BlueDBM: A Multi-access, Distributed Flash Store for Big Data Analytics

Arvind
Computer Science and Artificial Intelligence Laboratory
MIT

VLSI 2016, Kolkata, India
January 5, 2016


This talk is based heavily on our [ISCA, 2015] paper

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

and

Jamey Hicks, John Ankcorn, Myron King
Quanta Research

Big data analytics

- Analysis of previously unimaginable amount of data can provide deep insight
  - Google has predicted flu outbreaks a week earlier than the Center for Disease Control (CDC)
  - Analyzing personal genome can determine predisposition to diseases
  - Social network chatter analysis can identify political revolutions before newspapers
  - Scientific datasets can be mined to extract accurate models

Likely to be the biggest economic driver for the IT industry for the next decade

A currently popular solution: RAM Cloud

- Cluster of machines with large DRAM capacity and fast interconnect
  - + Fastest as long as data fits in DRAM
  - - Power hungry and expensive
  - - Performance drops when data doesn't fit in DRAM

What if enough DRAM isn't affordable?

- Flash-based solutions may be a better alternative
  - + Faster than Disk, cheaper than DRAM
  - + Lower power consumption than both
  - - Legacy storage access interface is burdening
  - - Slower than DRAM

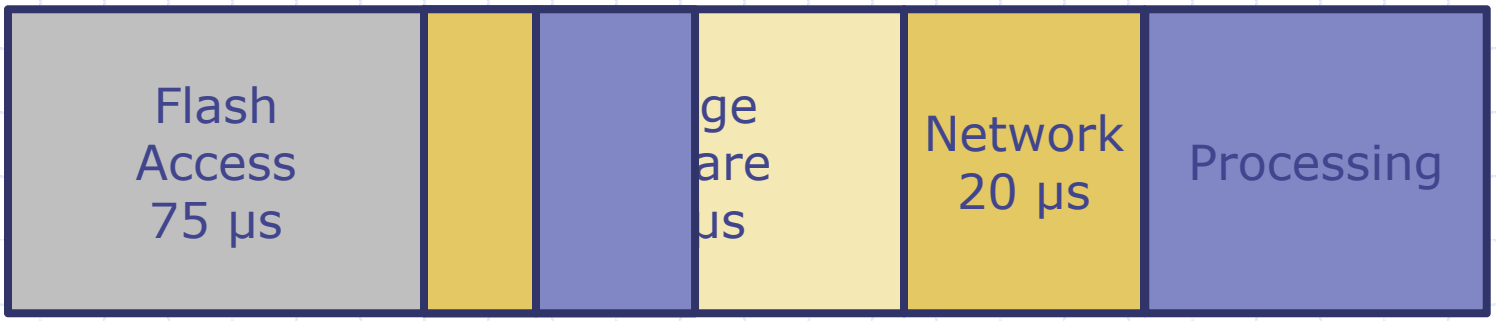
Latency profile of distributed flash-based analytics

- Distributed processing involves many system components
  - Flash device access
  - Storage software (OS, FTL, …)
  - Network interface (10gE, Infiniband, …)
  - Actual processing

Latency is additive

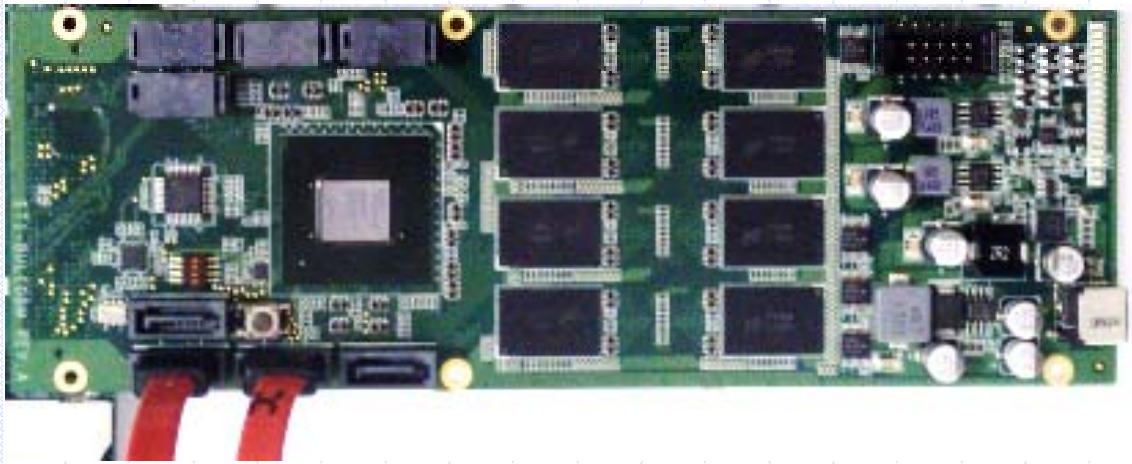
Latency profile of distributed flash-based analytics

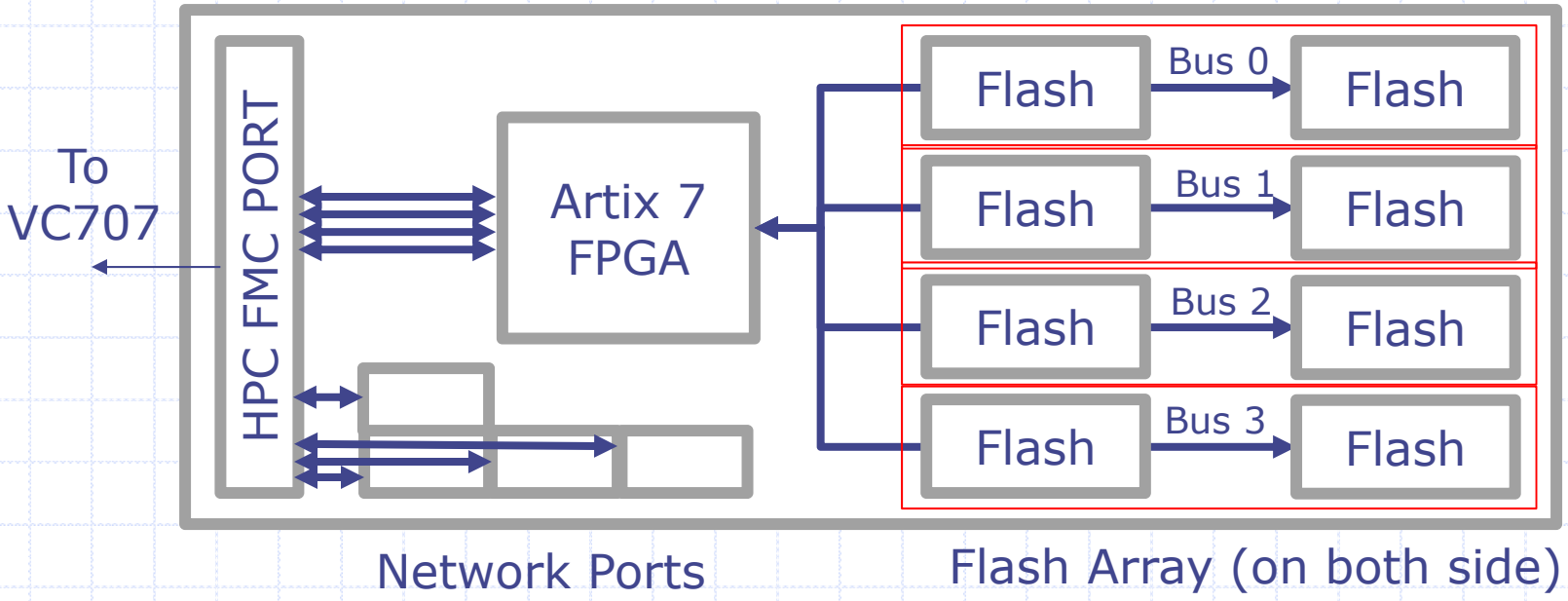
- Architectural modifications can remove unnecessary overhead
  - Near-storage processing
  - Cross-layer optimization of flash management software*
  - Dedicated storage area network
  - Accelerator



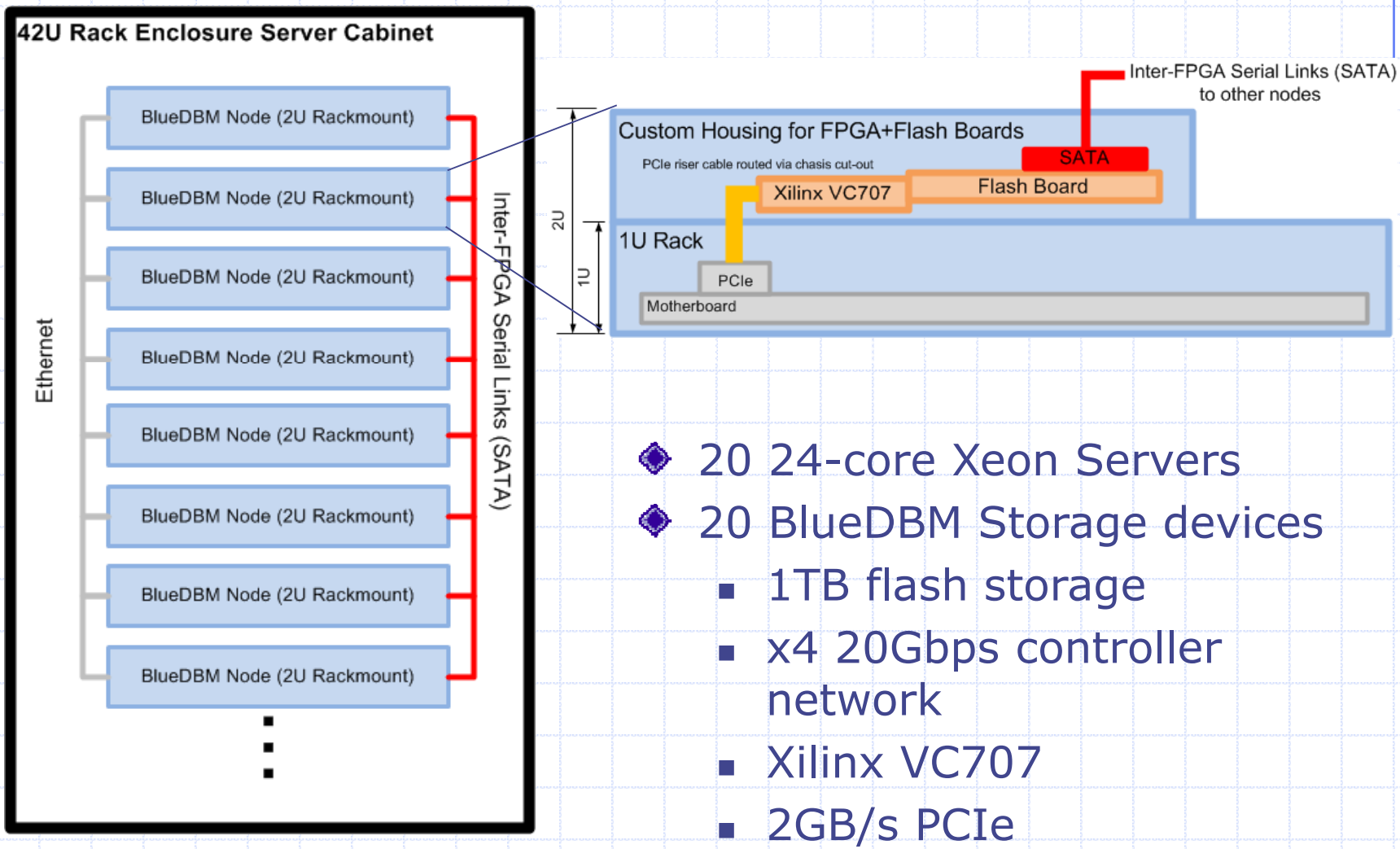
Difficult to explore using flash packaged as off-the-shelf SSDs

Custom flash card had to be built





BlueDBM: Platform with near-storage processing and inter-controller networks

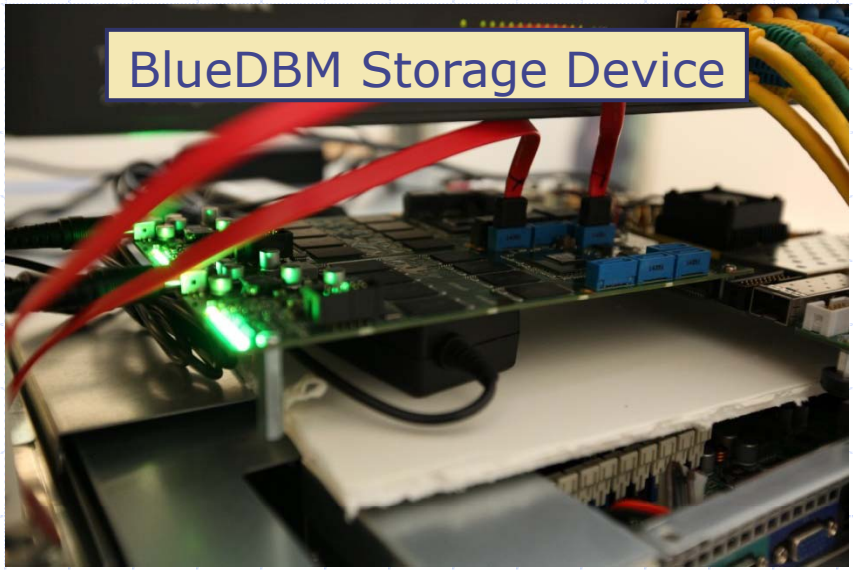


- 20 24-core Xeon Servers
- 20 BlueDBM Storage devices
  - 1TB flash storage
  - x4 20Gbps controller network
  - Xilinx VC707
  - 2GB/s PCIe

BlueDBM: Platform with near-storage processing and inter-controller networks

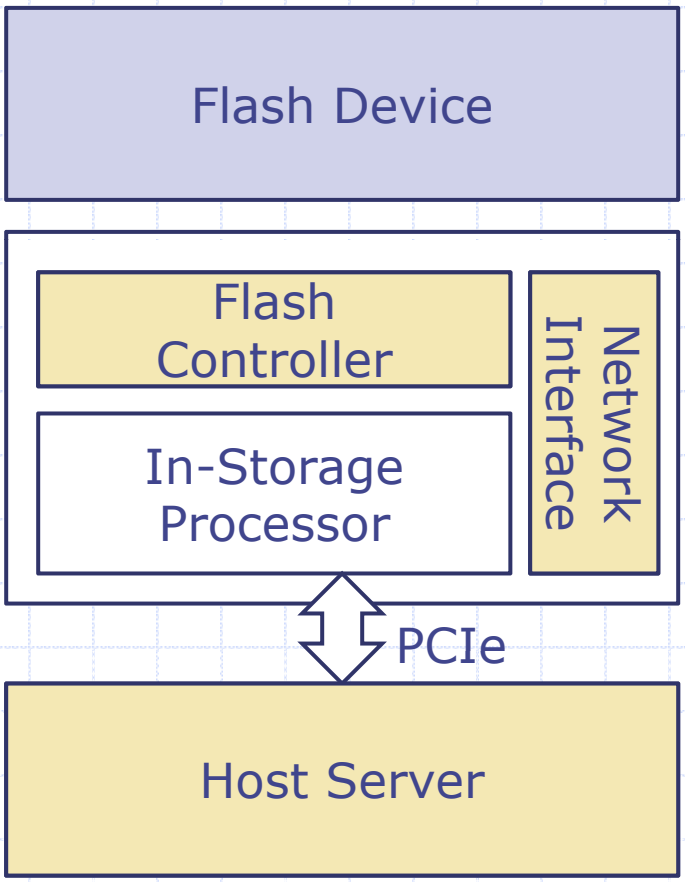
1 of 2 Racks (10 Nodes)

BlueDBM Storage Device



- 20 24-core Xeon Servers
- 20 BlueDBM Storage devices
  - 1TB flash storage
  - x4 20Gbps controller network
  - Xilinx VC707
  - 2GB/s PCIe

BlueDBM node architecture



- Lightweight flash management with low overhead
- Custom network protocol with low latency/high bandwidth
- Software has low-level access to flash storage
  - High-level information can be used for low-level management
  - FTL implemented inside file system

Power consumption is low

Component	Power (Watts)
VC707	30
Flash Board (x2)	10
Storage Device Total	40

Storage device power consumption is a very conservative estimate

Component	Power (Watts)
Storage Device	40
Xeon Server	200+
Node Total	240+

GPU-based accelerator will double the power

Applications

- High-dimensional nearest neighbor search
  - Faster flash with accelerators as replacement for DRAM-based systems
- BlueCache – An accelerated memcached
  - Dedicated network and accelerated caching systems with larger capacity
- Graph analytics
  - Benefits of lower latency access into distributed flash for computation on large graphs

High-dimensional nearest neighbor search

- Takes a query point in a high dimensional space and returns nearest points in a dataset of tens of millions of data points
  - We used images as an example of high-dimensional data
- Used a distance metric of difference between histograms of each image
  - Histogram is generated using RGB, HSV, “edgeness”, etc
  - Better algorithms are available!

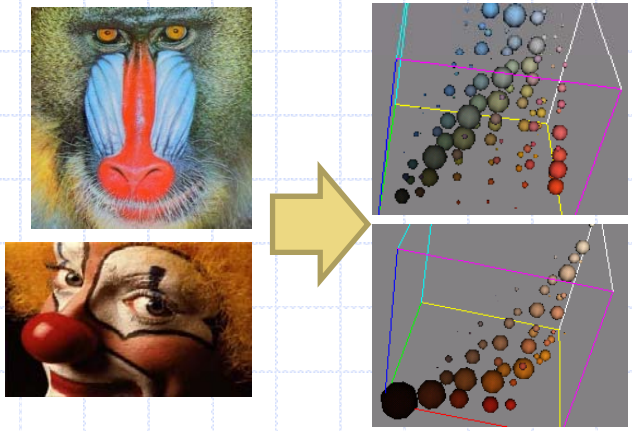
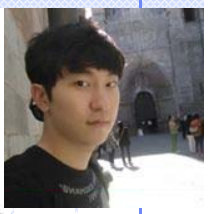


Image search accelerator

Sang woo Jun, Chanwoo Chung



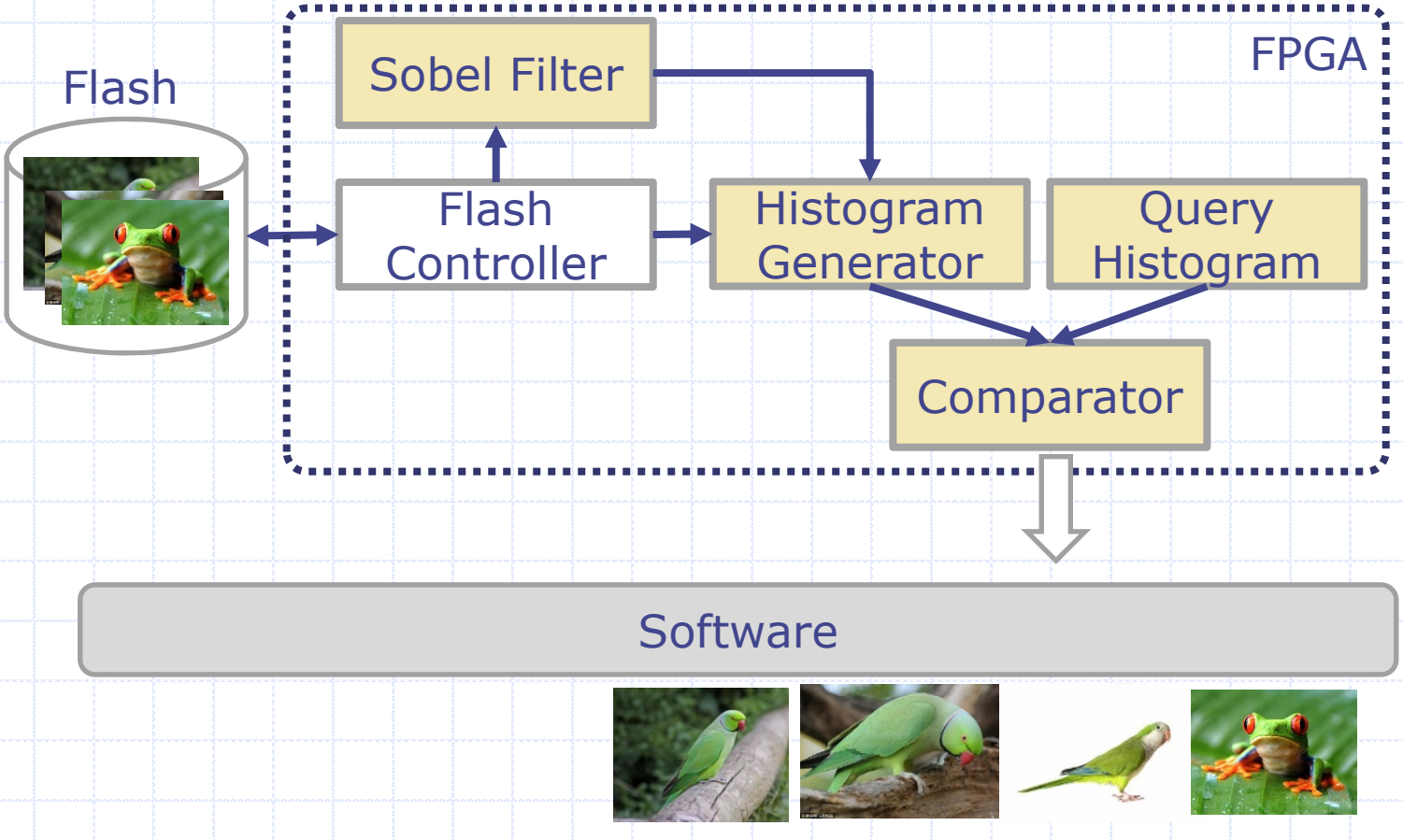
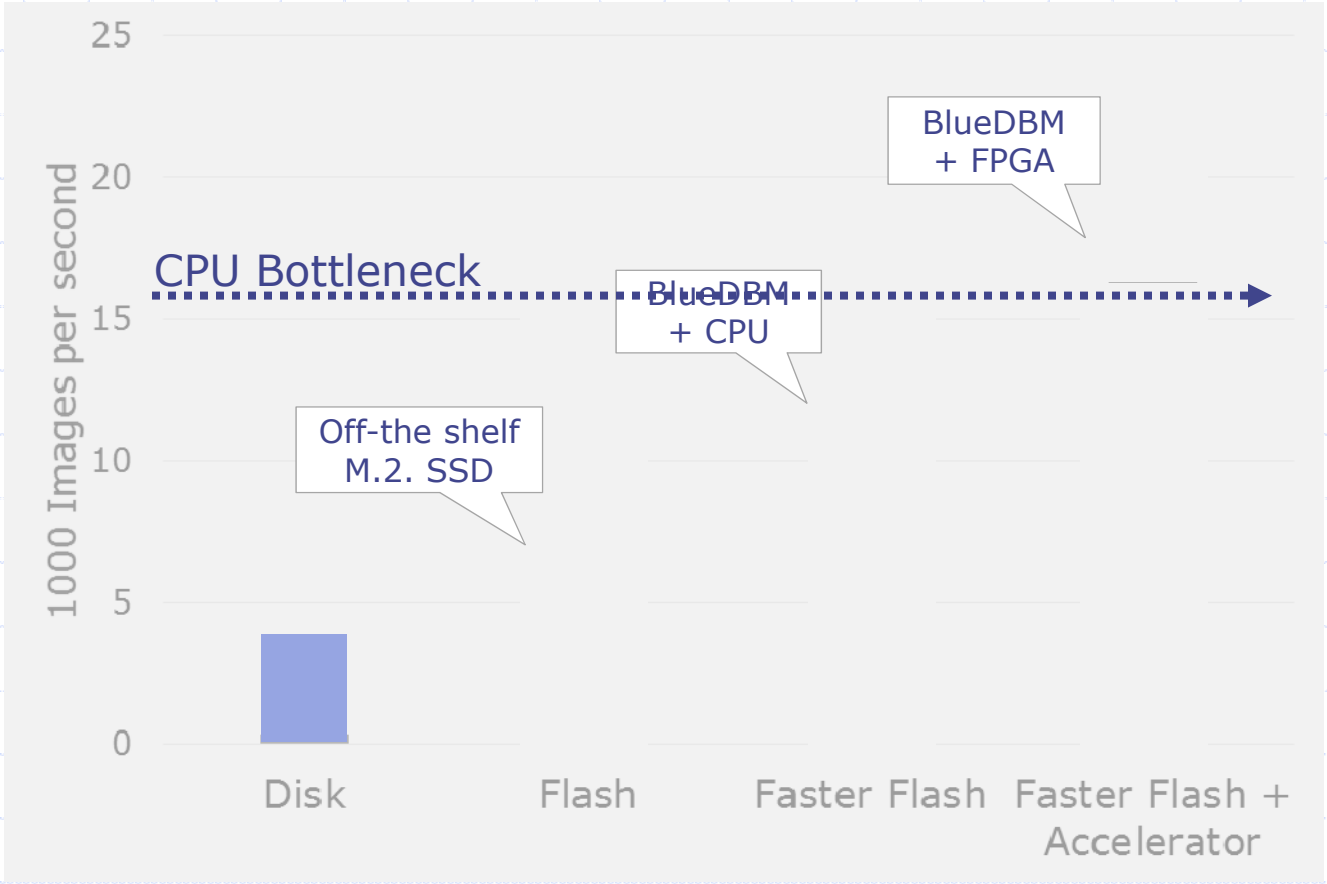


Image search accelerator

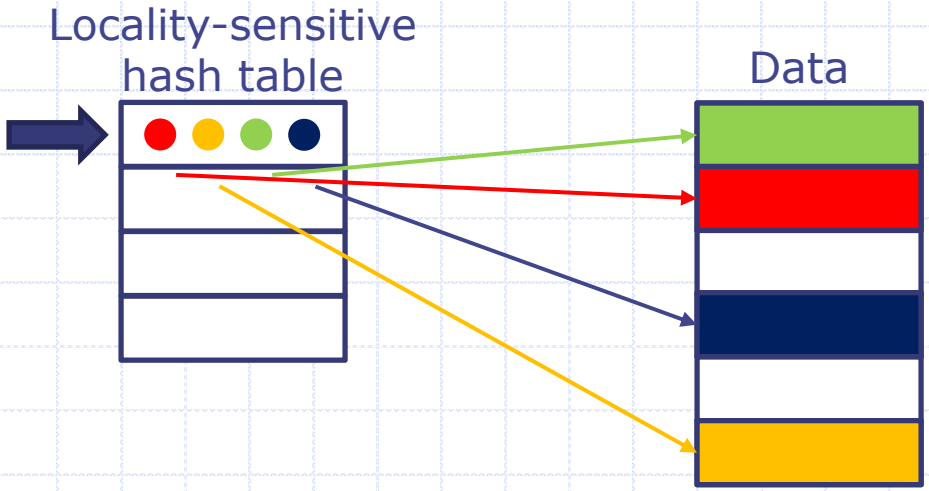
Sang woo Jun, Chanwoo Chung



- Faster flash with acceleration can perform at DRAM speed [ReConFig 2015]

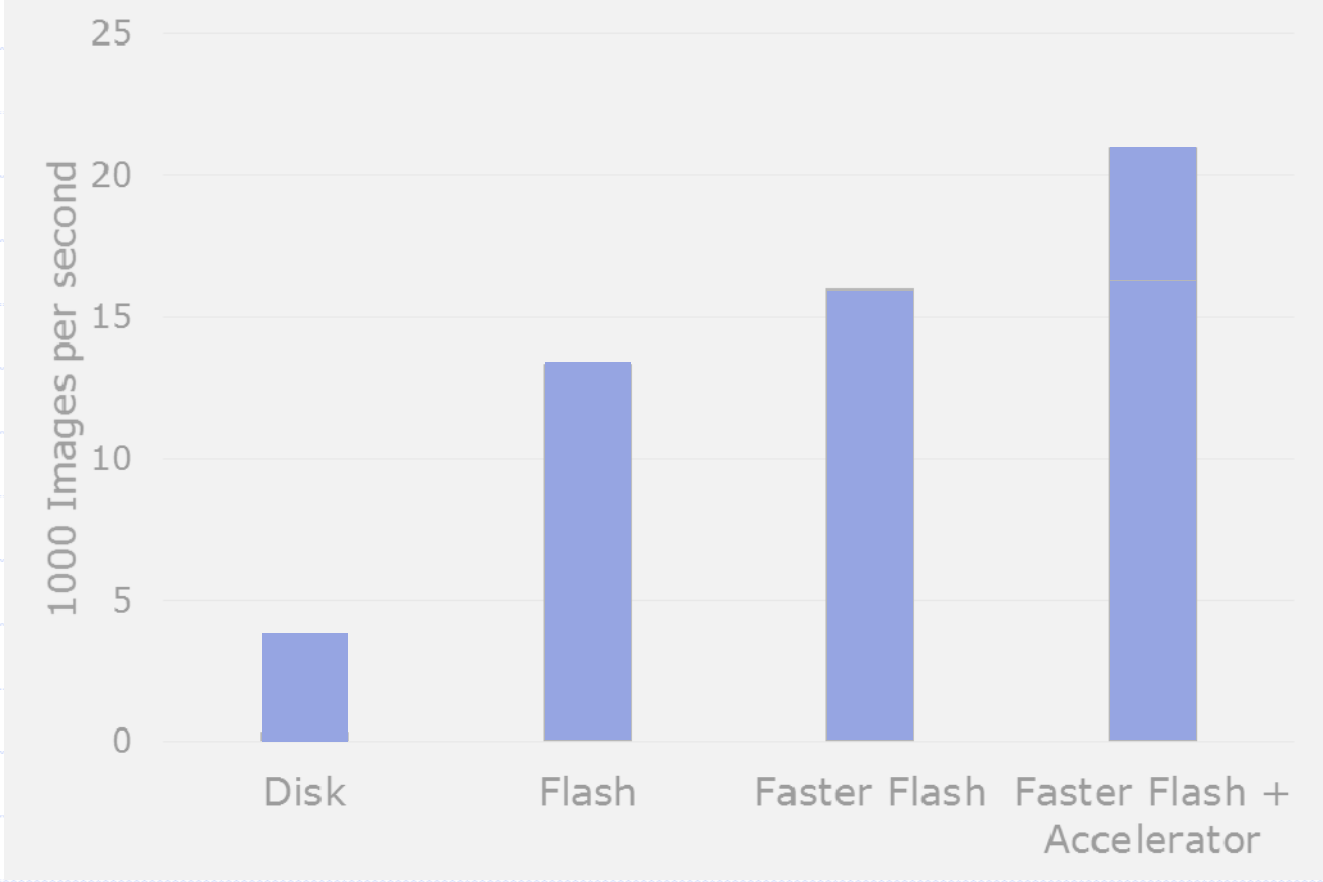
Approximate algorithms to improve performance

- Approximate search algorithms incorporate intelligent sampling methods and improve performance by dramatically reducing the search space
  - e.g., Locality Sensitive Hashing
  - *But introduces random access pattern*



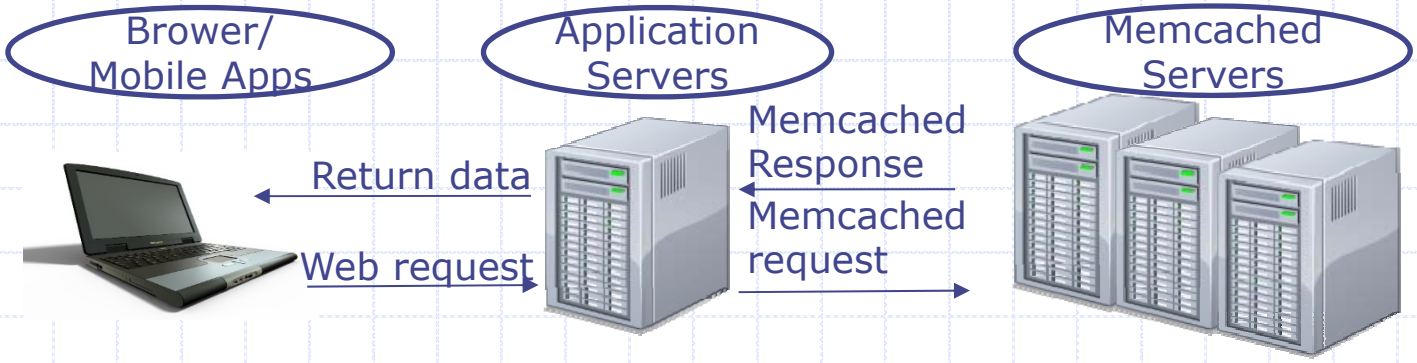
Data accesses corresponding to a single hash table entry results in a lot of random accesses

Disk-based system can't take advantage of sampling



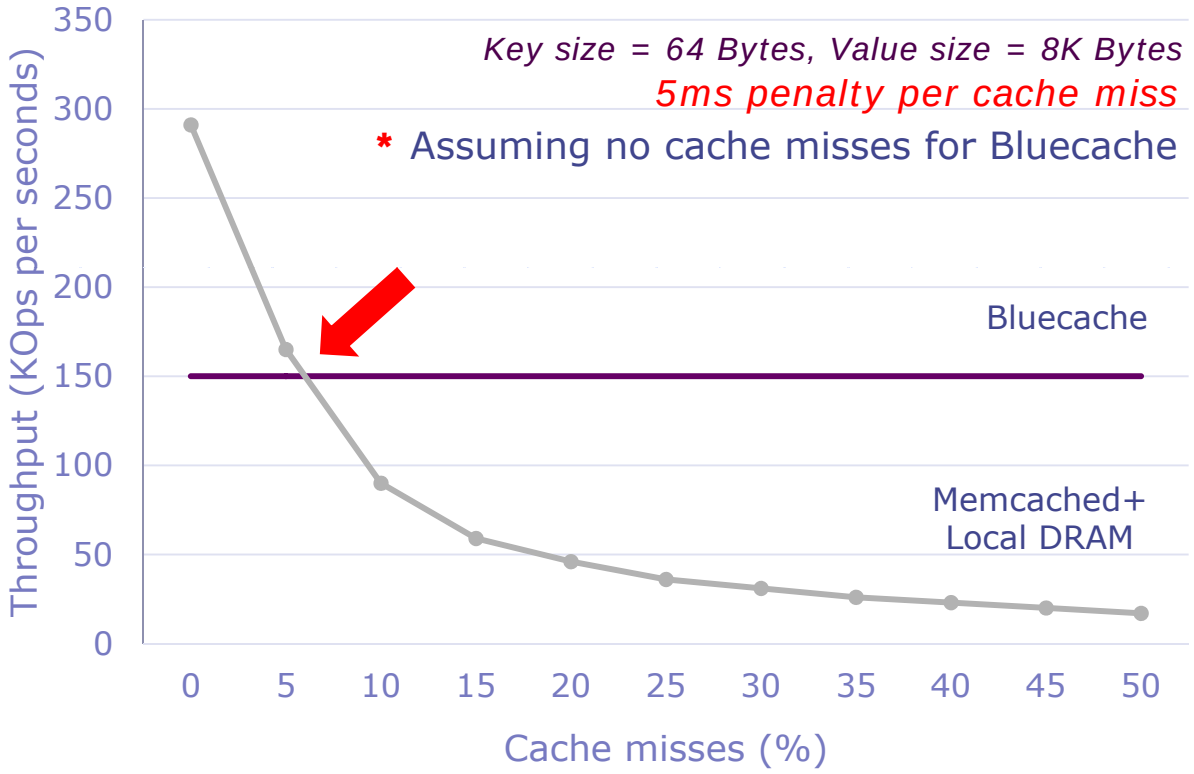
memcached service

- A distributed in-memory key-value store
  - caches DB results indexed by query strings
  - Accessed via socket communication
  - Uses system DRAM for caching (~256GB)
- Extensively used by database-driven websites
  - Facebook, Flicker, Twitter, Wikipedia, Youtube …



Networking contributes 90% of the overhead

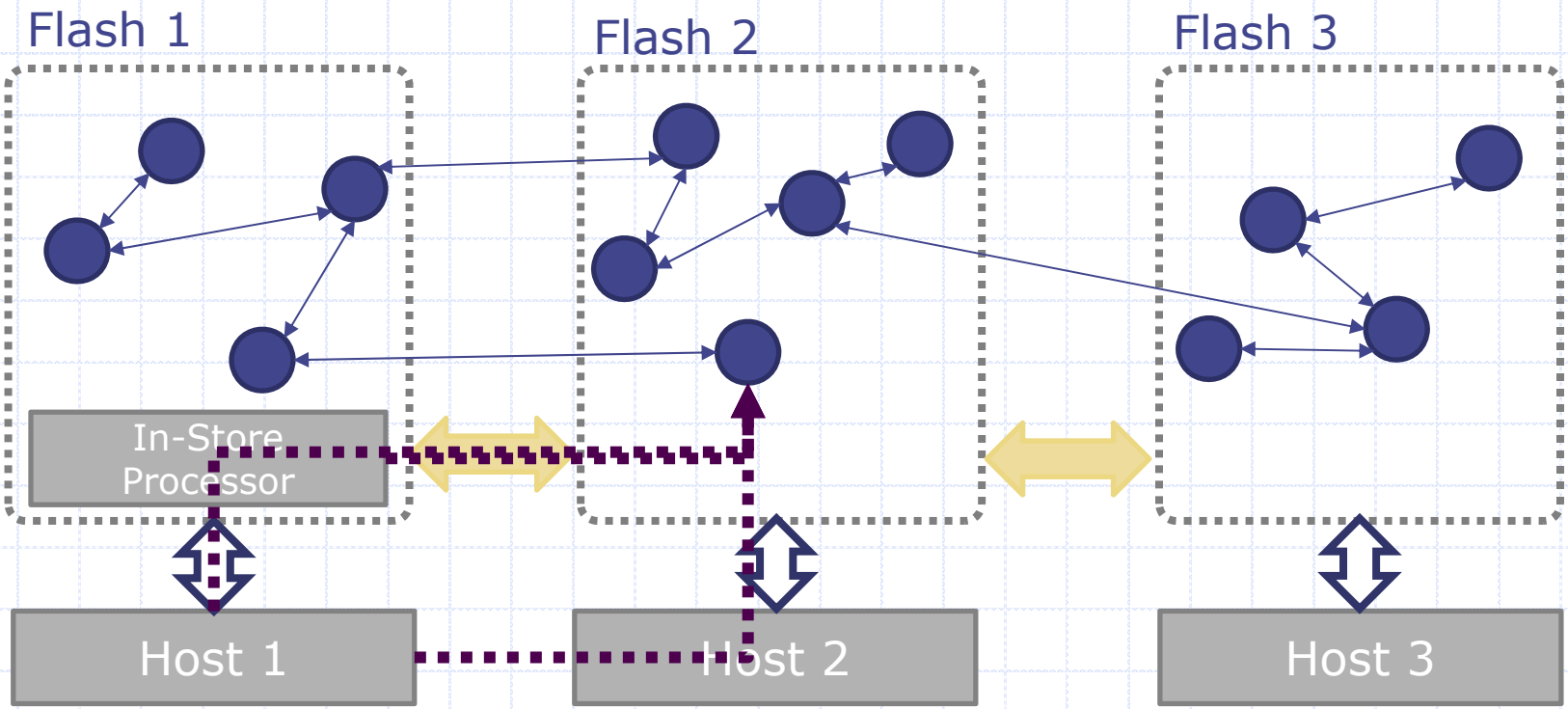
Bluecache: Accelerated memcached service Shuotao Xu



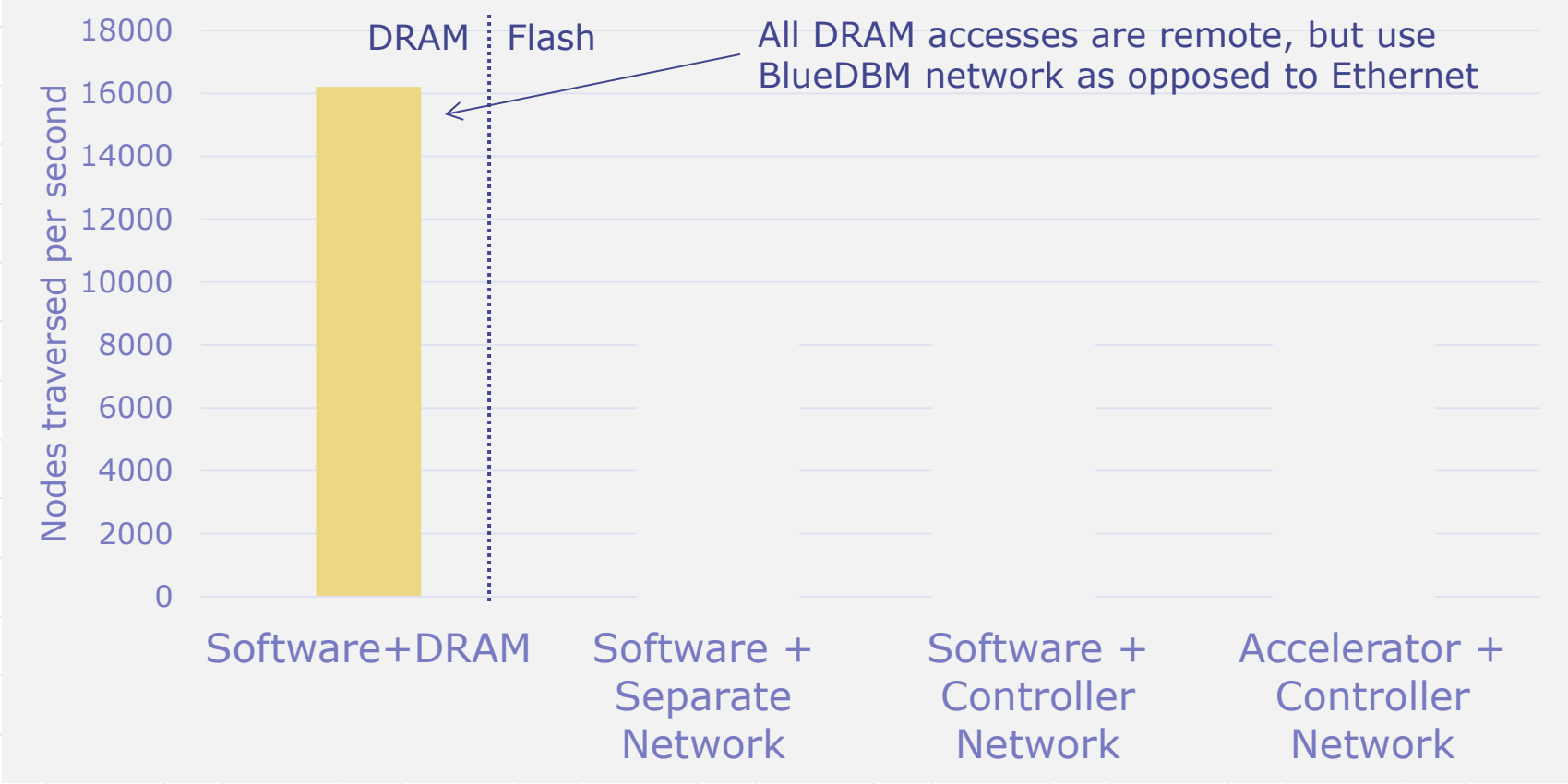
High cache-hit rate outweighs slow flash-accesses (small DRAM vs. large Flash)

Graph traversal

- Latency-bound problem because the next node to be visited cannot be predicted
  - Latency can be reduced by moving the computation closer to data



Graph traversal performance



Flash based system can achieve comparable performance with a much smaller cluster

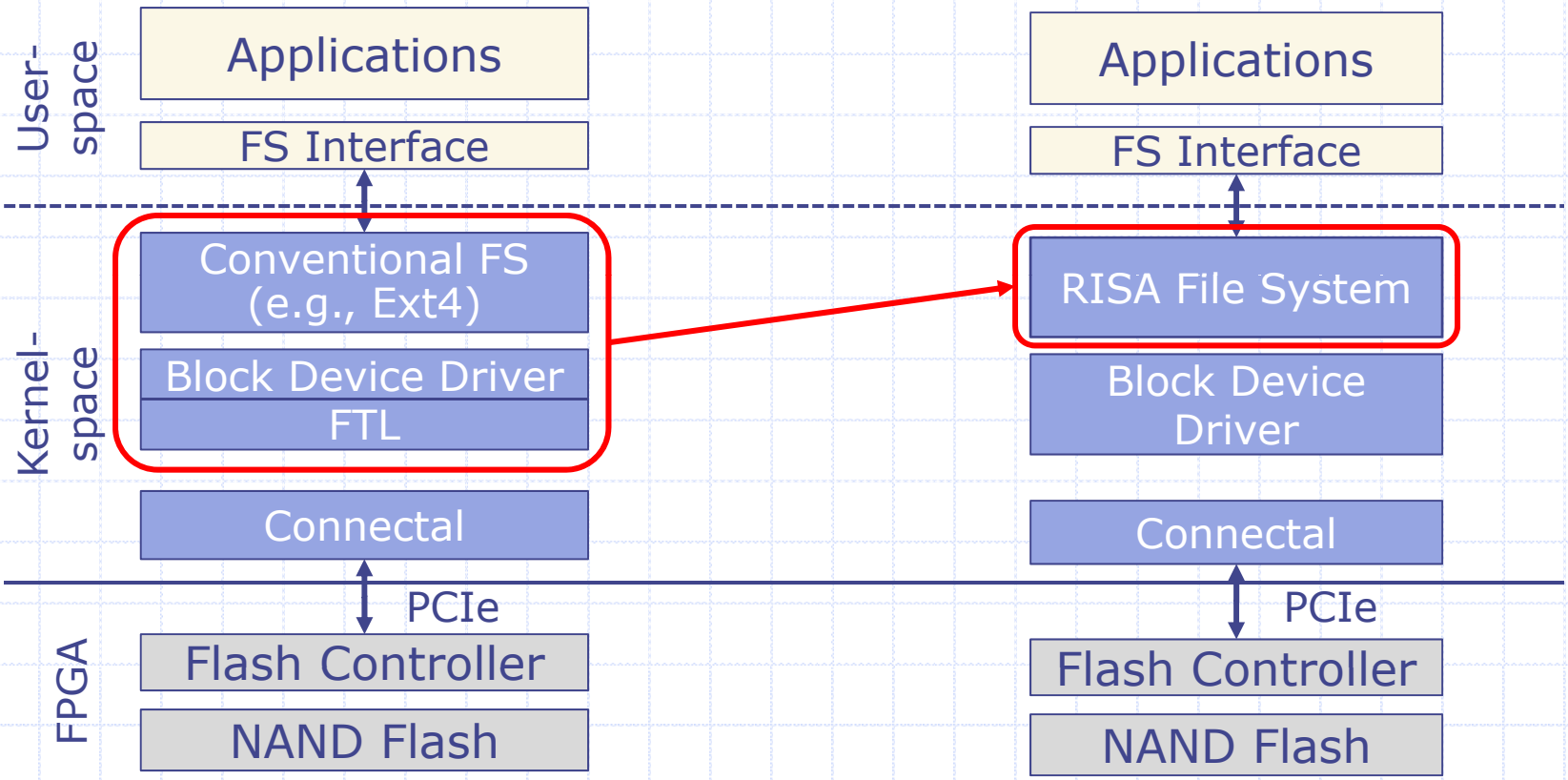
Other potential applications

- Deep machine learning – Convolutional Neural Networks
- Genomics
- Complex graph analytics
- Platform acceleration
  - Spark, MATLAB, SciDB, …

Infrastructure software to help application development

- RISA - distributed flash aware file system
- Connectal: Software to automate the generation of SW-to-HW and HW-to-SW codes for an interface definition
- Communications library
- map-reduce applications support
- ...

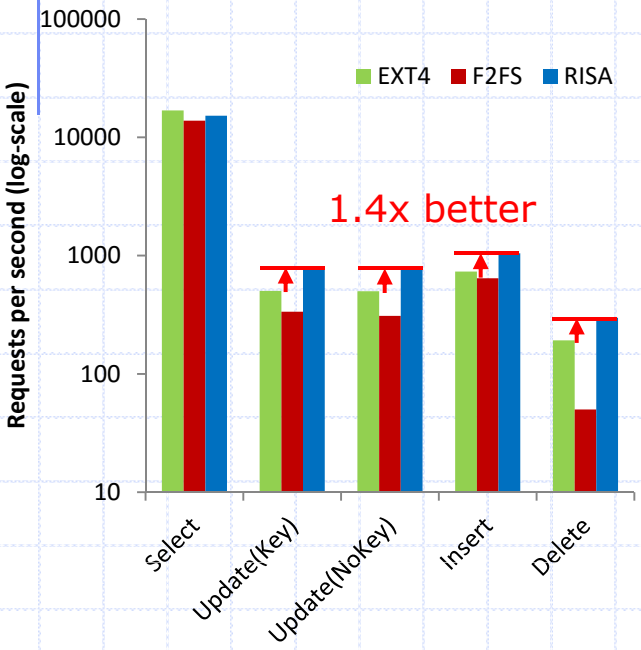
RISA: a new flash-optimized file system Sungjin Lee



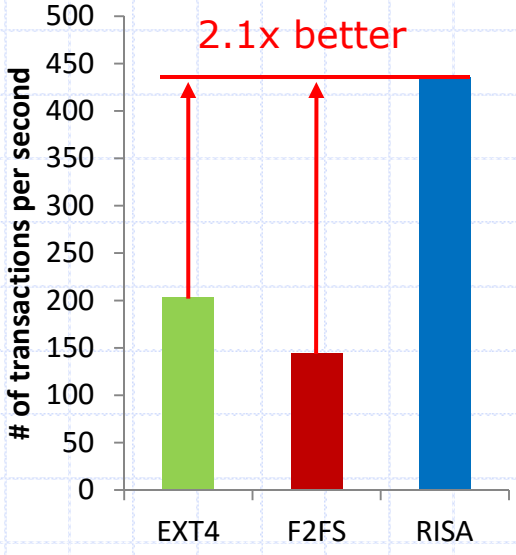
- Flash storage on all nodes is accessible from any node
- Currently, distributed storage supports shared read-space but disjoint write-spaces

Comparison of three different file systems [FAST 2016]

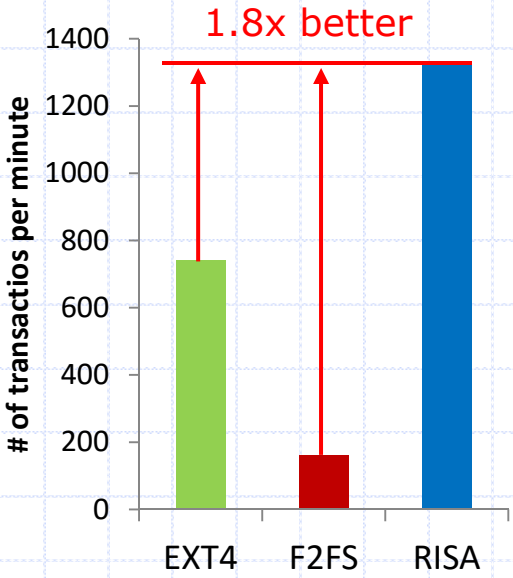
- EXT4: Well-known Linux EXT4 file system
- F2FS: Samsung's F2FS file system
- RISA: Our software solution optimized for BlueDBM



Non-Trans



OLTP



TPC-C

Conclusion

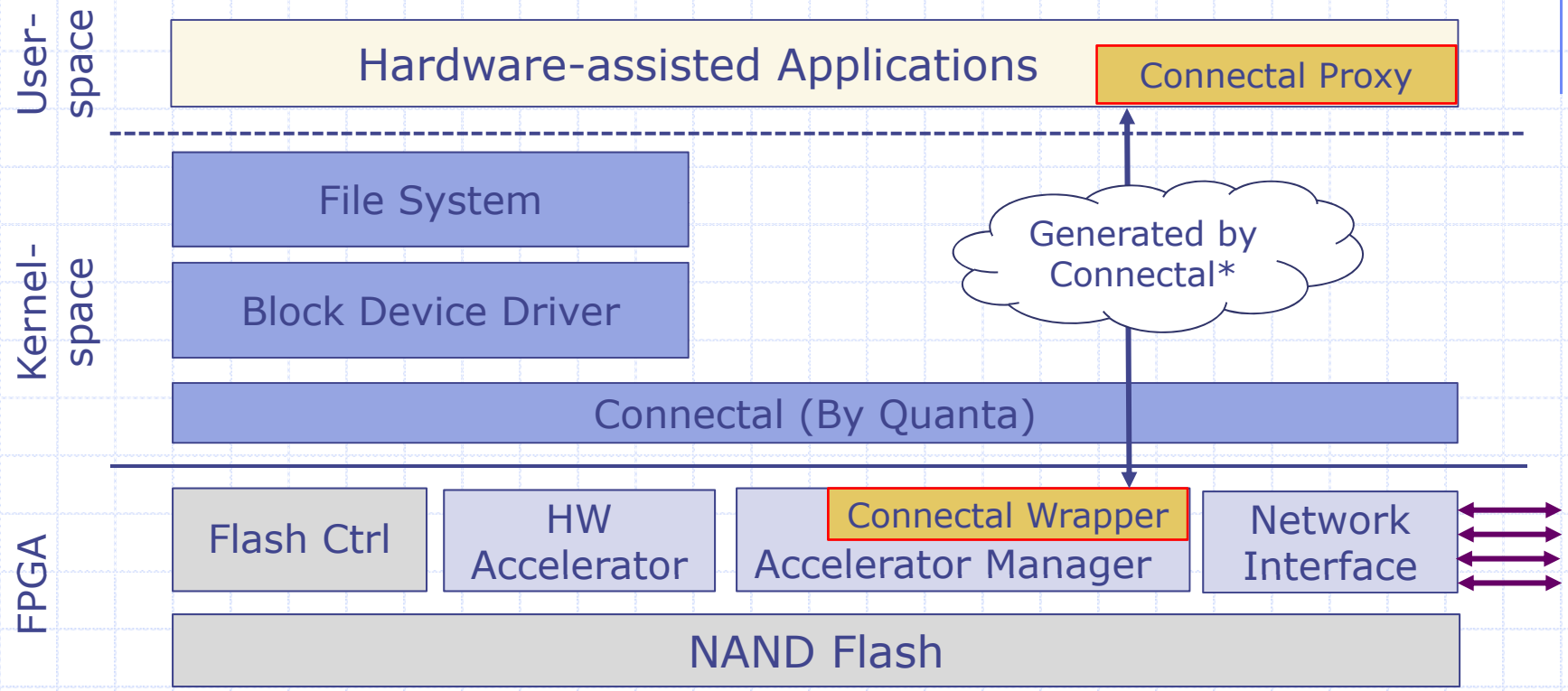
- Fast flash-based distributed storage systems with low-latency random access may be a good platform to support complex queries on Big Data
- Reducing access latency for distributed storage requires architectural modifications, including in-storage processors and fast storage networks
- Flash-based analytics hold a lot of promise, and we plan to continue demonstrating more application acceleration

Thank you

Related work

- Use of flash
  - SSDs, FusionIO, Purestorage
  - Zetascale
  - SSD for database buffer pool and metadata [SIGMOD 2008], [IJCA 2013]
- Networks
  - QuickSAN [ISCA 2013]
  - Hadoop/Spark on Infiniband RDMA [SC 2012]
- Accelerators
  - SmartSSD[SIGMOD 2013], Ibex[VLDB 2014]
  - Catapult[ISCA 2014]
  - GPUs

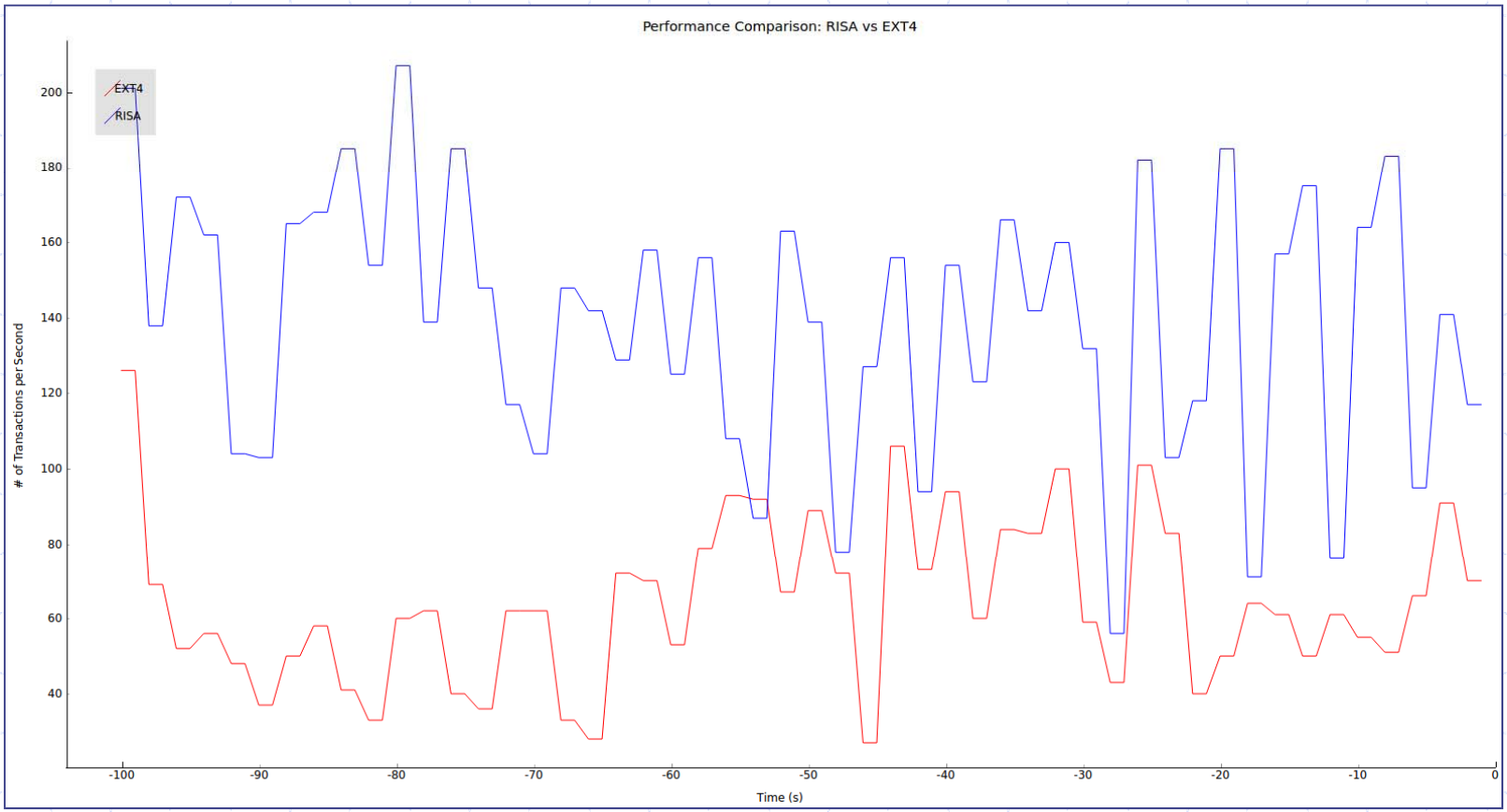
BlueDBM software view



- BlueDBM provides a generic file system interface as well as an accelerator-specific interface (Aided by Connectal)

Performance: RISA *vs* EXT4

TPC-C workloads



Hardware improvements

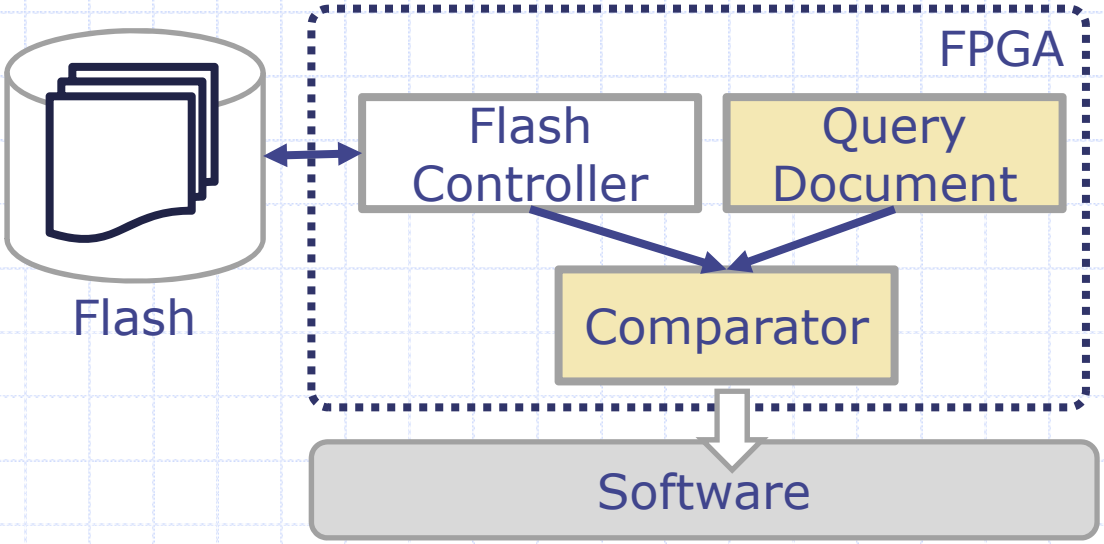
- FPGA is being used for three distinct purposes
  - Flash controller
  - Routers in a configurable network
  - Accelerators

These functionalities can be supported by different chips. For example, a general purpose processor may simplify accelerator programming

Document search accelerator

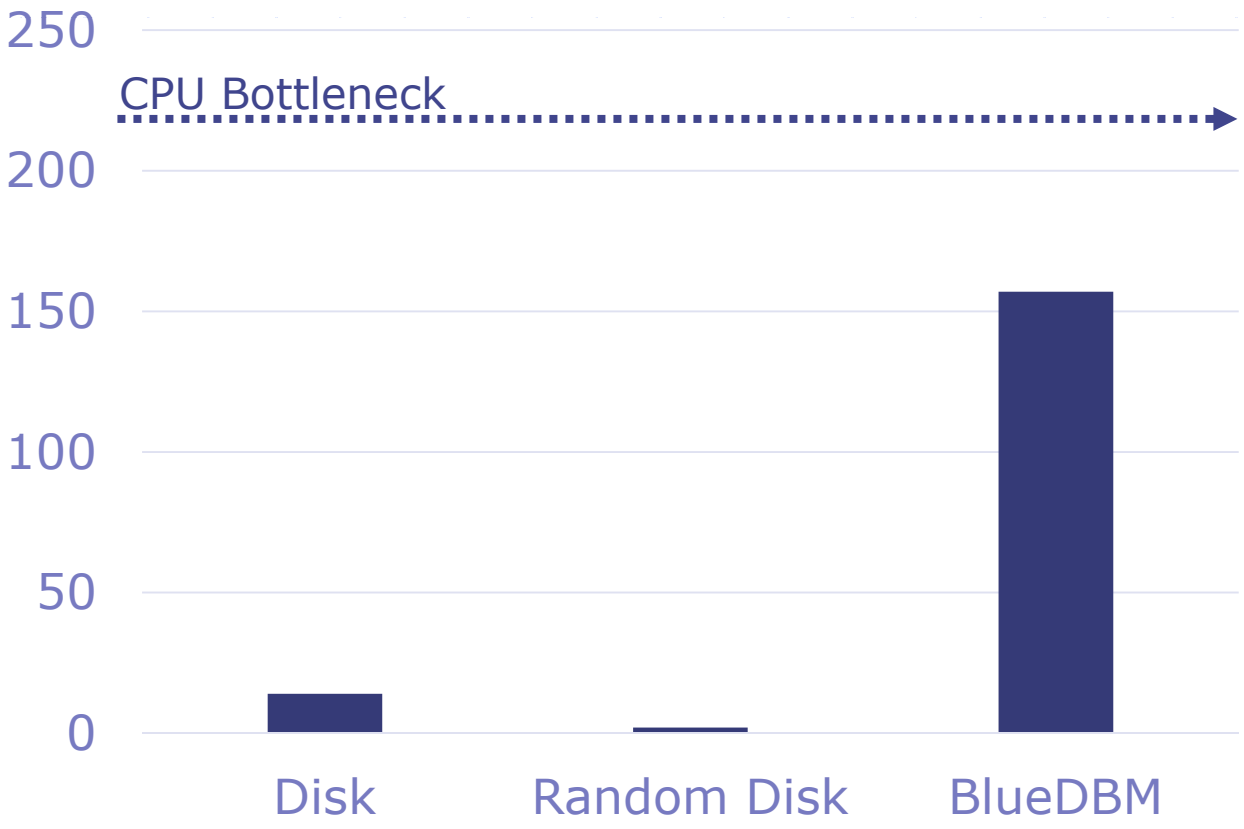
Sang woo Jun, Chanwoo Chung

- Useful for web search, plagiarism detection and document clustering
- Distance determined by the amount of words two documents share
  - Documents are pre-processed into tuples of words and occurrence count
  - Example of a much simpler distance metric

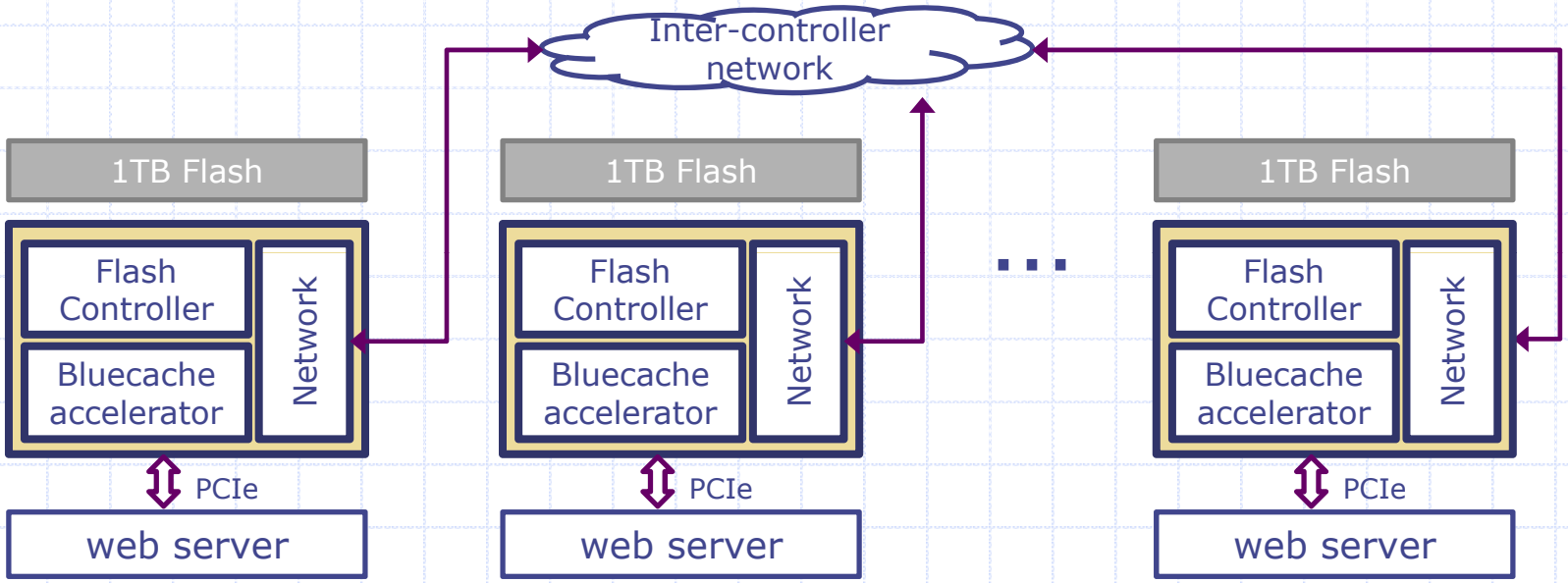


Document search performance

- Even with a very computationally light distance metric, BlueDBM performs comparable to a DRAM-based system

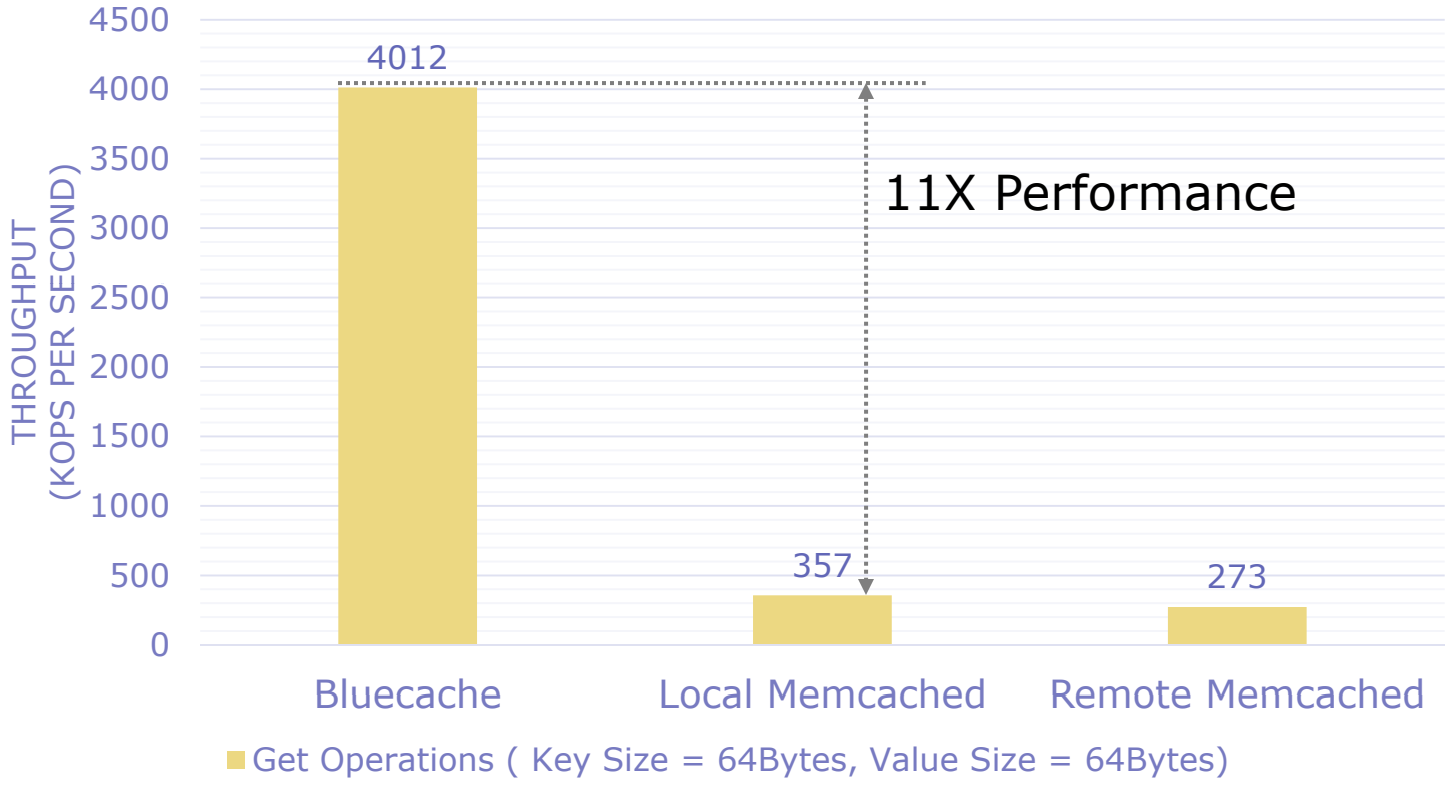


Bluecache: Accelerated memcached service Shuotao Xu



- Memcached server implemented in hardware
  - Hashing and flash management implemented in FPGA
  - 1TB hardware managed flash cache per node
- Hardware server accessed via local PCIe
- Direct network between hardware

Effect of hardware support for networking (no flash, only DRAM)



- PCIe DMA and inter-controller network reduces access overhead
- FPGA acceleration of memcached is effective