

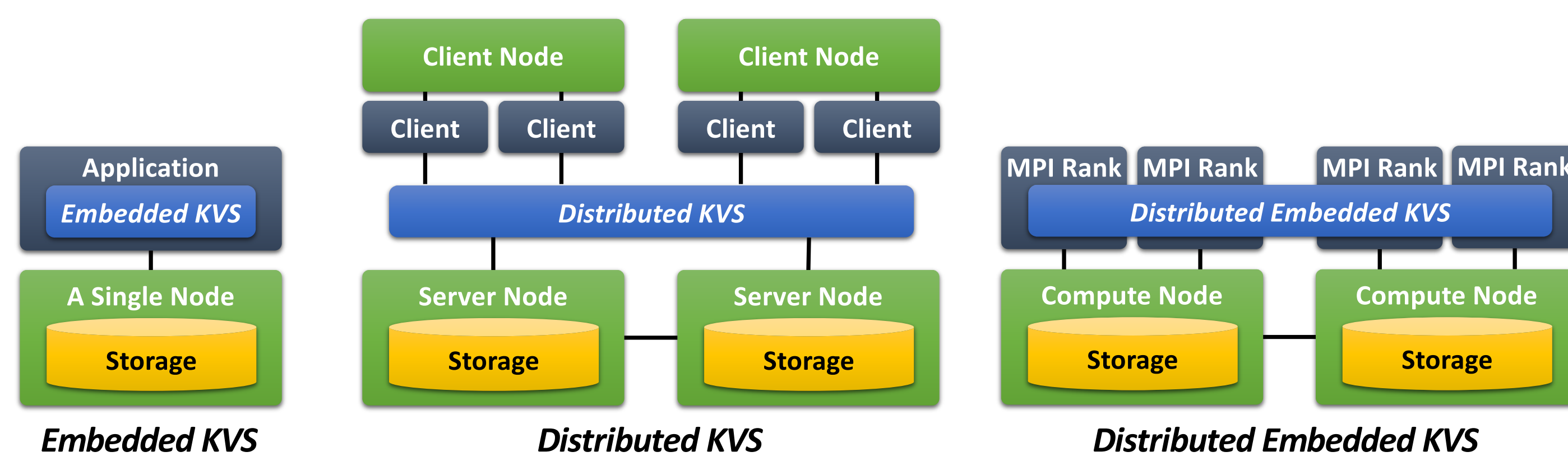
Implementing Efficient Data Compression and Encryption in a Persistent Key-Value Store for HPC

Jungwon Kim (kimj@ornl.gov) and Jeffrey S. Vetter (vetter@ornl.gov)
Oak Ridge National Laboratory

Motivation

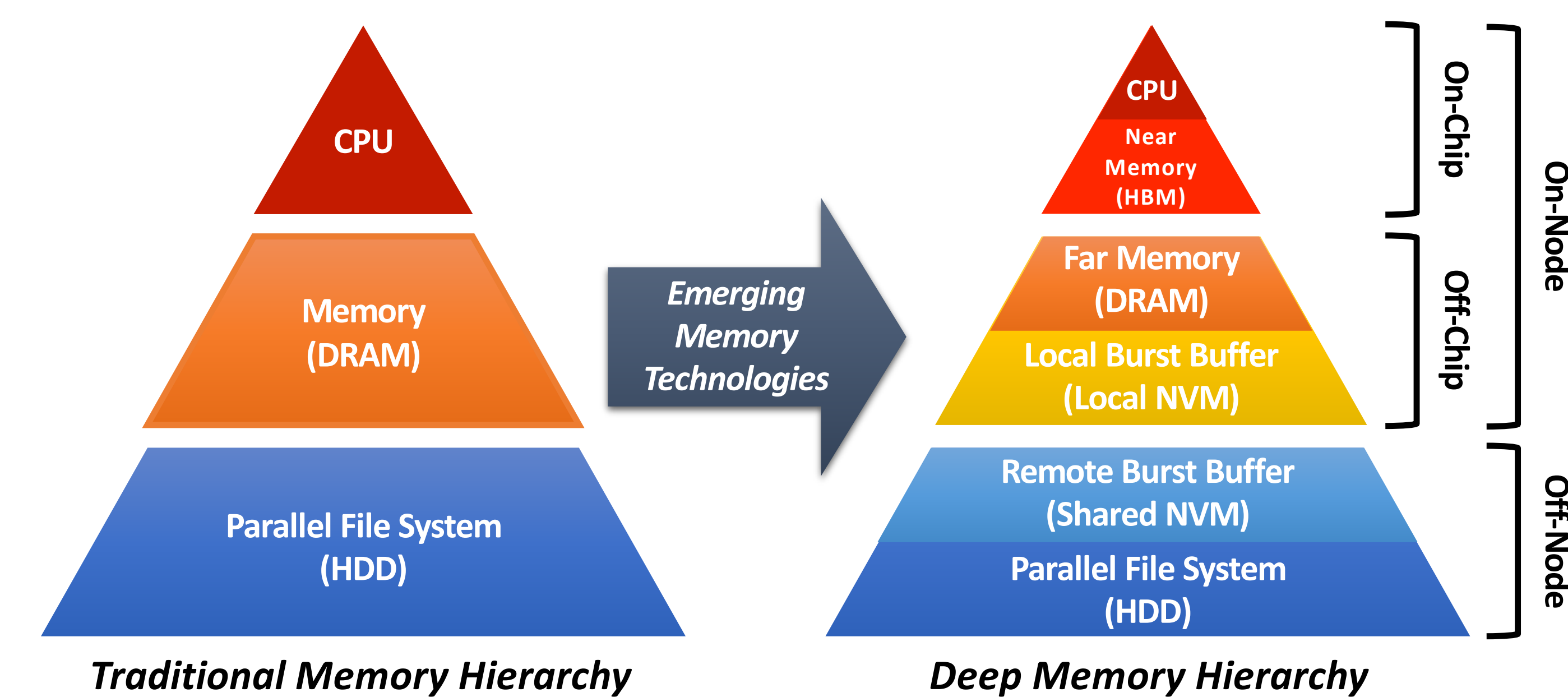
- Data Compression**
Contemporary scientific HPC simulations and experimental facilities are producing massive amounts of data at unprecedented rate.
- Data Encryption**
More sensitive data (e.g., health records, DNA data) is being used in distributed infrastructures and shared in complex scientific workflows.
- Key-Value Stores in HPC**
Coupling applications, storing intermediate results, and distributed programming model

Distributed Embedded KVS



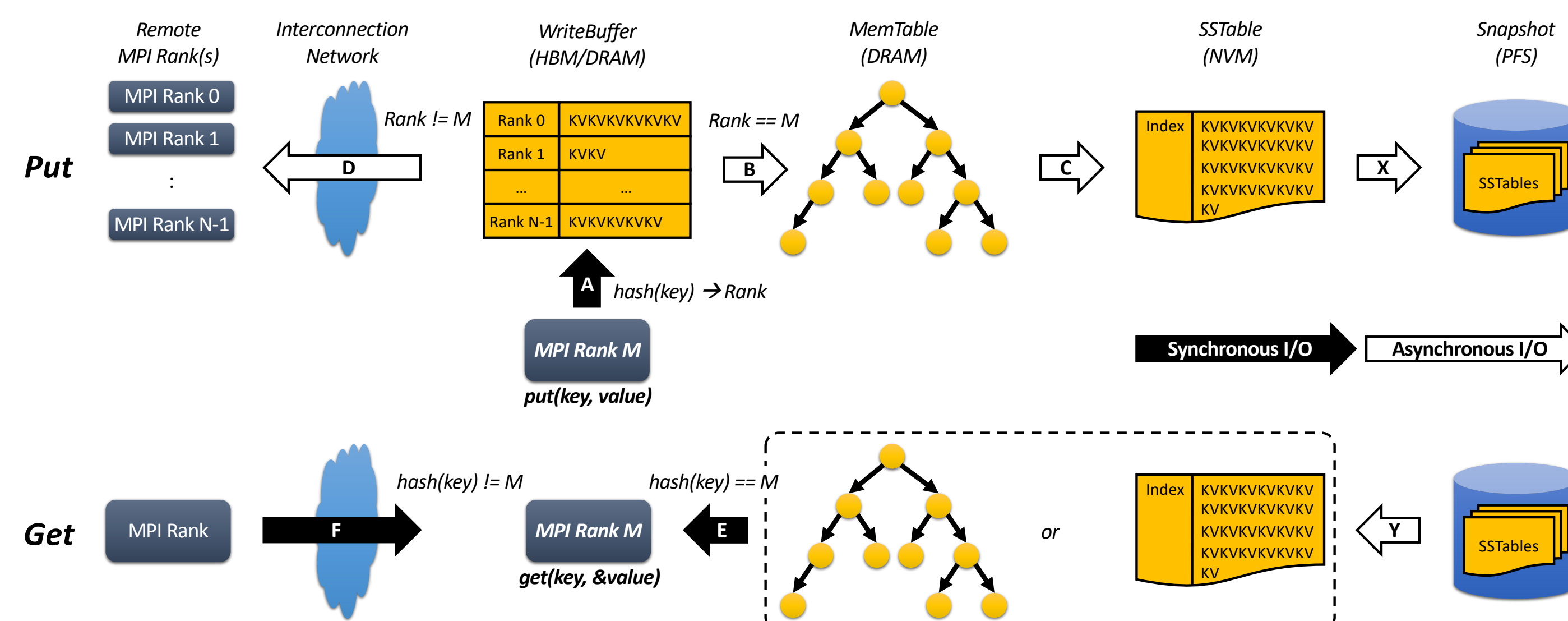
The distributed embedded KVS (DEKVS) inherits portability from the embedded KVS and scalability from the distributed KVS. The DEKVS is originally designed for HPC.

Deep Memory Hierarchies in HPC

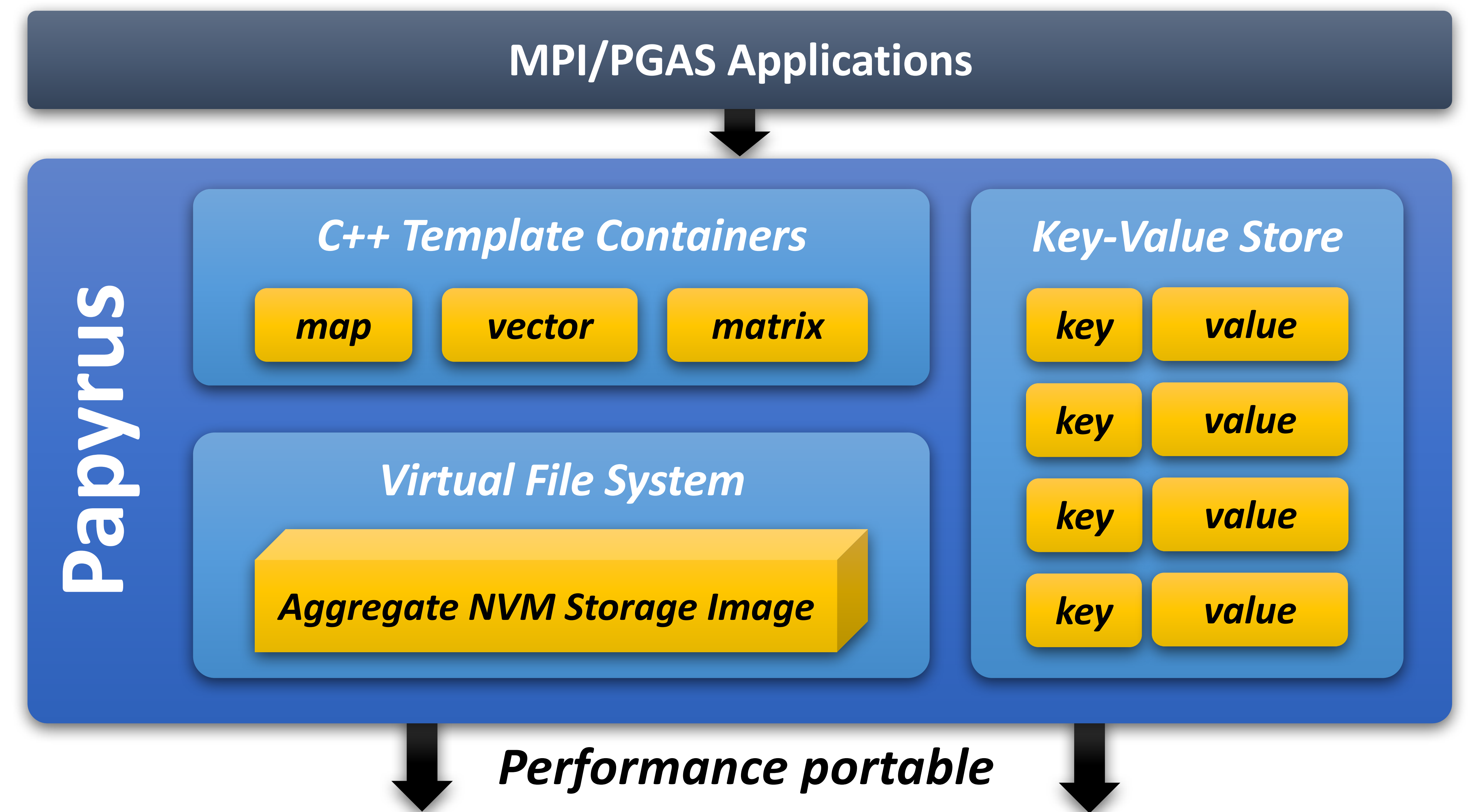


PapyrusKV presents data compression and encryption techniques that exploit deep memory hierarchy in HPC systems to achieve storage reduction, secure data sharing, and performance improvement.

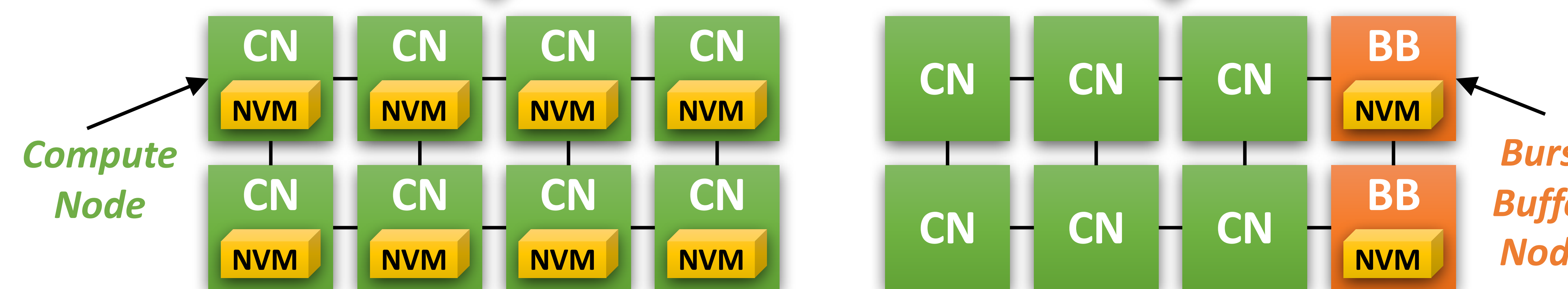
Data Structures and Flow Control



We evaluate the benefits and costs of several designs for implementing data compression and encryption capabilities along different points in the PapyrusKV dataflow path, illustrating differences in effective bandwidth, latency, and additional computational expense.



Performance portable



Node-Local NVM Architecture

(ALCF's Theta, LLNL's Sierra, OLCF's Summit, TACC's Stampede2)

Dedicated NVM Architecture

(NERSC's Cori, LANL/Sandia's Trinity)

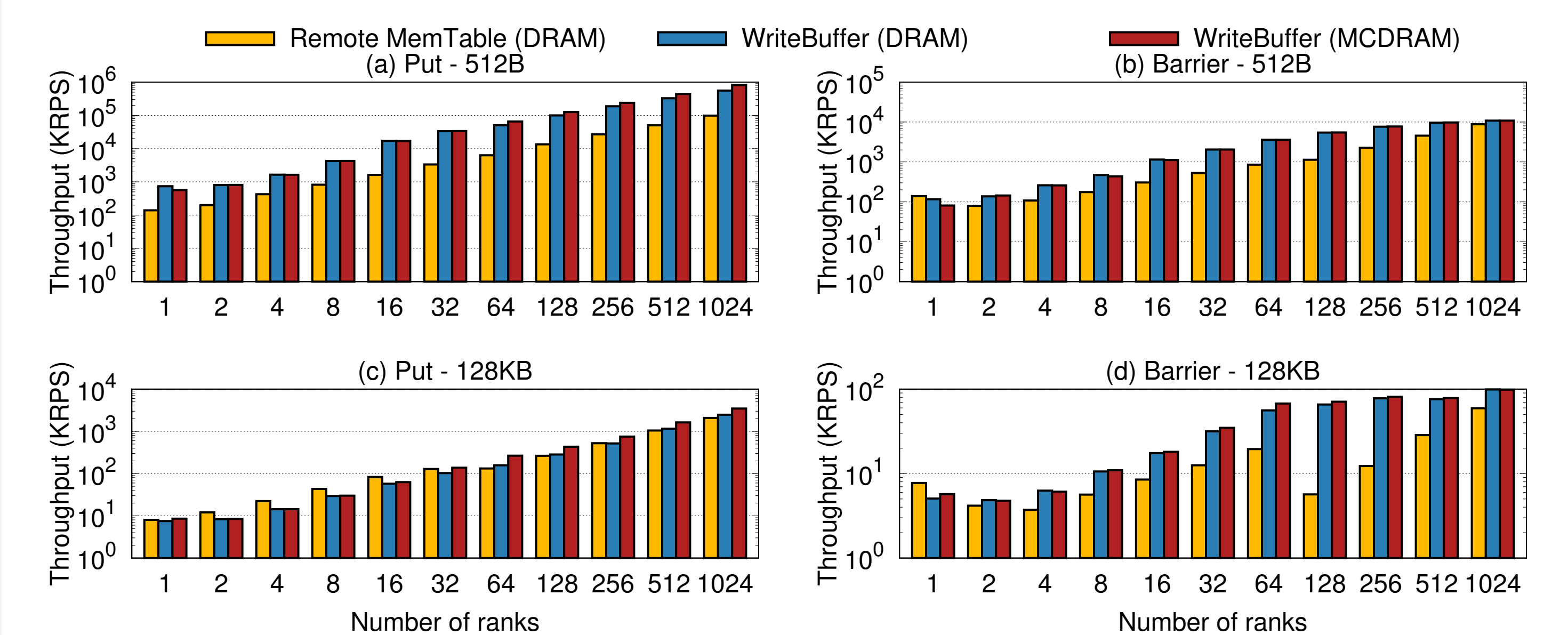
- Papyrus** [1] is a programming system that provides a portable and scalable programming interface to access and manage parallel data structures on the distributed NVM storage.
- PapyrusKV** [2] is a distributed embedded key-value store (DEKVS).
- PapyrusKV** supports BZ2, LZ4, LZ4HC, Snappy, Zlib, Zstandard, and SZ data compression algorithms.
- PapyrusKV** supports AES-128,192,256 ECB and CBC data encryption modes.

<https://code.ornl.gov/eck/papyrus>

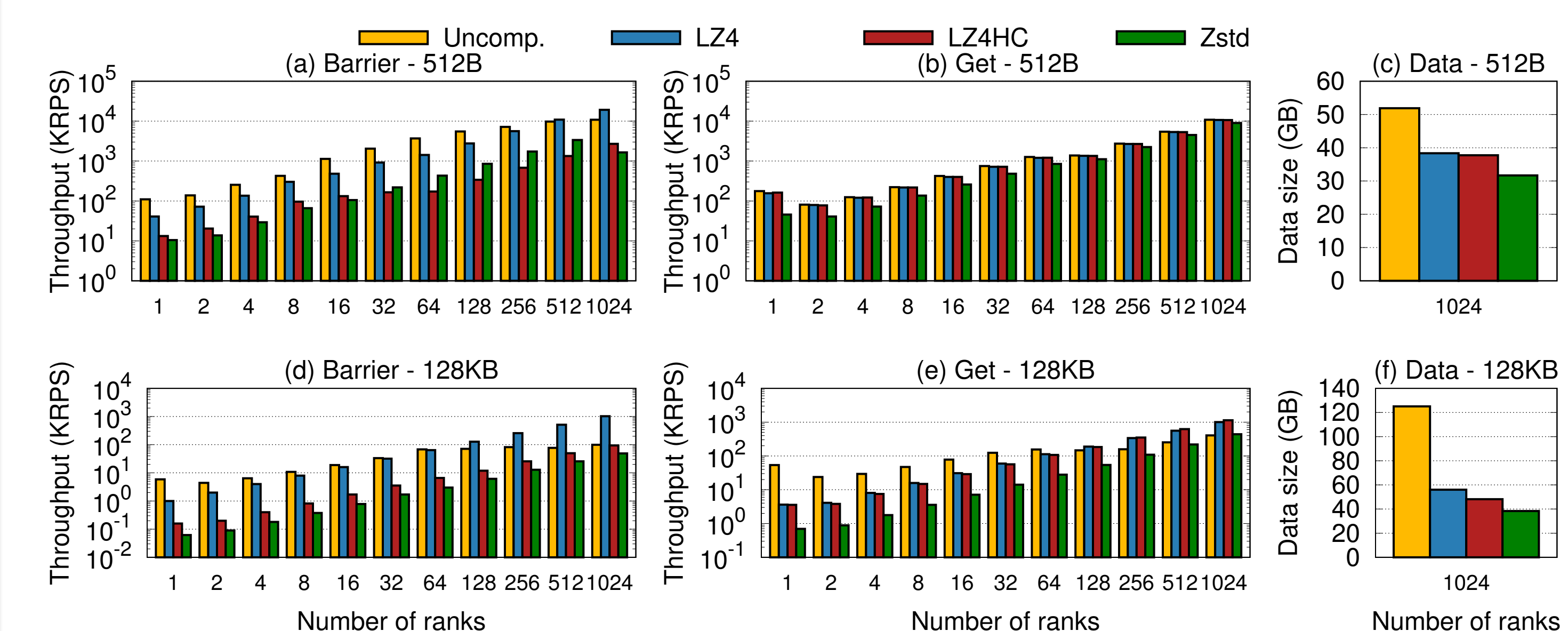
- [1] Jungwon Kim, Kittisak Sajjapongse, Seyong Lee, and Jeffrey S. Vetter. "Design and Implementation of Papyrus: Parallel Aggregate Persistent Storage". IPDPS 2017.
[2] Jungwon Kim, Seyong Lee, and Jeffrey S. Vetter. "PapyrusKV: A High-Performance Parallel Key-Value Store for Distributed NVM Architectures". SC 2017.

Preliminary Evaluation

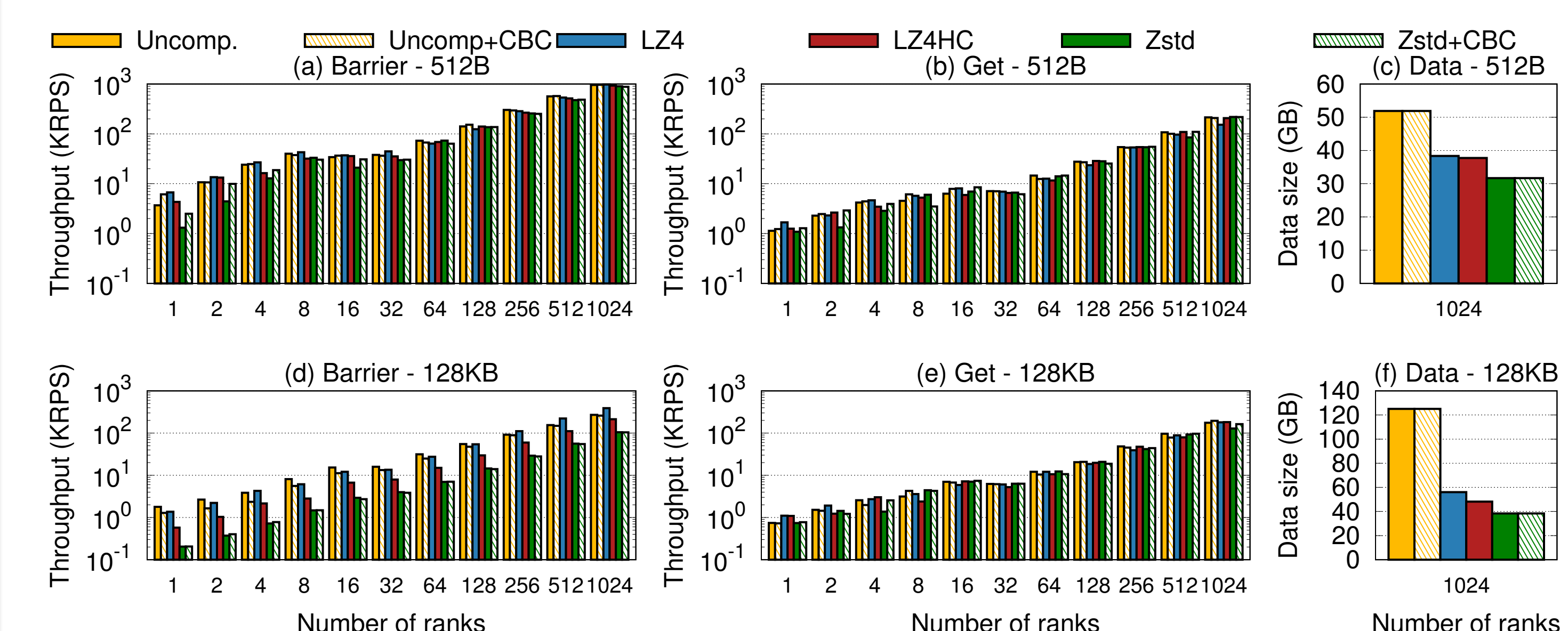
System	Grand Tave	Cori
Site	CSCS	NERSC
CPU	Intel Xeon Phi 7230 (KNL)	2 x Intel Xeon E5-2698 (Haswell)
CPU cores per node	64	32
HBM	16GB MCDRAM	-
Burst buffer	-	Shared 1.8PB aggregate SSDs on burst buffer nodes



Put and barrier operations performance on Grand Tave. Up to 64 MPI ranks run on a single node. For 512B, each rank performs 100K put operations with 32B keys and 512B values. For 128KB, each rank performs 1K put operations with 32B keys and 128KB values. We use KRPS (kilo requests per second) metric of performance. **5.6x** (WriteBuffer-DRAM) and **8.3x** (WriteBuffer-MCDRAM) speedups in 1,024 ranks.



Barrier and get operations performance on Grand Tave with various compression algorithms. Up to 64 MPI ranks run on a single node.
Barrier performance: LZ4-128KB: **2.9x** speedup with **0.4x** data size in 1,024 ranks.
Get performance: LZ4HC-128KB: **2.8x** speedup with **0.4x** data size in 1,024 ranks.



Barrier and get operations performance on Cori with various compression algorithms and AES-256 with CBC data encryption. Up to 32 MPI ranks run on a single node.
Barrier performance: LZ4-128KB: **1.4x** speedup with **0.4x** data size in 1,024 ranks.
Get performance: Zstd+CBC-128KB: **1.0x** speedup with **0.3x** data size in 1,024 ranks.