BlueDBM: Distributed Flash Store for Big Data Analytics

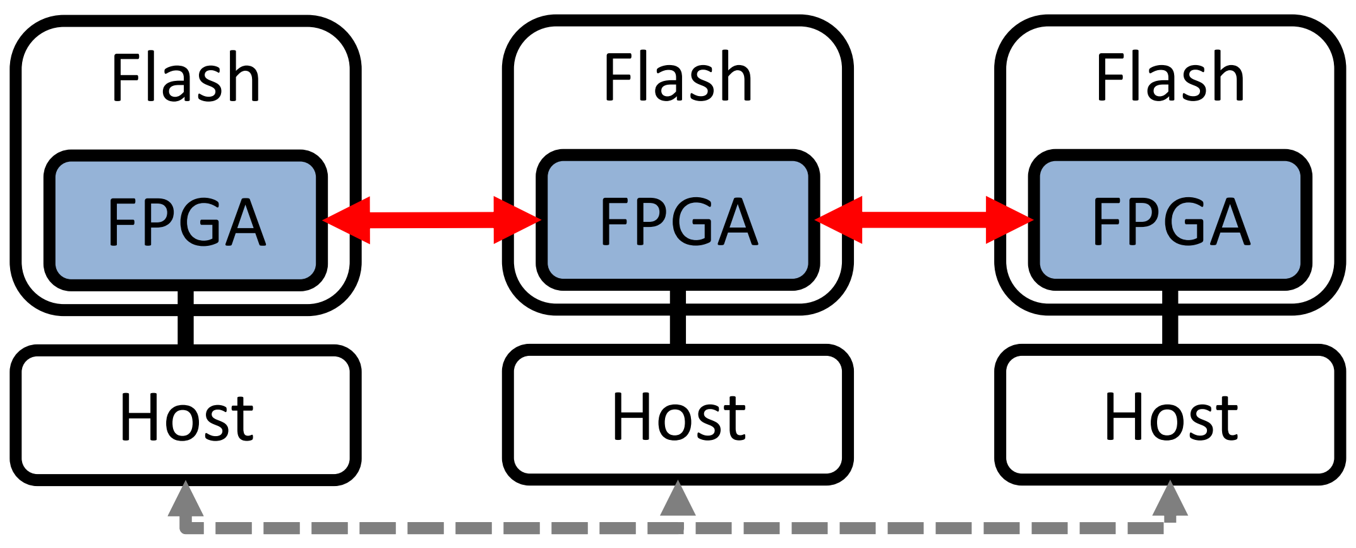
Sang-Woo Jun, Ming Liu, Shuotao Xu, Sungjin Lee, Arvind

Motivation

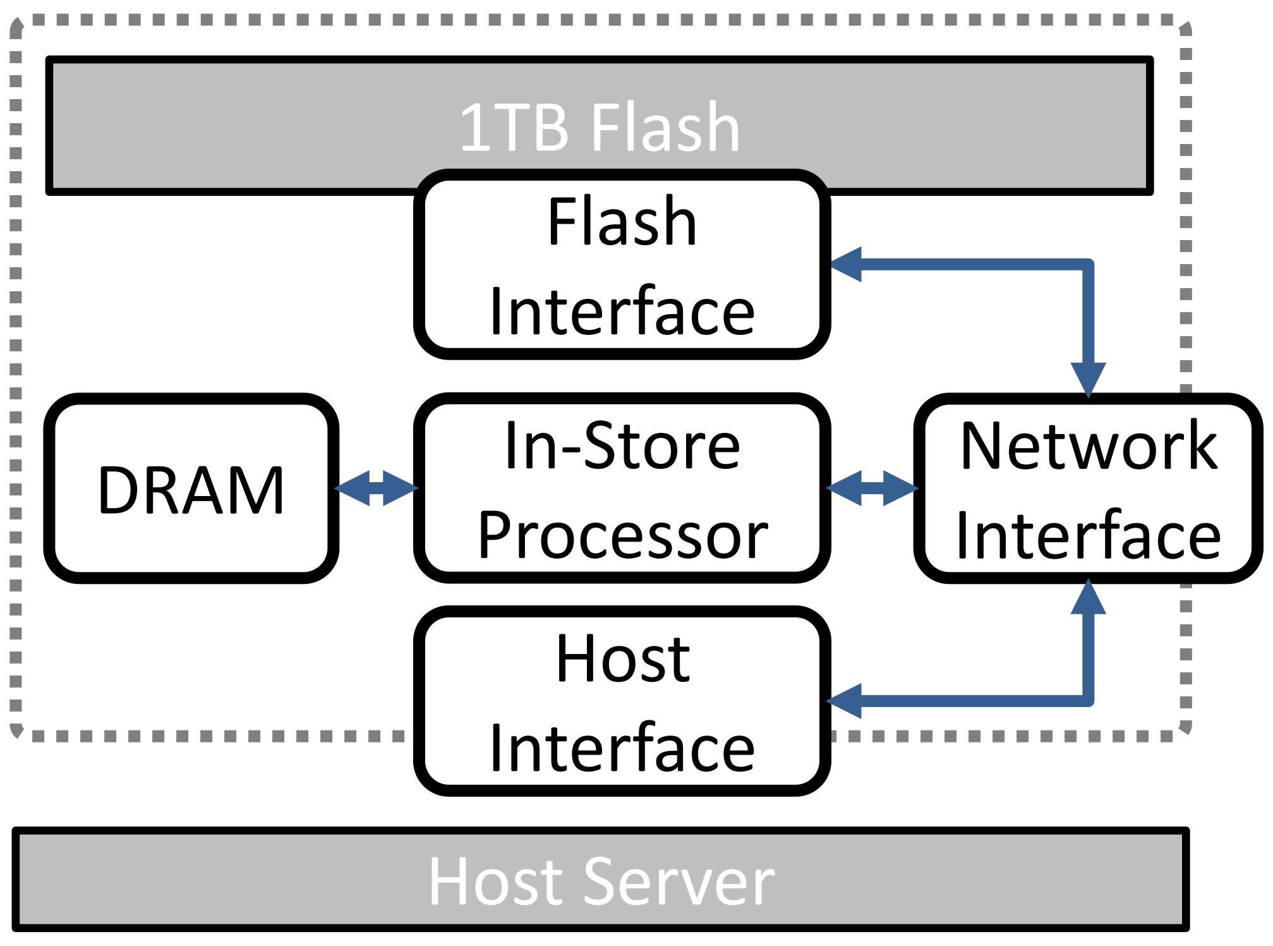
- Performance of many Big-Data applications are bound by capacity of fast random access memory, as performance drops sharply when even a small amount doesn't fit
- A system with enough DRAM to accommodate the entire working set is very expensive and power hungry
- Flash storage is cheap and fast, but using it as a disk replacement is inefficient due to translation overhead
- Flash is fast enough that other system components become bottlenecks

Platform Overview

- Rack-scale cluster of systems with enough flash capacity for Big Data workloads
- Fast flash storage devices with FPGA-based in-store processor and PCIe host link
- High-speed storage area network directly between storage devices
- Fast software with cross-layer optimizations



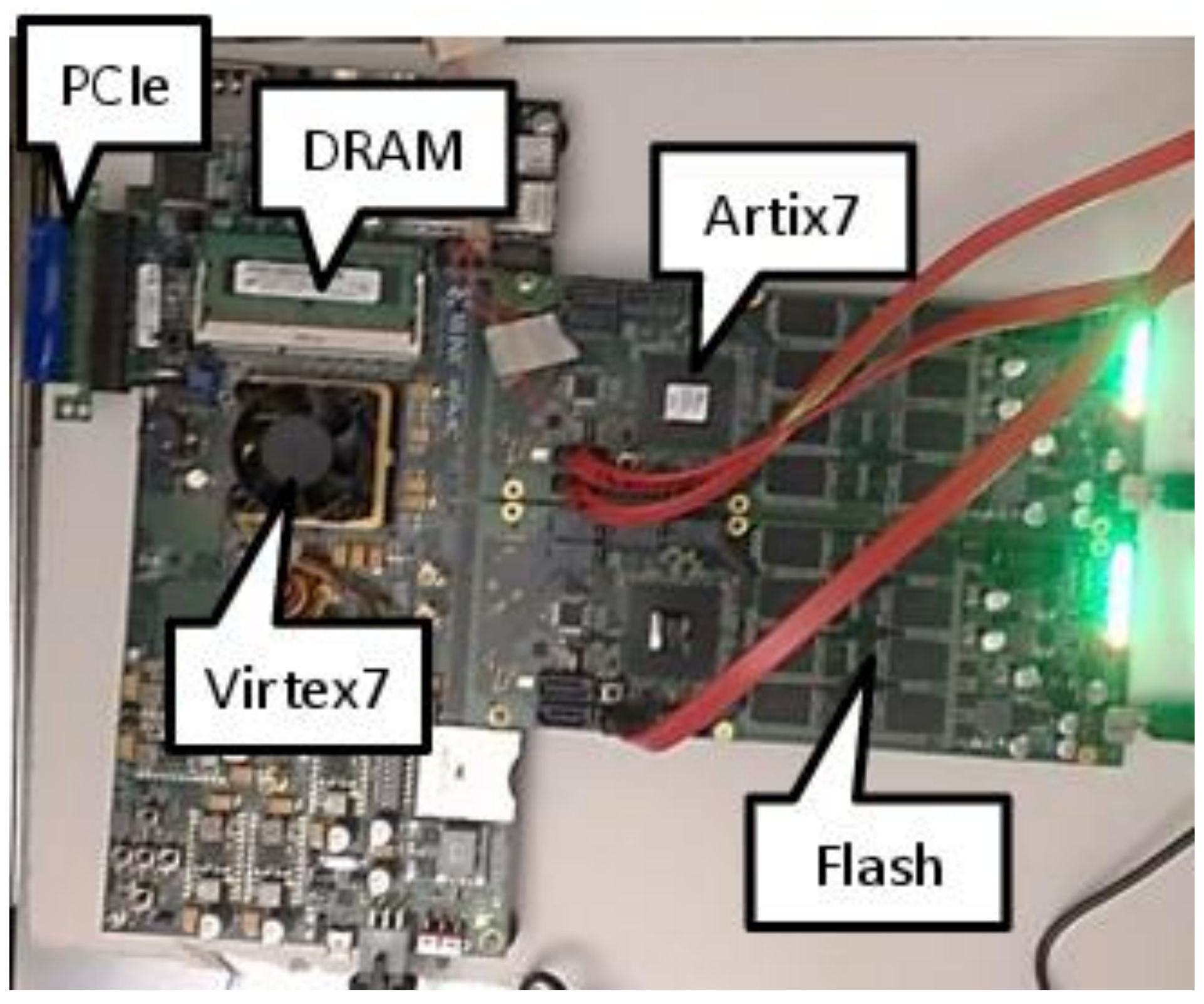
System Architecture



- Lightweight flash management
  - ✓ Adds almost no overhead
  - ✓ Exposes device organization to upper layers for exploiting parallelism
  - ✓ Bit-error corrected
- Low latency transport layer network protocol
  - ✓ Deterministic routing to simplify flow control while maximizing bandwidth
  - ✓ Virtual channel and flow control with very low protocol overhead (0.5us)
- Software has very low level access to flash
  - ✓ High level information can be used for flash management
  - ✓ Cross-layer optimizations, such as FTL function in file system

Hardware Description

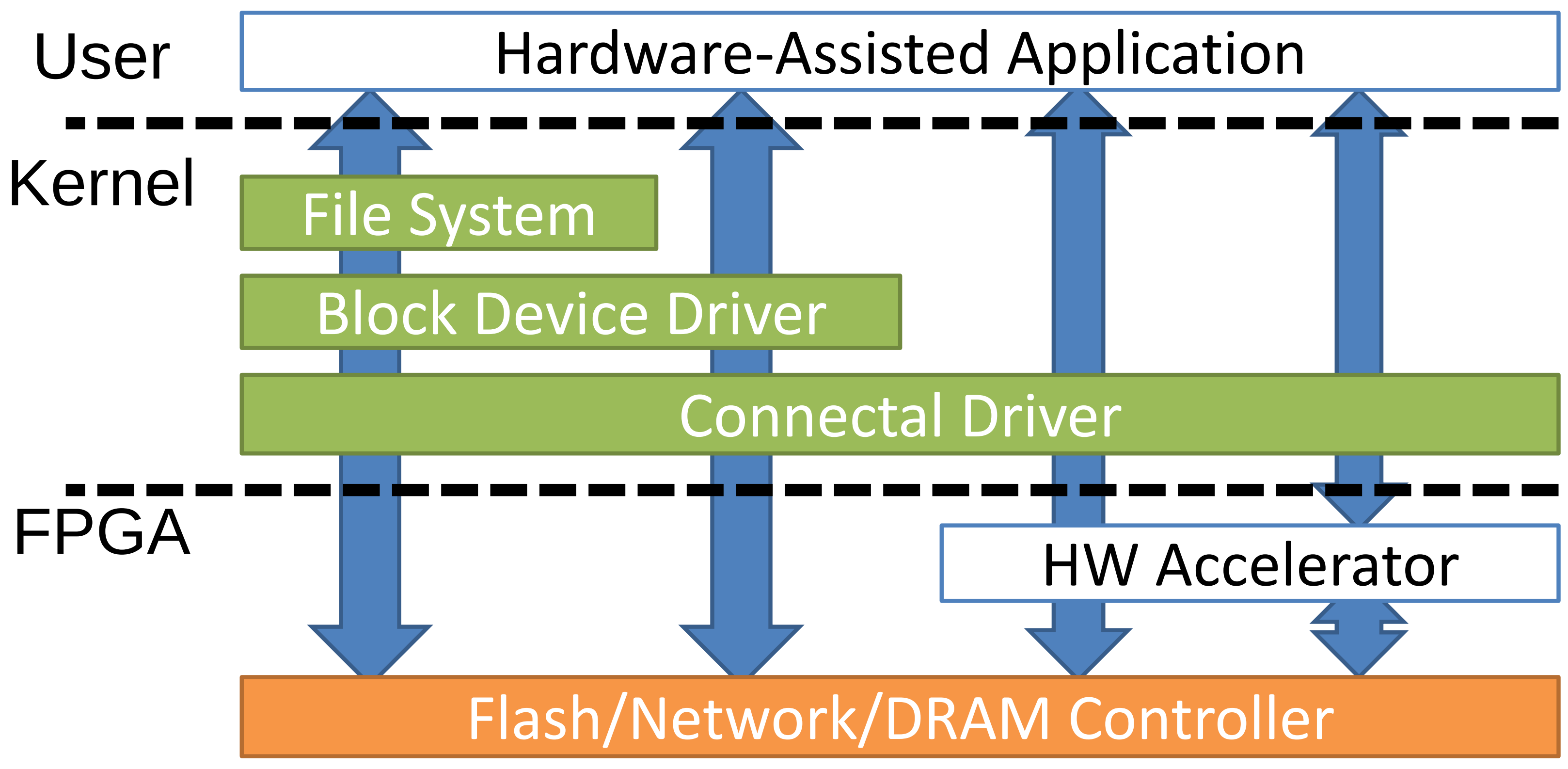
- 20-Node cluster across two racks
- Xilinx VC707 with two custom flash boards
  - ✓ Capacity: 1TB per node
  - ✓ Flash Bandwidth: 2x 1.6GB/s per node
  - ✓ Network: x4 20Gbps 0.5us serial links
  - ✓ Host interface: 8-lane PCIe Gen 1



User Model

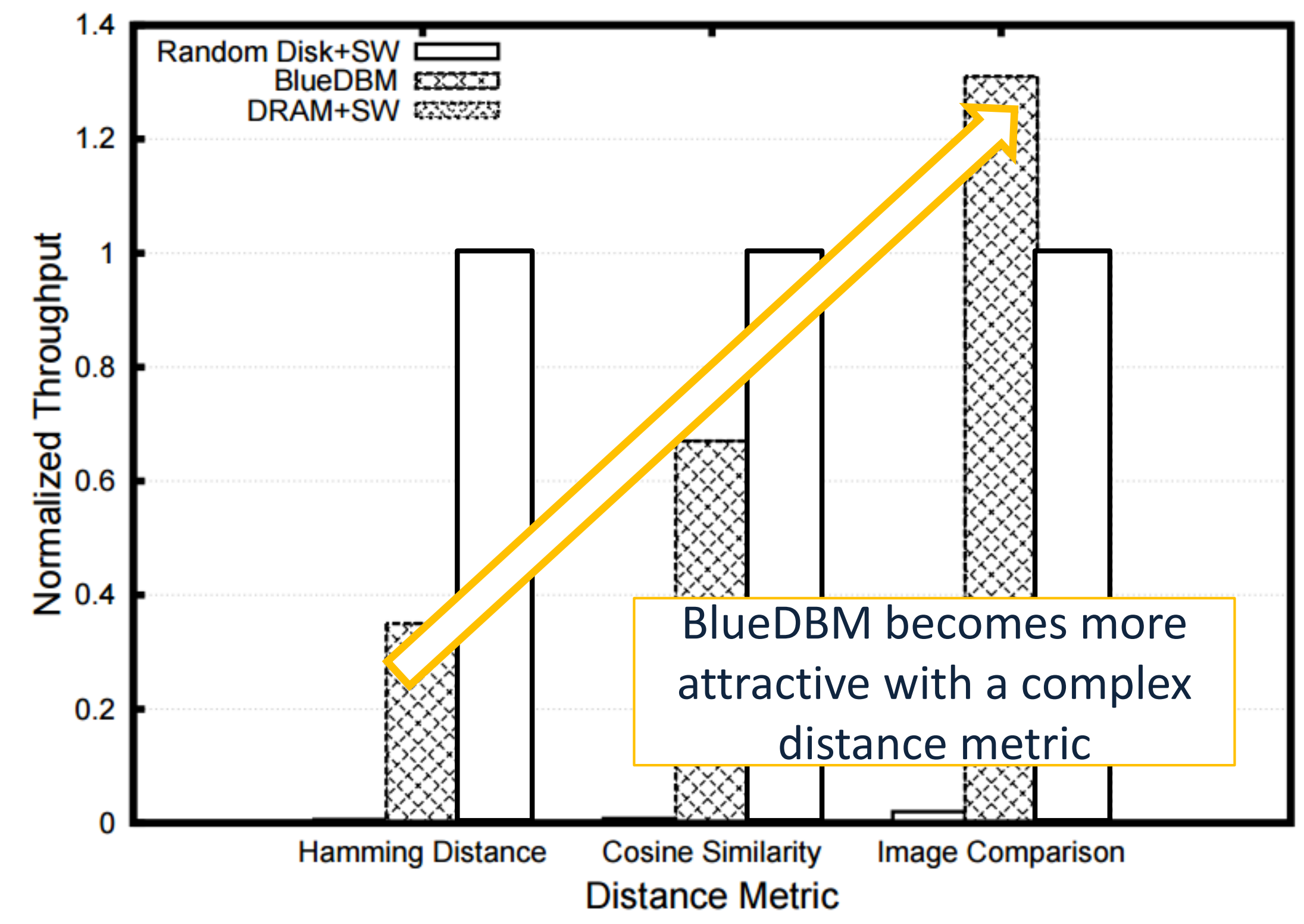
- SW application is augmented with HW accelerators interfaced over Connectal* interface abstraction
- In-store processor can access flash storage, network and DRAM via corresponding controllers

* Open-source RPC-style HW/SW interface library developed by Quanta Research Cambridge



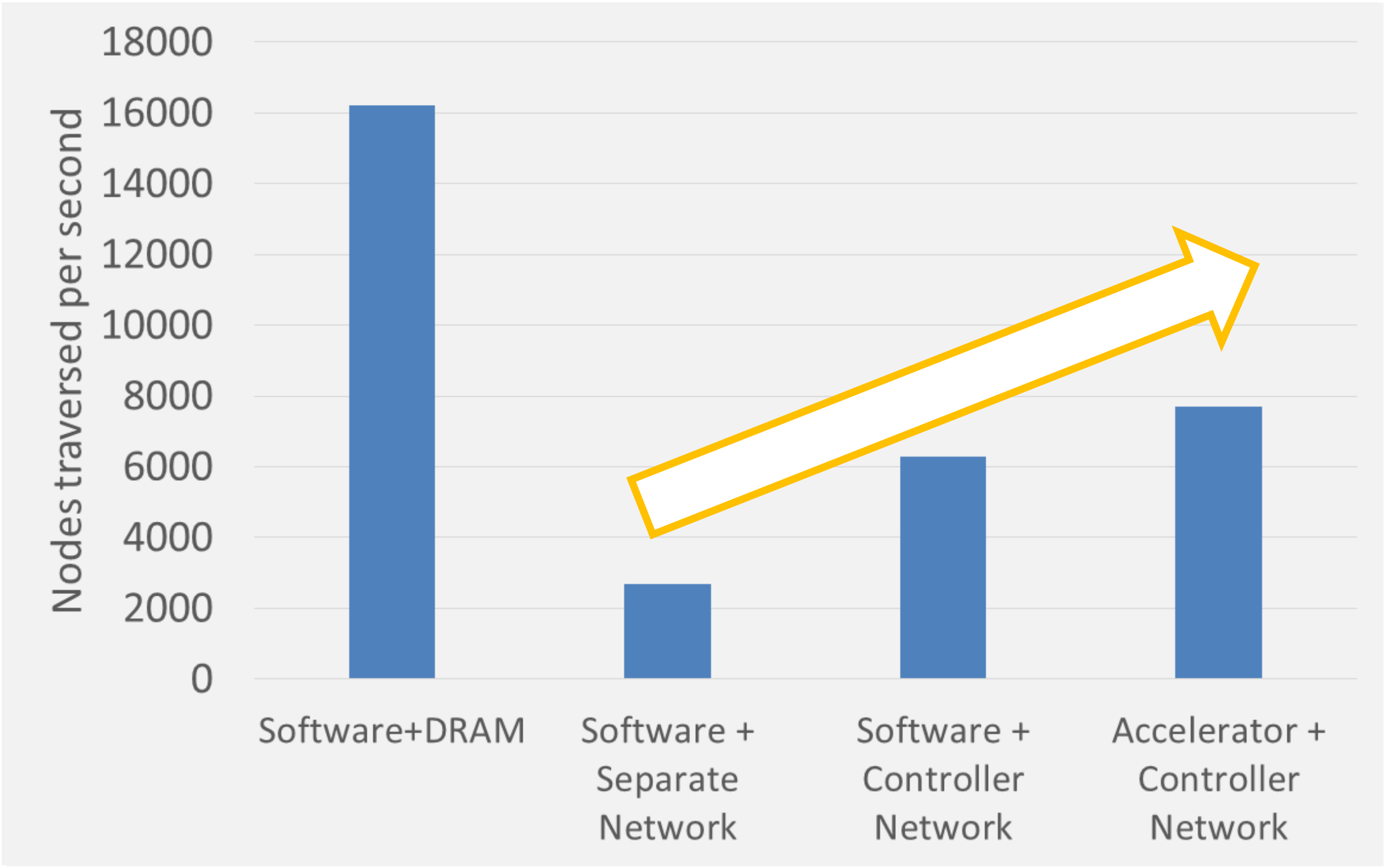
Example Applications – More on the way!

- **Nearest neighbor search**
  - ➢ Takes a query and finds similar data points from a large dataset according to some distance metric
  - ➢ Hamming distance: simple
  - ➢ Cosine similarity: moderate
  - ➢ image histogram comparison: complex

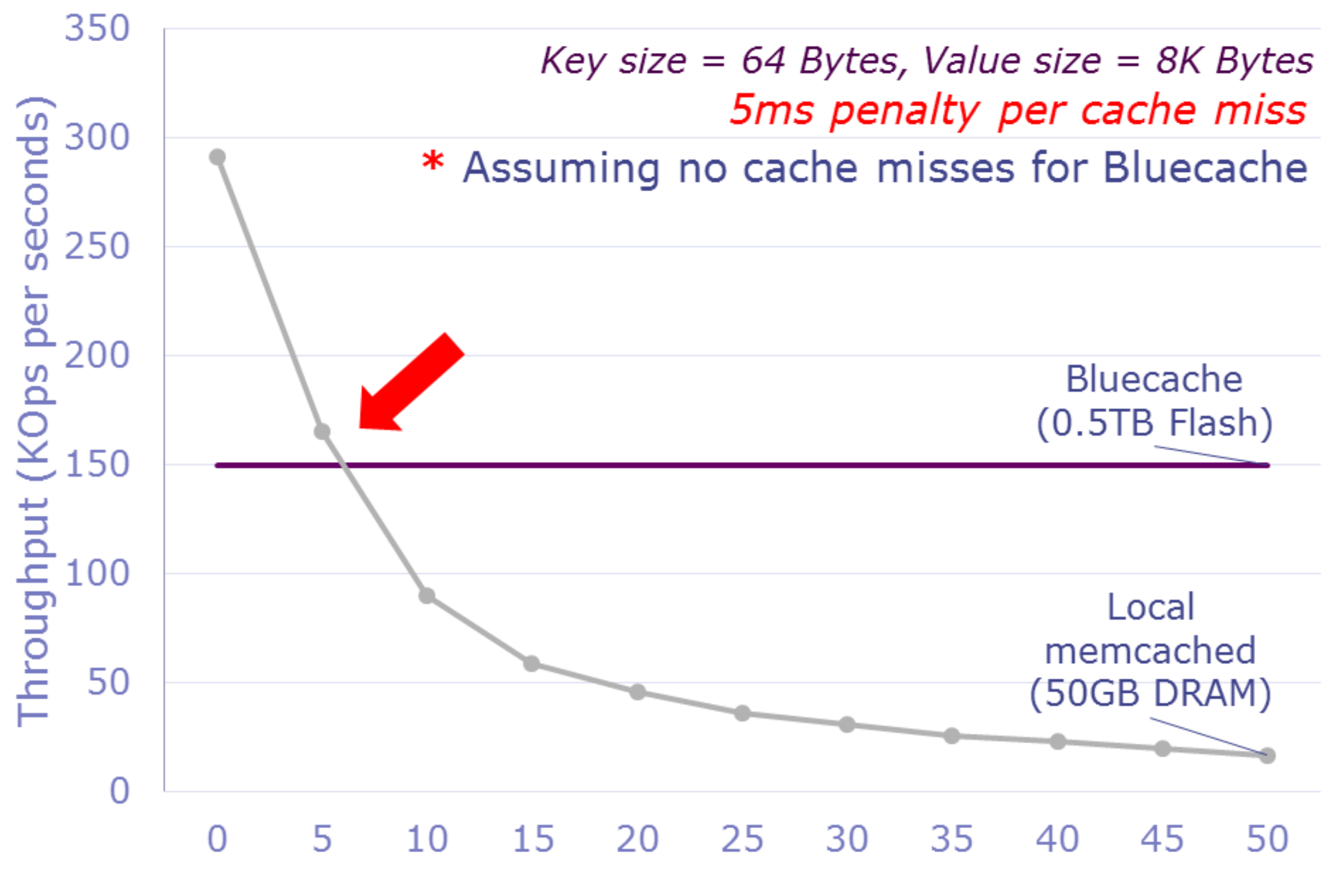


BlueDBM becomes more attractive with a complex distance metric

- **Graph traversal**
  - ➢ Very latency bound because the next node to visit often can only be known after reading previous node
  - ➢ Reduced latency using integrated network and in-storage processing
  - ➢ Comparable to a much more expensive DRAM system



- **Flash-based memcached**
  - ➢ Simple distributed key-value store implemented in hardware
  - ➢ Low-latency network between application server and memcached
  - ➢ Performance benefits of architecture modification and higher capacity makes it attractive


Key size = 64 Bytes, Value size = 8K Bytes
5ms penalty per cache miss
* Assuming no cache misses for Bluecache
Throughput (KOps per seconds)
Bluecache (0.5TB Flash)
Local memcached (50GB DRAM)