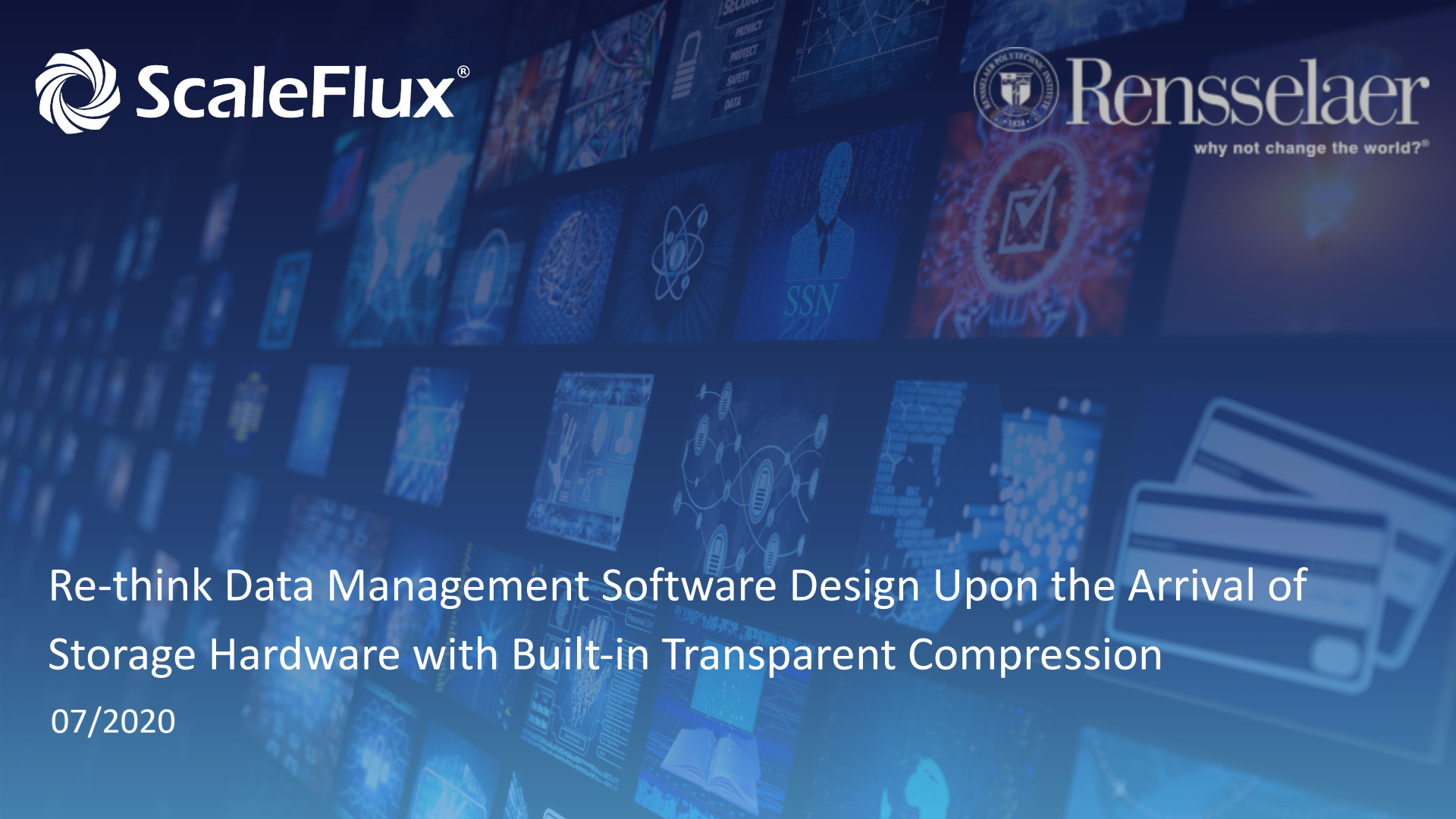
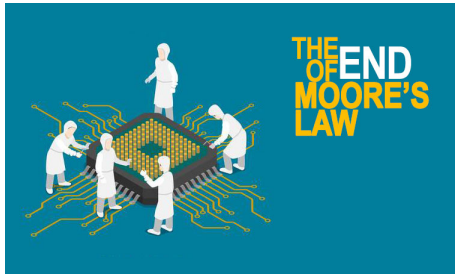
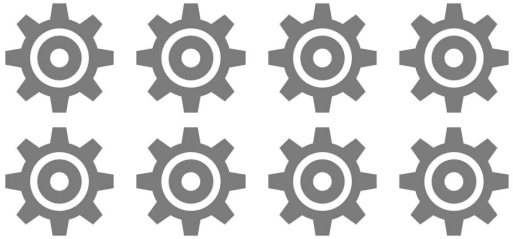


Re-think Data Management Software Design Upon the Arrival of Storage Hardware with Built-in Transparent Compression

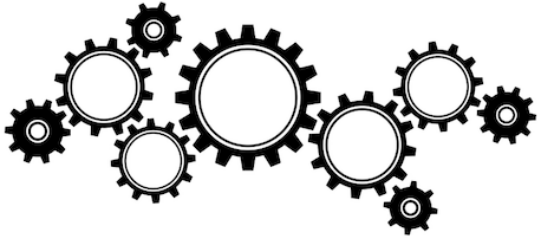
07/2020

The Rise of Computational Storage





Homogeneous Computing



Heterogenous Computing

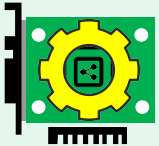
Compute

FPGA/GPU/TPU

End of Moore's Law

Networking

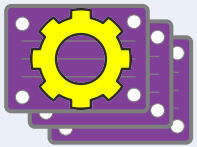
SmartNICs



10 → 100-400Gb/s

Storage

Computational Storage

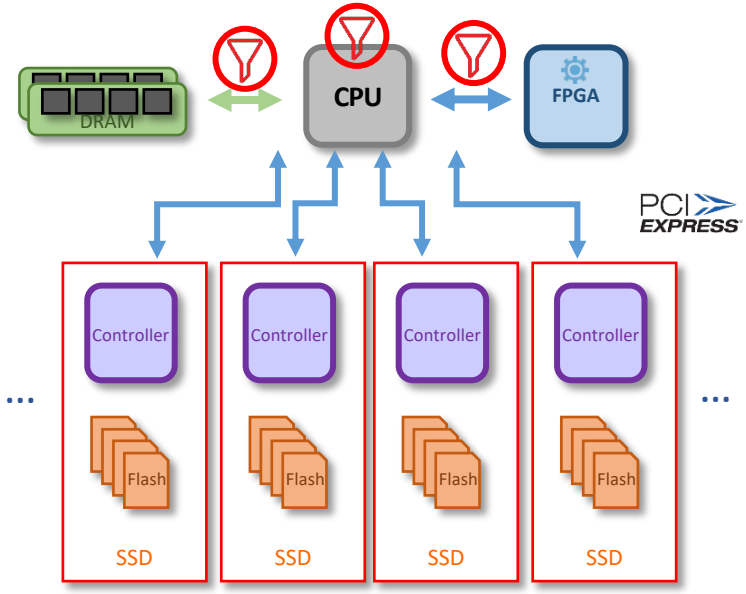


Fast & Big Data Growth

Domain Specific Compute

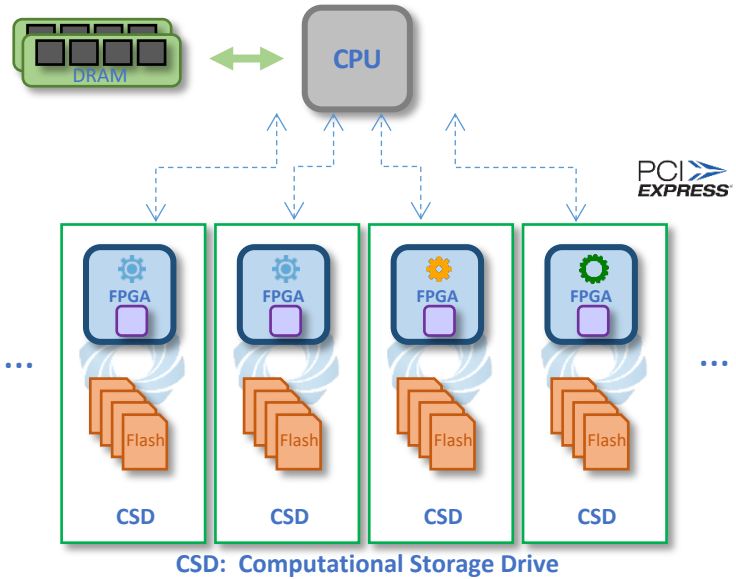
The Rise of Computational Storage

from Processor-Driven…



- CPU & Memory I/O bottlenecks
- Limited FPGAs, specific sockets required
- Massive data movement
- **No compute parallelism**

to Data-Driven



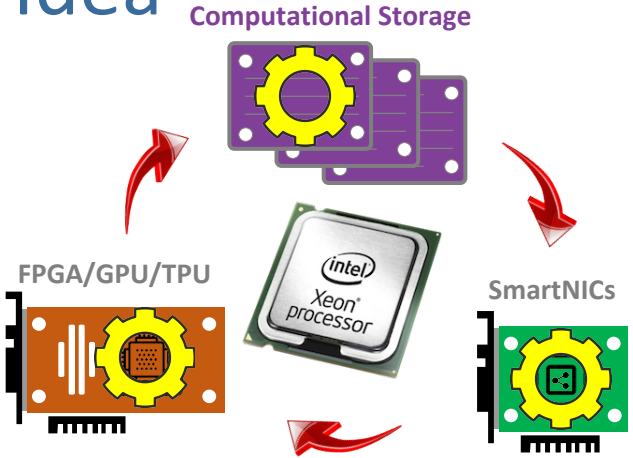
CSD: Computational Storage Drive

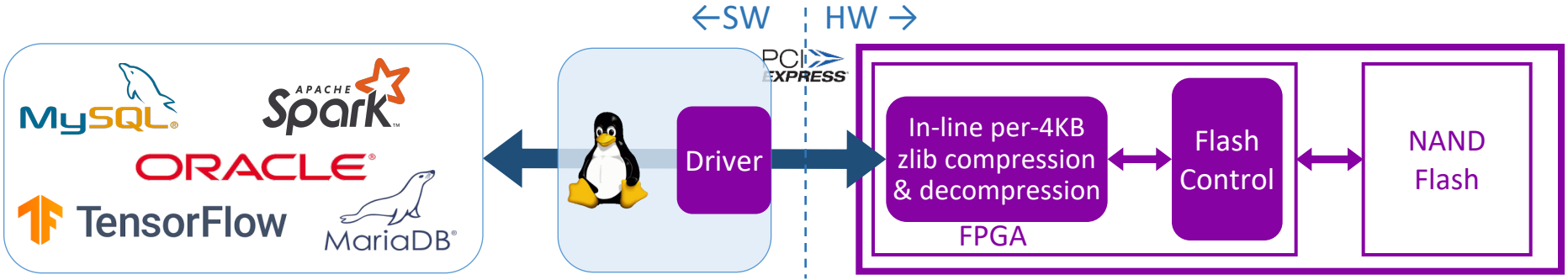
- Balanced compute & storage I/O
- Multiple FPGAs, easily plug-in via storage
- Minimize data movement
- **Maximum compute parallelism**

Computational Storage: A Simple Idea

- End of Moore's Law ➔ heterogeneous computing

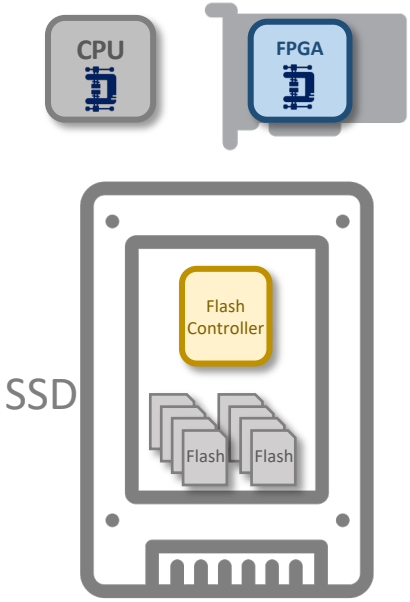
Low-hanging fruits



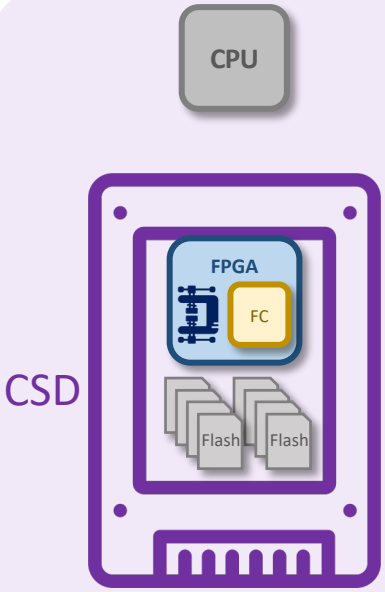


Computational Storage Drive (CSD) with Data Path Transparent Compression

ScaleFlux Computational Storage Drive



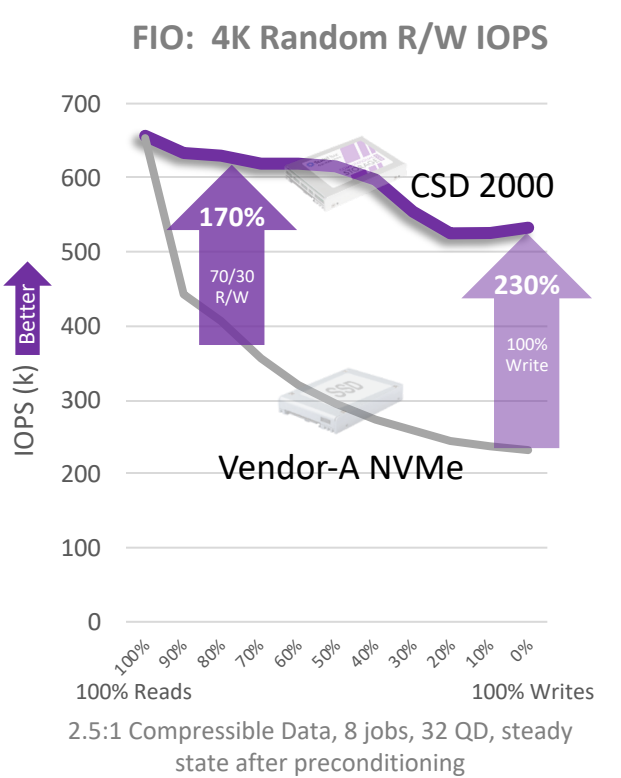
Multiple, discrete components for Compute and SSD Functions



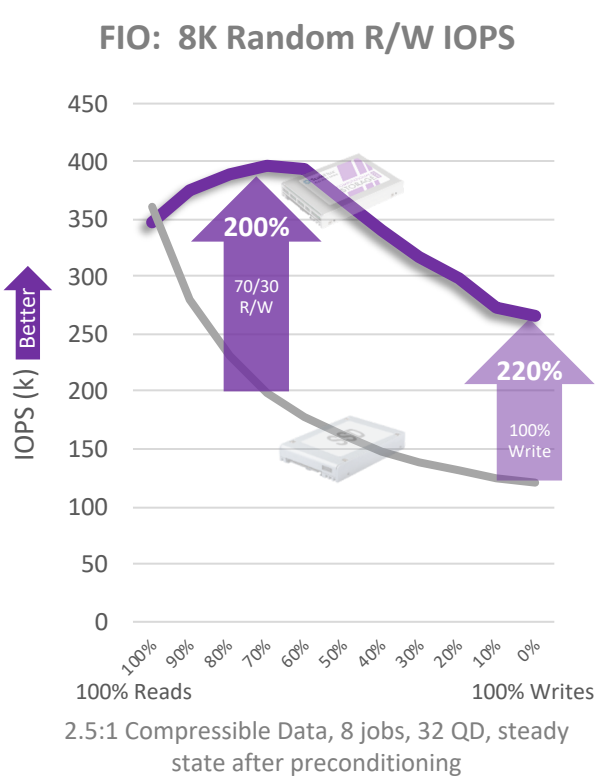
Single FPGA combines Compute and SSD Functions

- ✓ Complete, validated solution
  - ✓ Pre-Programmed FPGA
  - ✓ Hardware
  - ✓ Software
  - ✓ Firmware
- ✓ No FPGA knowledge or coding
- ✓ Field upgradeable
- ✓ Standard U.2 & AIC form factors

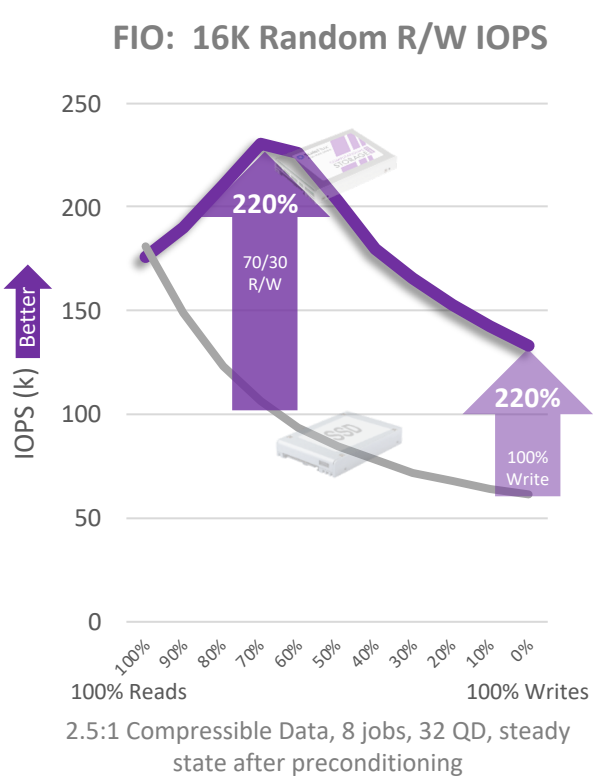
CSD 2000: Data Path Compression/Decompression



2.5:1 Compressible Data, 8 jobs, 32 QD, steady state after preconditioning



2.5:1 Compressible Data, 8 jobs, 32 QD, steady state after preconditioning



2.5:1 Compressible Data, 8 jobs, 32 QD, steady state after preconditioning

Increasing Mix R/W Performance Advantage with Larger Block Sizes

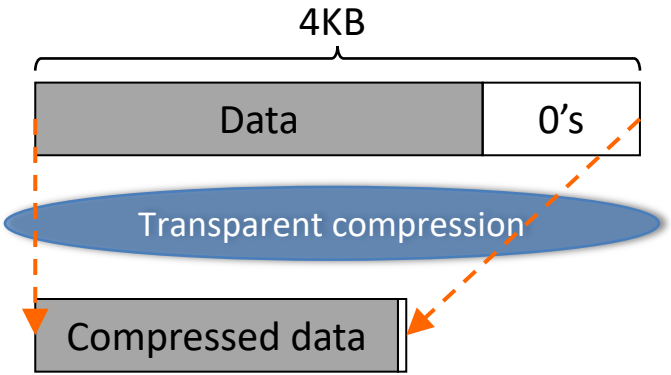
Open a Door for System-level Innovations

Logical storage space utilization efficiency

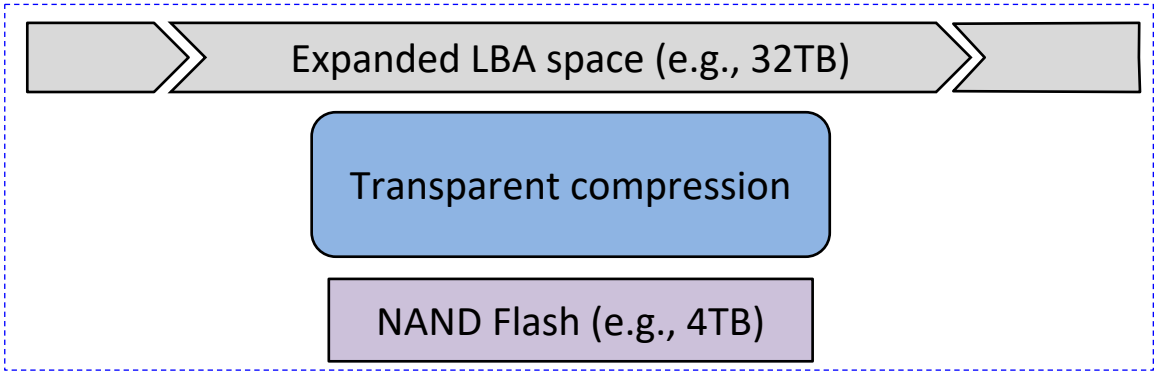
Data path compression

Physical storage space utilization efficiency

OS/Applications can ***purposely waste*** logical storage space to gain performance benefits

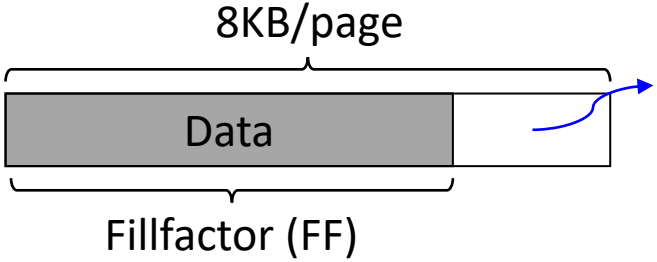


Unnecessary to completely fill each 4KB sector with user data



Unnecessary to use all the LBAs

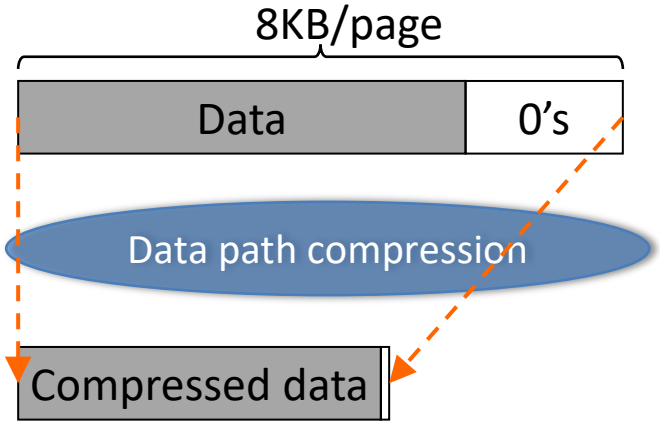
PostgreSQL

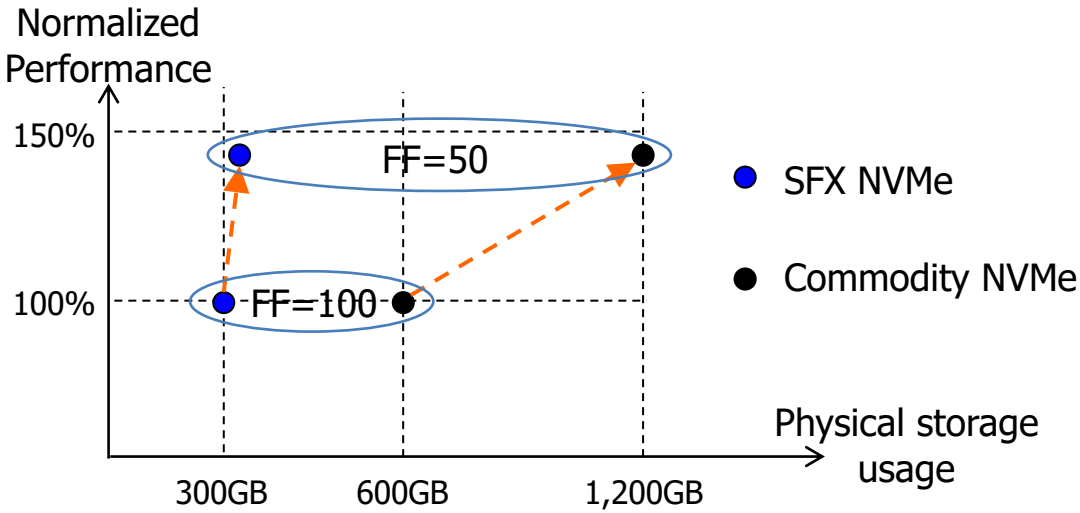


Reserved for future update

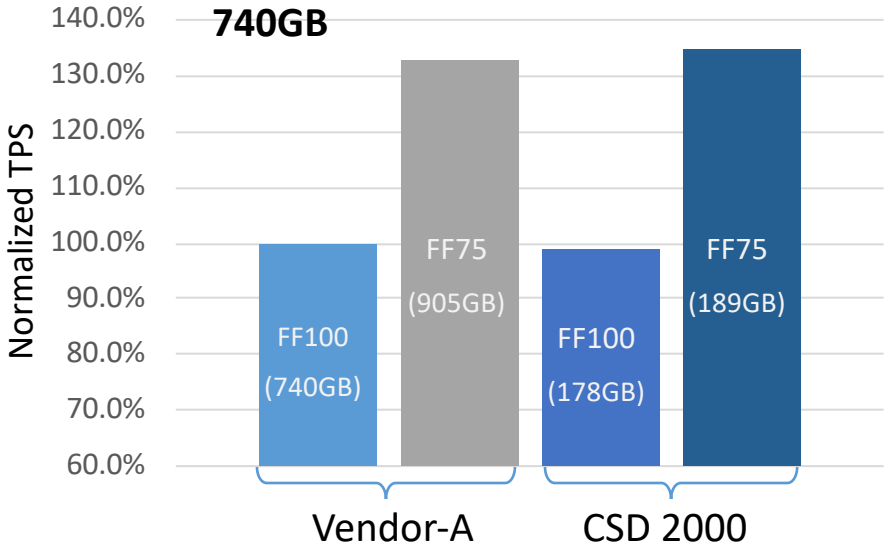
FF ↓ → Performance ↑

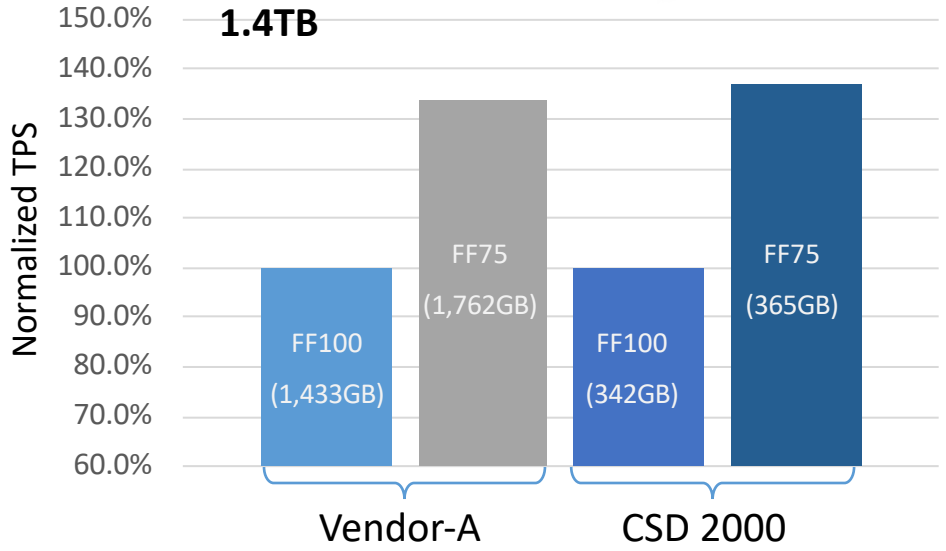
FF ↓ → Storage space ↑





PostgreSQL (Sysbench-TPCC)

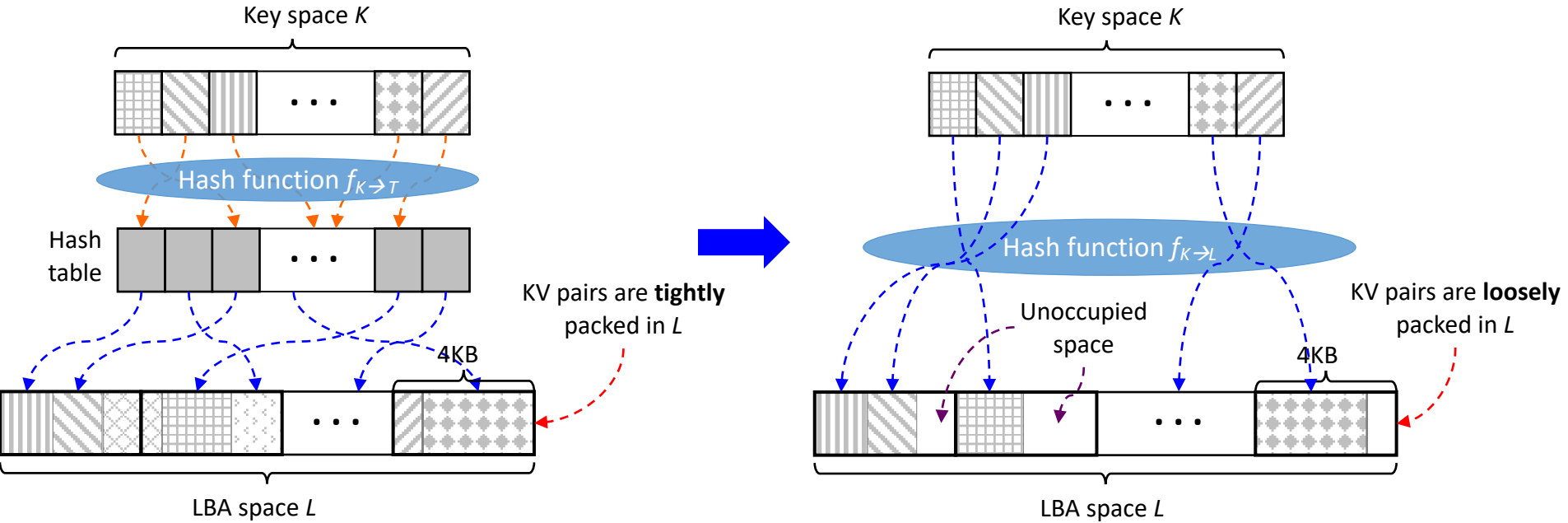




Fillfactor	Drive	Logical size (GB)	Physical size (GB)	Comp Ratio
100	Vendor-A	**740**	740	1.00
	CSD 2000		**178**	4.12
75	Vendor-A	**905**	905	1.00
	CSD 2000		**189**	4.75

Fillfactor	Drive	Logical size (GB)	Physical size (GB)	Comp Ratio
100	Vendor-A	**1,433**	1,433	1.00
	CSD 2000		**342**	4.19
75	Vendor-A	**1,762**	1,762	1.00
	CSD 2000		**365**	4.82

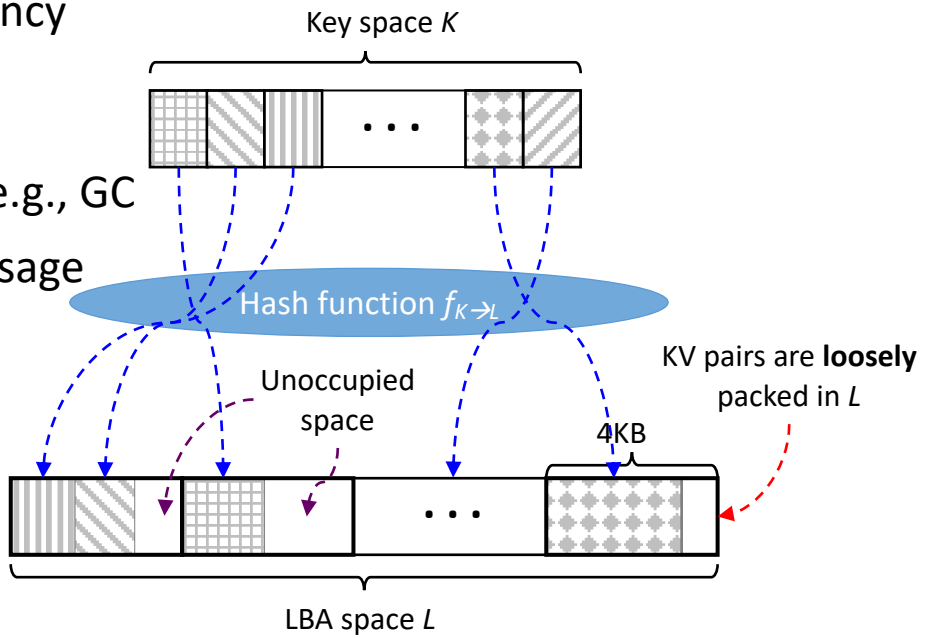
Table-less Hash-based Key-Value Store



KV store *purposely* **under-utilizes** logical storage space to **eliminate hash table** without sacrificing physical storage utilization

Table-less Hash-based Key-Value Store

- ✓ Simple code base & high operational concurrency
- ✓ Very small memory footprint
- ✓ Absence of frequent background operations (e.g., GC and compaction) ➔ low and consistent CPU usage



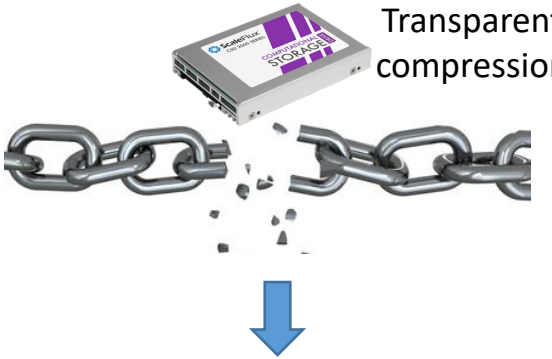
Compared with RocksDB

- ✧ >2x ops/s improvement
- ✧ >2x less average CPU usage

We will open source and are looking for collaborations to together grow the community!

Summary

Logical storage space utilization efficiency



Physical storage space utilization efficiency

Unique opportunities to re-think the data management software design

www.scaleflux.com

tong.zhang@scaleflux.com

ScaleFlux™