Towards "Intelligence, Storage, Network": Characterizing, Optimizing, and Outlooking

— A Systems Researcher's Perspective

Rong Chen

Institute of Parallel and Distributed Systems, SJTU

Huawei STW, 2023

Joint work with *Xingda, Xiating, Rongxin, Yuhan, Haibo, Binyu, and members of IPADS*

Who AM I

Rong Chen (陈榕) / IPADS, SJTU

https://ipads.se.sjtu.edu.cn/rong_chen

- Research Interest: Building efficient, scalable, and reliable distributed systems
- Publications and awards in systems conferences (OSDI, SOSP, EuroSys)
- Huawei OlympusMons Pioneer Award, 2020

  “Efficient Data Processing System based on New Heterogeneous Hardware”

Disclaimers:

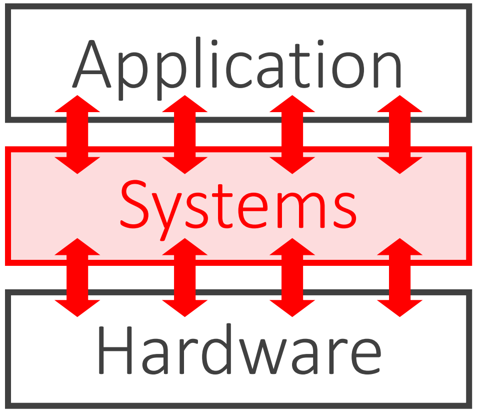
I am a Systems person, not a Network/Storage/AI expert ☺

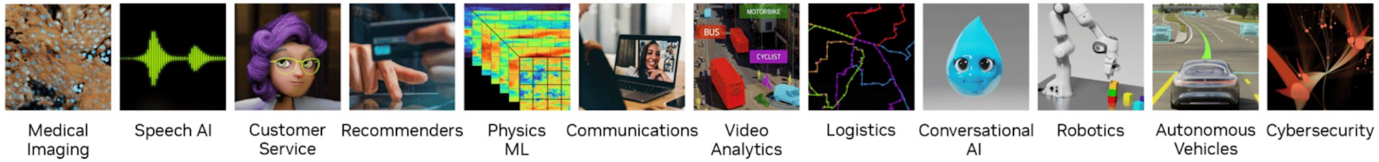
This Talk

My view:

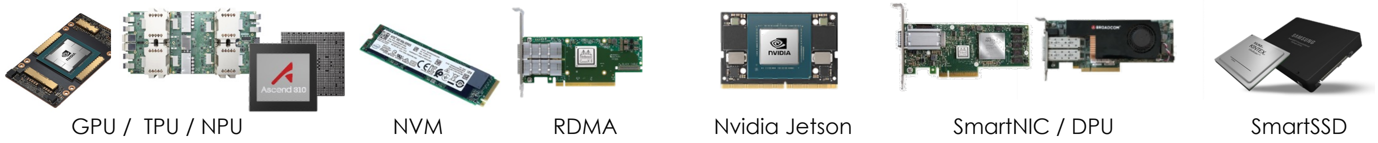
How to (re)build high-performance system software stack

by exploiting new hardware of “Intelligence, Storage, Network”

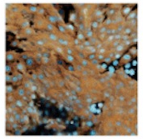




kernel, compiler, toolkit, framework . . .



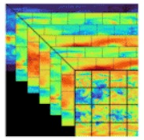
Application Demands


Medical Imaging

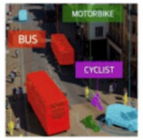
Speech AI

Customer Service

Recommenders


Physics ML

Communications


Video Analytics

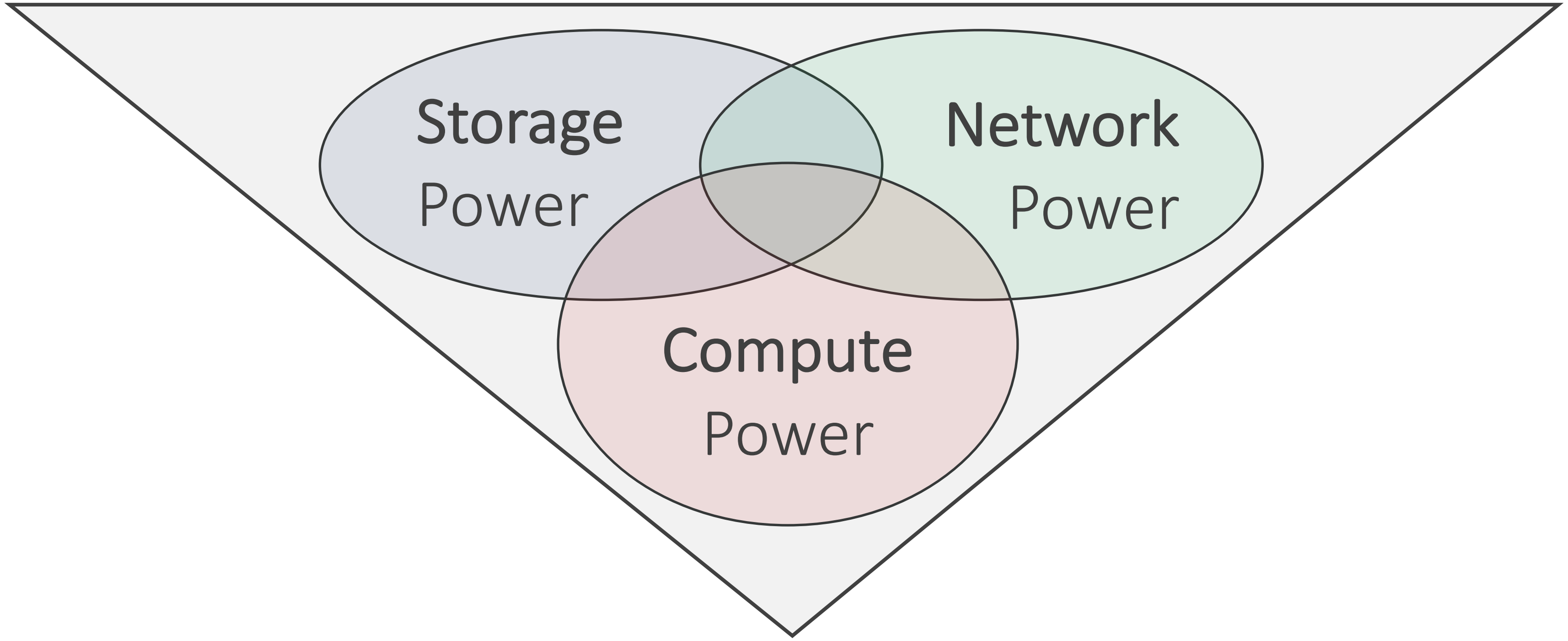
Logistics

Conversational AI

Robotics

Autonomous Vehicles

Cybersecurity

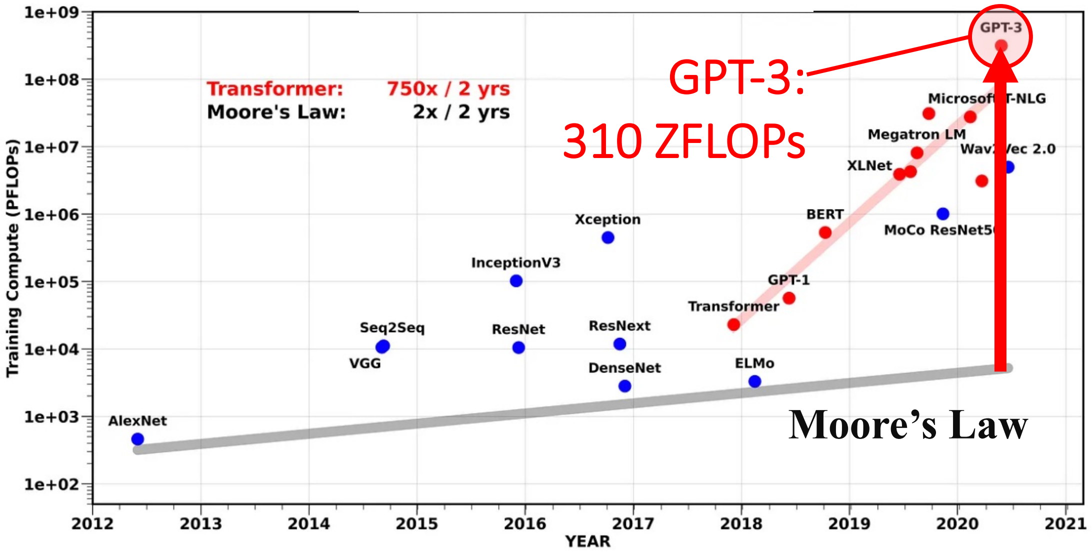

Storage Power

Network Power

Compute Power

Application Demands

Compute Power



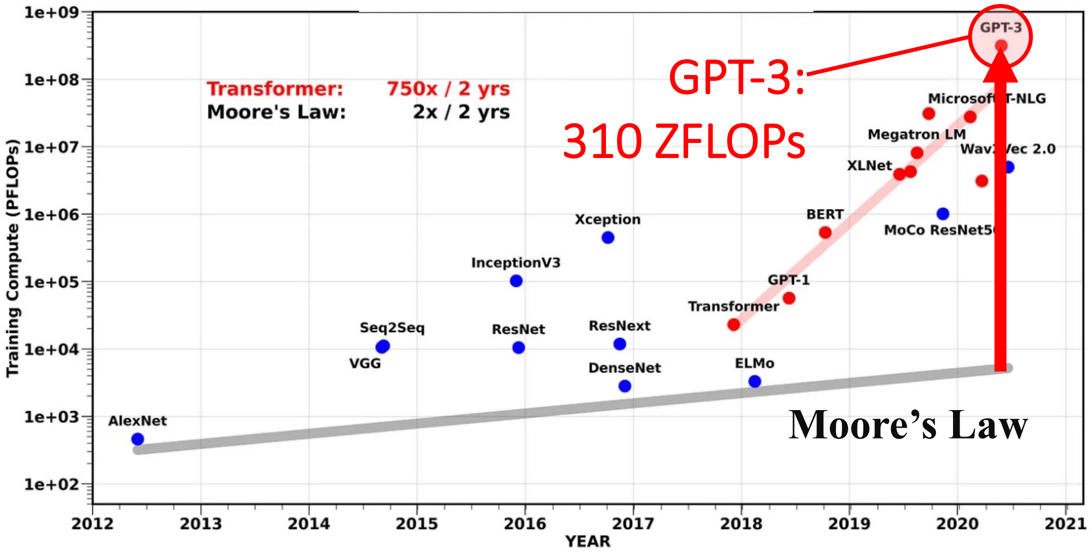
GPT-3:
310 ZFLOPs

Moore's Law

Source: "AI and Memory Wall", 2021. https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Application Demands

Compute Power

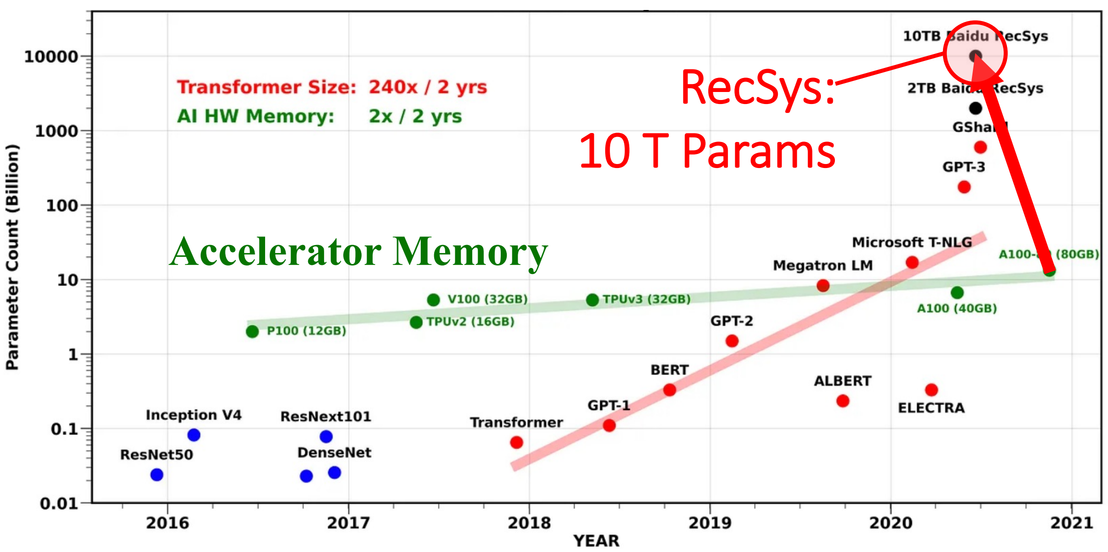


GPT-3:

310 ZFLOPs

Moore's Law

Storage Power



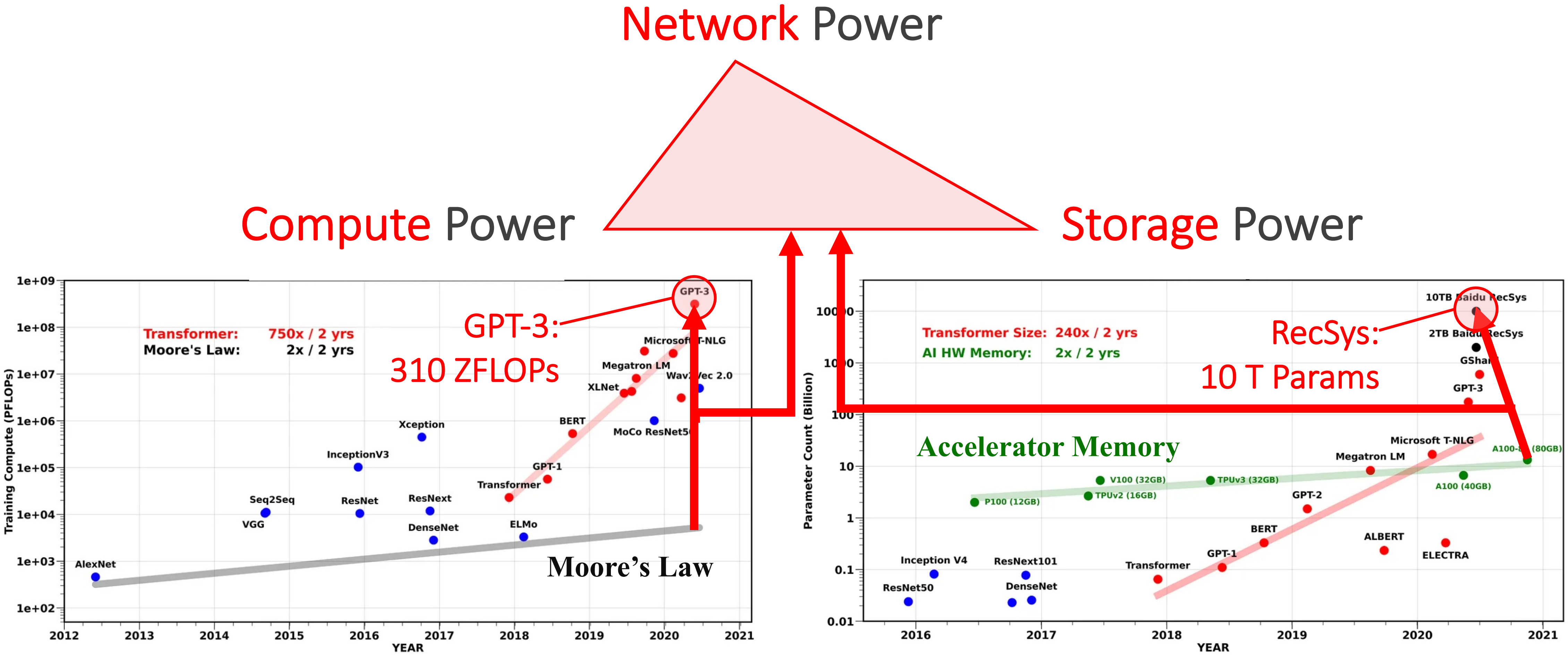
Accelerator Memory

RecSys:

10 T Params

Source: "AI and Memory Wall", 2021. https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Application Demands



Network Power

Compute Power

Storage Power

GPT-3:
310 ZFLOPs

RecSys:
10 T Params

Source: "AI and Memory Wall", 2021. https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

This Talk

My view:

How to (re)build high-performance system software stack

by exploiting **new hardware of “Intelligence, Storage, Network”**

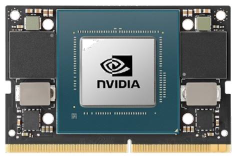
GPU / TPU / NPU

NVM

RDMA

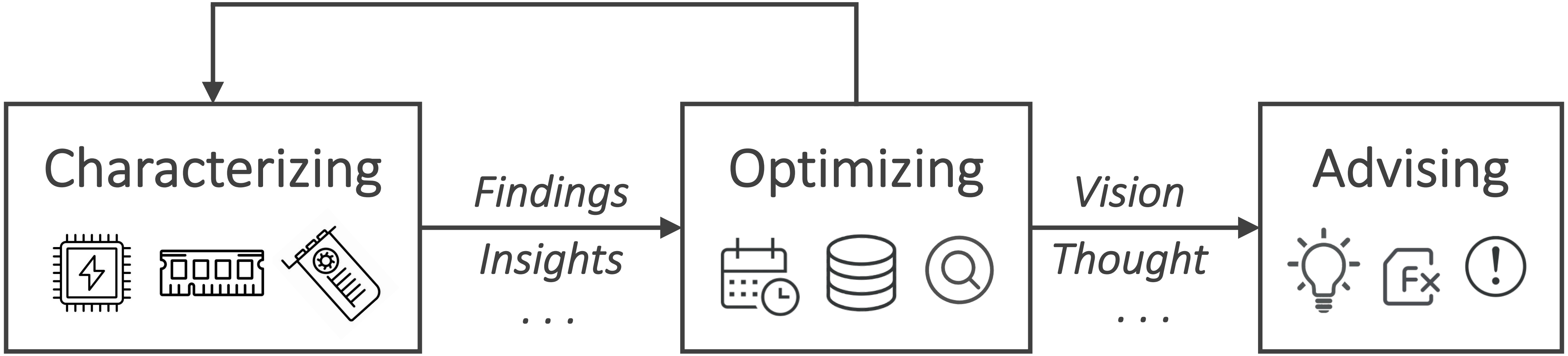
DPU / SmartNIC

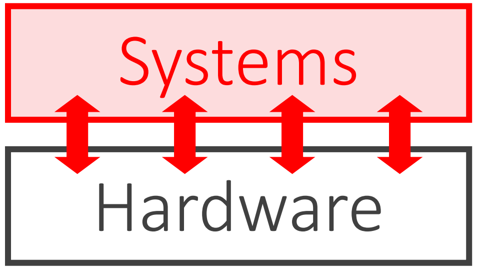
SmartSSD



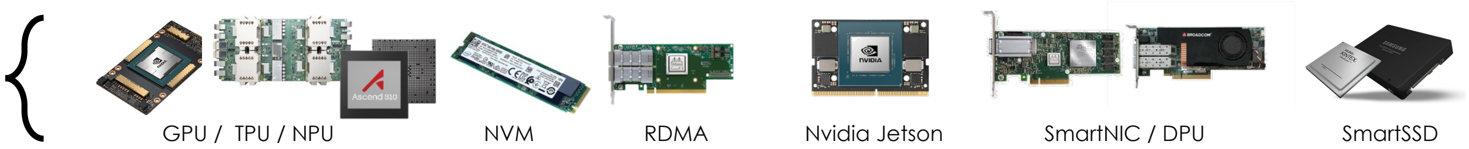
Nvidia Jetson

Our Approach





kernel, compiler, toolkit, framework . . .

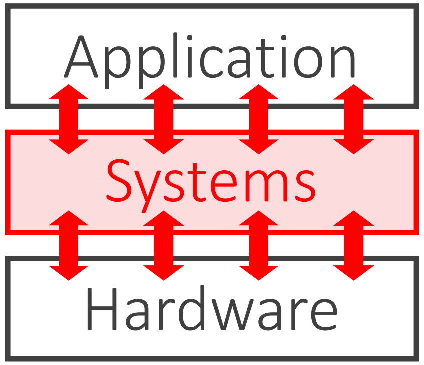


Outline

Case #1: Collaborative offloading

Case #2: Cooperative offloading

Outlooking systems research for DPU

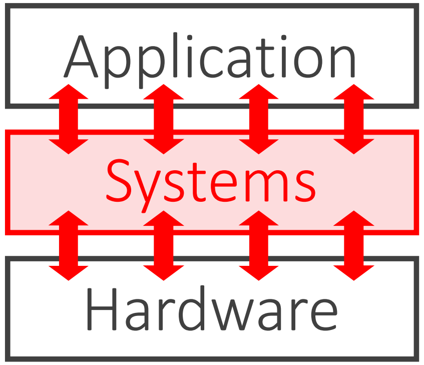


Outline

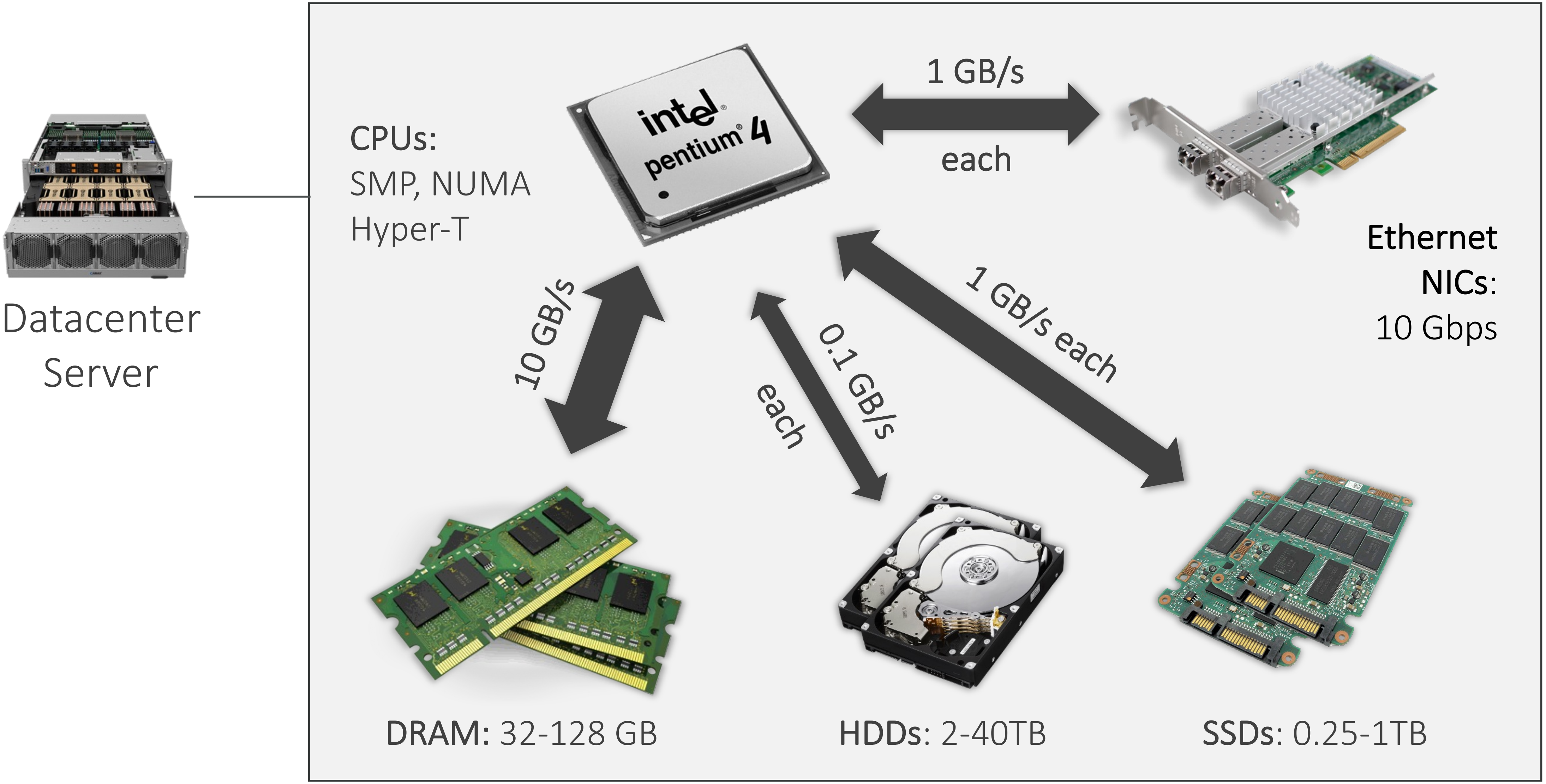
Case #1: Collaborative offloading

Case #2: Cooperative offloading

Outlooking systems research for DPU



Hardware in DC



Datacenter Server

CPUs:
SMP, NUMA
Hyper-T

1 GB/s
each

Ethernet NICs:
10 Gbps

10 GB/s

0.1 GB/s
each

1 GB/s each

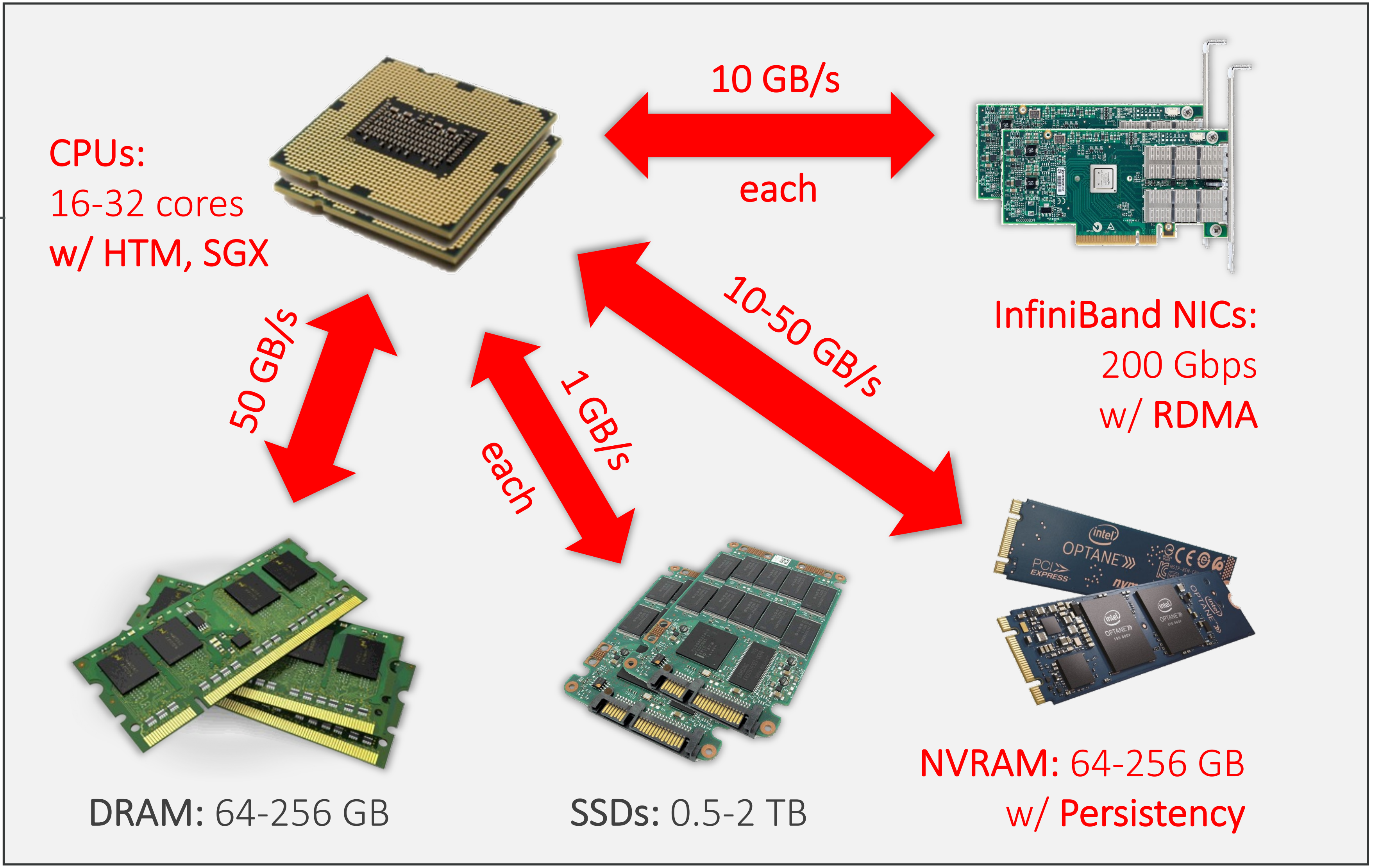
DRAM: 32-128 GB

HDDs: 2-40TB

SSDs: 0.25-1TB

Hardware in DC

Datacenter Server



CPUs:
16-32 cores
w/ HTM, SGX

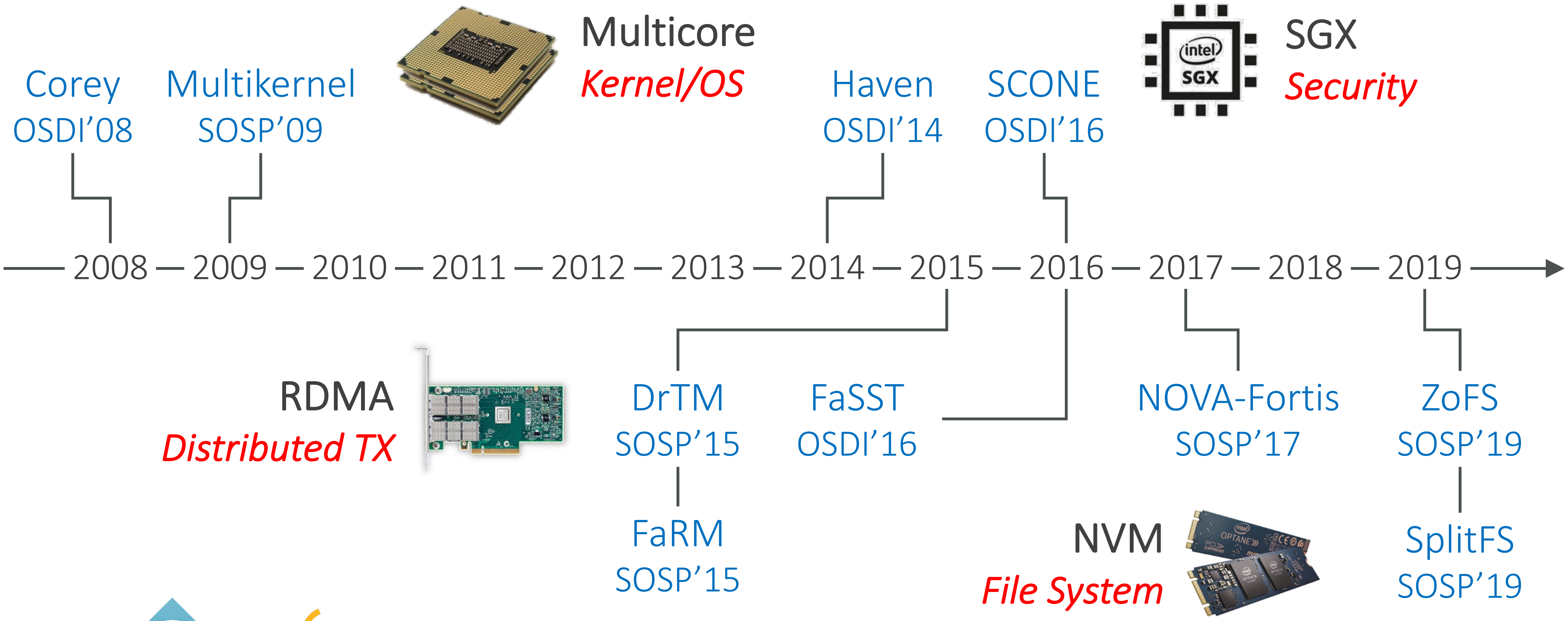
InfiniBand NICs:
200 Gbps
w/ RDMA

DRAM: 64-256 GB

SSDs: 0.5-2 TB

NVRAM: 64-256 GB
w/ Persistency

Common Practice: Offloading



Multicore
Kernel/OS

Corey
OSDI'08

Multikernel
SOSP'09

Haven
OSDI'14

SCONE
OSDI'16

SGX
Security

2008 — 2009 — 2010 — 2011 — 2012 — 2013 — 2014 — 2015 — 2016 — 2017 — 2018 — 2019

RDMA
Distributed TX

DrTM
SOSP'15

FaRM
SOSP'15

FaSST
OSDI'16

NOVA-Fortis
SOSP'17

NVM
File System

ZoFS
SOSP'19

SplitFS
SOSP'19

Systems
Community

Opportunity: Collaboration

Different hardware devices can work together

- **Case**: RDMA NIC (**RNIC**) can directly access **NVM**

  → “Remote Persistent Memory”

- **Scenarios**: distributed logging in FS, TX, ..

Distributed Filesystems

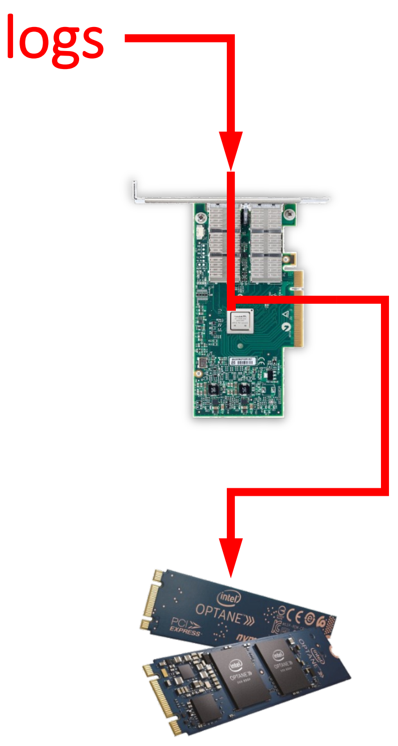
Octopus [ATC’17]

Orion [FAST’19]

Distributed Transactions

DrTM+R [EuroSys’16]

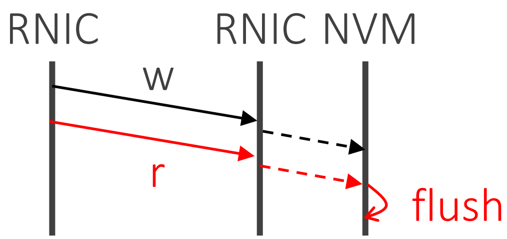
FaRMv2 [SIGMOD’19]



Challenge: Compatibility

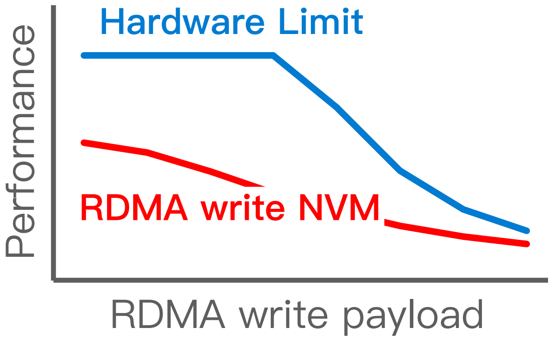
Functional flaw: remote write is NOT persistent

- ► Solution[1]: + remote read (two network roundtrips)



Performance pitfall: remote write is inefficient

- ► < 29% of NVM thpt limit (15M vs. 52M reqs/s)



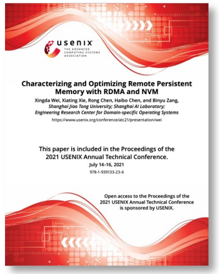
New hardware features are unaware of each other

[1] Intel. The librpmem library. https://pmem.io/pmdk/librpmem/

Our work

Collaborative offloading for the concurrent use of RDMA & NVM

- ▶ Characterizing RDMA+NVM for optimization hints
- ▶ Case studies: distributed TX (DrTM+H) and FS (Octopus)
- ▶ Suggestions to RDMA/NVM hardware designers



usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Characterizing and Optimizing Remote Persistent Memory with RDMA and NVM

Xingda Wei, Xiating Xie, Rong Chen, Haibo Chen, and Binyu Zang, *Shanghai Jiao Tong University; Shanghai AI Laboratory; Engineering Research Center for Domain-specific Operating Systems*

https://www.usenix.org/conference/atc21/presentation/wei

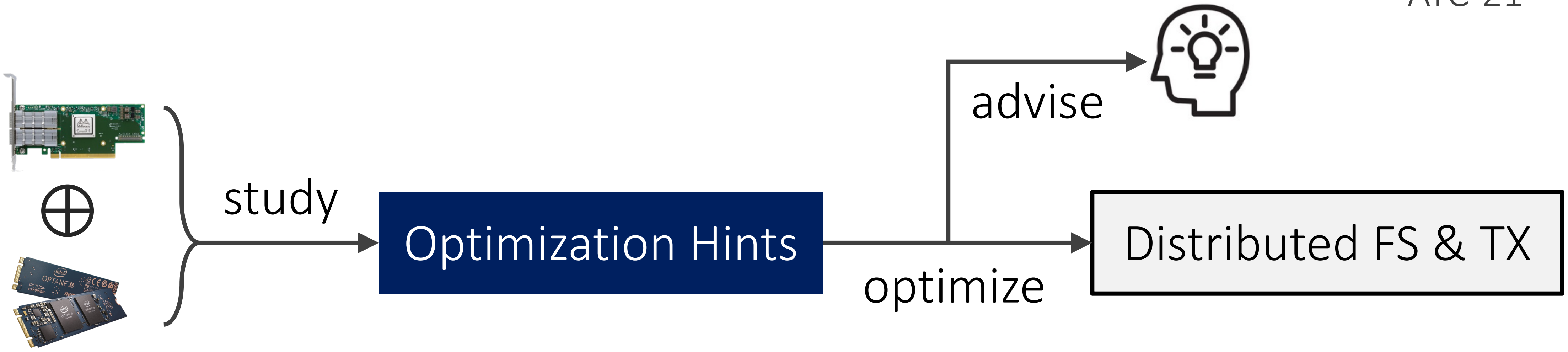
This paper is included in the Proceedings of the 2021 USENIX Annual Technical Conference.

July 14–16, 2021

978-1-939133-23-6

Open access to the Proceedings of the 2021 USENIX Annual Technical Conference is sponsored by USENIX.

USENIX ATC’21



Characterizing

Characterizing remote persistent memory w/ RDMA and NVM

- ▶ A systematic study of the collaboration btw. RDMA and NVM
- ▶ Tools: https://github.com/SJTU-IPADS/librdpma

An Empirical Guide to the Behavior and Use of Scalable Persistent Memory

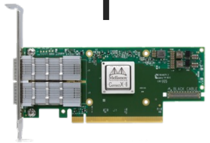
Challenges and Solutions for Fast Remote Persistent Memory Access

Persistent Memory Replication Over Traditional RDMA Part 1: Understanding Remote Persistent Memory

prior work

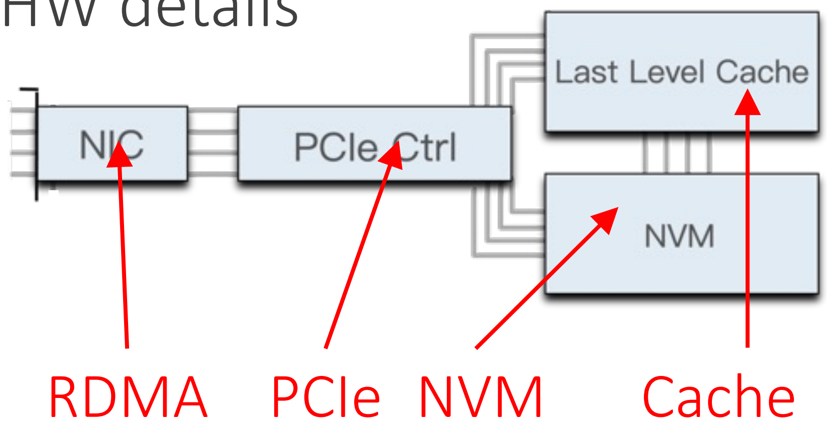
Optimization Hints

study



⊕

HW details



Example 1

Optimization Hint

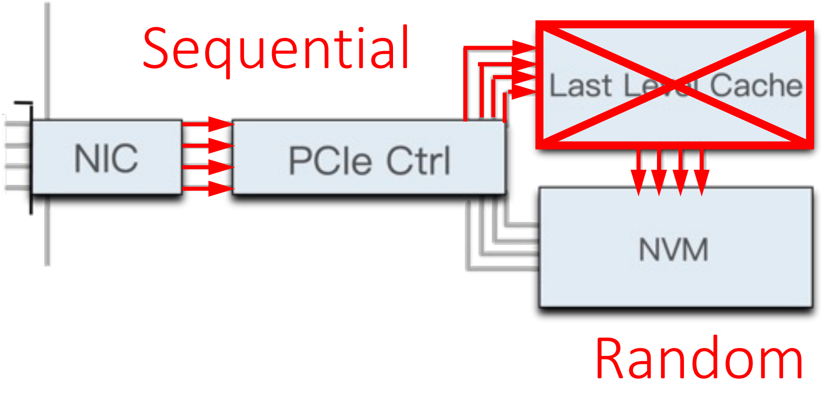
- ▶ Disable DDIO to skip LLC for large writes

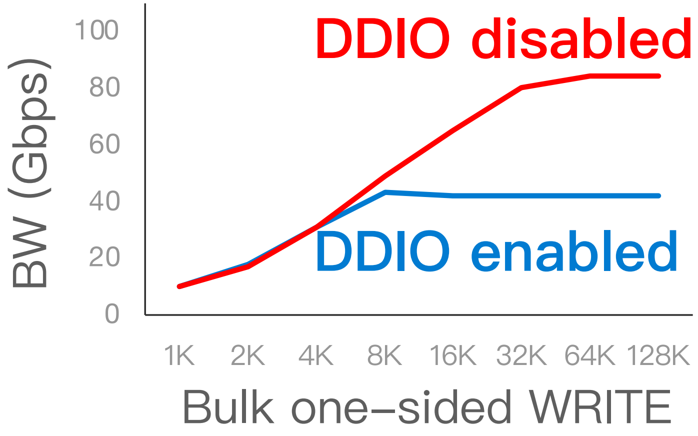
NVM feature

- ▶ Random I/O causes **write amplification**

Performance pitfall

- ▶ RNIC **sequentially writes** the data to LLC
- ▶ Then, LLC **randomly evicts** the data to NVM





Example 2

Optimization Hint

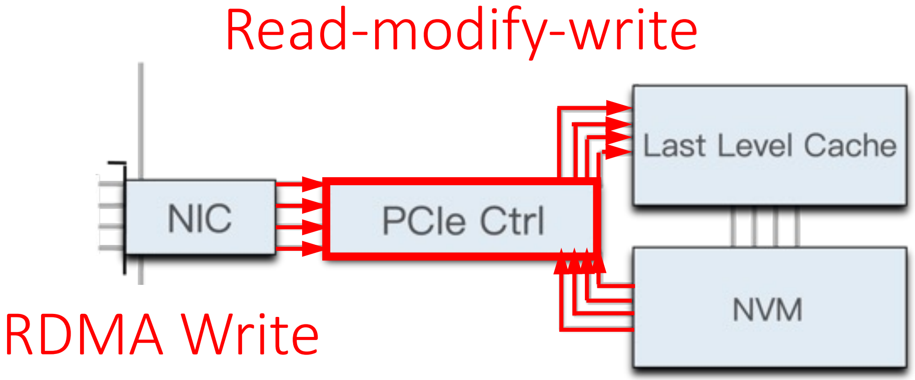
- ▶ Use 64B granularity for small writes

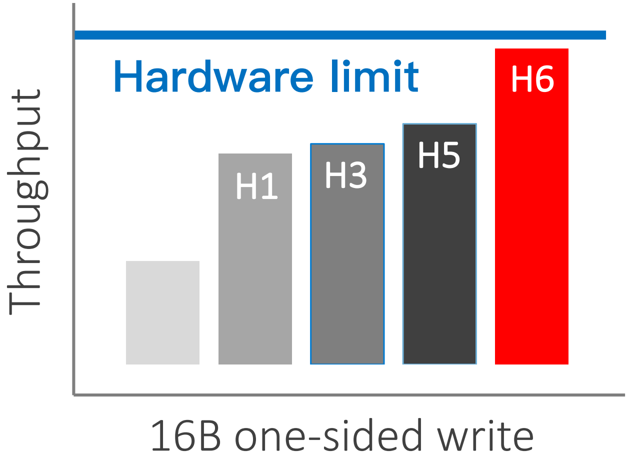
NVM feature

- ▶ Read-modify-write pattern *(PCIe partial-write)*

Performance pitfall

- ▶ An **extra read** to NVM if write does not fit **PCIe data word granularity** (e.g., 64B)





Characterizing

Characterizing remote persistent memory w/ RDMA and NVM

- A design guideline: 9 optimization hints in 3 aspects
- Achieve 87% of NVM thpt limit (from 15M to 45M reqs/s)

An Empirical Guide to the Behavior and Use of Scalable Persistent Memory

Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson

Challenges and Solutions for Fast Remote Persistent Memory Access

Anuj Kalia, David Andersen, Michael Kaminsky

Persistent Memory Replication Over Traditional RDMA Part 1: Understanding Remote Persistent Memory

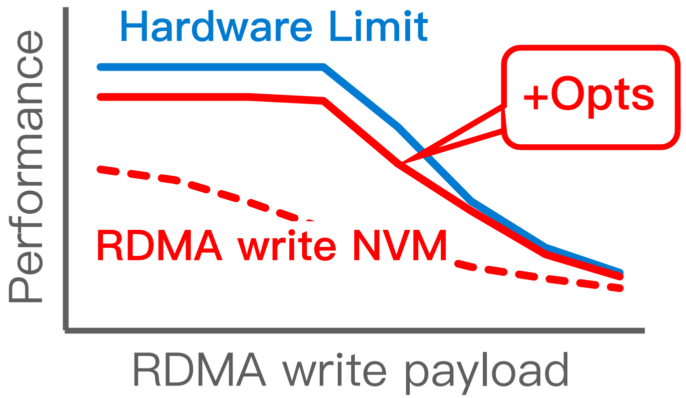
By Tomasz Gromadzki, Jan Marian Michalski

prior work

Optimization Hints

study

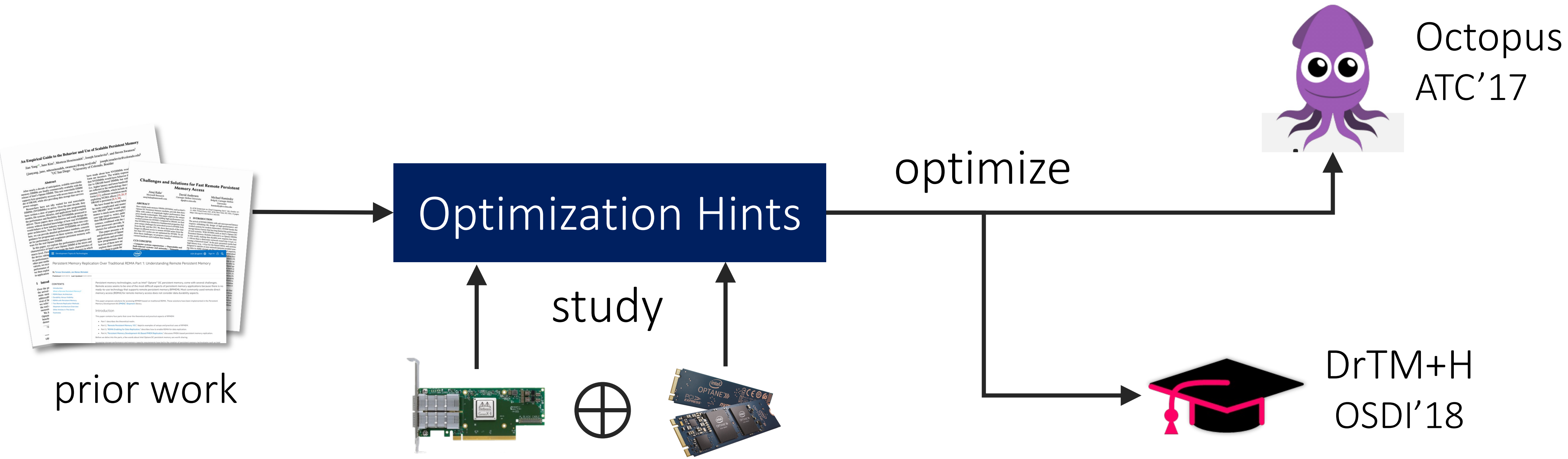
⊕



Optimizing

Applying our performance hints to existing RDMA-NVM systems

- **DrTM+H** (distributed TX) by 1.44×/2.09× for TPC-C/SmallBank
- **Octopus** (distributed FS) by 2.40× for Data I/O



prior work

Optimization Hints

study

optimize

Octopus
ATC'17

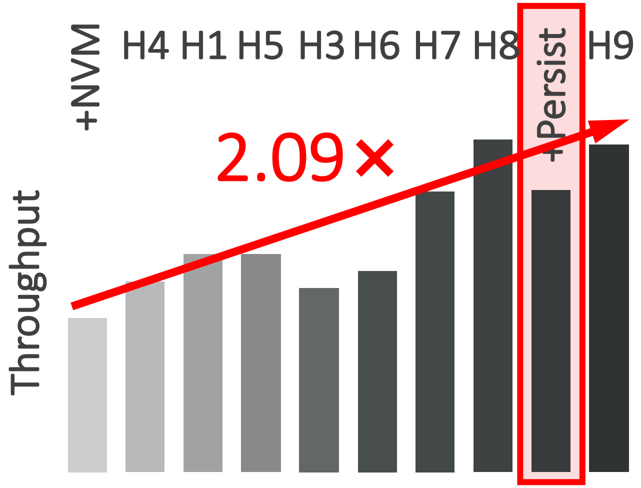
DrTM+H
OSDI'18

Case Study: Distributed Transaction

Applying our performance hints cumulatively on DrTM+H

Hints	Optimizations
H1	Separate memory pool from different sockets to avoid cross-socket NVM access
H3	Configure database with DDIO disabled
H4	Use ntstore to optimize the commit phase
H5	Align and pad logs/records larger than 256 B to XPLine granularity
H6+H7	Align and pad logs/records smaller than 256 B to 64 B granularity
H8	Implement a DRAM-based lock service for the validation phase
H9	Implement remote persistent log with H9 in one roundtrip

Improve perf. & enable persist



Factor Analysis of SmallBank

Optimization Hints

optimize

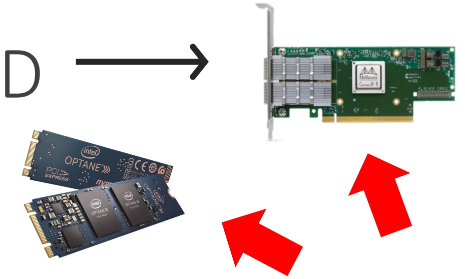
DrTM+H
OSDI'18

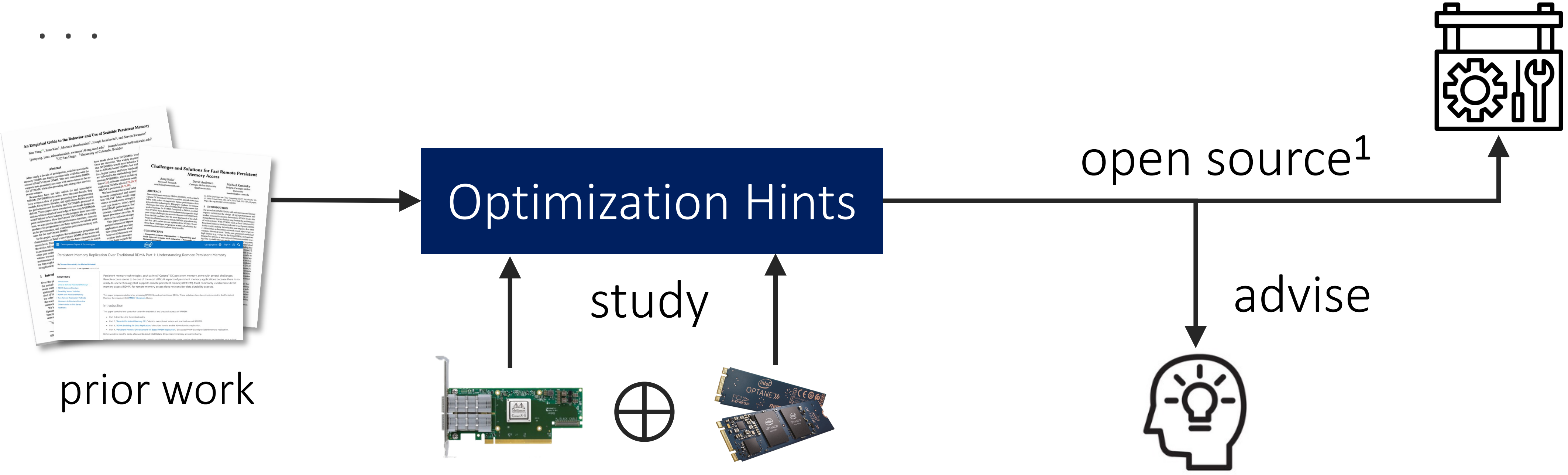
Advising

Suggestions to hardware designers

- RDMA (persistent) WRITE: avoid extra RDMA READ →
- RDMA-version nstore: avoid disabling DDIO →

. . .





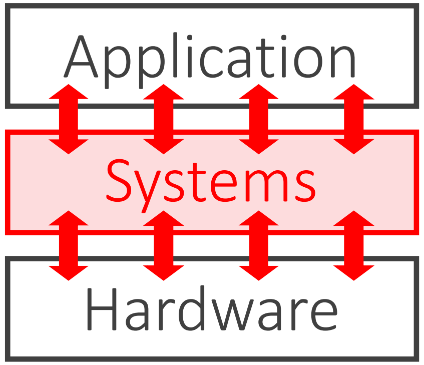
[1] Our open-sourced toolkit: https://github.com/SJTU-IPADS/librdpma

Outline

Case #1: Collaborative offloading

Case #2: Cooperative offloading

Outlooking systems research for DPU



New Trend : Capability Integration

Intelligent Hardware ➔ Network + Computation

Storage

CPU, FPGA, ASIC

. . .

SmartNIC

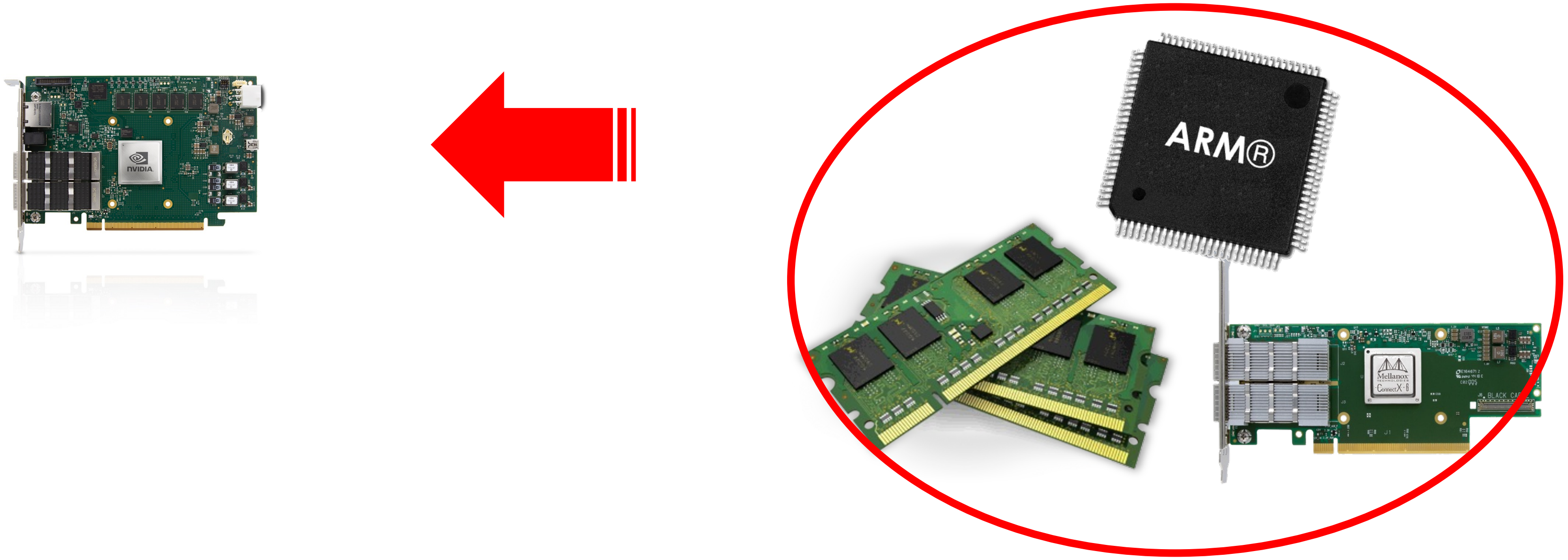
SmartSSD

Smart + X

New Trend : Capability Integration

Integrating multiple capabilities into a single device

- Typical case: DPU/SmartNIC (e.g., Nvidia BlueField)



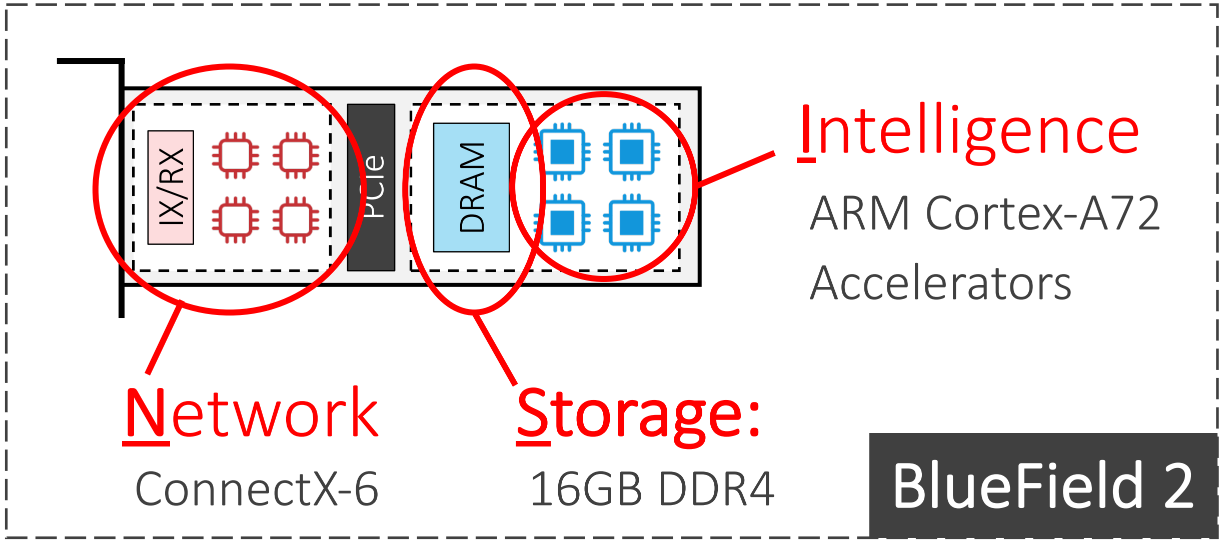
New Trend : Capability Integration

Integrating multiple capabilities into a single device

- Typical case: DPU/SmartNIC (e.g., Nvidia BlueField)

Nvidia BlueField-2

- ConnectX-6 (2x 100Gbps)
- 16 GB of on-board DRAM
- ARM Cortex-A72 (8 cores)



Intelligence
ARM Cortex-A72
Accelerators

Network
ConnectX-6

Storage:
16GB DDR4

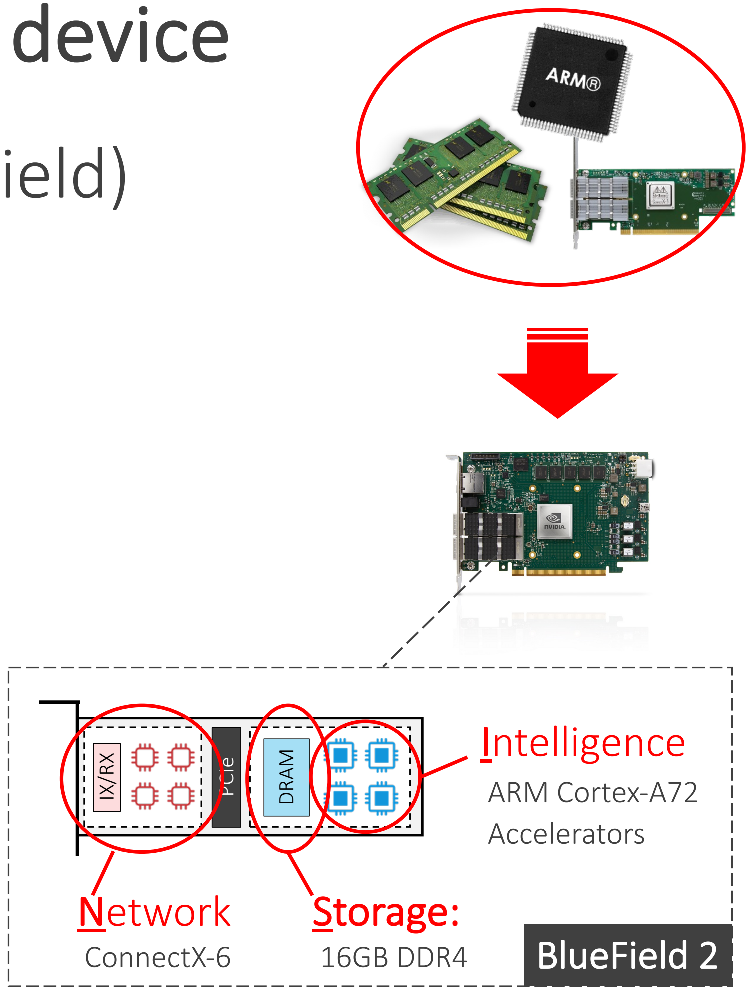
BlueField 2

New Trend : Capability Integration

Integrating multiple capabilities into a single device

- Typical case: DPU/SmartNIC (e.g., Nvidia BlueField)
- Good: Innately immune to compatibility issues
- Bad: (much) higher cost, compared to RNIC

		BlueField-2[1]	ConnectX-6[2]
Price	1.5×	$ 3615	$ 2,440
Space	2.0×	6.6 in. x 4.53 in.	6.6 in. x 2.71 in.
Power	3.2×	75W	23.6W



[1] NVIDIA MBF2H516A-EEEOT BlueField-2

[2] NVIDIA MCX653106A-HDAT ConnectX-6

Challenge: Underutilization

DPU is inferior in *every* single capability

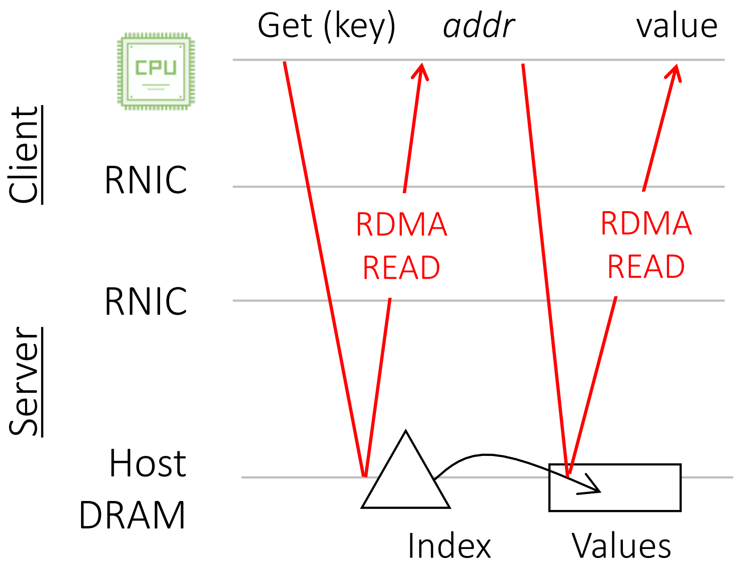
- Wimpy cores (e.g., 8-core ARM) and small memory (e.g., 16GB)
- Net. perf. degradation (BF-2 vs. CX-6): latency (+6~30%), thpt (-15~36%)

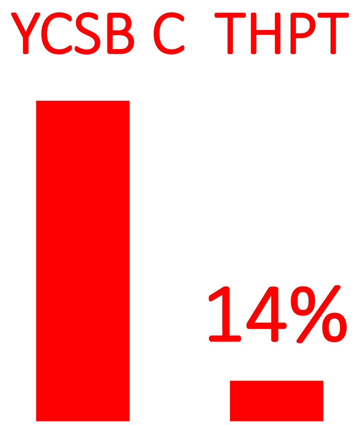
Challenge: Underutilization

Case study: Get (k) in distributed key/value store (KVS)

RNIC-based KVS

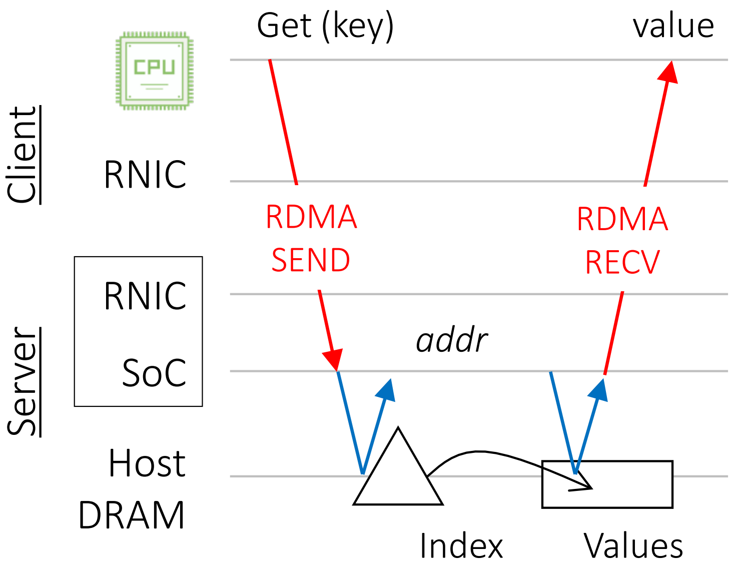
2x RDMA READs (1 for index, 1 for value)





DPU-based KVS

1x SEND/RECV, offload indexing to DPU



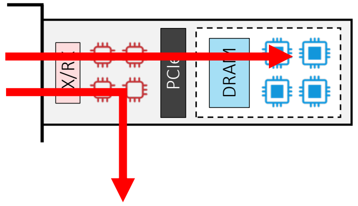
Challenge: Underutilization

DPU is inferior in *every* single capability

- Wimpy cores (e.g., 8-core ARM) and small memory (e.g., 16GB)
- Net. perf. degradation (BF-2 vs. CX-6): latency (+6~30%), thpt (-15~36%)

Existing systems only utilize a portion of DPU device

- Only NIC-Host path, treated as RNIC
- Only computing resource (SoC), treated as accelerator

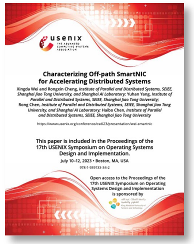


A systematic way to fully utilize integrated capabilities

Our work

Cooperative offloading for fully utilizing DPU

- ▶ Characterizing: study offloading paths, rather than HW components
- ▶ A step-by-step optimization guideline for DS designer
- ▶ Case studies: DPU-accelerated distributed FS and KV
- ▶ Open-source toolkit: https://github.com/smartnickit-project



usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Characterizing Off-path SmartNIC for Accelerating Distributed Systems

Xingda Wei and Rongxin Cheng, *Institute of Parallel and Distributed Systems, SEIEE, Shanghai Jiao Tong University, and Shanghai AI Laboratory;* Yuhan Yang, *Institute of Parallel and Distributed Systems, SEIEE, Shanghai Jiao Tong University;* Rong Chen, *Institute of Parallel and Distributed Systems, SEIEE, Shanghai Jiao Tong University, and Shanghai AI Laboratory;* Haibo Chen, *Institute of Parallel and Distributed Systems, SEIEE, Shanghai Jiao Tong University*

https://www.usenix.org/conference/osdi23/presentation/wei-smartnic

This paper is included in the Proceedings of the 17th USENIX Symposium on Operating Systems Design and Implementation.

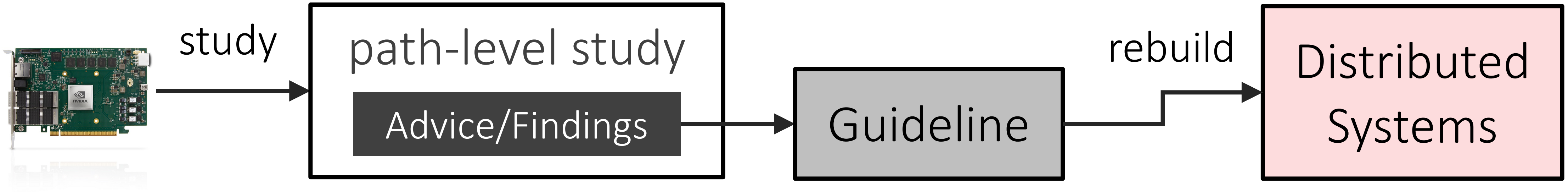
July 10–12, 2023 • Boston, MA, USA

978-1-939133-34-2

Open access to the Proceedings of the 17th USENIX Symposium on Operating Systems Design and Implementation is sponsored by

King Abdullah University of Science and Technology

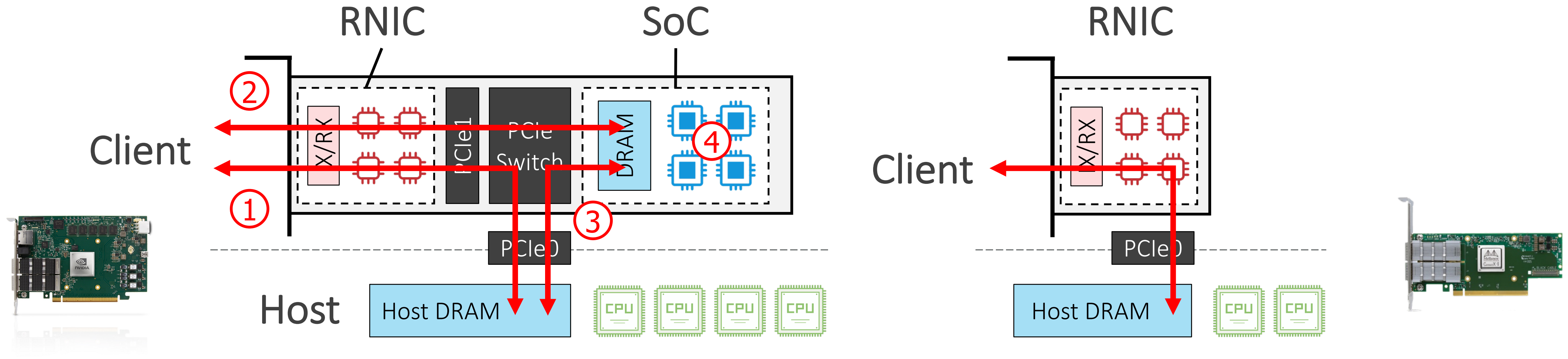
USENIX
OSDI’23



Characterizing

Characterizing DPU (i.e., BlueField 2) in path level

- Study offloading paths, rather than HW components
- Four paths: NIC-Host (①), NIC-SoC (②), SoC-Host (③), SoC-only (④)
- Performance implications: bottlenecks, anomalies, and takeaways



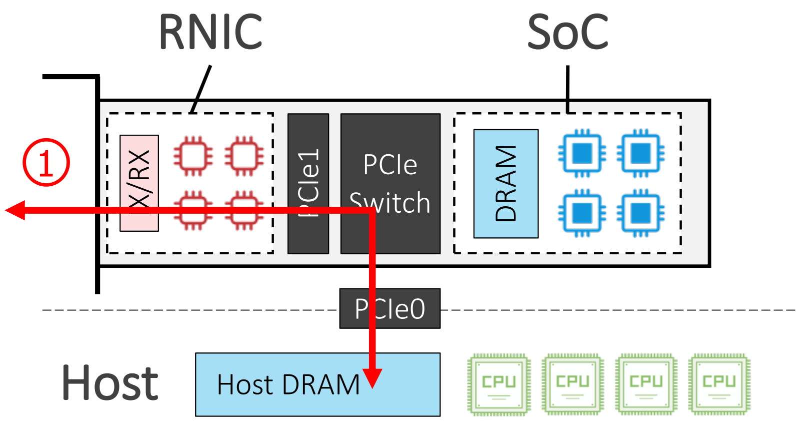
Example 1

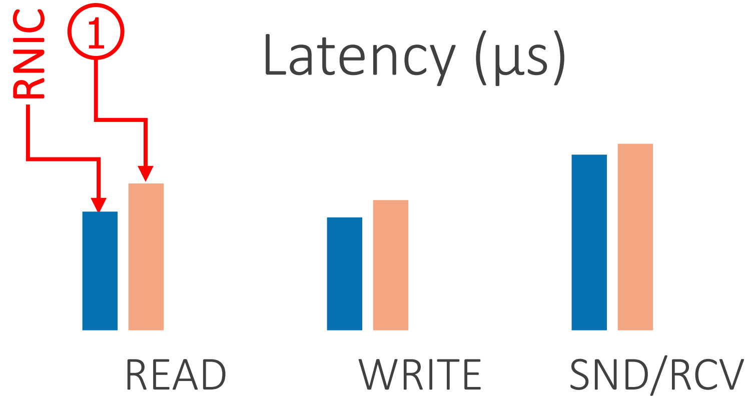
Findings

- NIC-Host is slower than RNIC
- Overhead: PCIe latency (300ns x4)
- Non-trivial for small request (1-2μs)

Takeaway

- If only NIC-Host is used, select RNIC as it is faster, cheaper, and saves power



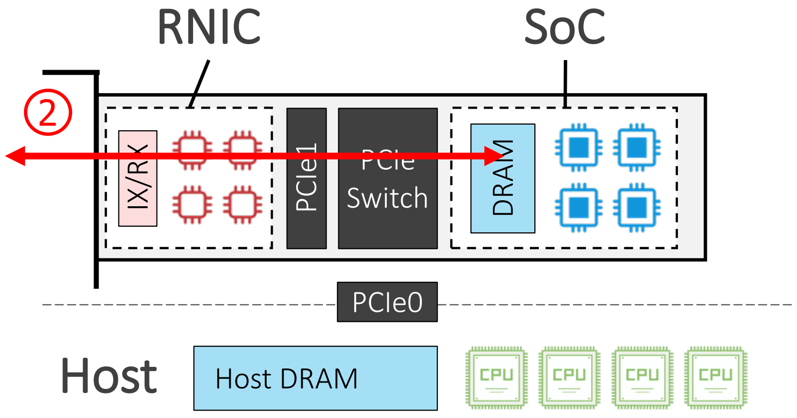


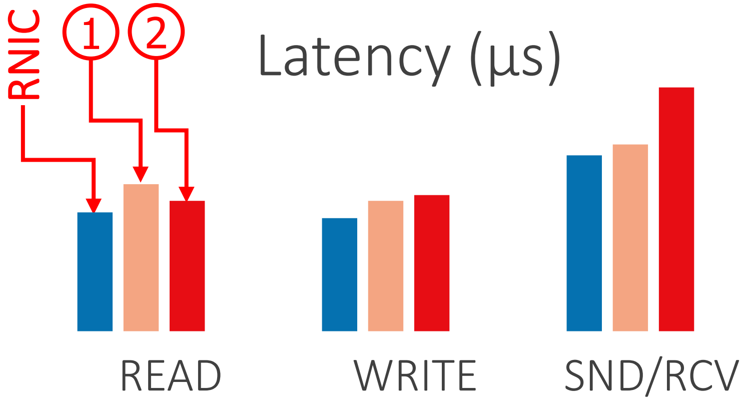
(Payload = 64B)

Example 2

Findings

- NIC-SoC is faster than NIC-host (no PCIe0), but still slower than RNIC (PCIe switch)
- SEND/RECV is much slow (wimpy SoC cores)



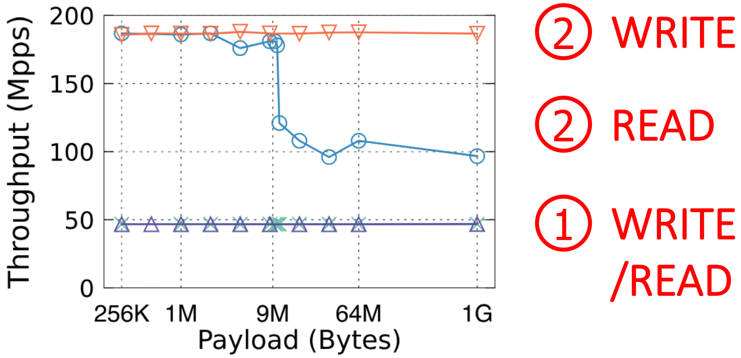


(Payload = 64B)

Example 3

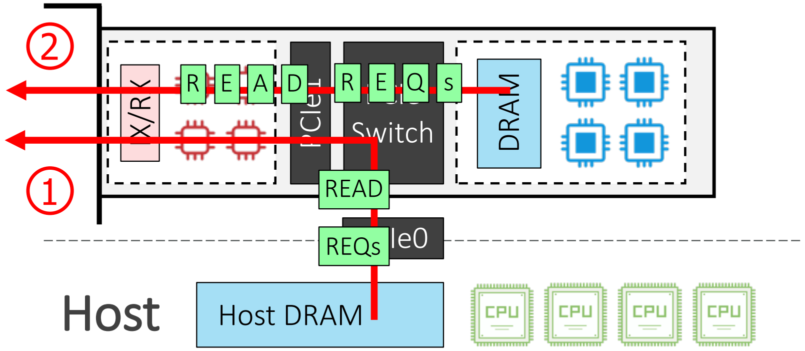
Findings

- RDMA READ performance of NIC-SoC collapses w/ large request (>=9MB)



Advice: avoid large READ requests

- PCIe MTU: Host (512B) vs. SoC (128B)
- NIC-SoC READ: 4× PCIe packets for large requests → HoL blocking



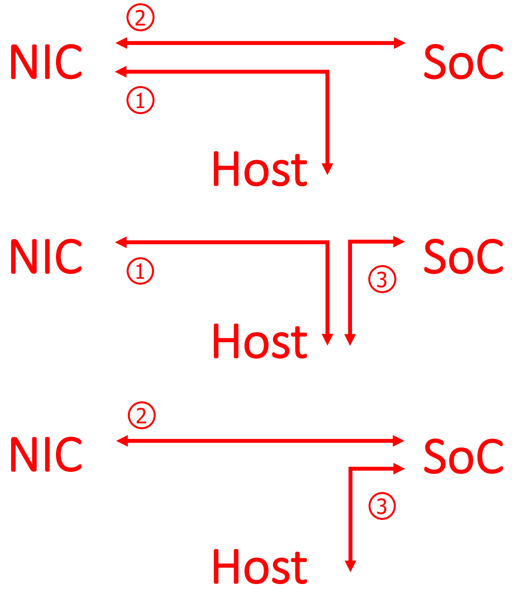
Example 4

Characterizing concurrent paths in DPU

- DPU is always underutilized when only using a single path
- Study the concurrent use of multiple offloading paths (e.g., ①+②)

Takeaway

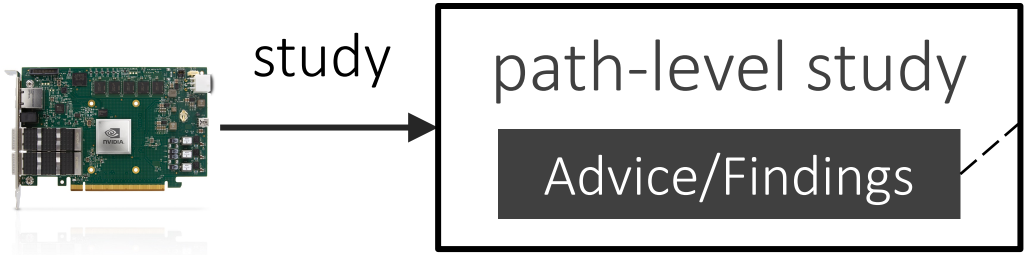
- Concurrent offloading can better utilize DPU, esp. when used in opposition directions (R+W)
- But, carefully avoid interference btw. paths, e.g., NIC cores (①+②) and PCIe switch (②+③)



Characterizing

Our path-level DPU study

- A comprehensive perf. study on *offloading paths* (6) × *primitives* (3)
- 11 *findings/advice* for either individually using a single path or concurrently using multiple paths



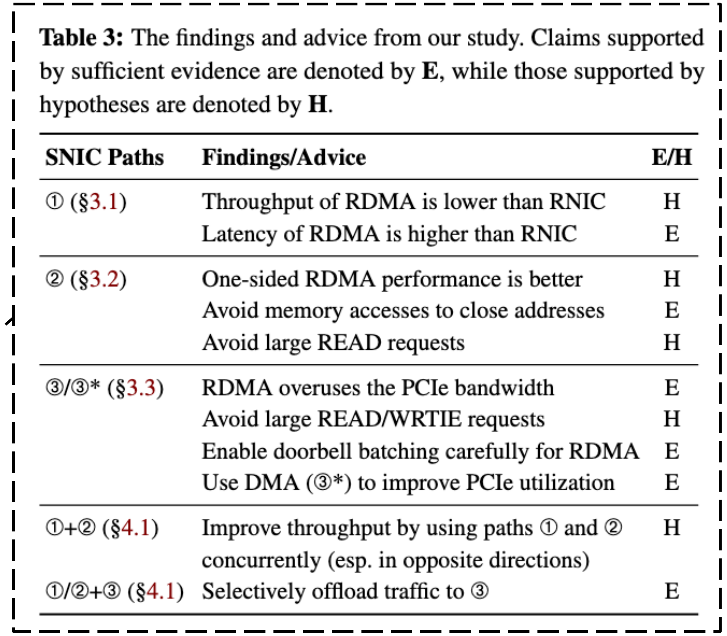
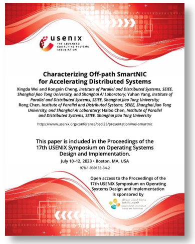


Table 3: The findings and advice from our study. Claims supported by sufficient evidence are denoted by **E**, while those supported by hypotheses are denoted by **H**.

SNIC Paths	Findings/Advice	E/H
① (§3.1)	Throughput of RDMA is lower than RNIC	H
	Latency of RDMA is higher than RNIC	E
② (§3.2)	One-sided RDMA performance is better	H
	Avoid memory accesses to close addresses	E
	Avoid large READ requests	H
③/③* (§3.3)	RDMA overuses the PCIe bandwidth	E
	Avoid large READ/WRTIE requests	H
	Enable doorbell batching carefully for RDMA	E
	Use DMA (③*) to improve PCIe utilization	E
①+② (§4.1)	Improve throughput by using paths ① and ② concurrently (esp. in opposite directions)	H
①/②+③ (§4.1)	Selectively offload traffic to ③	E

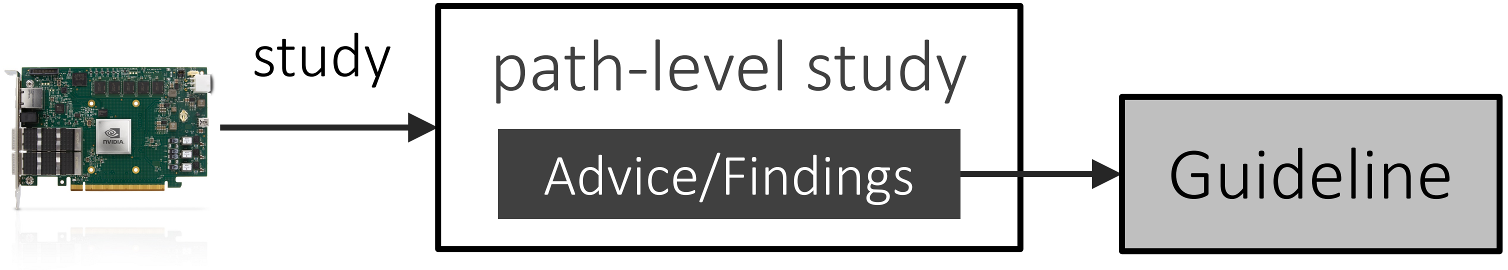


USENIX OSDI’23

Optimizing

A step-by-step optimization guideline for system designers

1. Devise potential alternatives for DPU to support the given functionality, and optimize them based on our study
2. Evaluate and rank alternatives based on system-specific criteria
3. Select and combine alternatives in turn until DPU is saturated



study

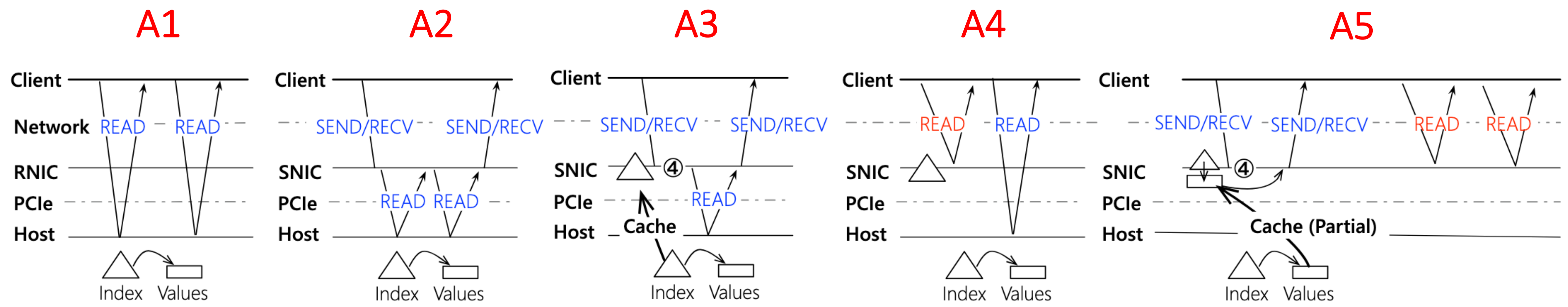
path-level study

Advice/Findings

Guideline

Case Study: Get(k) in Key/Value Store

1. Devise alternatives (A1-A5) and optimize them

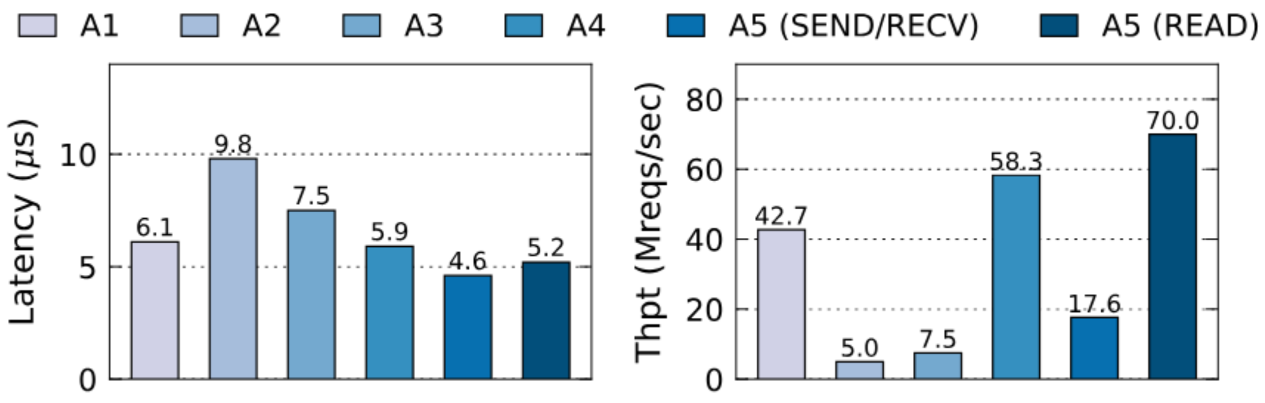


①READx2 ②S/R ③READx2 ②S/R ④ ③READ ②READ ①READ ②S/R ④ + ②READx2

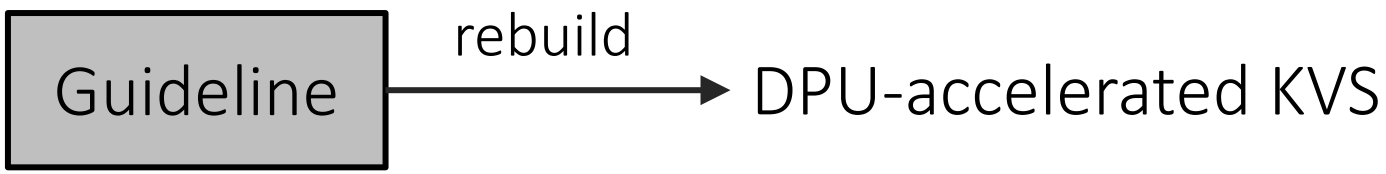
Guideline —rebuild→ DPU-accelerated KVS

Case Study: Get(k) in Key/Value Store

2. Evaluate and rank alternatives based on high performance



Rank: A5 > A4 > A1 > A3 > A2

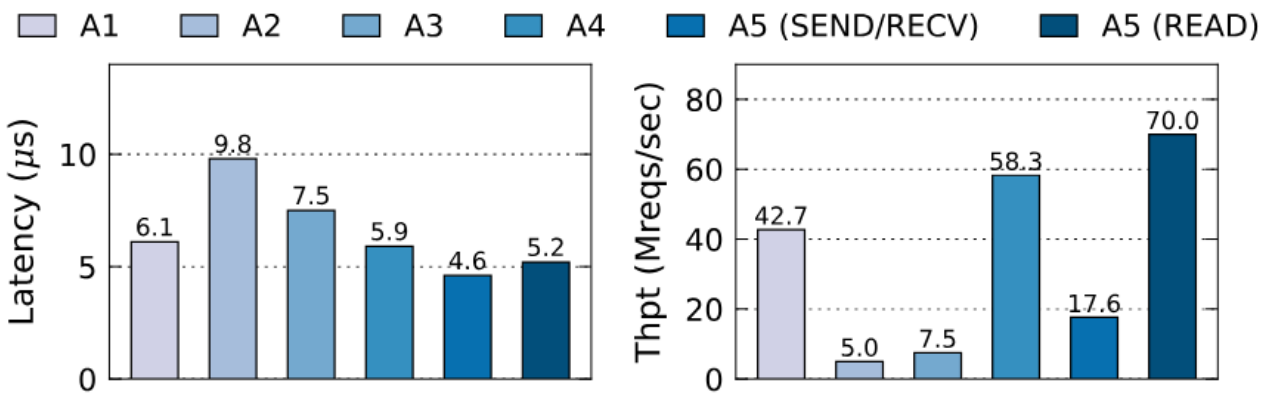

Guideline

rebuild

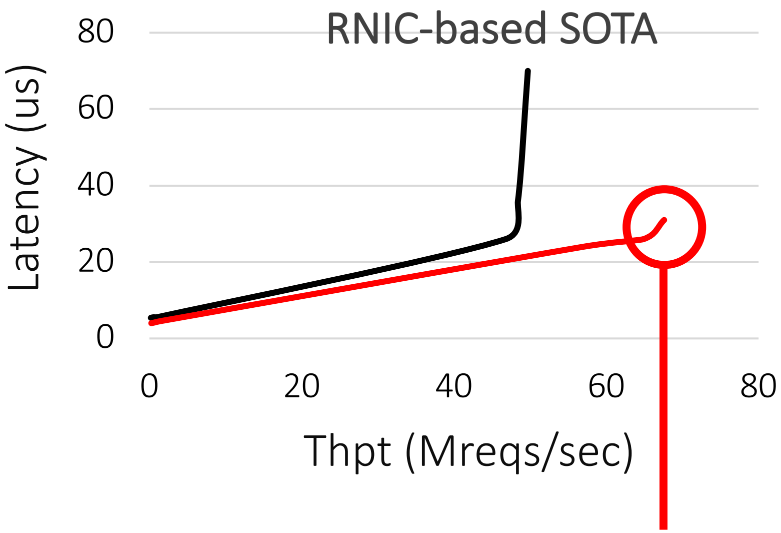
DPU-accelerated KVS

Case Study: Get(k) in Key/Value Store

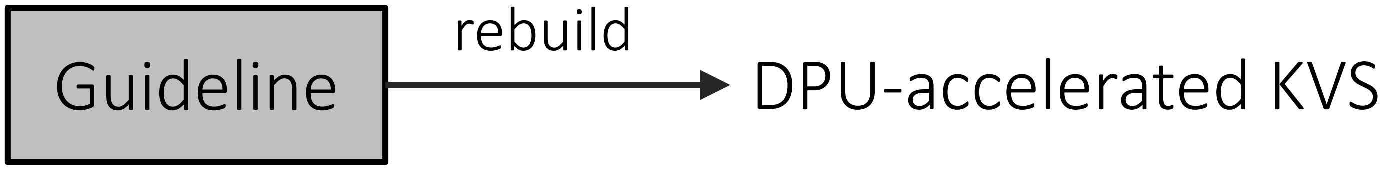
3. Select and combine alternatives in turn until DPU is saturated



Rank: A5 > A4 > A1 > A3 > A2



A5+A4: use A5 first until SoC is saturated, and then select A4


Guideline

rebuild

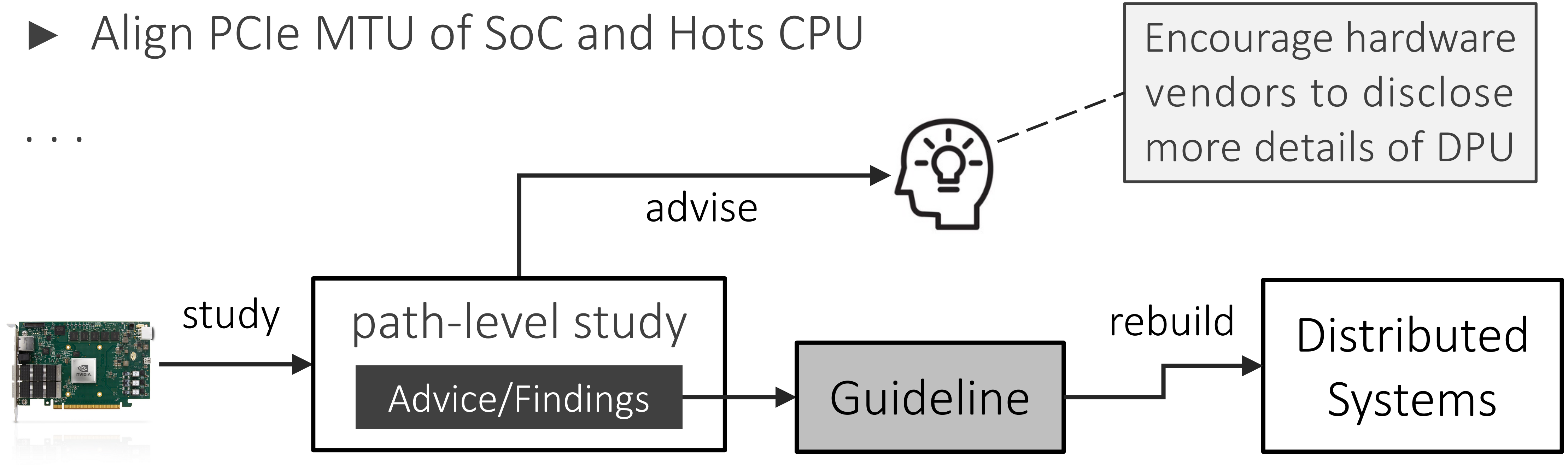
DPU-accelerated KVS

Advising

Suggestions to hardware designers

- ▶ Support CXL to relieve the pressure on SoC cores
- ▶ Support ARM CCI (similar to DDIO on host CPU)
- ▶ Align PCIe MTU of SoC and Hots CPU

. . .

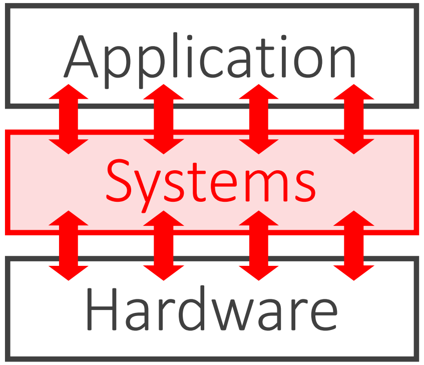


Outline

Case #1: Collaborative offloading

Case #2: Cooperative offloading

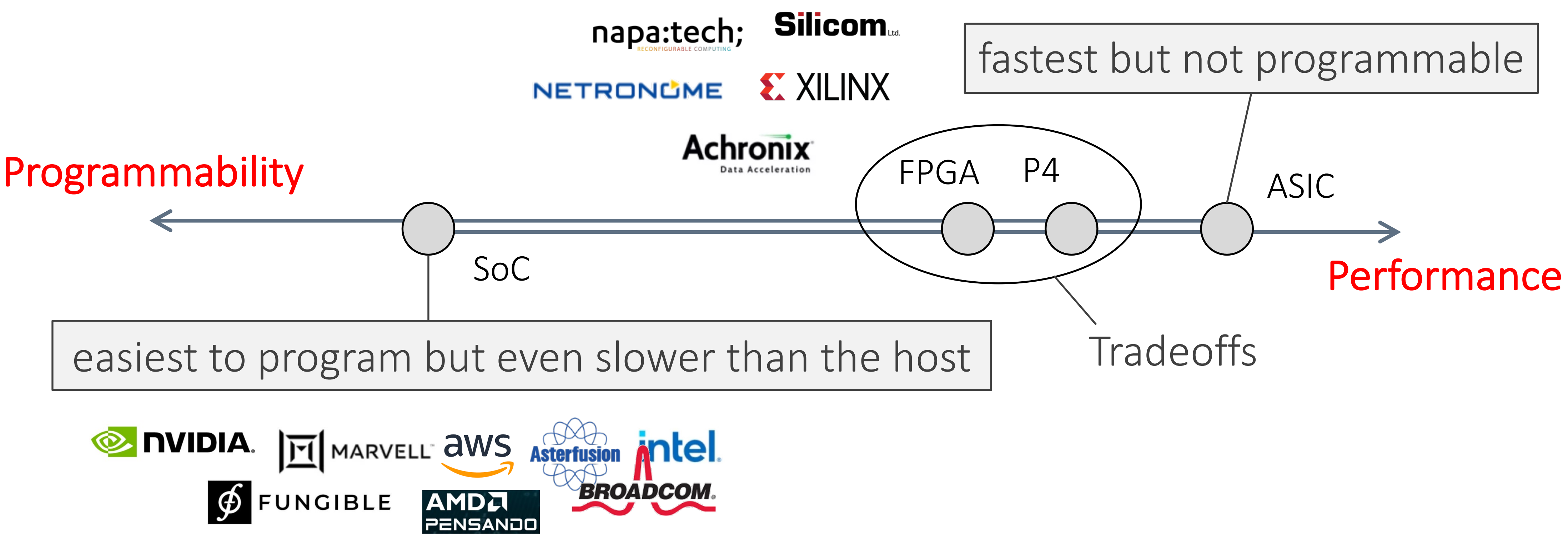
Outlooking systems research for DPU



Outlooking

Which type of processor should be selected, SoC, FPGA, or ASIC?

- An inherent trade-off in programmability and performance



napa:tech; Silicom Ltd.

NETRONOME XILINX

Achronix Data Acceleration

fastest but not programmable

Programmability

FPGA P4

ASIC

SoC

Performance

easiest to program but even slower than the host

Tradeoffs

NVIDIA MARVELL aws Asterfusion intel

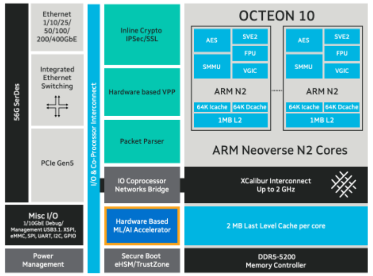
FUNGIBLE AMD PENSANDO BROADCOM

Outlooking

Which type of processor should be selected, SoC, FPGA, or ASIC?

- “Don’t want to CHOOSE, want BOTH”: SoC + FPGA/ASIC/...

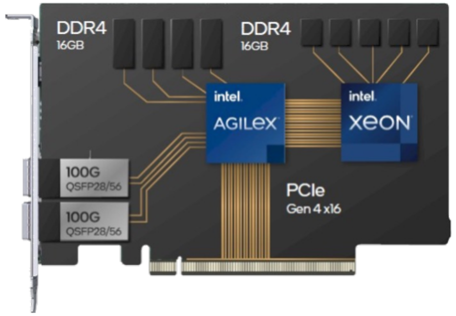
Marvell OCTEON


OCTEON 10
56G SerDes
Ethernet 1/10/25/ 50/100/ 200/400GbE
Integrated Ethernet Switching
PCIe Gen5
I/O & Co-Processor Interconnect
Inline Crypto IPSec/SSL
Hardware based VPP
Packet Parser
AES
SVE2
FPU
SMMU
VGIC
ARM N2
64K Icache
64K Dcache
1MB L2
ARM Neoverse N2 Cores
IO Coprocessor Networks Bridge
XCalibur Interconnect Up to 2 GHz
Misc I/O 1/10GbE Debug/ Management USB3.1, XSPI, eMMC, SPI, UART, I2C, GPIO
Hardware Based ML/AI Accelerator
2 MB Last Level Cache per core
Power Management
Secure Boot eHSM/TrustZone
DDR5-5200 Memory Controller

SoC (ARM)

VPP Accelerators

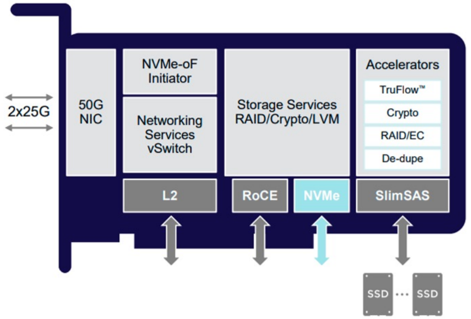
Intel IPU



Soc (Xeon)

Agilex FPGA

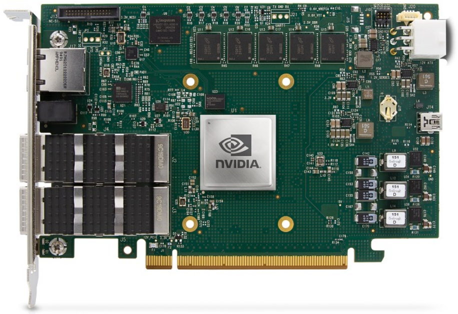
Broadcom Stingray



SoC (ARM)

ASIC Accelerators

NVIDIA BlueField



SoC (ARM)

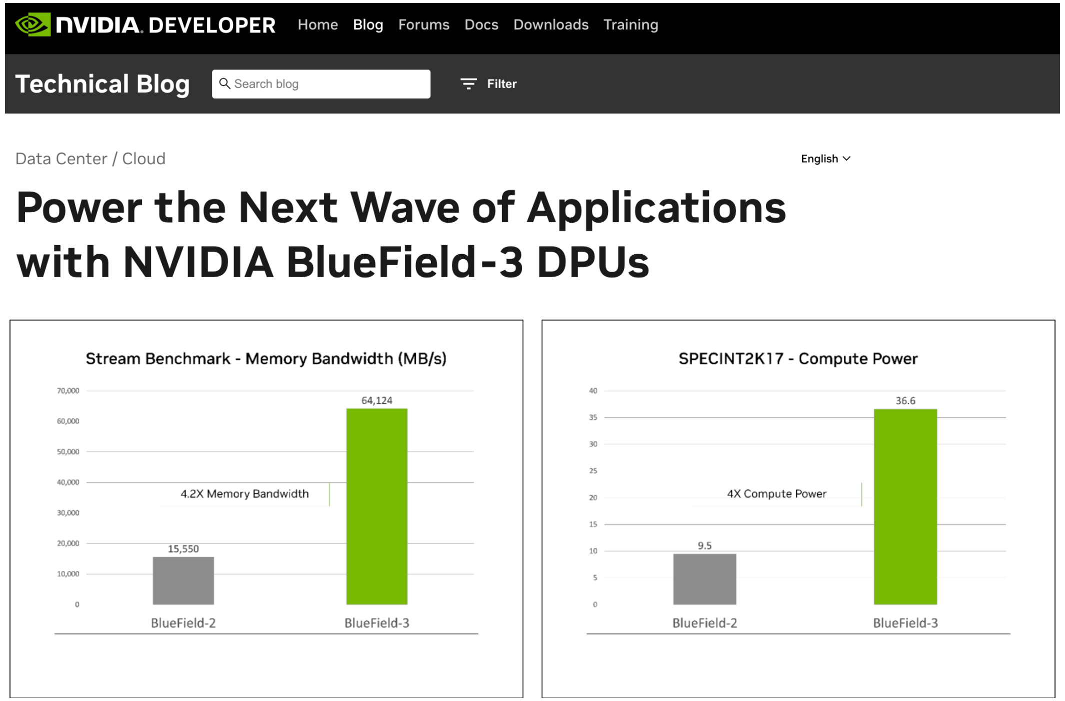
ASIC Accelerators

DPA (RISC-V)

Outlooking

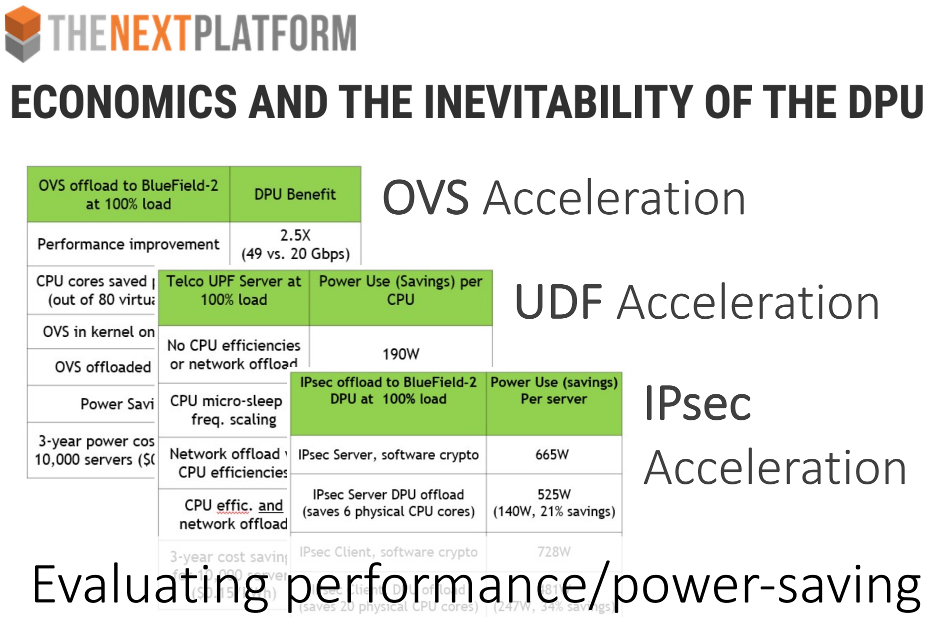
How to measure integrated hardware components in DPU?

- New *metrics*, *benchmarks*, and *toolkits*?



Data Center / Cloud

Power the Next Wave of Applications with NVIDIA BlueField-3 DPUs


THENEXTPLATFORM

ECONOMICS AND THE INEVITABILITY OF THE DPU

OVS offload to BlueField-2 at 100% load	DPU Benefit
Performance improvement	2.5X (49 vs. 20 Gbps)
CPU cores saved p (out of 80 virtu	
OVS in kernel on	
OVS offloaded	
Power Savi	
3-year power cos 10,000 servers ($C	

OVS Acceleration

Telco UPF Server at 100% load	Power Use (Savings) per CPU
No CPU efficiencies or network offload	190W
CPU micro-sleep freq. scaling	
Network offload v CPU efficiencies	
CPU effic. and network offload	
3-year cost savin for 10,000 server ($0.15/kWh)	

UDF Acceleration

IPsec offload to BlueField-2 DPU at 100% load	Power Use (savings) Per server
IPsec Server, software crypto	665W
IPsec Server DPU offload (saves 6 physical CPU cores)	525W (140W, 21% savings)
IPsec Client, software crypto	728W
IPsec Client DPU offload (saves 20 physical CPU cores)	481W (247W, 34% savings)
3-year savings over 10,000 machines ($0.15/kWh)	$5.5M on server side $8.[illegible]M on client side

IPsec Acceleration

Evaluating performance/power-saving of accelerators for network functions

Outlooking

Which domain-specific accelerators deserve to be integrated?

The killer applications of DPUs

datacenter networking, storage, security, and virtualization workloads

Power the Next Wave of Applications with NVIDIA BlueField-3 DPUs

- CPUs that are used for serial processing and running hyperthreaded applications.
- GPUs that excel at parallel processing and are optimized for accelerating modern workloads.
- DPUs that are ideal for infrastructure computing tasks; used to offload, accelerate, and isolate data center networking, storage, security, and manageability workloads.

HPC/AI

Cloud Computing

Storage

Marvell's OCTEON 10 Challenges All Comers For DPU Supremacy

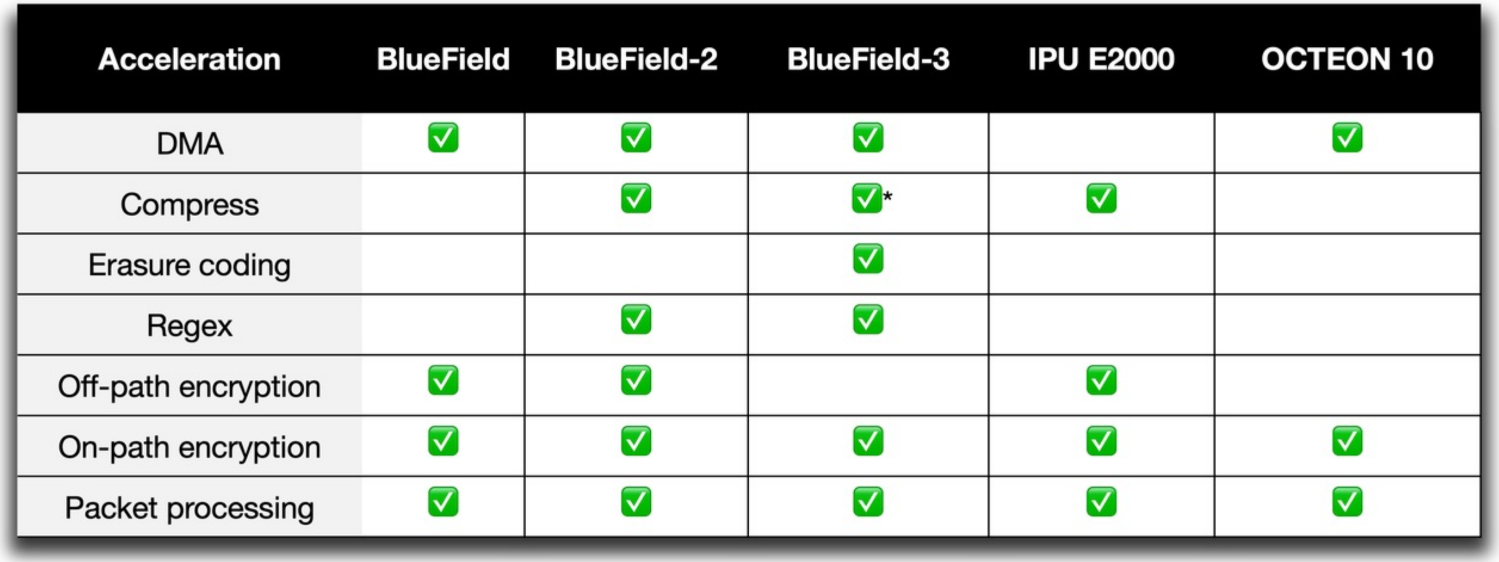
- **Cloud and data center** servers to offload virtual overlay and cryptographic processing for multi-tenant VM, container, and storage services.
- **LTE and 5G vRAN** implementations when paired with Marvell's Fusion-O baseband processor providing a 5G and LTE-A PHY with the OCTEON used for CU or vRAN offload processing.
- **Enterprise router-firewall and SD-WAN** appliances using NFV service chaining to deliver L2/L3 forwarding, VPN termination, SPI, and new AI-based applications and security services.

Source: https://developer.nvidia.com/blog/power-the-next-wave-of-applications-with-nvidia-bluefield-3-dpus/
https://packetpushers.net/marvells-octeon-10-challenges-all-comers-for-dpu-supremacy/

Outlooking

Which domain-specific accelerators deserve to be integrated?

- Compression, encryption, virtualization, packet processing, . . .



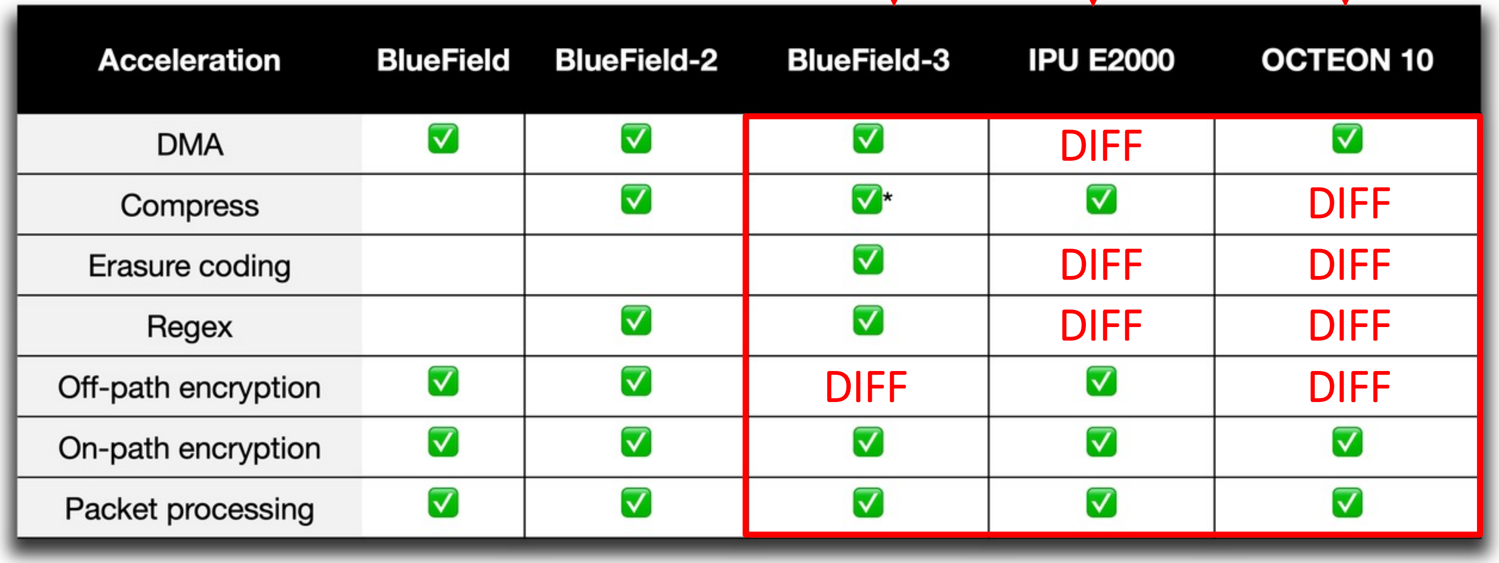
Acceleration	BlueField	BlueField-2	BlueField-3	IPU E2000	OCTEON 10
DMA	✓	✓	✓		✓
Compress		✓	✓*	✓	
Erasure coding			✓		
Regex		✓	✓		
Off-path encryption	✓	✓		✓	
On-path encryption	✓	✓	✓	✓	✓
Packet processing	✓	✓	✓	✓	✓

Year	2016	2020	2023	2023	2021

Outlooking

Which domain-specific accelerators deserve to be integrated?

- Compression, encryption, virtualization, packet processing, . . .



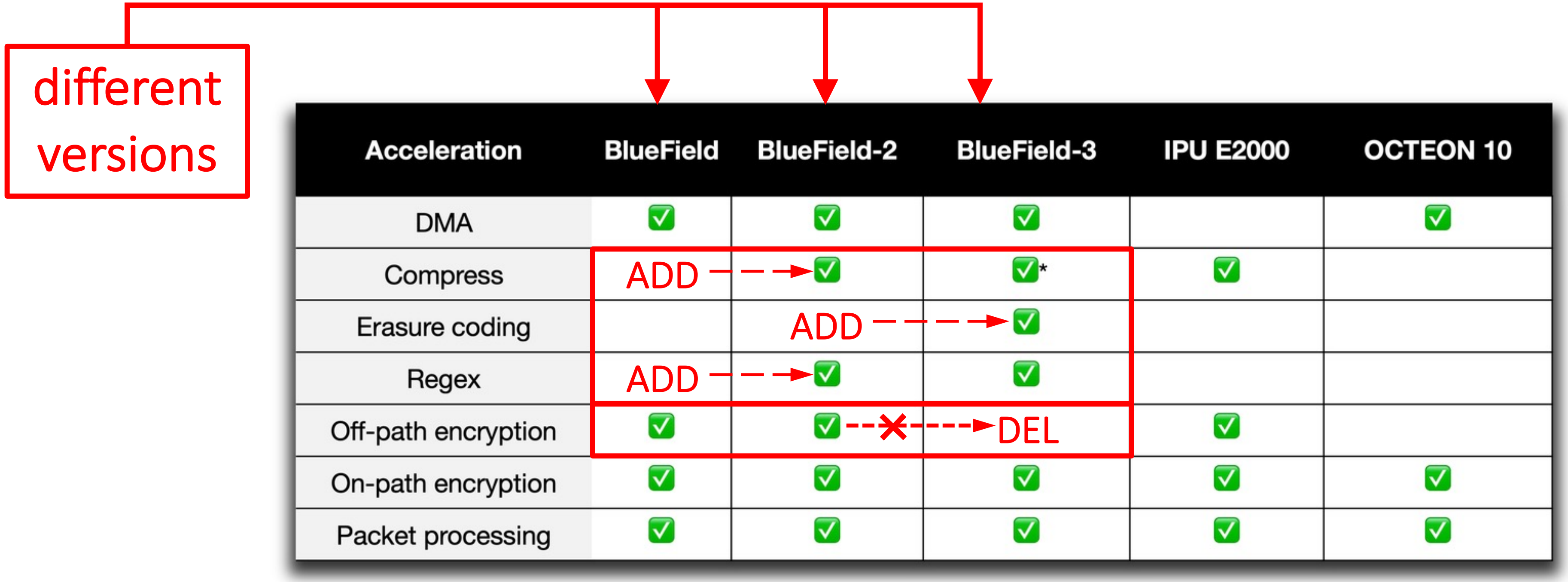
Acceleration	BlueField	BlueField-2	BlueField-3	IPU E2000	OCTEON 10
DMA	✓	✓	✓	DIFF	✓
Compress		✓	✓*	✓	DIFF
Erasure coding			✓	DIFF	DIFF
Regex		✓	✓	DIFF	DIFF
Off-path encryption	✓	✓	DIFF	✓	DIFF
On-path encryption	✓	✓	✓	✓	✓
Packet processing	✓	✓	✓	✓	✓
Year	2016	2020	2023	2023	2021

different vendors

Outlooking

Which domain-specific accelerators deserve to be integrated?

- Compression, encryption, virtualization, packet processing, . . .



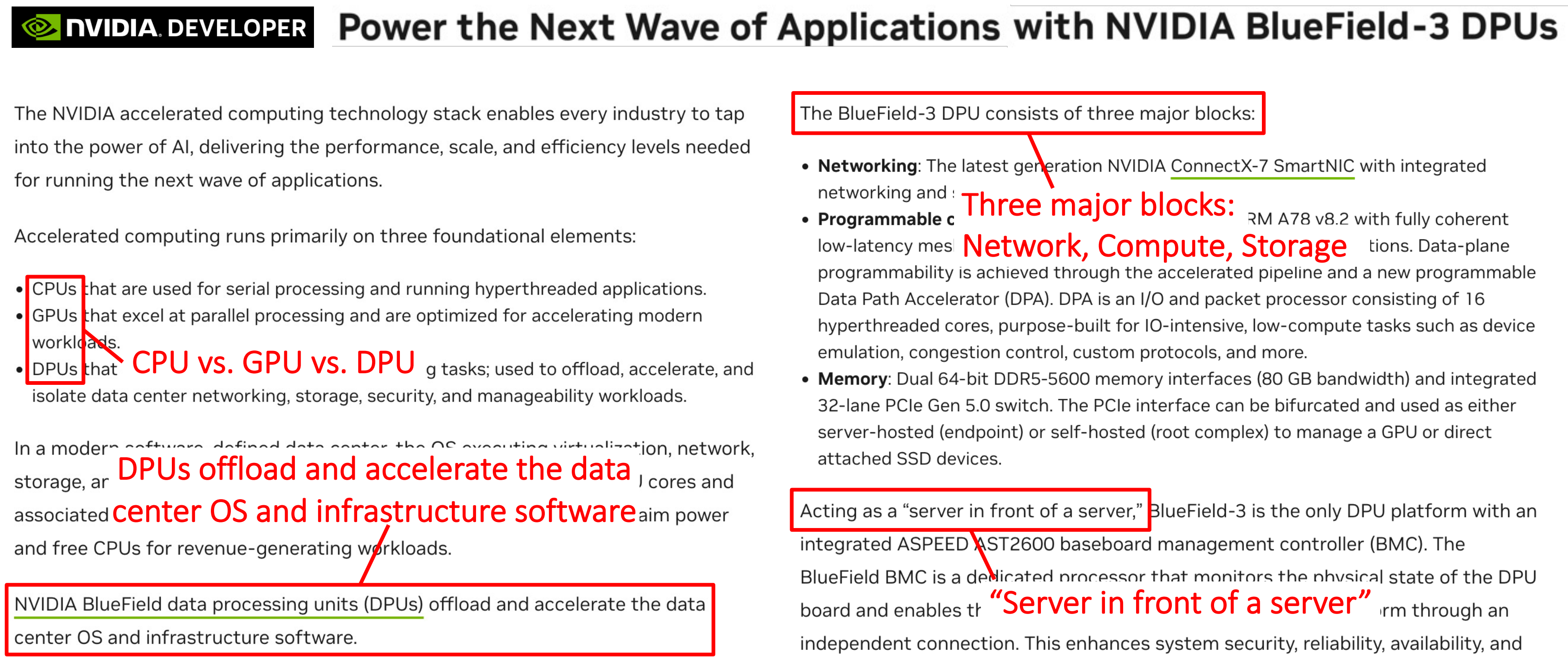
different versions

Acceleration	BlueField	BlueField-2	BlueField-3	IPU E2000	OCTEON 10
DMA	✓	✓	✓		✓
Compress	ADD →	✓	✓*	✓	
Erasure coding		ADD →	✓		
Regex	ADD →	✓	✓		
Off-path encryption	✓	✓ --✗--→	DEL	✓	
On-path encryption	✓	✓	✓	✓	✓
Packet processing	✓	✓	✓	✓	✓
Year	2016	2020	2023	2023	2021

Outlooking

How to unify system abstraction & programming interface?

- e.g., BlueField: PCIe accelerator vs. a standalone server



NVIDIA DEVELOPER **Power the Next Wave of Applications with NVIDIA BlueField-3 DPUs**

The NVIDIA accelerated computing technology stack enables every industry to tap into the power of AI, delivering the performance, scale, and efficiency levels needed for running the next wave of applications.

Accelerated computing runs primarily on three foundational elements:

- CPUs that are used for serial processing and running hyperthreaded applications.
- GPUs that excel at parallel processing and are optimized for accelerating modern workloads.
- DPUs that g tasks; used to offload, accelerate, and isolate data center networking, storage, security, and manageability workloads.

CPU vs. GPU vs. DPU

In a modern software-defined data center, the OS executing virtualization, network, storage, ar cores and associated aim power and free CPUs for revenue-generating workloads.

DPUs offload and accelerate the data center OS and infrastructure software

NVIDIA BlueField data processing units (DPUs) offload and accelerate the data center OS and infrastructure software.

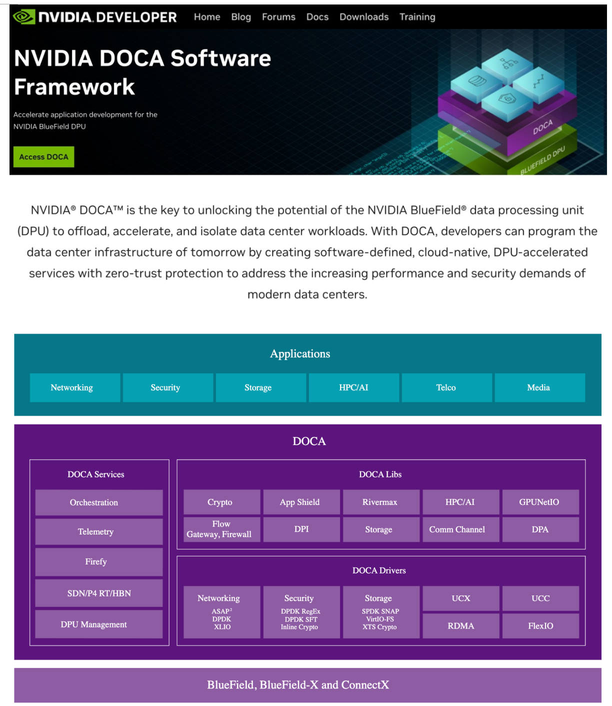
The BlueField-3 DPU consists of three major blocks:

- **Networking**: The latest generation NVIDIA ConnectX-7 SmartNIC with integrated networking and
- **Programmable c** RM A78 v8.2 with fully coherent low-latency mes tions. Data-plane programmability is achieved through the accelerated pipeline and a new programmable Data Path Accelerator (DPA). DPA is an I/O and packet processor consisting of 16 hyperthreaded cores, purpose-built for IO-intensive, low-compute tasks such as device emulation, congestion control, custom protocols, and more.
- **Memory**: Dual 64-bit DDR5-5600 memory interfaces (80 GB bandwidth) and integrated 32-lane PCIe Gen 5.0 switch. The PCIe interface can be bifurcated and used as either server-hosted (endpoint) or self-hosted (root complex) to manage a GPU or direct attached SSD devices.

Three major blocks:
Network, Compute, Storage

Acting as a “server in front of a server,” BlueField-3 is the only DPU platform with an integrated ASPEED AST2600 baseboard management controller (BMC). The BlueField BMC is a dedicated processor that monitors the physical state of the DPU board and enables th rm through an independent connection. This enhances system security, reliability, availability, and

“Server in front of a server”



NVIDIA DEVELOPER Home Blog Forums Docs Downloads Training

NVIDIA DOCA Software Framework

Accelerate application development for the NVIDIA BlueField DPU

Access DOCA

NVIDIA® DOCA™ is the key to unlocking the potential of the NVIDIA BlueField® data processing unit (DPU) to offload, accelerate, and isolate data center workloads. With DOCA, developers can program the data center infrastructure of tomorrow by creating software-defined, cloud-native, DPU-accelerated services with zero-trust protection to address the increasing performance and security demands of modern data centers.

Applications
Networking | Security | Storage | HPC/AI | Telco | Media

DOCA
DOCA Services: Orchestration, Telemetry, Firefy, SDN/P4 RT/HBN, DPU Management
DOCA Libs: Crypto, App Shield, Rivermax, HPC/AI, GPUNetIO, Flow Gateway, Firewall, DPI, Storage, Comm Channel, DPA
DOCA Drivers: Networking (ASAP², DPDK, XLIO), Security (DPDK RegEx, DPDK SFT, Inline Crypto), Storage (SPDK SNAP, VirtIO-FS, XTS Crypto), UCX, UCC, RDMA, FlexIO

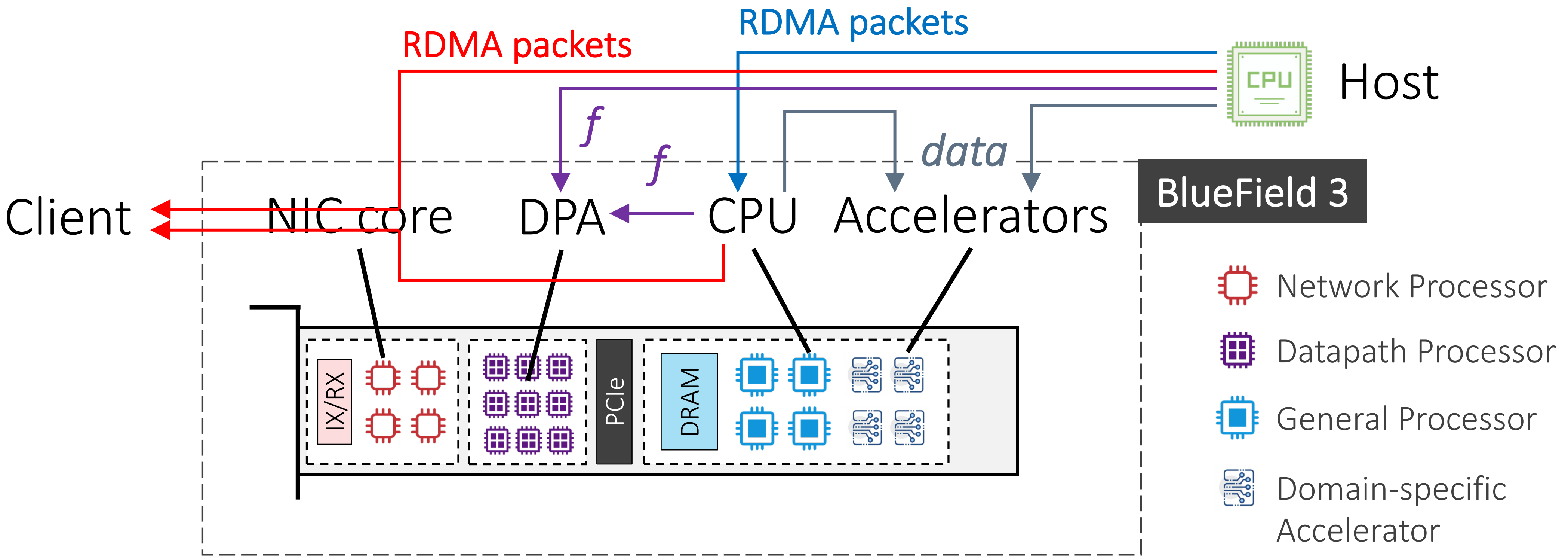
BlueField, BlueField-X and ConnectX

Source: https://developer.nvidia.com/blog/power-the-next-wave-of-applications-with-nvidia-bluefield-3-dpus/

Outlooking

How to unify system abstraction & programming interface?

- e.g., BlueField: PCIe accelerator vs. a standalone server



Conclusion & Thanks

Hardware evolution:

single capability breakthrough & *multiple capability integration*

Our approach: characterizing, optimizing, and advising

- ▶ Collaborative offloading for multiple devices (e.g., RDMA & NVM)
- ▶ Cooperative offloading for intelligent devices (e.g., DPU)

Our outlook on systems research for DPU

See more at https://ipads.se.sjtu.edu.cn/rong_chen