Affordable Analytics Using Hardware Accelerated Flash Storage

Sang-Woo Jun

Assistant Professor

Department of Computer Science

University of California, Irvine

2019-10-15

Presentation Overview

Cost/Power-Efficient Analytics

↑

FPGA-Based Hardware Acceleration

+

Processing In Flash Storage

+

Magic Sauce What is interesting!

Stating The Obvious

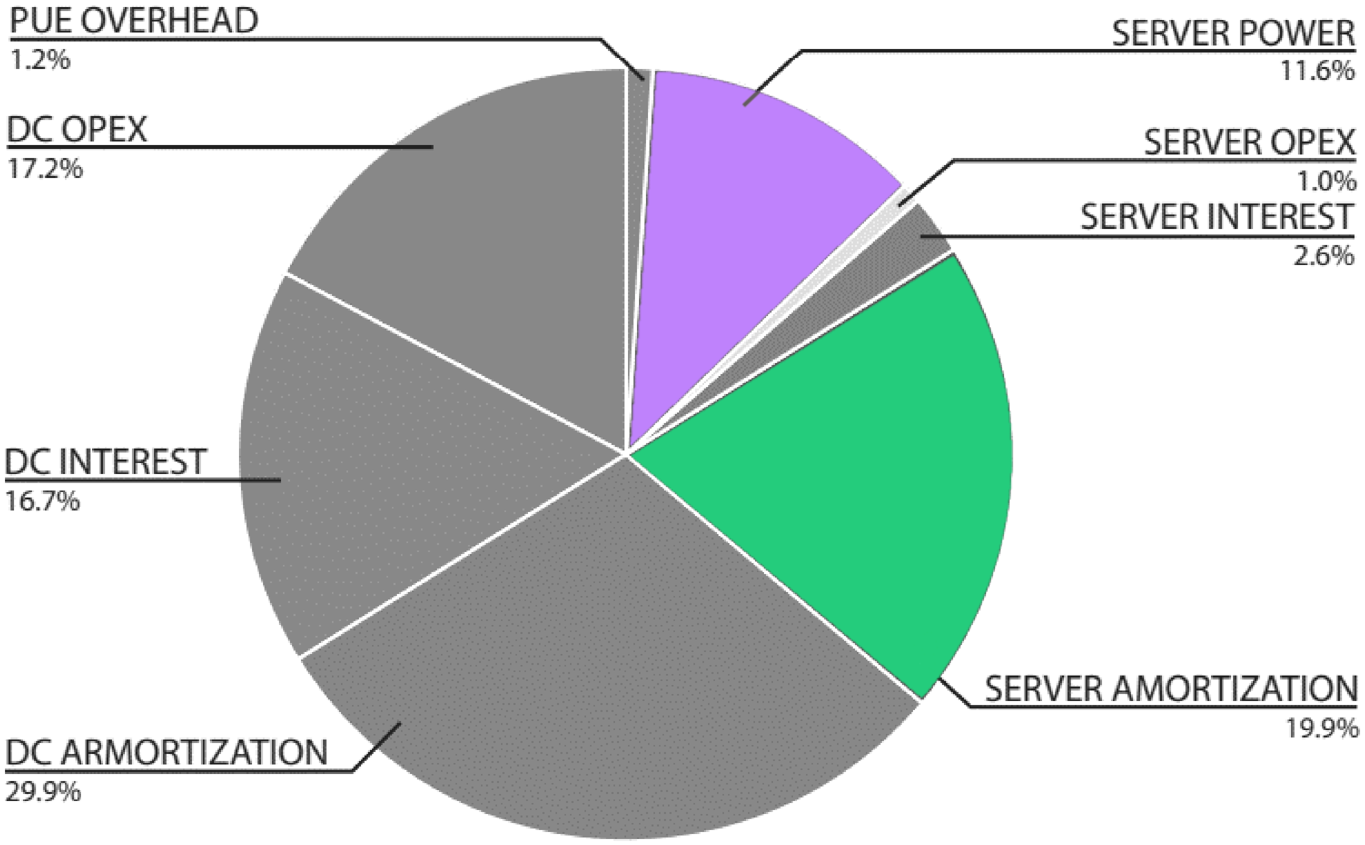
“We need to build efficient systems”

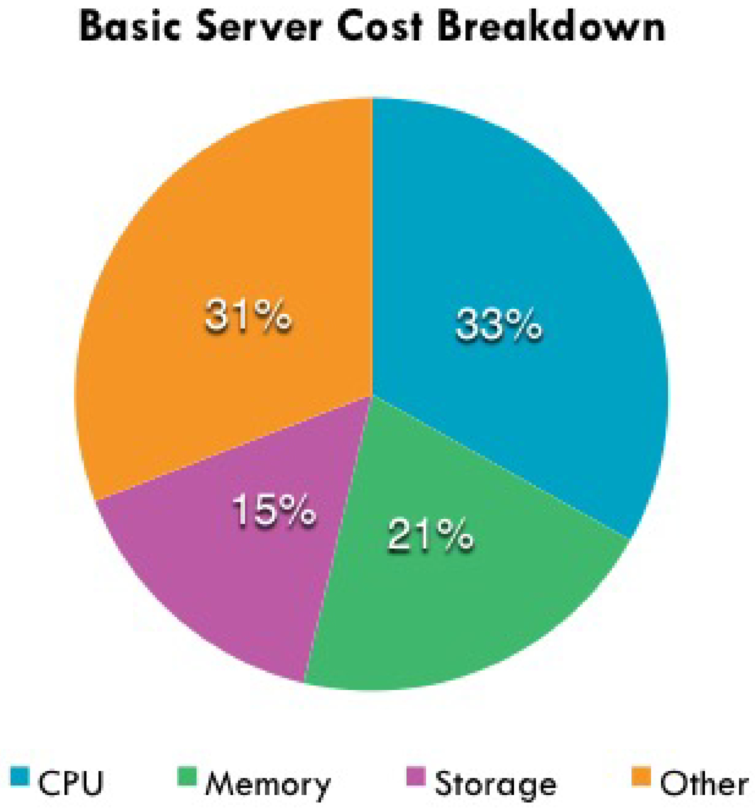
- Captain Obvious

Hotels.com mascot

Where Does The Cost Come From?

(TCO of a realistic, partially filled data center)



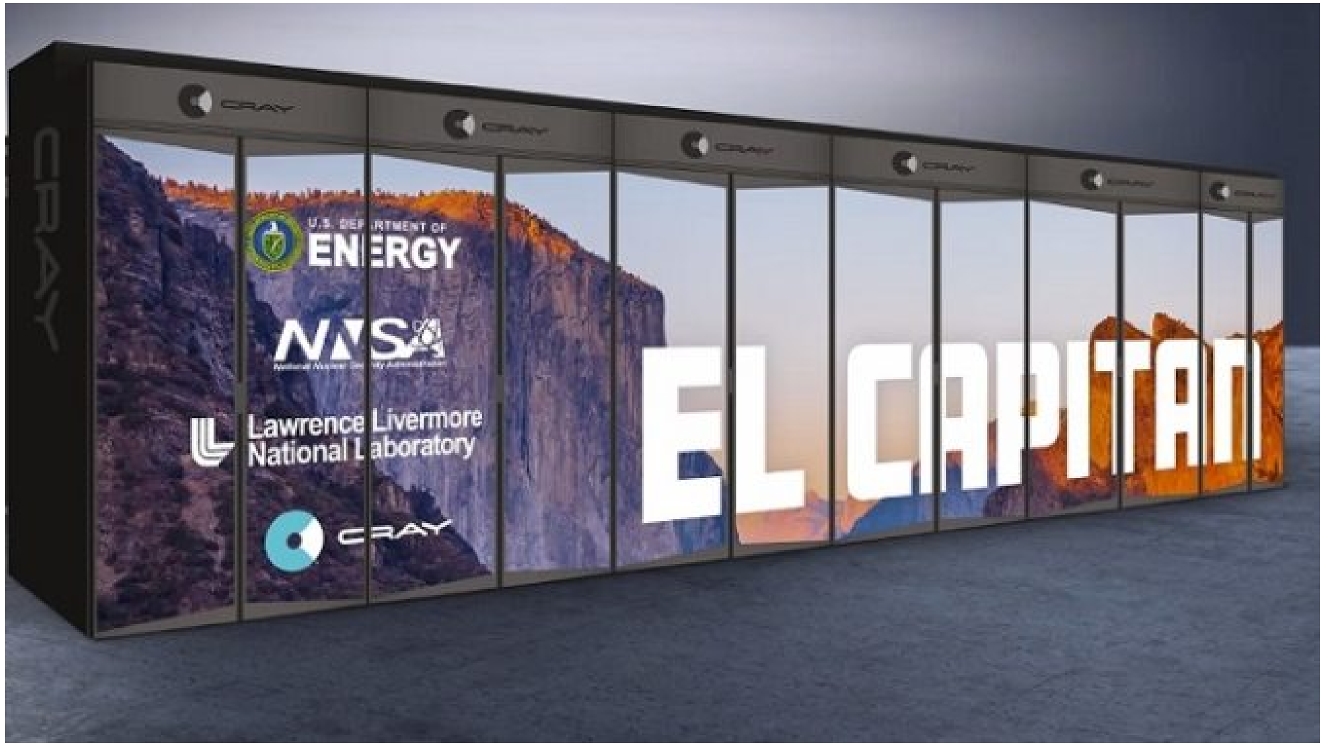


Source: ARK Investment Management LLC | ark-invest.com

Barroso et.al., “The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition,” 2018

The Scale of Power Consumption

Department of Energy requests an exaflop machine by 2020



1,000,000,000,000,000,000 floating point operations per second

Using 2016 technology, 600 MW

Using 2019 technology, 200 MW

Palo Verde Nuclear Generating Station

MIT Research nuclear reactor

3,937 MW

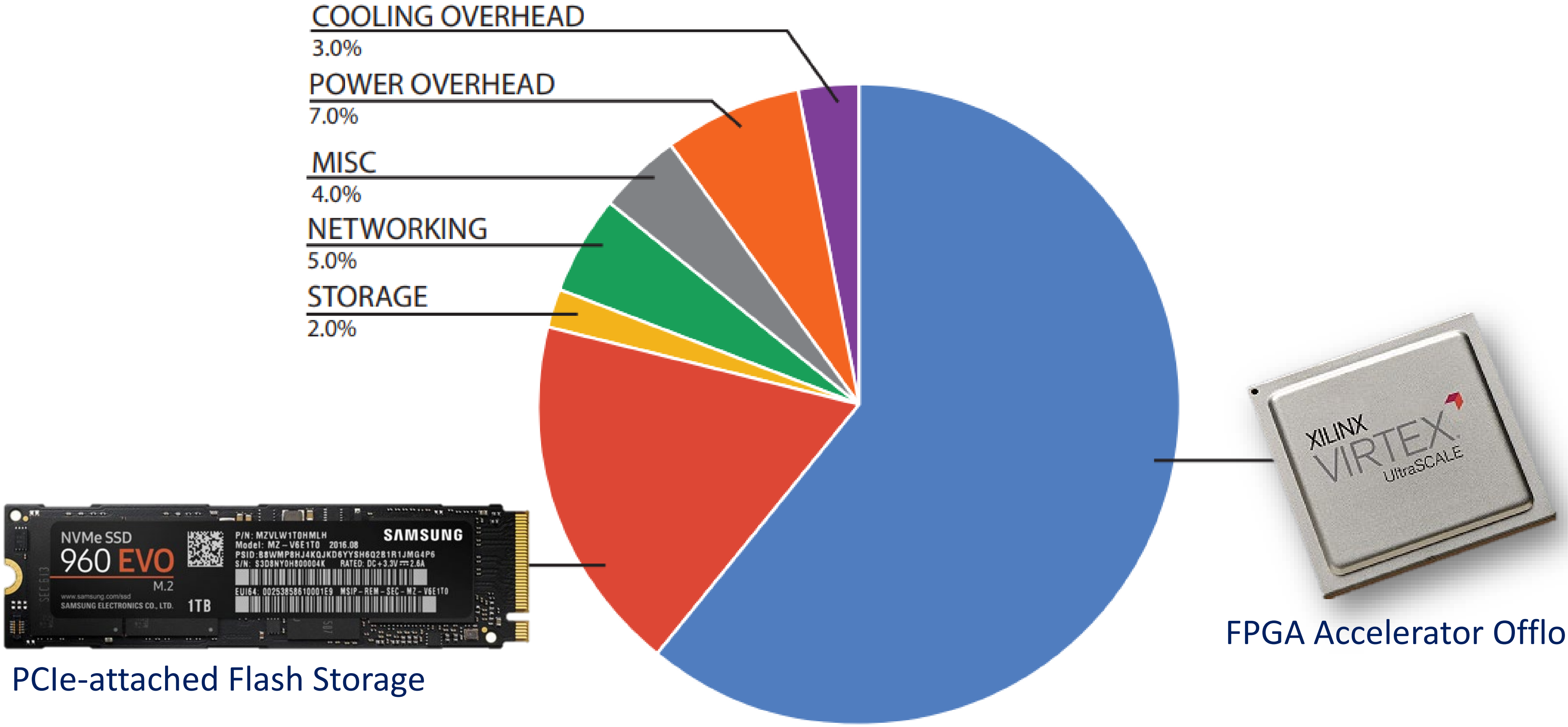
6 MW

Total residential power consumption of San Francisco: 168 MW

(Source: California Energy Commission 2018)

Lynn Freeny, Department of Energy

Warehouse-Scale Computer Power Consumption Profile



PCIe-attached Flash Storage

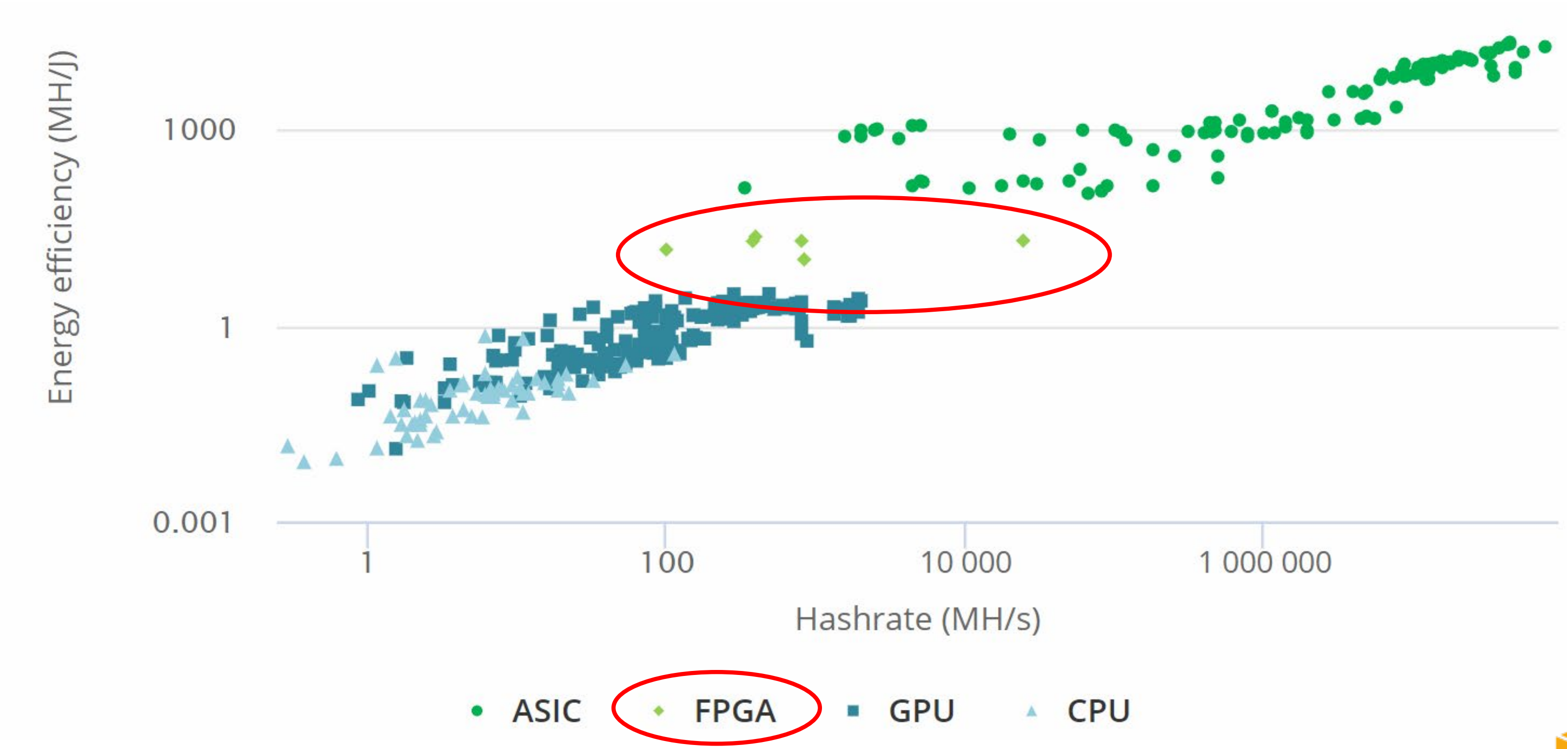
FPGA Accelerator Offloading

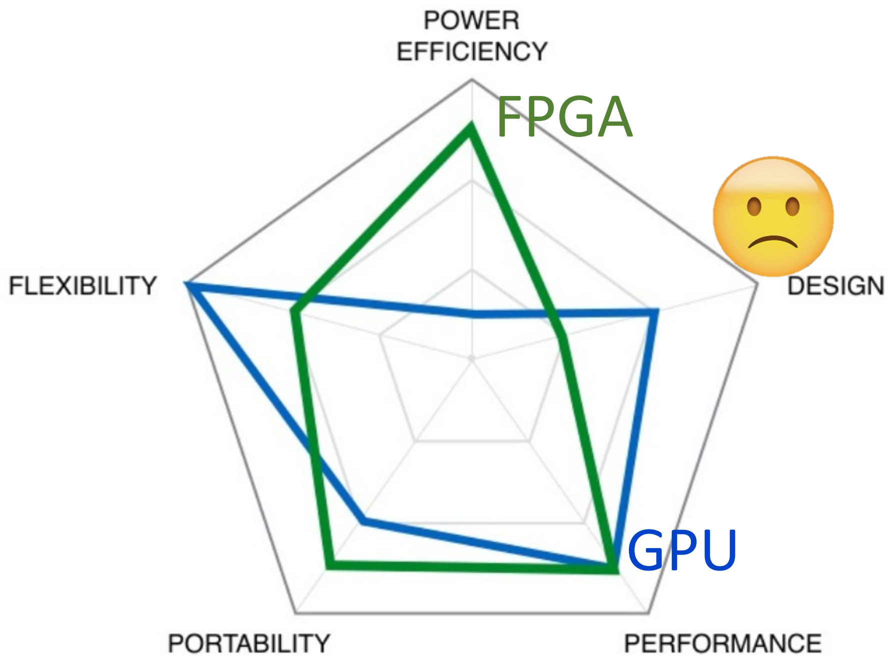
Info-Tech, “Top 10 energy-saving tips for a greener data center,” Info-Tech Research Group, 2010
Source: Google, 2017 -- Barroso et.al., “The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition,” 2018

FPGA-Based
Reconfigurable Hardware Acceleration

- ***Field-Programmable*** Gate Array
- Can be configured to act like any circuit – Optimized for the application
- High performance, Low power

An Application Anecdote: Bitcoin Mining!





amazon web services™ EC2

Azure

Baidu Cloud

International Energy Agency, “Commentary: Bitcoin energy use - mined the gap”, 2019

Bracco Filippo, “Rationale behind FPGA”, 2017

Flash Storage for Analytics

Fine-grained,

Irregular access

Terabytes in size

TB of DRAM

$$$ $5,000/TB, >200W

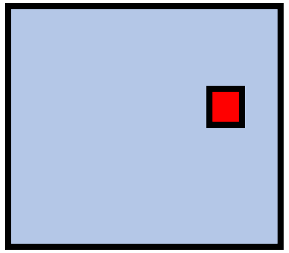
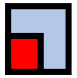
Our goal:

$ $300/TB, <10W

$100/TB, <5W

Drop-in replacement causes sharp performance decline

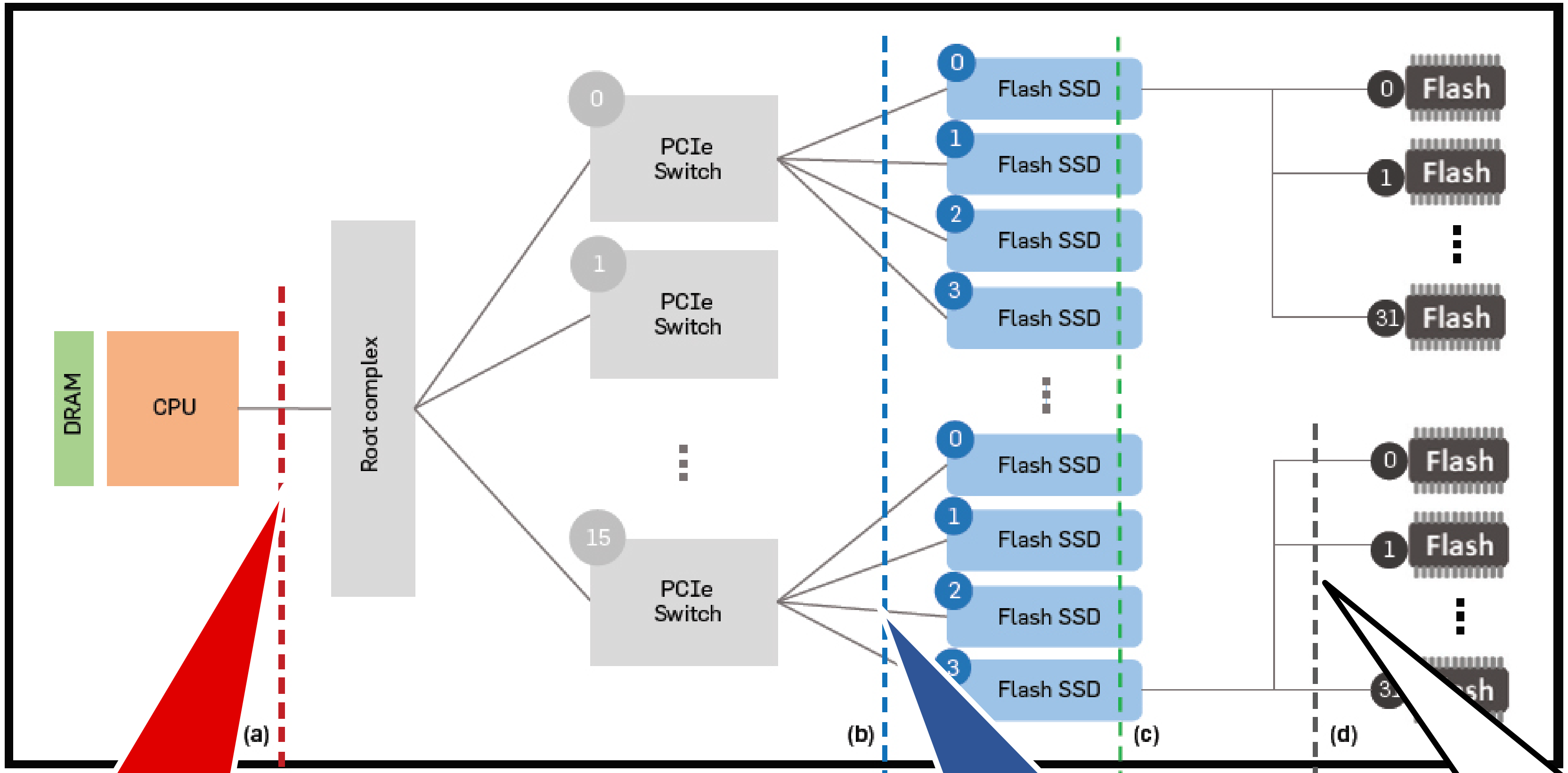
Challenge 1: Random Access Performance

	Flash	DRAM
Bandwidth:	~10 GB/s	~100 GB/s
Latency:	~10 us	~100 ns
Access Granularity:	 8192 Bytes	 128 Bytes

Wastes performance by not using most of fetched page

Using 8 bytes in a 8192 Byte page uses 1/1024 of bandwidth!

Challenge 2: Data Movement



16 lanes PCIe Gen 3
= ~16 GB/s

8x Bandwidth Gap!

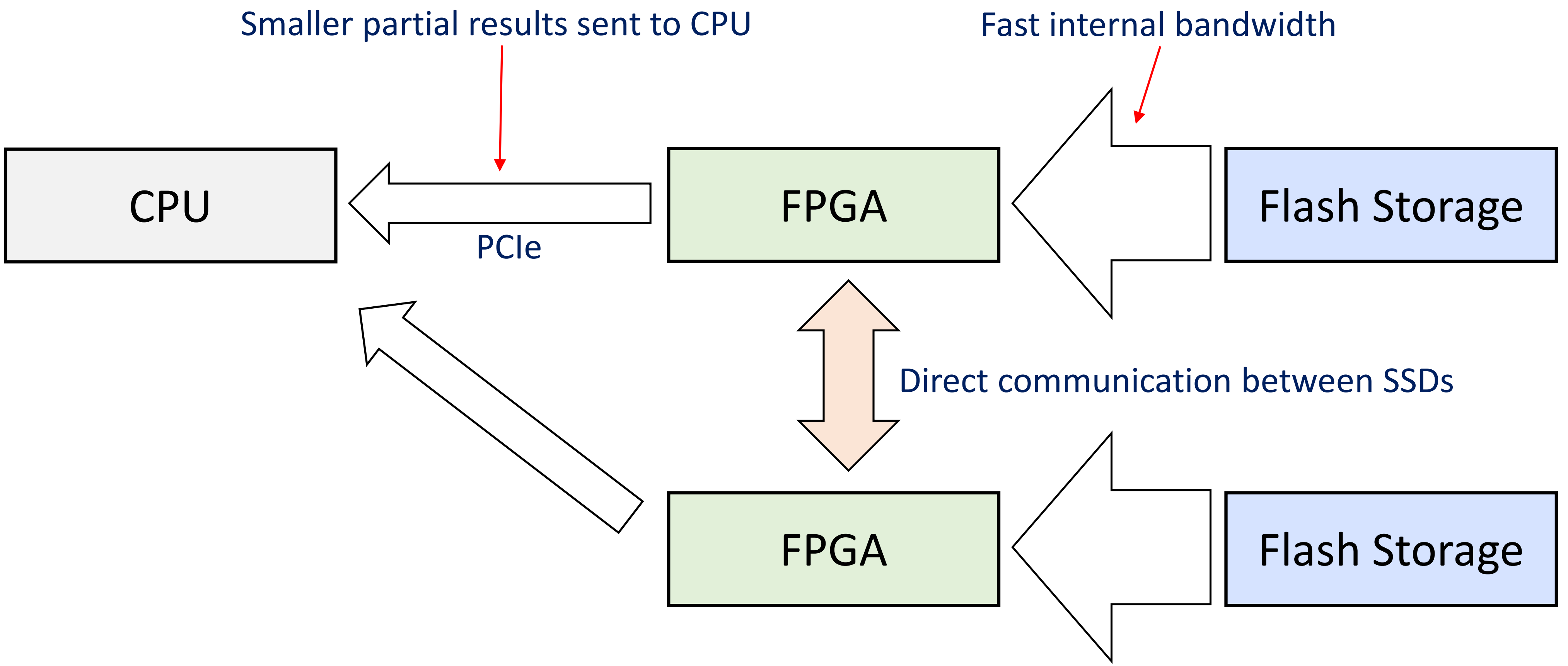
64 SSDs x ~2 GB/s
= ~128 GB/s

Internal bandwidth
faster than its PCIe

2.5x Internal read BW
(Source: Samsung)

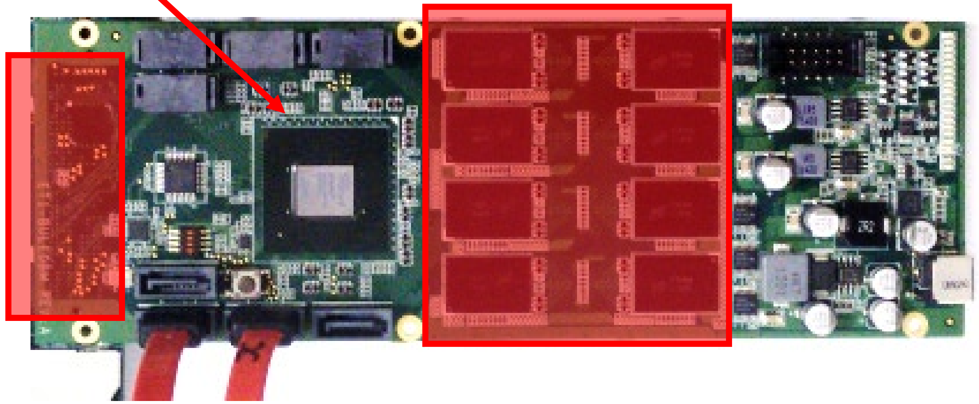
Jaeyoung Do et.al., “Programmable Solid-State Storage in Future Cloud Datacenters,” Communications of the ACM 2019

Solution Part 1/2: Handling Data Movement With Near-Storage Computation


Smaller partial results sent to CPU
Fast internal bandwidth
CPU
PCIe
FPGA
Flash Storage
Direct communication between SSDs
FPGA
Flash Storage

BlueDBM: Custom Flash Card for Distributed Accelerated Flash Architectures (2015)

- ❑ Requirement 1: Modify flash management
- ❑ Requirement 2: Dedicated storage-area network
- ❑ Requirement 3: In-storage hardware accelerator



Artix 7 FPGA
(Flash Controller)

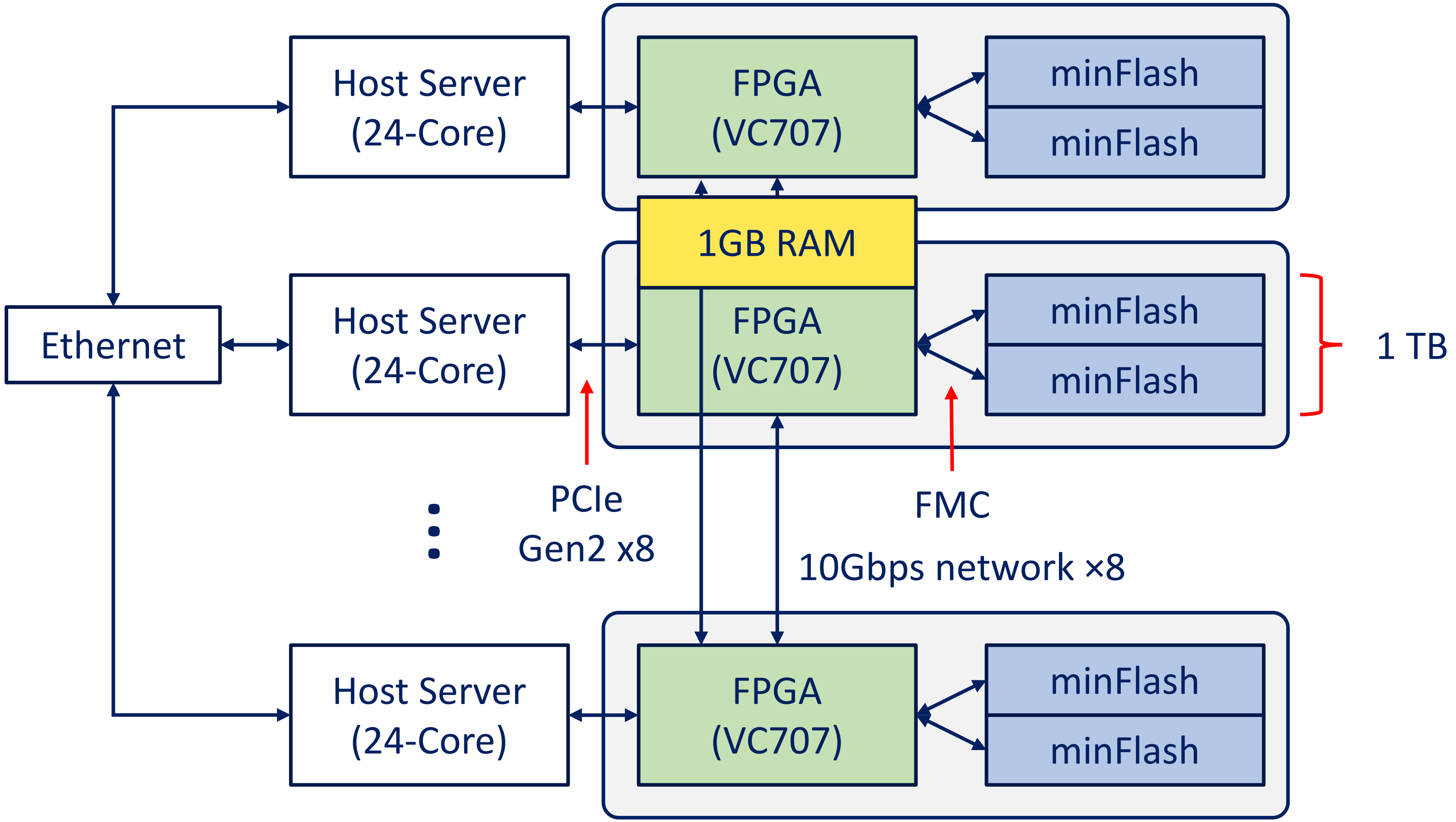
0.5 TB Flash

FMC Connection
to FPGA board
(40 Gbps)

Inter-Controller Network
(10Gbps/lane ×4)

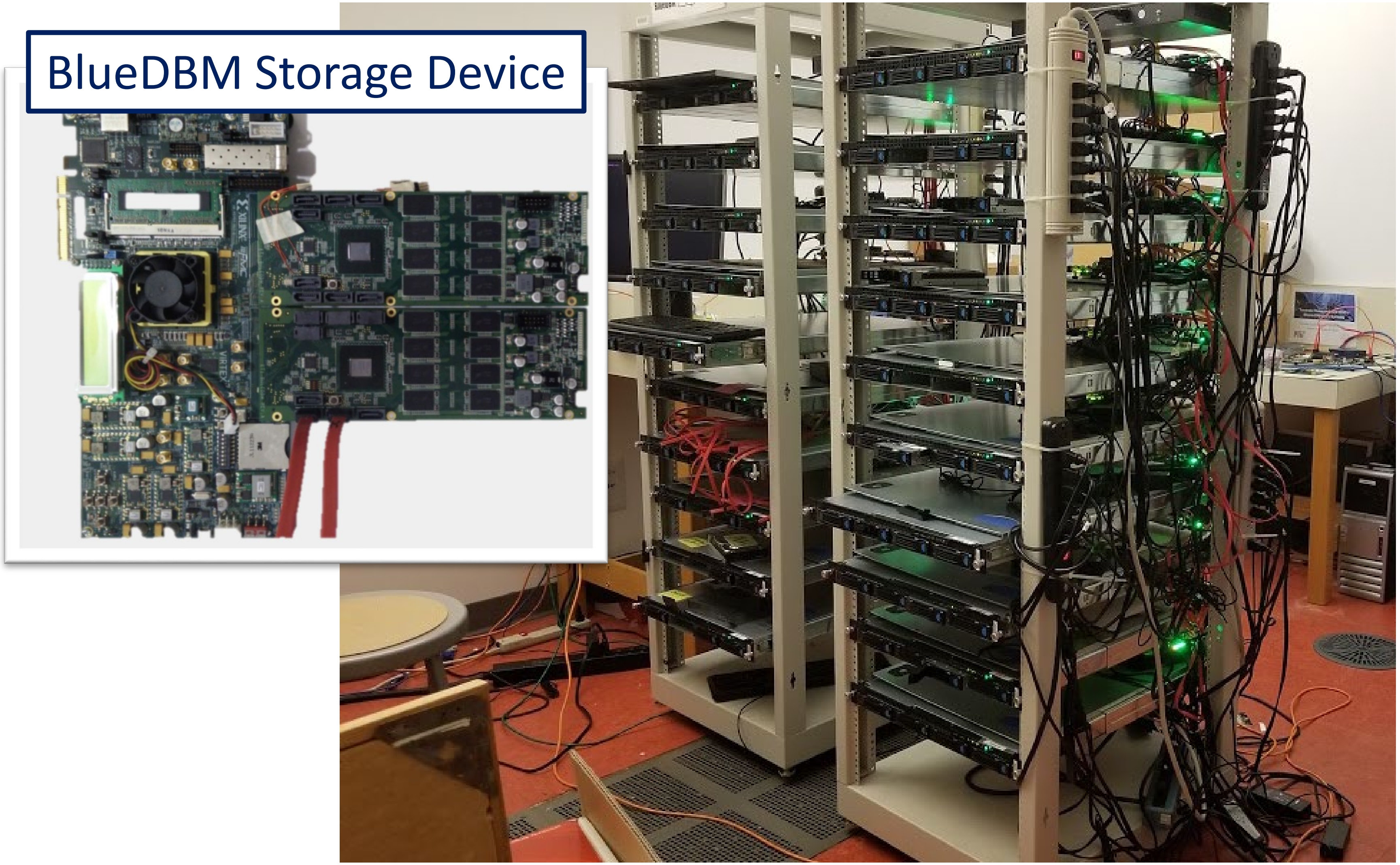
"minFlash: A Minimalistic Clustered Flash Array," DATE 2016

BlueDBM Cluster Architecture



Uniform latency of 100 μs!

The BlueDBM Cluster



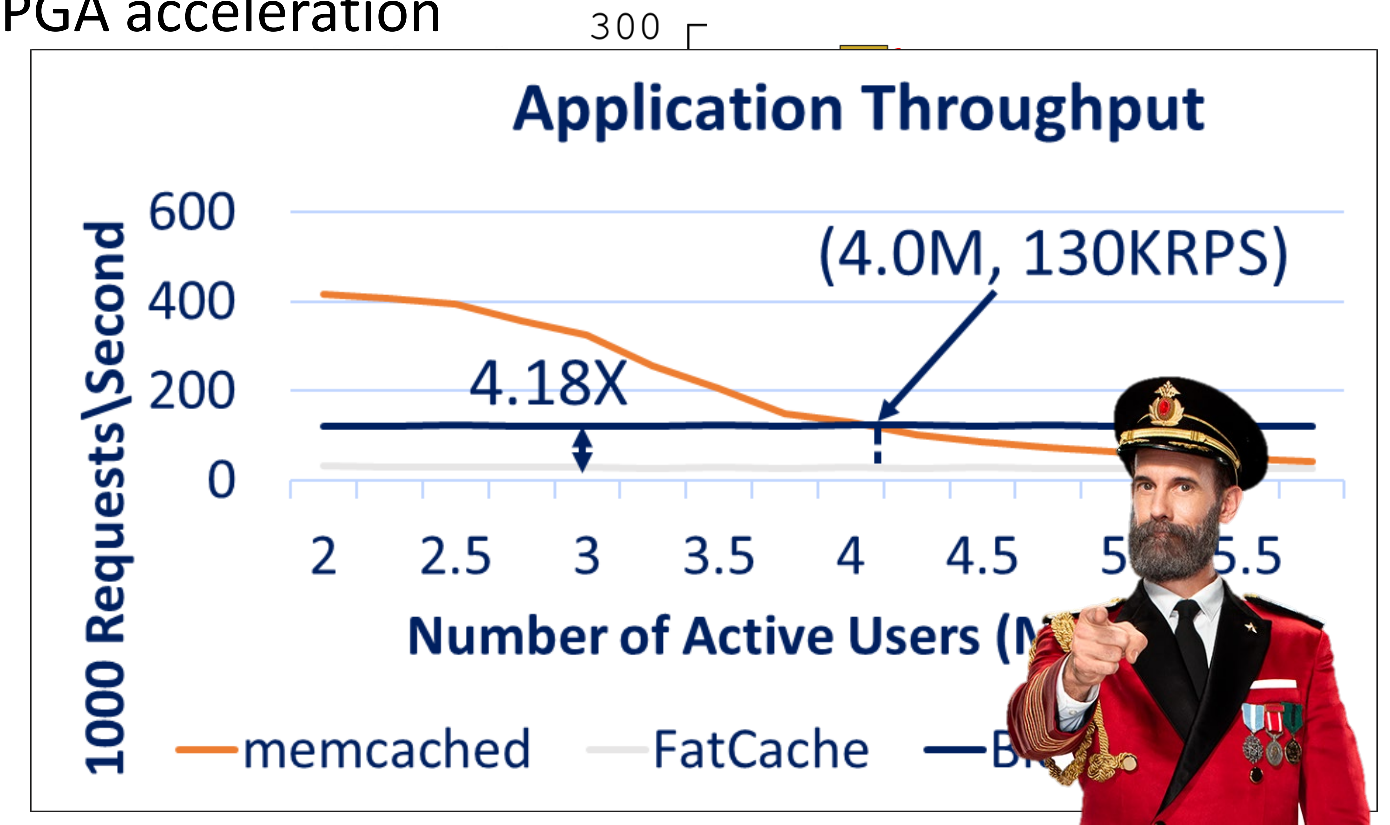
Research Enabled by BlueDBM

1. “Scalable Multi-Access Flash Store for Big Data Analytics,” FPGA 2012
2. “BlueDBM: An Appliance for Big Data Analytics,” ISCA 2015
3. “A Transport-Layer Network for Distributed FPGA Platforms,” FPL 2015
4. “Large-scale high-dimensional nearest neighbor search using Flash memory with in-store processing,” ReConFig 2015
5. “minFlash: A Minimalistic Clustered Flash Array,” DATE 2016
6. “Application-managed flash,” FAST 2016
7. “In-Storage Embedded Accelerator for Sparse Pattern Processing,” HPEC 2016
8. “Terabyte Sort on FPGA-Accelerated Flash Storage,” FCCM 2017
9. “BlueCache: A Scalable Distributed Flash-based Key-value Store,” VLDB 2017
10. “GraFBoost: Using accelerated flash storage for external graph analytics,” ISCA 2018
11. “LightStore: Software-defined Network-attached Key-value Drives,” ASPLOS 2019

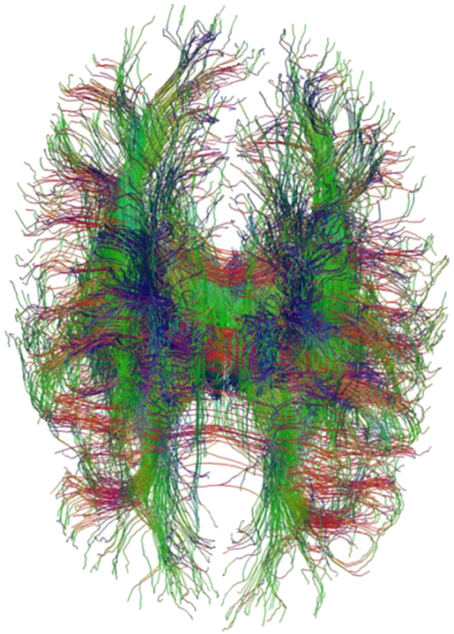
More to come!

Solution Part 2/2: Handling Access Granularity

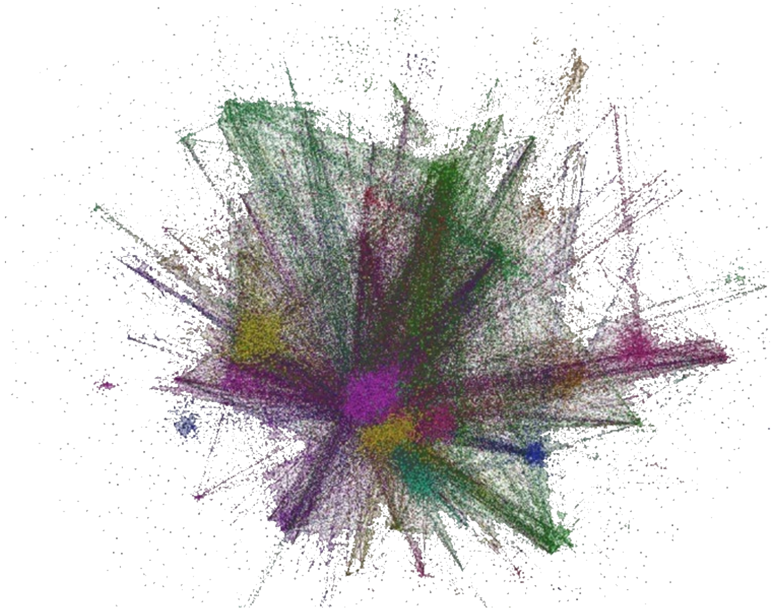
- ❑ The Magic Sauce …is nothing special
- ❑ "Use algorithms optimized for system characteristics"
  - ○ e.g., Flash storage, Near-storage FPGA acceleration
- ❑ Applications
  - ○ Database joins
  - ○ Genomic mutation detection
  - ○ Key-value cache



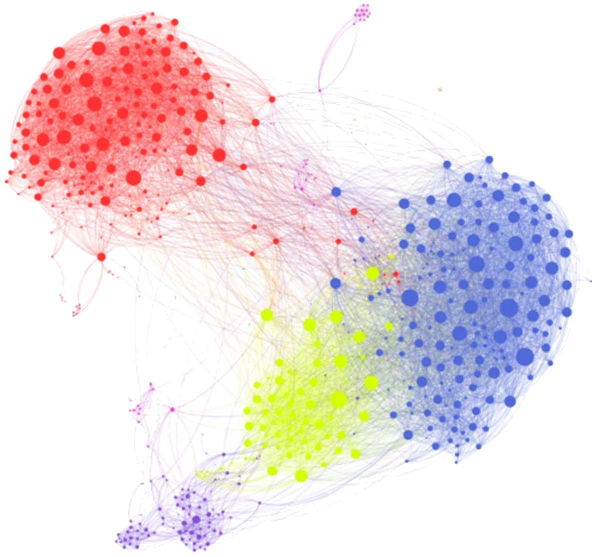
A Detailed Example: Graph Analytics



Human neural network



Structure of the internet



Social networks

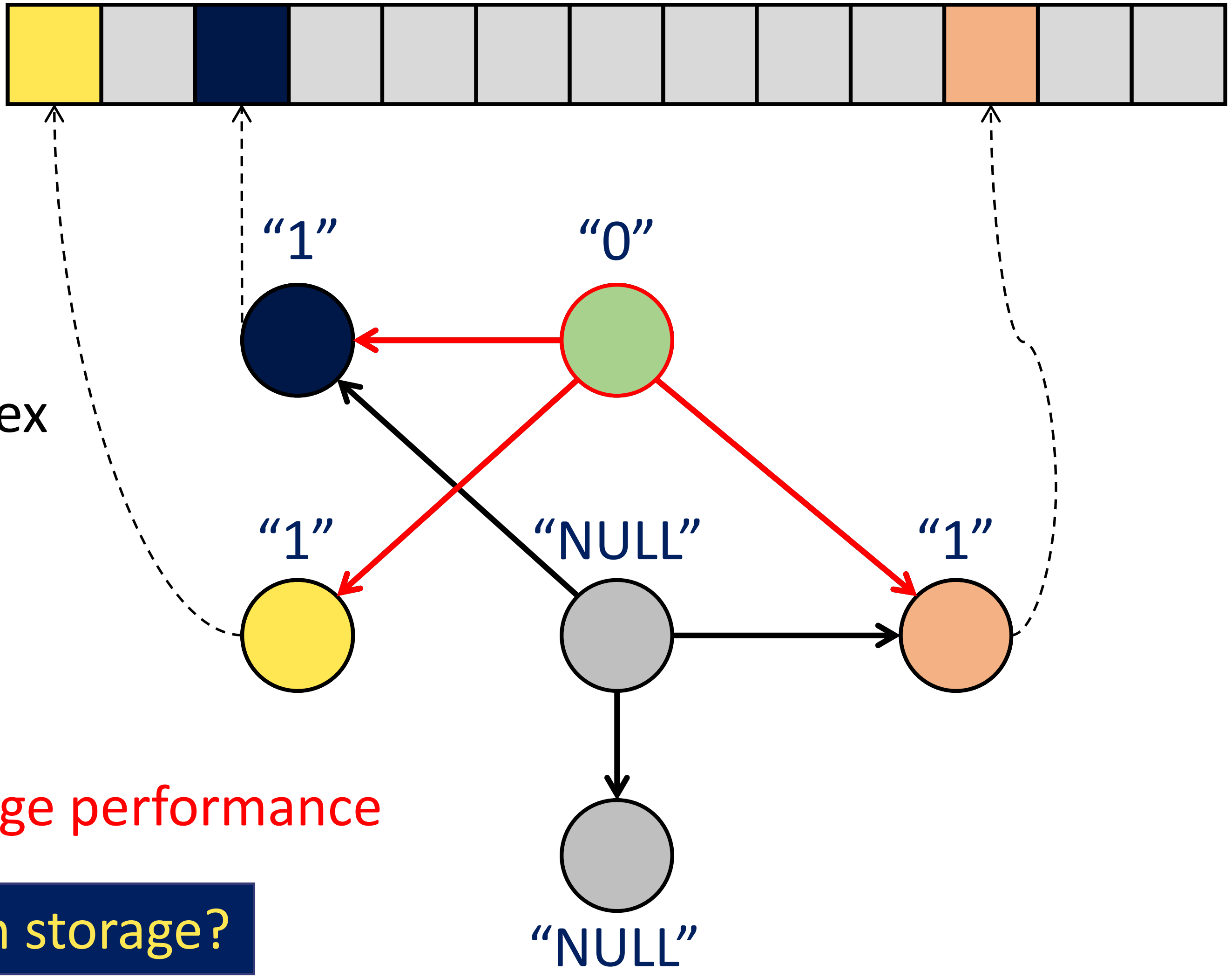
TB to 100s of TB in size

Notoriously sparse/irregular

1) Connectomics graph of the brain - Source: “Connectomics and Graph Theory” Jon Clayden, UCL
2) (Part of) the internet - Source: Criteo Engineering Blog
3) The Graph of a Social Network – Source: Griff’s Graphs

A Short Background on Graph Analytics

- ❑ Graph data consists of
  - ○ Graph structure (edges)
  - ○ Algorithmic state (vertex)
- ❑ Graphs are often sparse
  - ○ Edge data is much larger than vertex
- ❑ Vertex data access is irregular
  - ○ Edge data access is irregular to a lesser degree



Access granularity mismatch kills storage performance

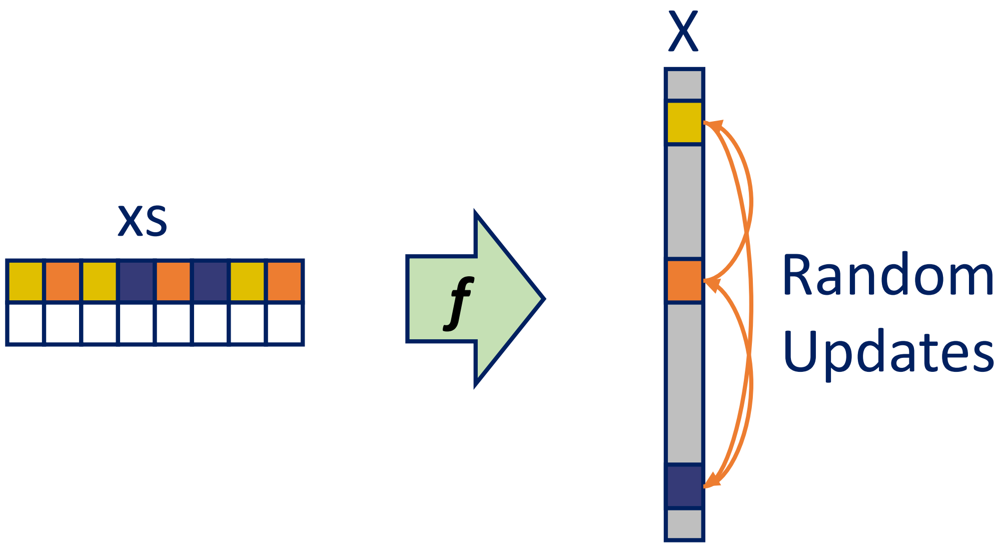
Can we still do graph analytics in storage?

General Problem of Irregular Array Updates

Updating an array ***x*** with a stream of update requests ***xs*** and update function ***f***

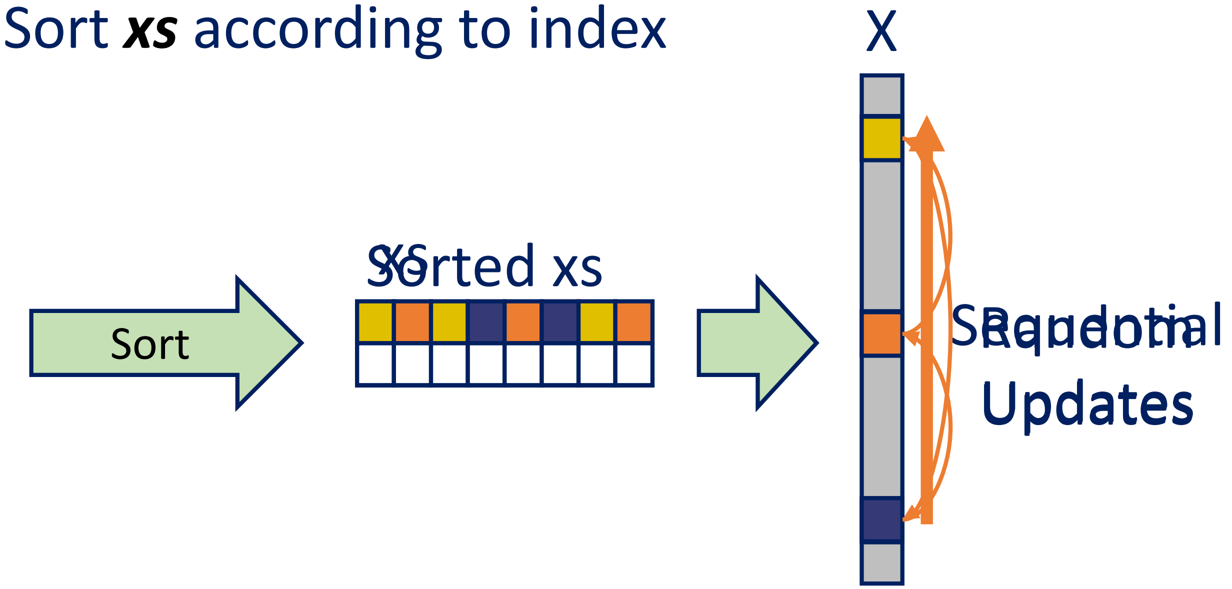
$\boldsymbol{For\ each} < idx, arg > \boldsymbol{in}\ xs$:

$x[idx] = \boldsymbol{f}(x[idx], arg)$



Solution Part One - Sort

Sort ***xs*** according to index



Much better than naïve random updates

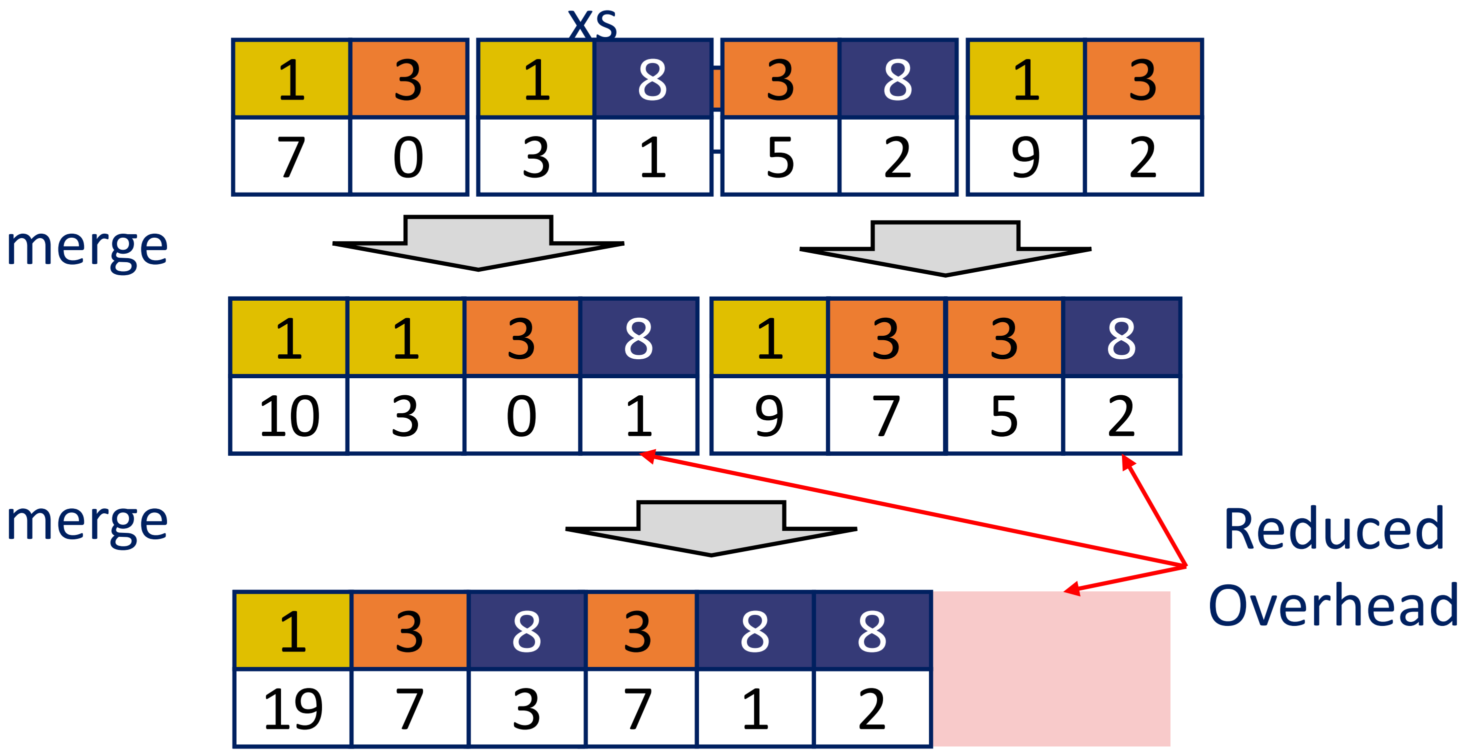
Terabyte graphs can generate terabyte logs

Significant sorting overhead

Solution Part Two - Reduce

Associative update function ***f*** can be interleaved with sort

e.g., (A + B) + C = A + (B + C)



xs

1	3	1	8	3	8	1	3
7	0	3	1	5	2	9	2

merge

1	1	3	8	1	3	3	8
10	3	0	1	9	7	5	2

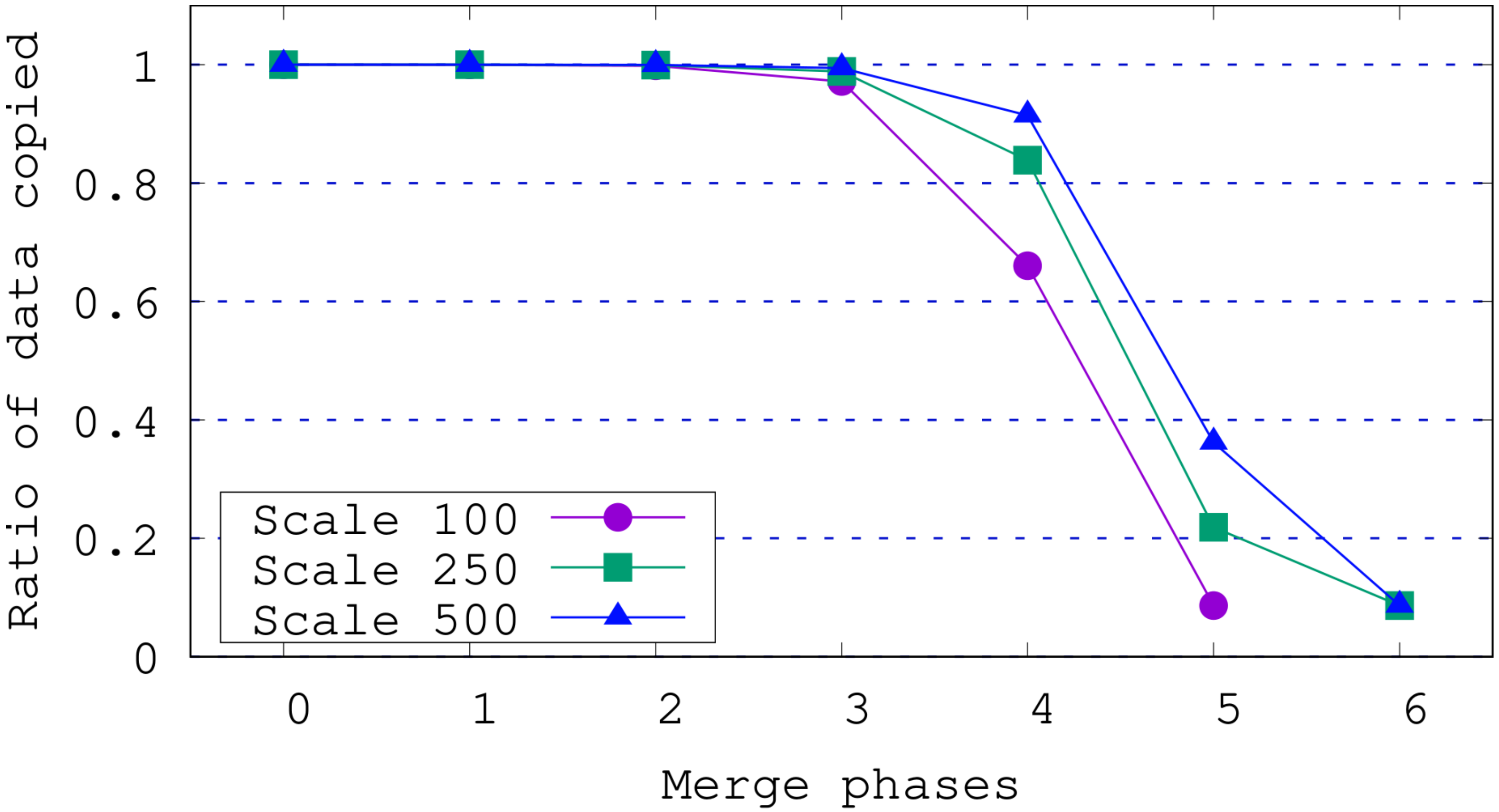
merge

1	3	8	3	8	8
19	7	3	7	1	2

Reduced Overhead

Big Benefits from Interleaving Reduction

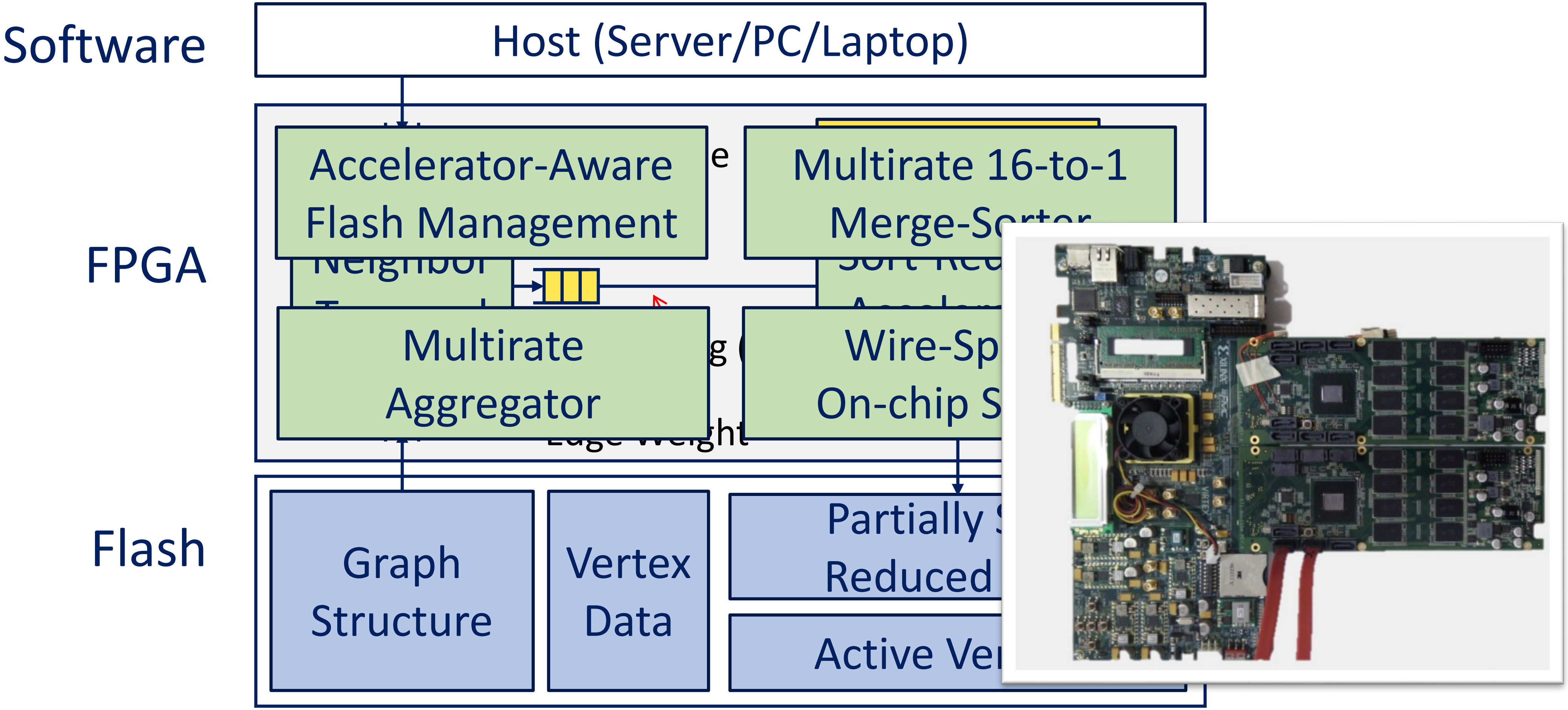
Ratio of data copied at each sort phase



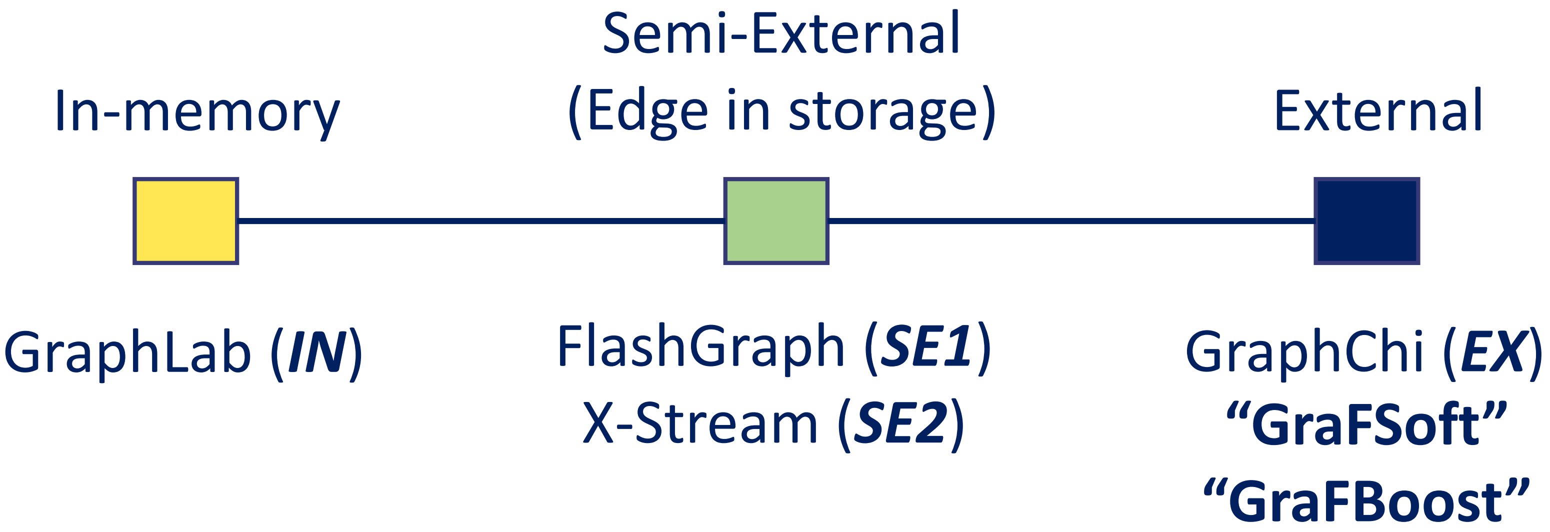
*Sneak peek into database join acceleration

Accelerated Graph Analytics Architecture

In-storage accelerator reduces data movement and cost



Evaluated Graph Analytic Systems



In-memory

Semi-External
(Edge in storage)

External

GraphLab (***IN***)

FlashGraph (***SE1***)
X-Stream (***SE2***)

GraphChi (***EX***)
“GraFSoft”
“GraFBoost”

“Distributed GraphLab: a framework for machine learning and data mining in the cloud,” VLDB 2012
“FlashGraph: Processing billion-node graphs on an array of commodity SSDs,” FAST 2015
“X-Stream: edge-centric graph processing using streaming partitions,“ SOSP 2013
“GraphChi: Large-scale graph computation on just a PC,“ USENIX 2012

Evaluation Environment

4-core i5
4 GB DRAM

$500

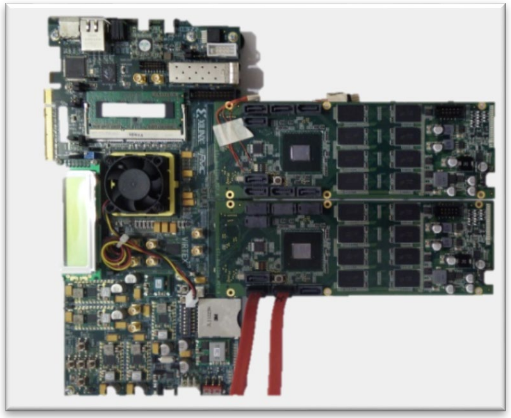
+

Virtex 7 FPGA
1 GB on-board DRAM
1 TB on-board flash

$???

$1,500

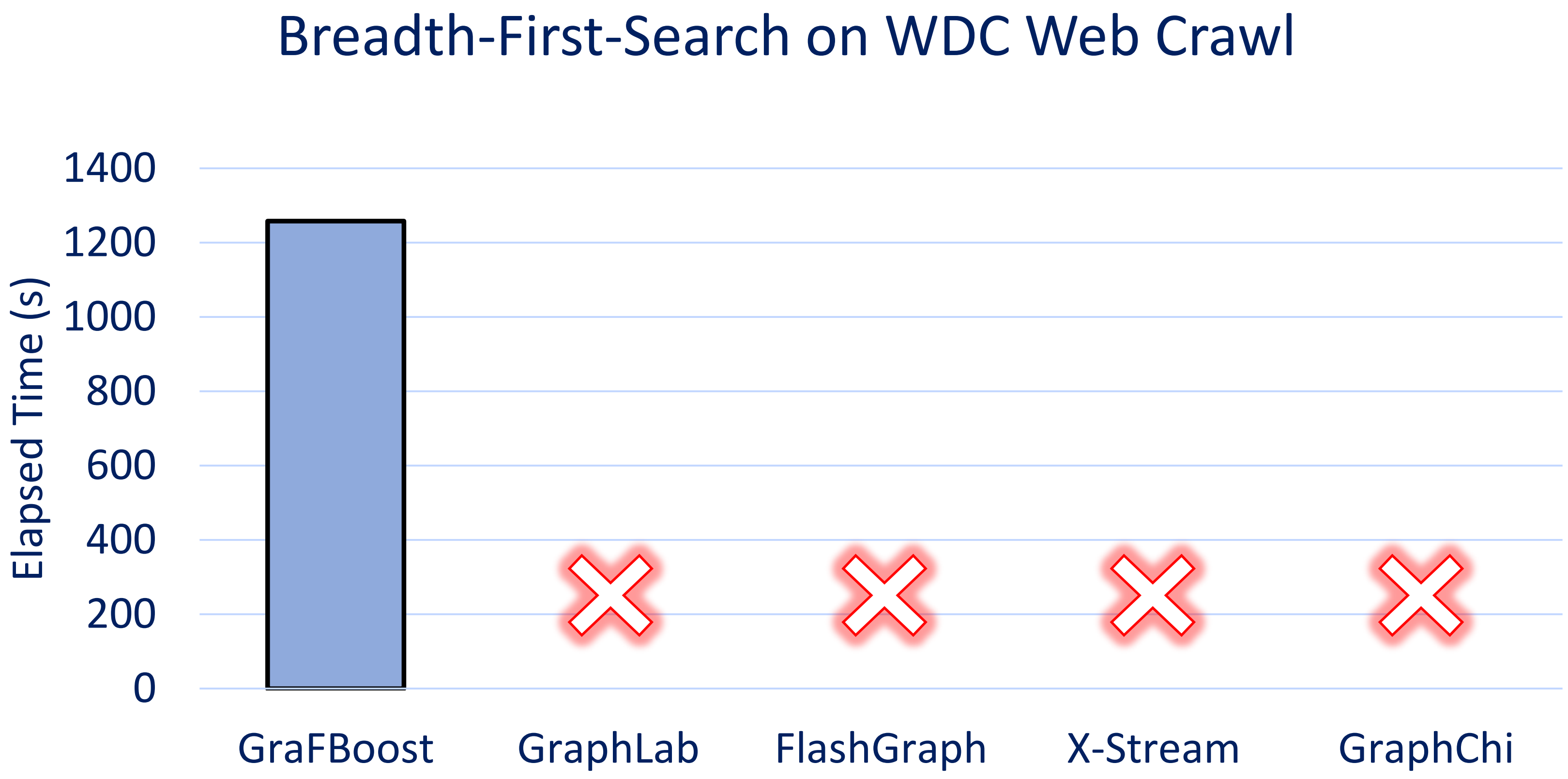
BlueDBM Card



The Graphs

	Capacity	Vertices	Edges
Web Data Commons Web Crawl	2 TB	3 Billion	128 Billion
Graph 500 Synthetic Kronecker	0.5 TB	4 Billion	32 Billion

Evaluation Result



This was a bit unfair…

Evaluation Environment

4-core i5
4 GB DRAM
1 TB PCIe Flash
Virtex 7 FPGA

$2,000

GraFBoost

16-core (32T) Xeon
128 GB DRAM
5x 0.5TB PCIe Flash

$9,000

GraFSoft, GraphChi
FlashGraph, X-Stream

5 Node Cluster
60-core (120T) Xeon
240 GB DRAM

$10,000

GraphLab

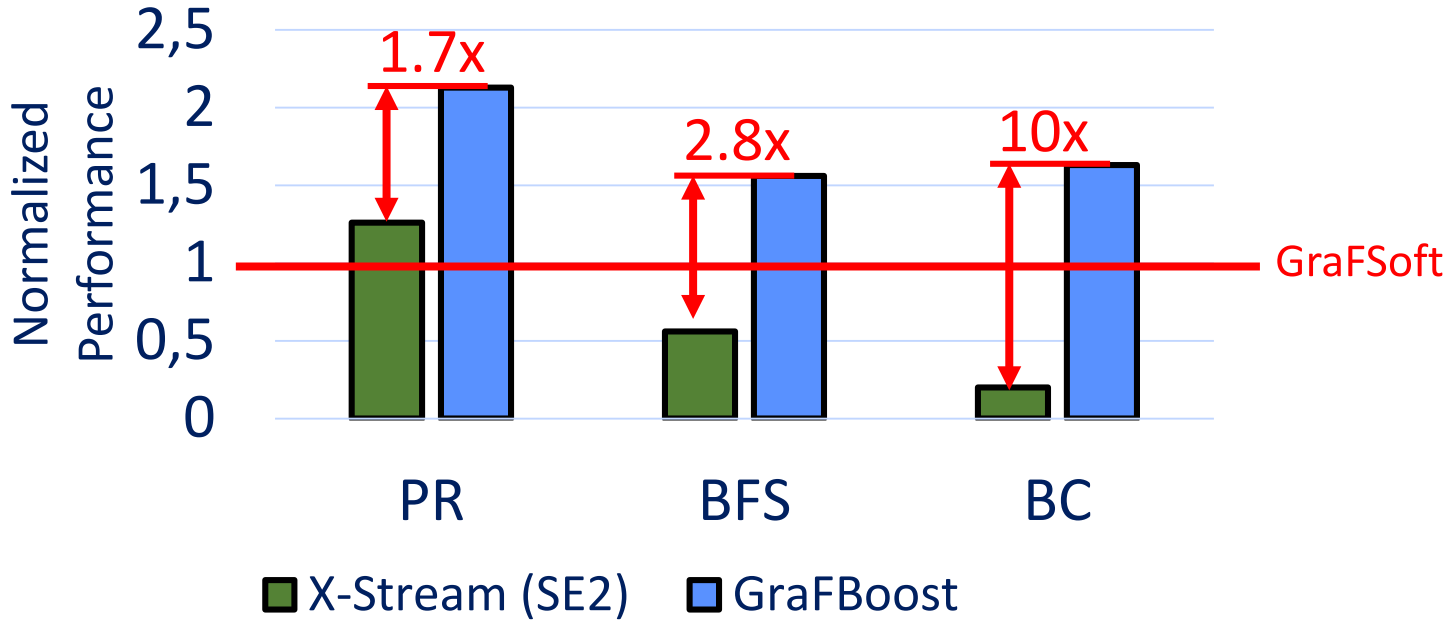
Results with a Large Graph: Synthetic Scale 32 Kronecker Graph

0.5 TB in text, 4 Billion vertices

GraphLab (***IN***) out of memory

FlashGraph (***SE1***) out of memory

GraphChi (***EX***) did not finish

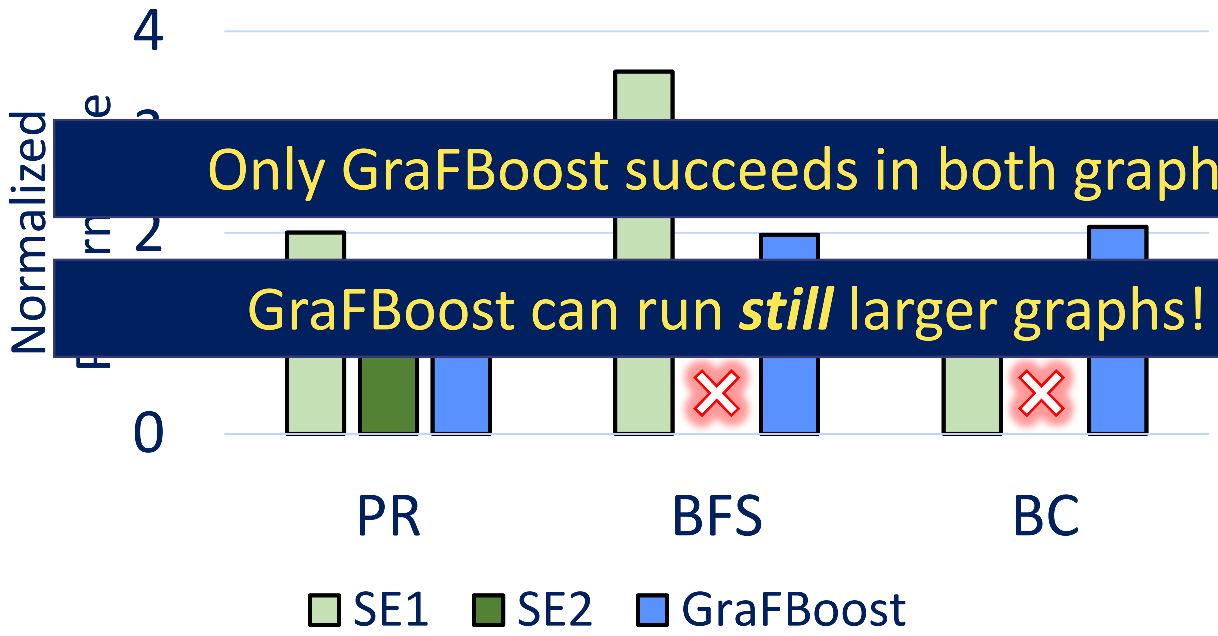


Results with a Large Graph: Web Data Commons Web Crawl

2 TB in text, 3 Billion vertices

GraphLab (***IN***) out of memory

GraphChi (***EX***) did not finish



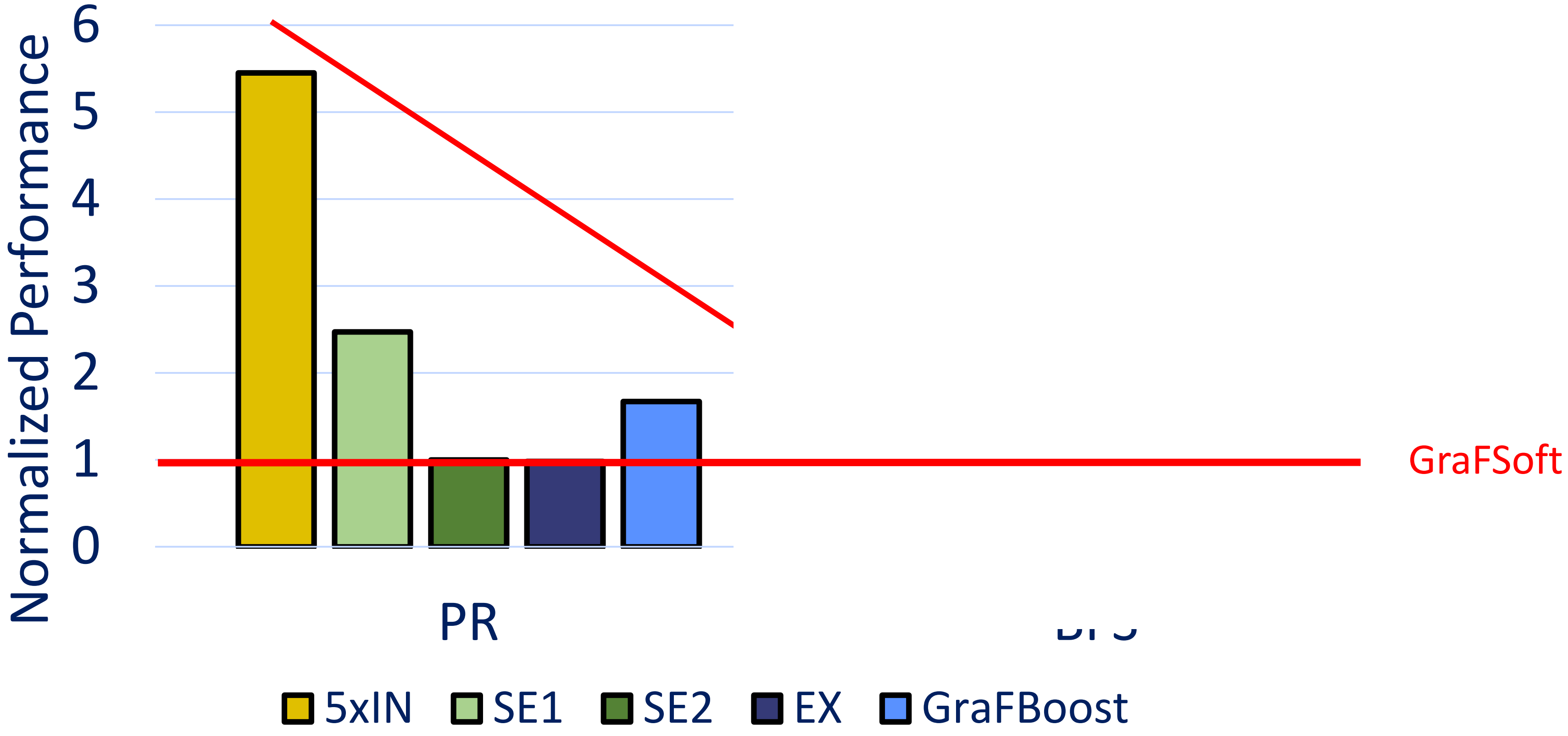
Only GraFBoost succeeds in both graphs

GraFBoost can run ***still*** larger graphs!

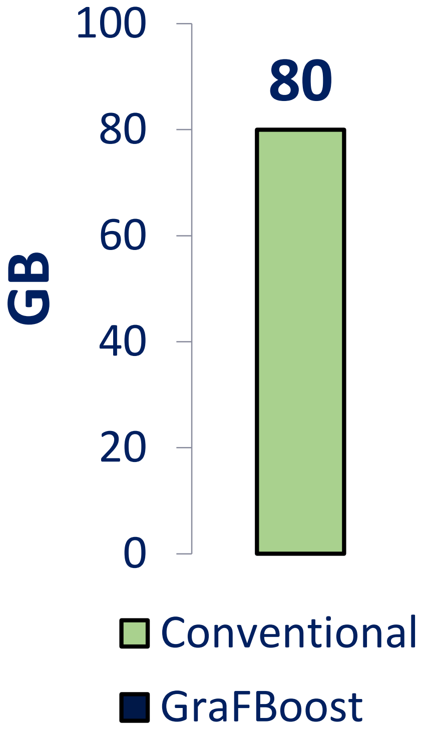
Results with a Medium Graph: Against an In-Memory Cluster

Synthesized Kronecker Scale 28

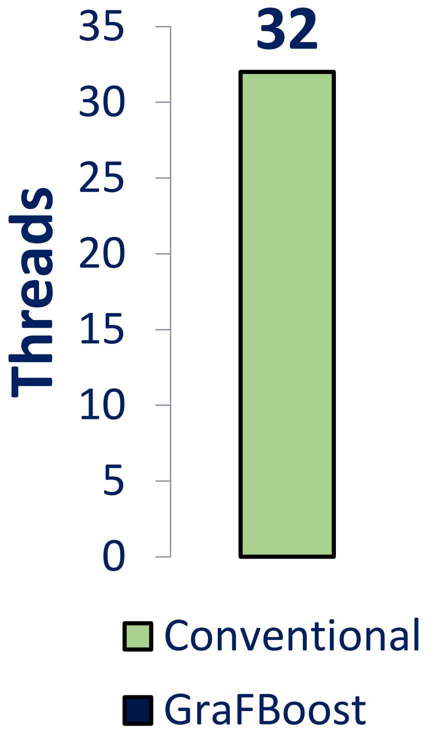
0.09 TB in text, 0.3 Billion vertices



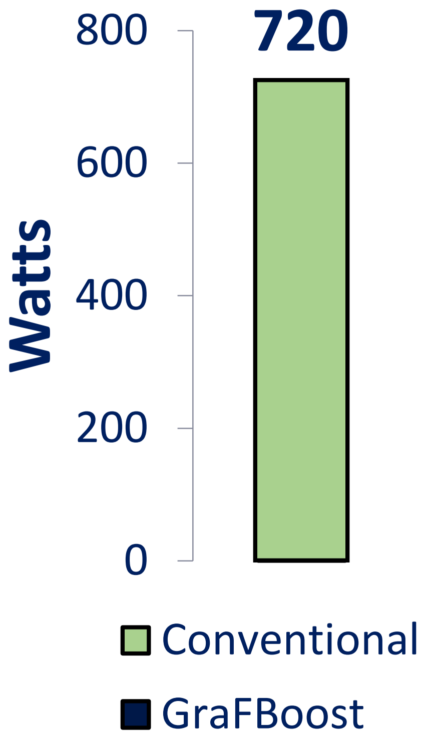
GraFBoost Reduces Resource Requirements



External analytics



Hardware Acceleration



External Analytics
+
Hardware Acceleration

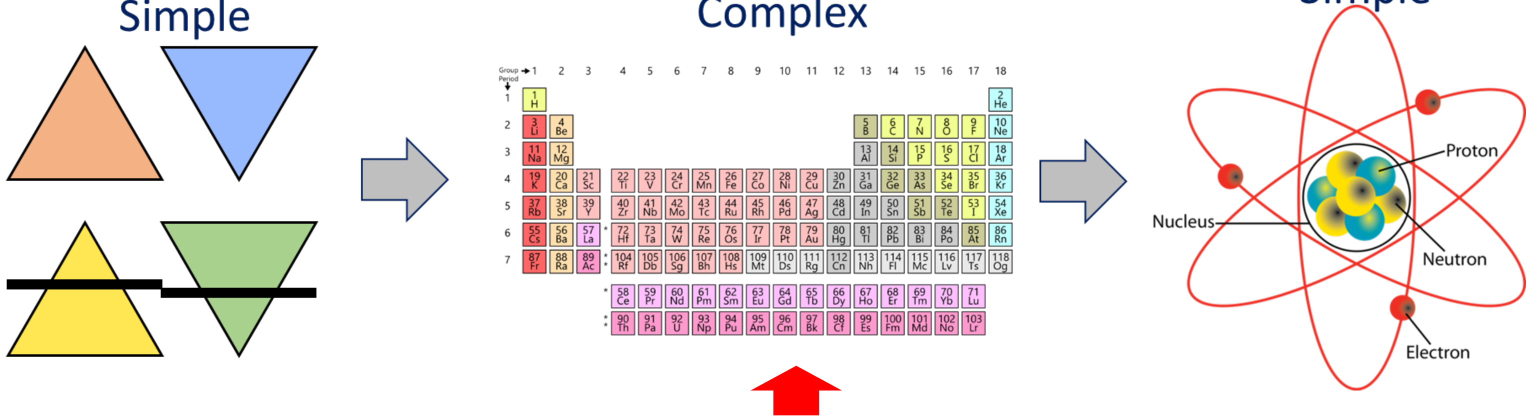
Future Directions – Short Term

- ❑ Next Generation Platform
  - BlueDBM was a custom design – difficult to disseminate results
  - Original prototype flash chips are aging
  - Newer SSD/FPGAs are much faster
- ❑ More Applications
  - Bioinformatics – Personalized cancer genomics
  - Scientific Computing – Stencil codes for physics simulations
  - Machine Learning – Low-power inference at the edge
  - More graph analytics – Exploring applications in security
  - And More!

Future Directions – Mid Term

- ❑ Programming models for accelerated computational storage
  - ○ The right abstraction and interface for efficient implementations
  - ○ In graph analytics, the associativity restriction made sort-reduce possible
  - ○ Exploring applications to build experience/data/intuition

Process of Model Discovery



Simple

Complex

Simple

I am here

The Long Term Goal

Thank you!