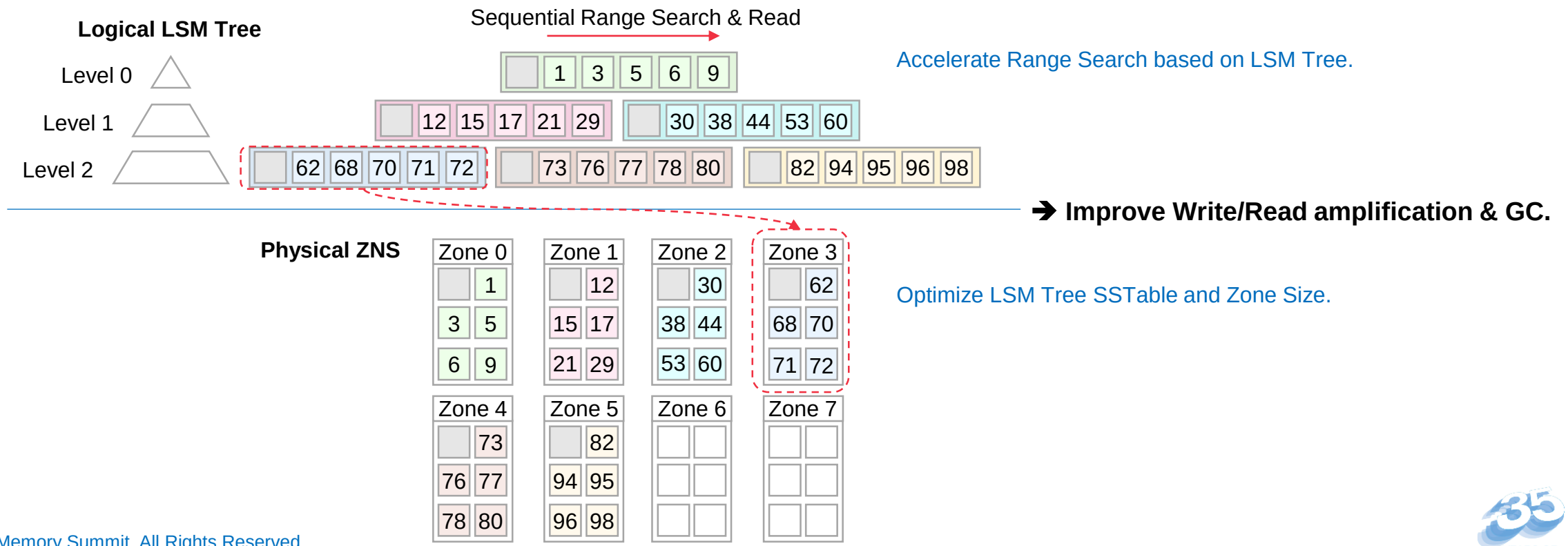
<KVCSD Prototype>



1. **Key-Value Interface**
   - Eliminate duplicate layer and fine-grained management.
2. **Analytics Query Acceleration**
   - Support Range Query, 2nd Index, and Histogram.
3. **LSM Data Structure**
   - Offload sorting data and building indexes into KVCSD.
4. **ZNS Optimization**
   - Improve Write/Read amplification and remove GC.

Optimization: LSM Data Structure + Zone

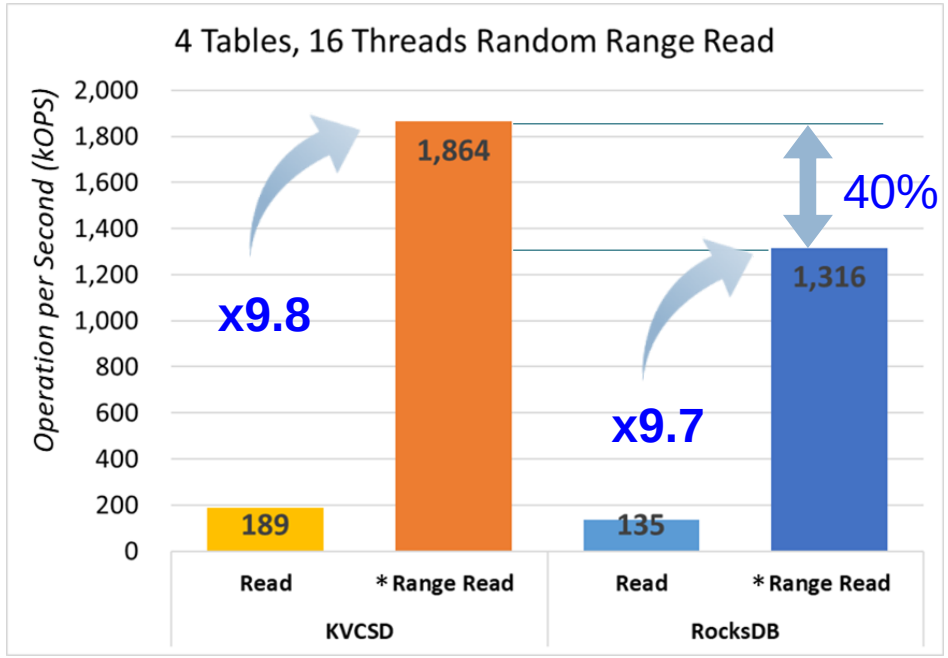
- Optimized LSM data structure and ZNS architecture.
  - KVCSD achieves optimized I/O and search performance through storing sorted data by zone unit.
  - Using ZNS, sorted data can be stored in one physical area, Zone



Evaluation: Benefits of Range Query

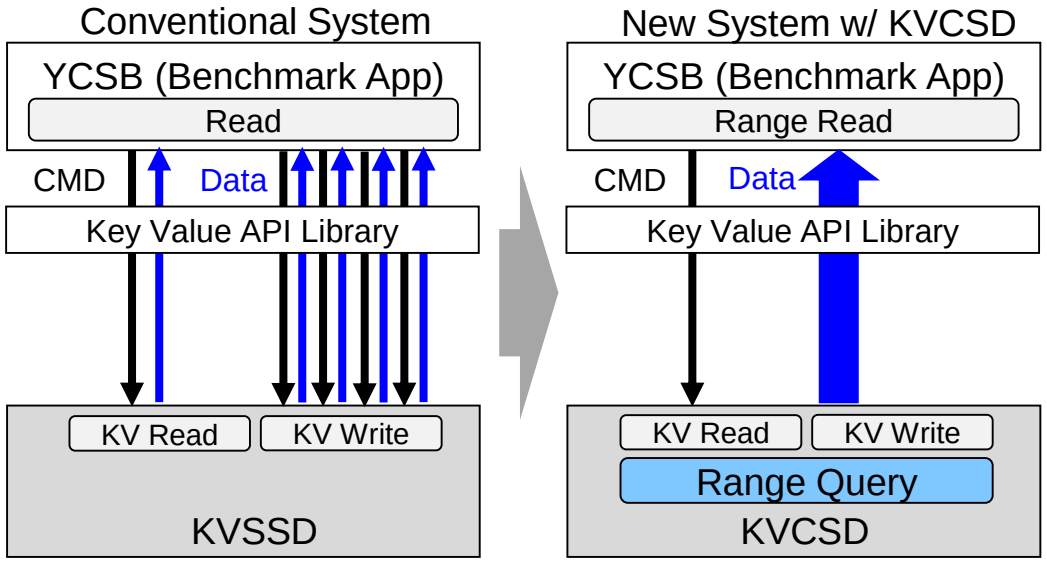
- KVCSD can increase 10X higher search performance by supporting “Range Query”.
  - About 40% performance improvement compared to host database, RocksDB.

Evaluation Result – Range Query



Results Analysis

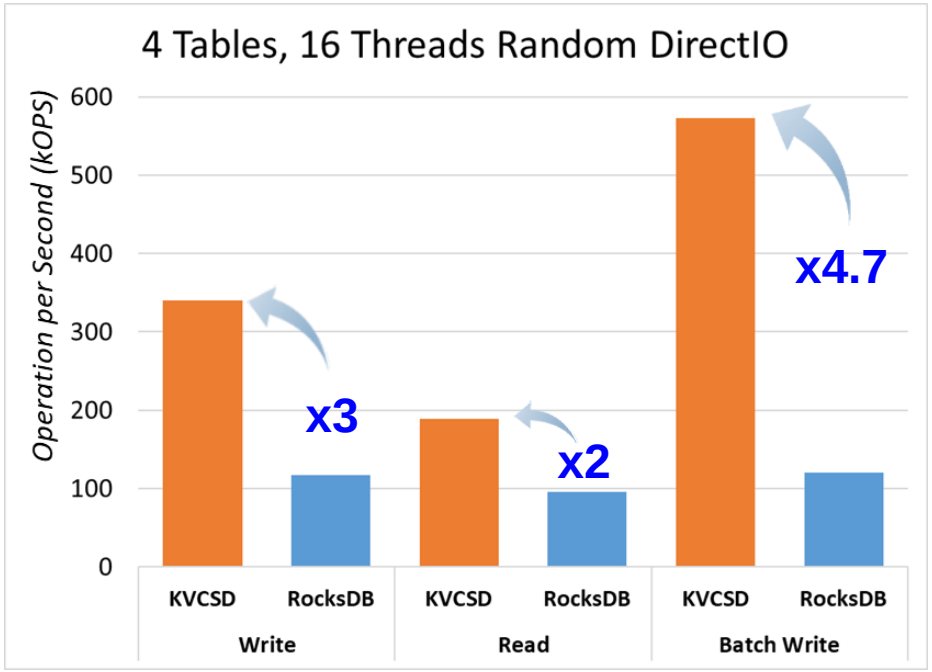
- Range Read command can read every data that satisfies a specific range condition by single command.



Evaluation: I/O Performance

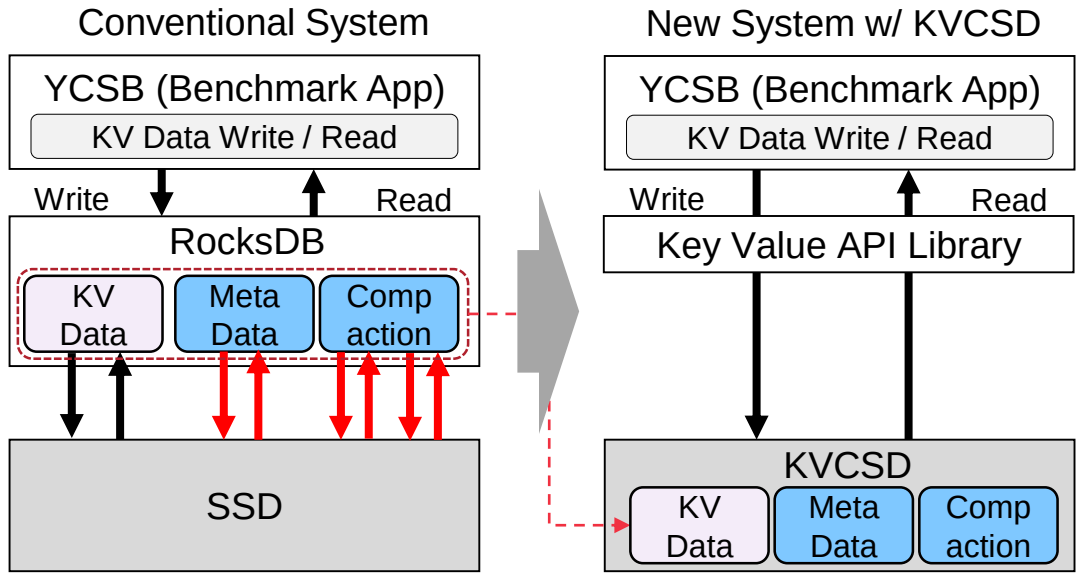
- KVCSD shows higher I/O performance than RocksDB.
  - I/O Improvement due to WAF & RAF reduction: Write: x3, read: x2, Bulk Store: x4.7

Evaluation Result – I/O Performance



Results Analysis

- KVCSD manages metadata about data location internally and performs compaction on its own. → Reduce number of I/O and increase performance.



Summary

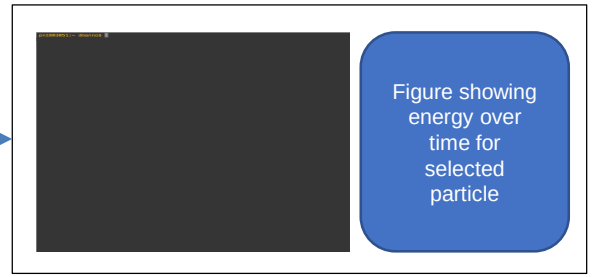
- Scientific simulation and analysis
  - An index accelerates a range query for analysis, but host-side indexing degrades write performance for simulation.
  - KVSSD can be a candidate for solving the host-side indexing problem. However, hash-based KVSSD has an aging problem and it cannot accelerate the range query.
- Ordered KVCSD
  - Offload indexing and range query processing to KVCSD.
  - Adopt ordered data structure and offer ZNS optimization.
  - Support fast range query without indexing overhead and aging problem.

Co-demonstration with Los Alamos National Lab

Video Demo
(SKH & LANL Video)

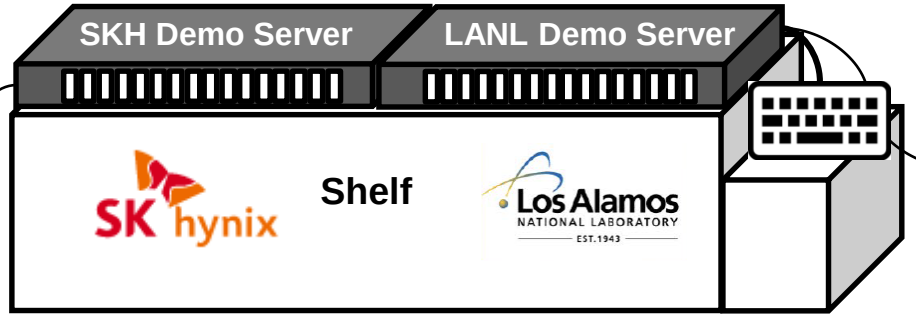
SKH Live Demo

LANL Live Demo



SKH Benchmark Application

SKH KVCSDs



LANL VPIC Application

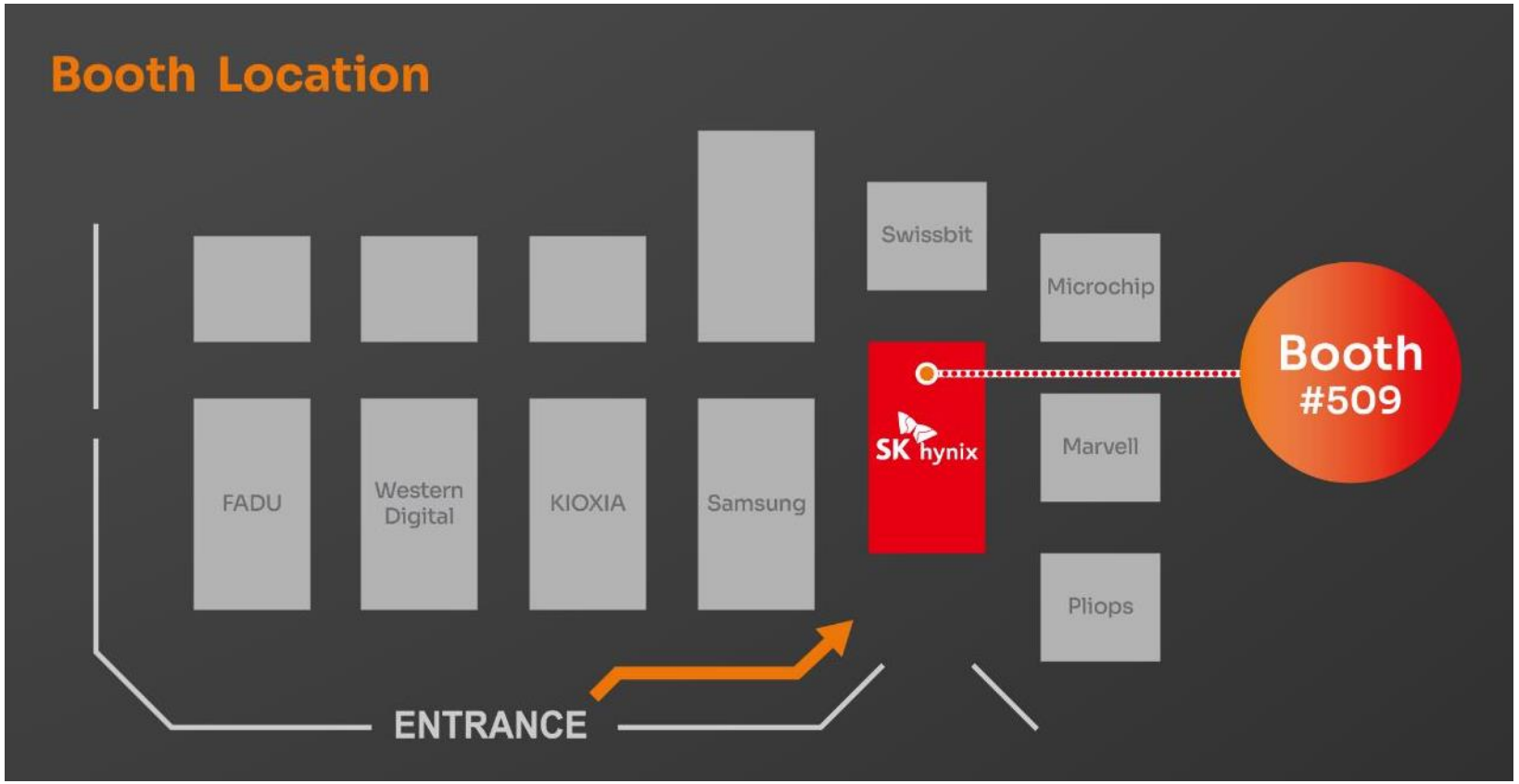
SKH KVCSDs

SKH Demo shows read and range query performance according to time.

In video demo, SKH describes KVCSD main features and its effect. LANL introduces VPIC and describes the need for KVCSD and its effect.

LANL Demo shows particle information and search latency.

Learn more about SK hynix



Visit Booth #509 and Experience SK hynix products and demos & get a free giveaway!

