HPC Driven Motivations for Ordered Key-Value Based Computational Storage

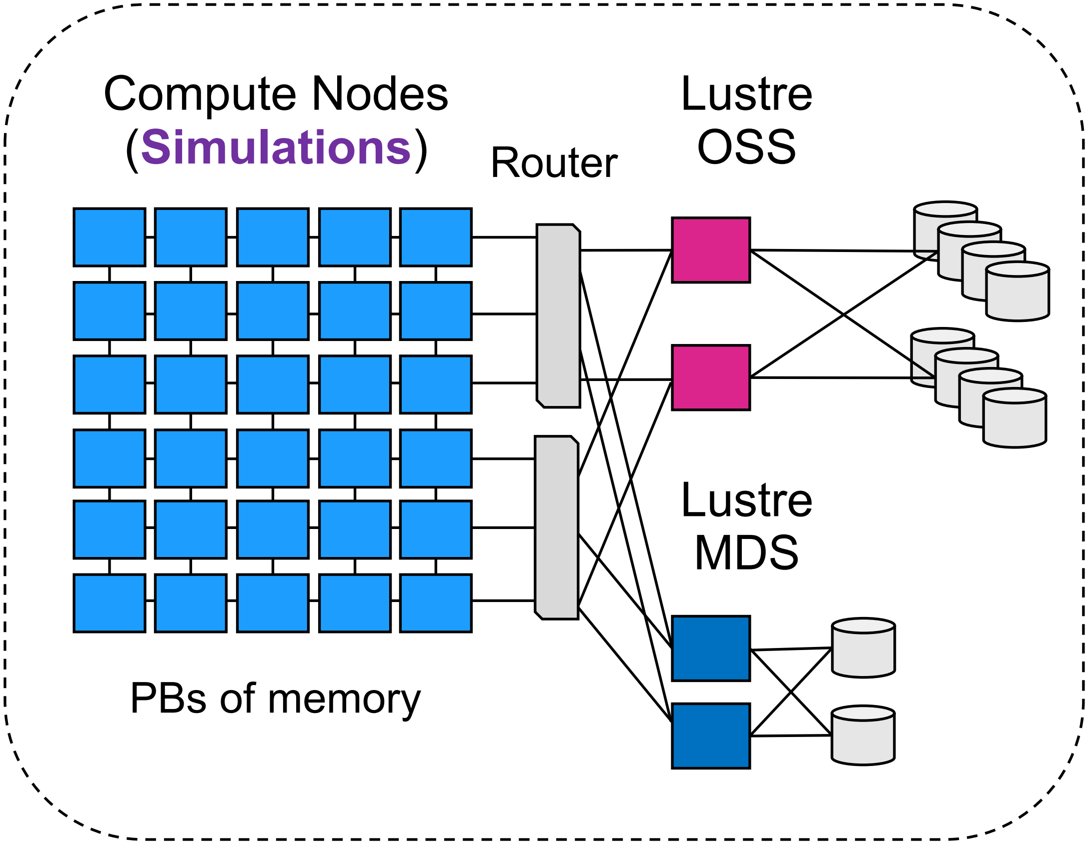
Qing Zheng

Scientist, Los Alamos National Laboratory (LANL)

LA-UR-22-27931

Typical HPC Simulation Workflow at LANL

- Simulation writes state to storage periodically
- Analysis code later reads data back for in-mem operations (e.g.: movie making)
- Data may not compress
- Performance depends on fully utilizing available storage bandwidth



Current HPC Platform

Emerging Trends: Analysis Increasingly Selective

- Analysis used to require seeing all data records
- Today: queries tend only to hit a small subset of data
- Problem: how to retrieve just interesting rows?

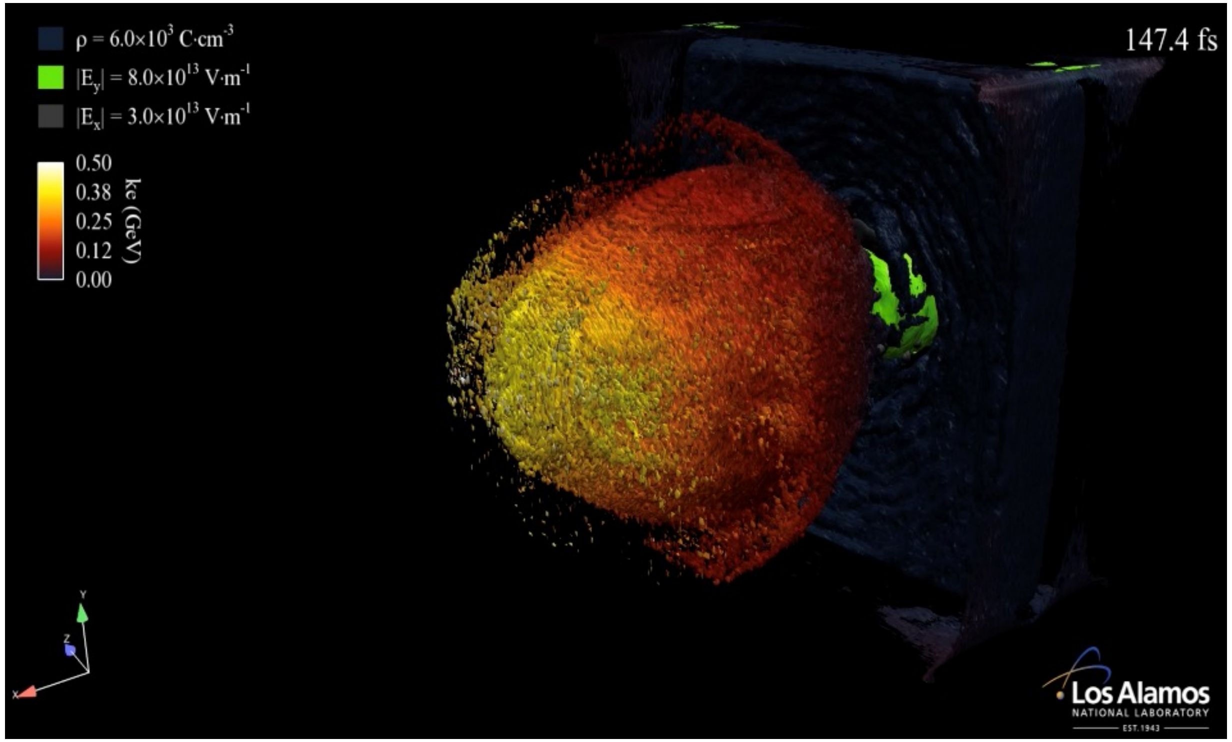

$\rho = 6.0\times10^{3}$ C·cm^{-3}
$|E_y| = 8.0\times10^{13}$ V·m^{-1}
$|E_x| = 3.0\times10^{13}$ V·m^{-1}
0.50
0.38
0.25
0.12
0.00
ke (GeV)
147.4 fs
Y
Z
X
Los Alamos NATIONAL LABORATORY EST.1943

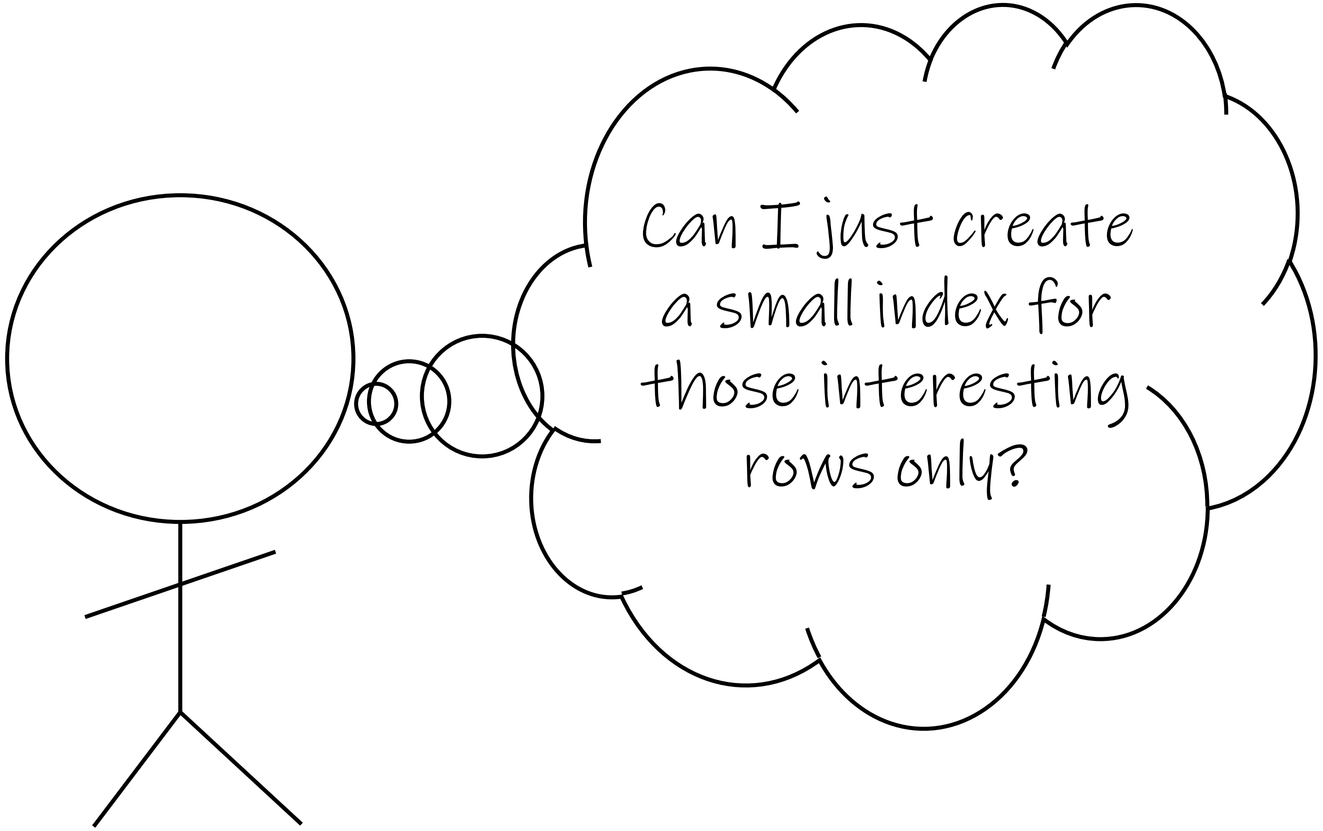
Image from LANL VPIC simulation done by L. Yin, et al at SC10

Example: SELECT X, Y, Z FROM particles **WHERE** E >= 1.5

Less than **0.1%** needs to be read from storage

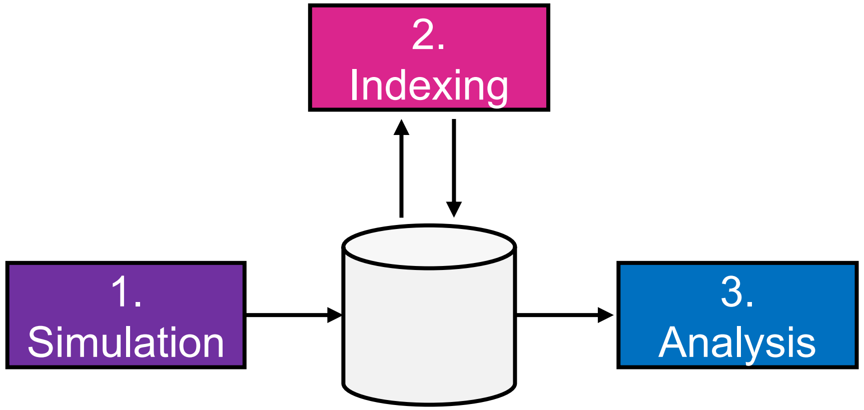
Reading Back Just Interesting Data is Non-Trivial

- Data known to be interesting only at simulation end
- Indexing only works when all rows are indexed at all timesteps
- Compute node resources are limited
- Sorting only helps one query



Existing Solutions Fall Short in Different Ways

Post-processing



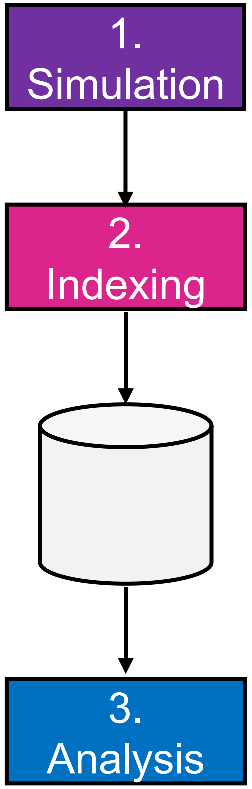
1. Simulation

2. Indexing

3. Analysis

Excessive data movement

In-transit processing



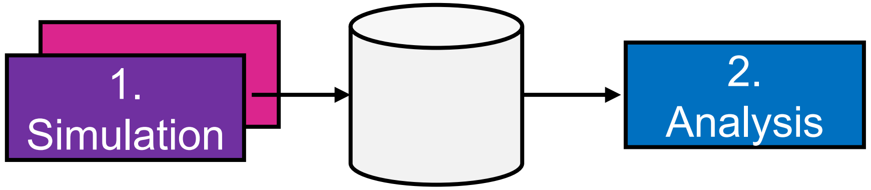
1. Simulation

2. Indexing

3. Analysis

Requires additional compute nodes than the job
Does not work for larger jobs

In-situ processing



1. Simulation

2. Analysis

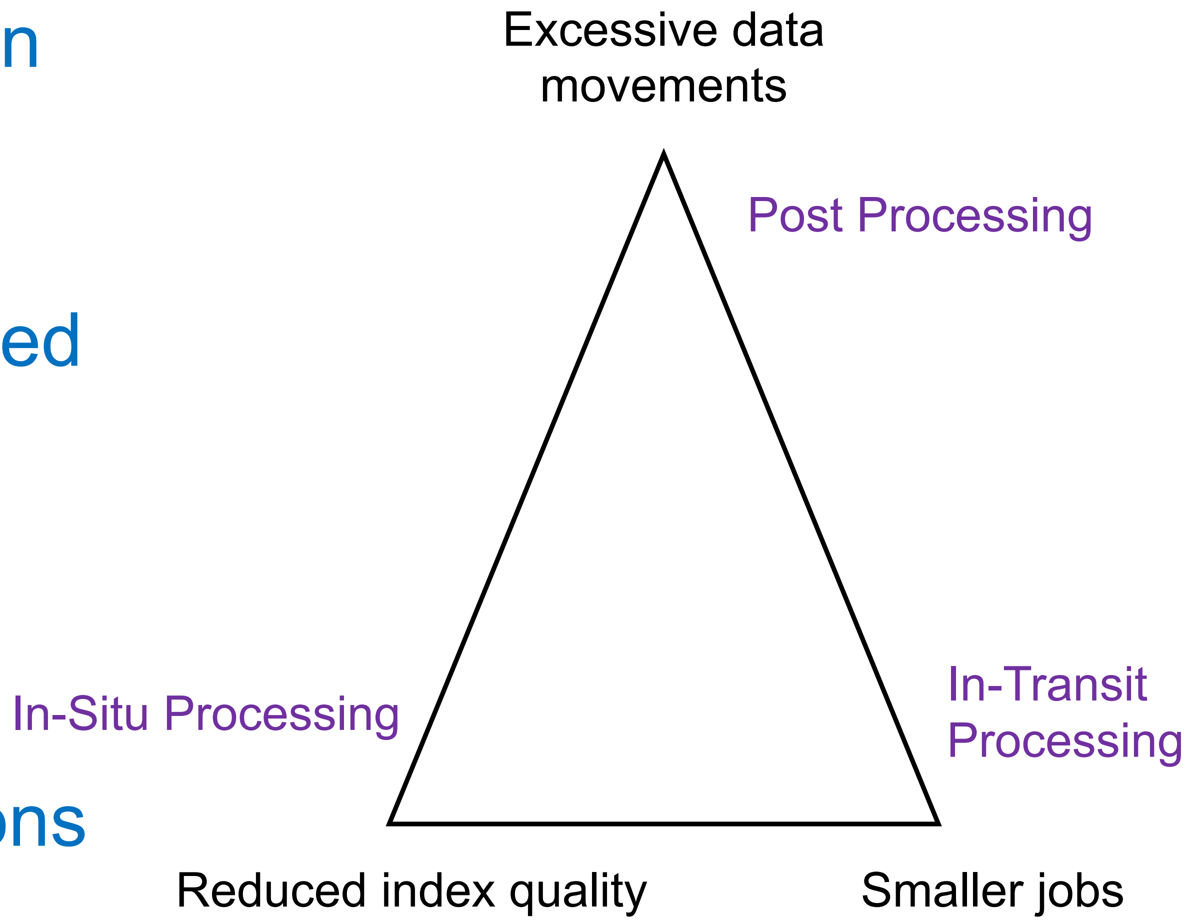
May only produce indexes
on 1 or few columns

Opportunities for Rapid Query Acceleration

- Today: all computation takes place on compute nodes

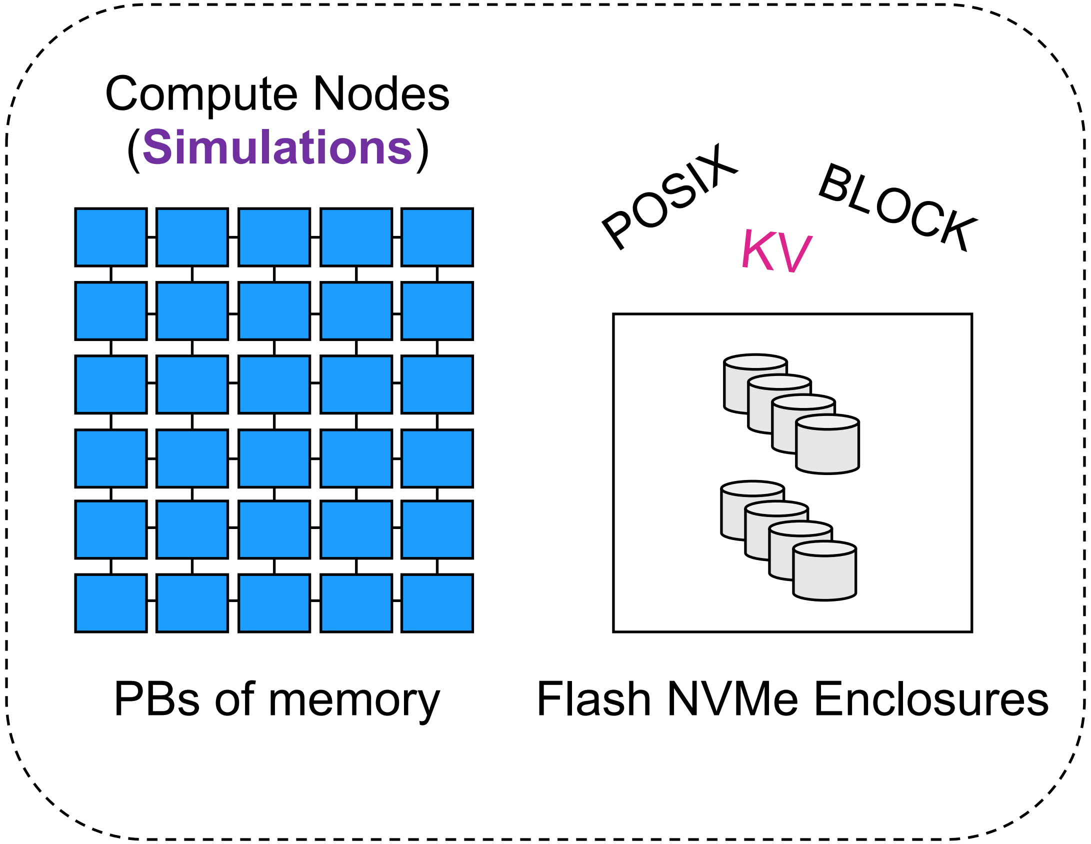
- Excessive data movements or reduced index quality or increased per-job resource footprint

- Computational storage allows for overcoming existing solution limitations



Towards KV-Based Storage Spaces for HPC

- KV namespaces in addition to POSIX and block for accelerated data indexing & analytics
- No one-size-fits-all: app chooses the best abstraction for the job at hand
- Dynamic platform: portions of KV change over time



Next-Gen HPC Platform

HPC-Driven KV Storage API

- Data insertion:
  - Bulk KV put operations
- Reads:
  - Range queries
  - Secondary indexes
  - Histogram construction
- Management:
  - Compaction control
  - Per key space data export

SK hynix | PAVILION

Los Alamos National Laboratory | Los Alamos National Laboratory

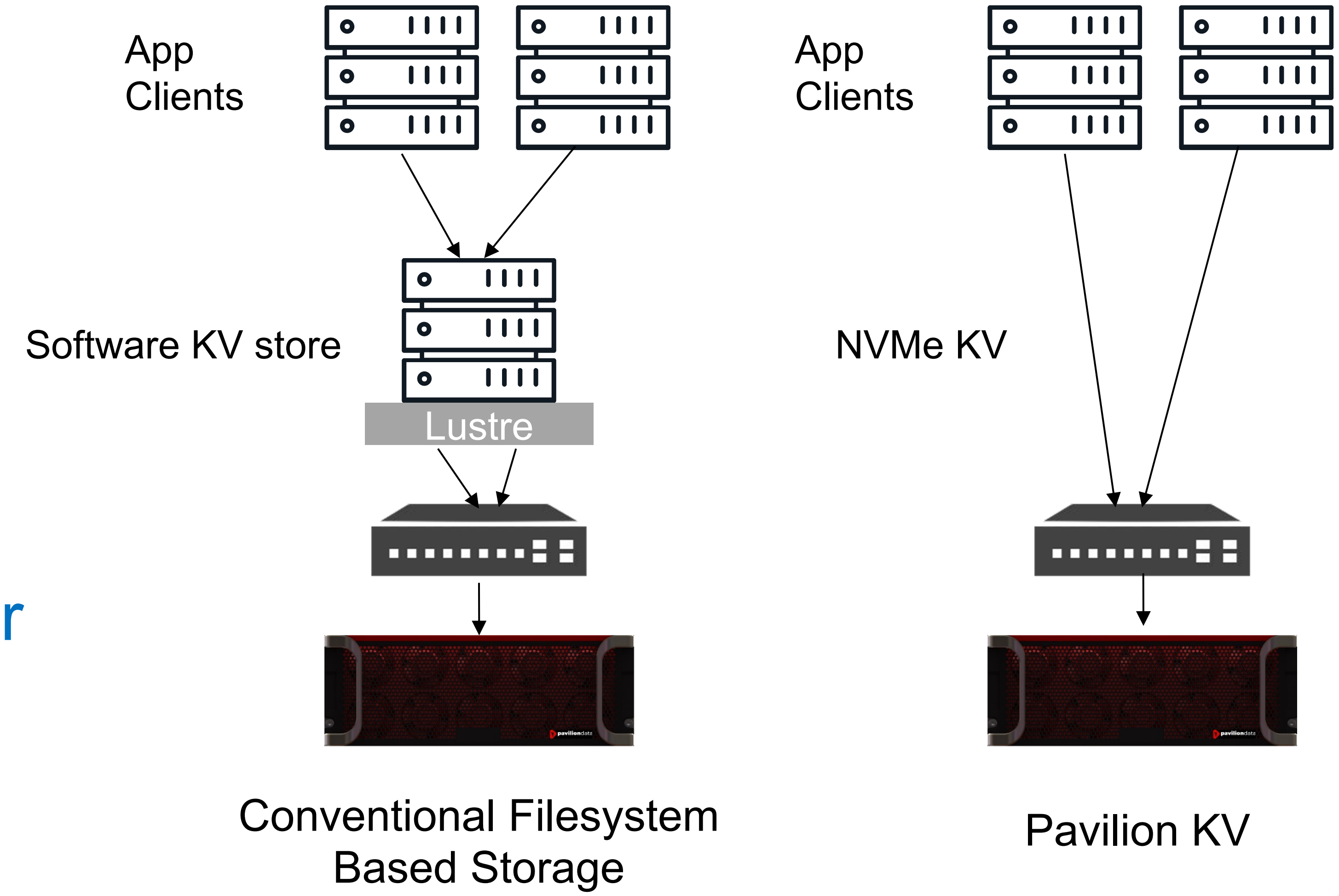
LANL is collaborating with industry for accelerated KV storage that speeds up scientific discovery

Pavilion Next-Gen KV Storage

- Server-based accelerated KV storage

- Access via NVMeOF

- Orders of magnitude faster than software KV



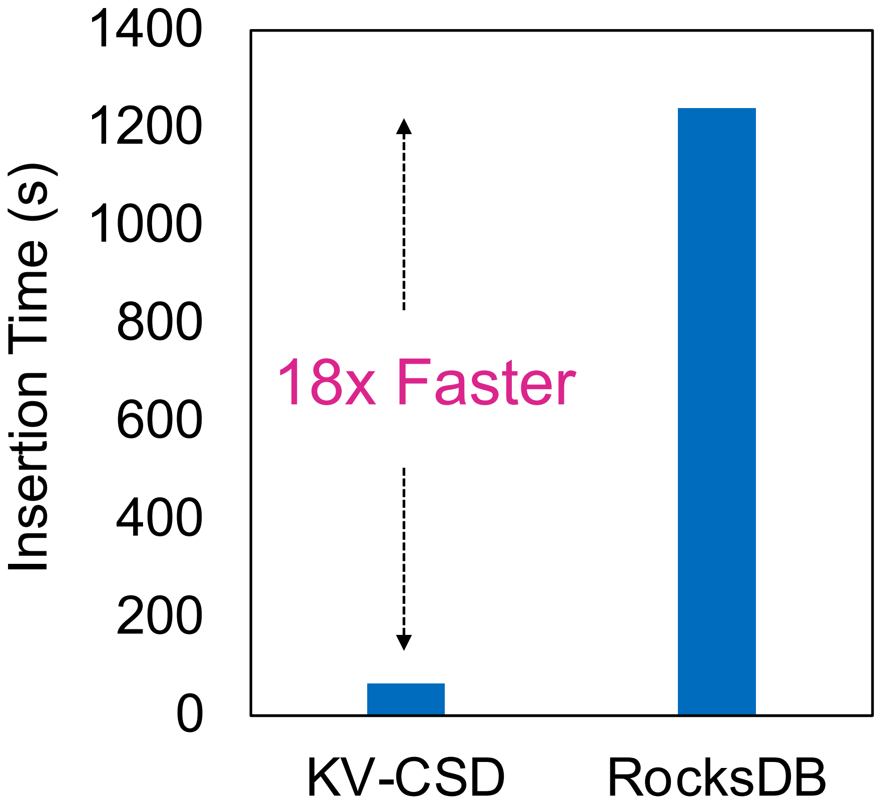
SK Hynix KV-CSD Prototype

- FPGA-based, hardware accelerated KV SSD
- Access via local PCIe
- ZNS storage for increased performance and longer SSD life span

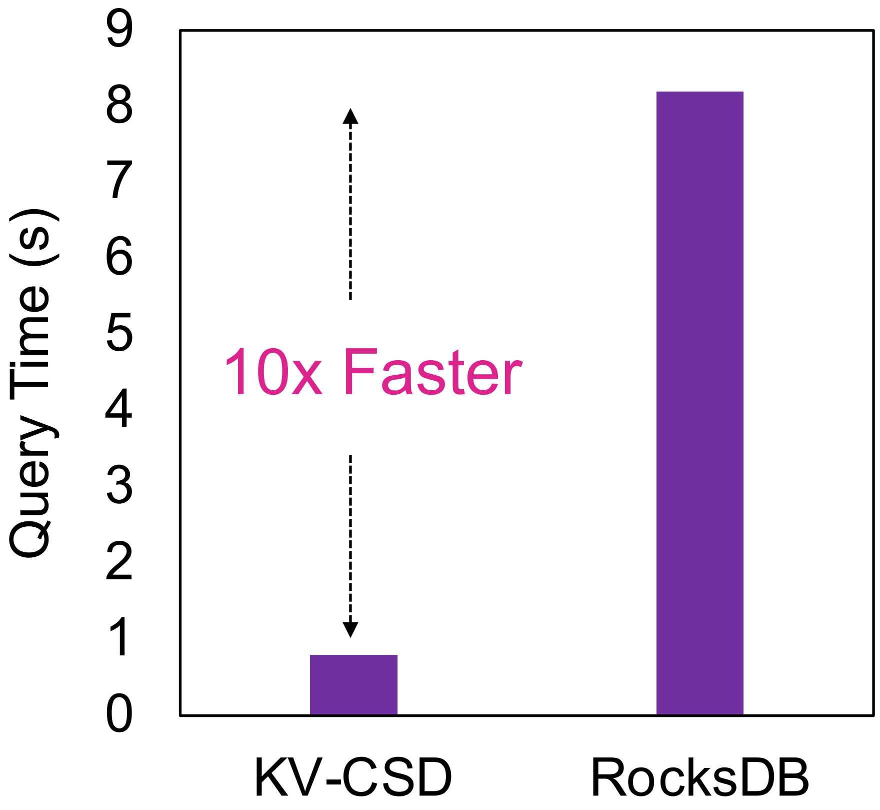
More info: SARC-302-1: Computational Storage Solutions

1:25pm Ballroom G

Preliminary Results: SK KV-CSD vs RocksDB



Data Insertion: Up to 18x faster

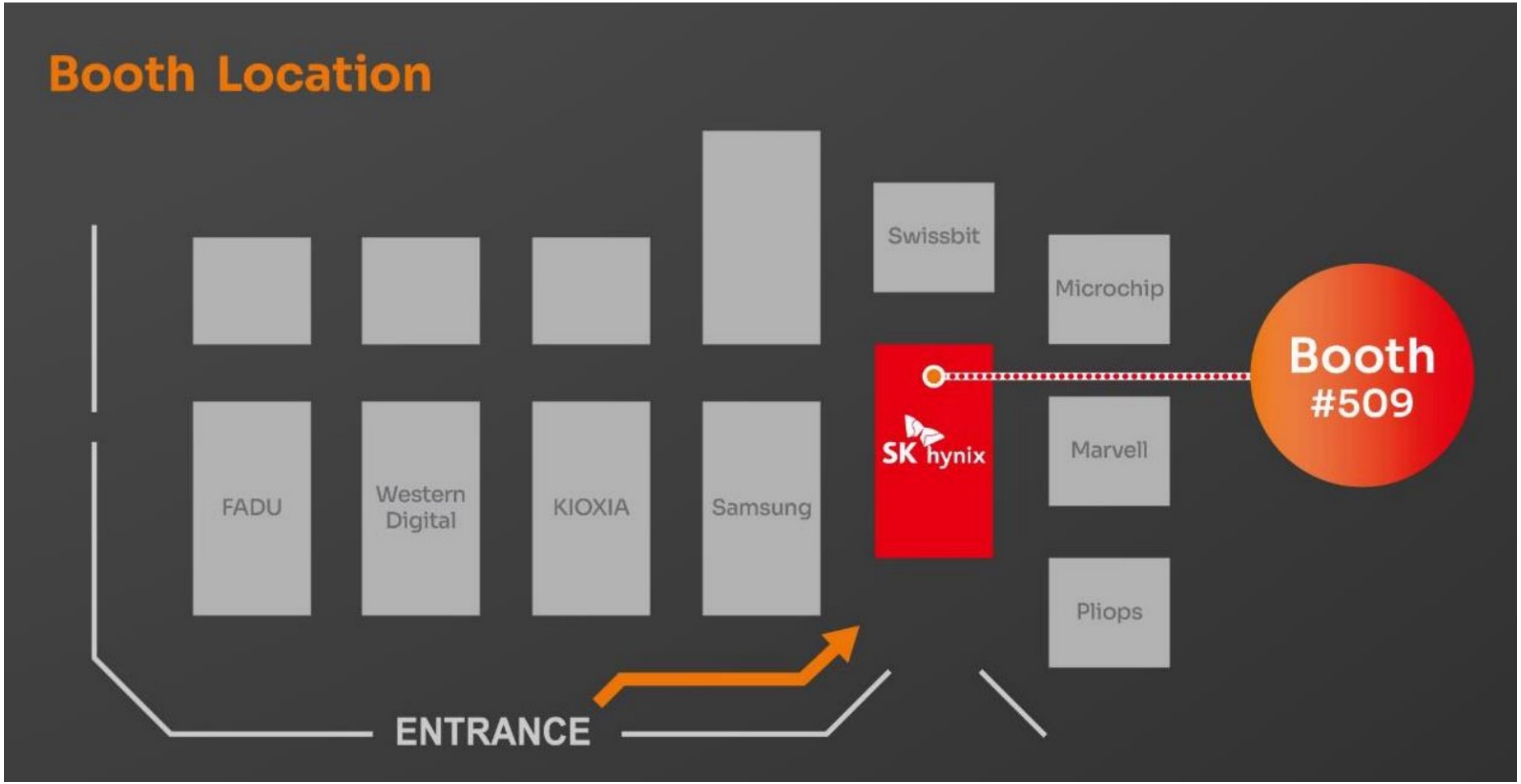


Queries: Up to 10x faster

Conclusion

- Massively-parallel computing and full bandwidth utilization will continue to matter
- But efficiently handling massive amounts of small objects and highly selective queries will be as critical going forward
- Implications: more diverse storage abstractions, more extensive processing offloading

Co-Demonstration with SK Hynix



See you there!