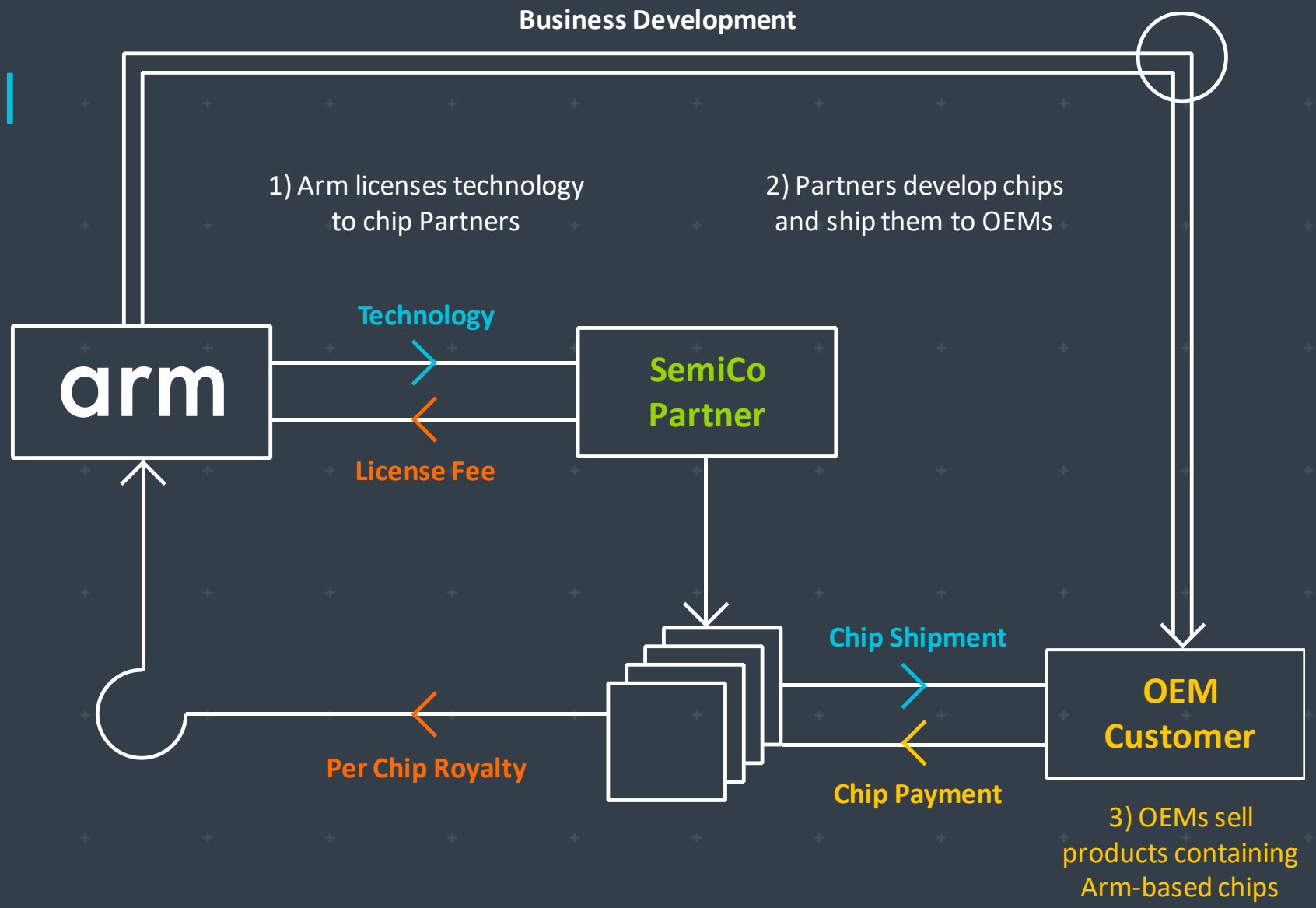
A Continuous Partnership Model

Arm develops technology that is licensed to semiconductor companies.

Arm receives an upfront license fee and a royalty on every chip that contains its technology.



arm

The Architects of Global Possibilities

The global leader in the development of licensable technology

- R&D outsourcing for semiconductor companies

Focused on freedom and flexibility to innovate

- Technology reused across multiple applications

With a partnership based culture & business model

- Licensees take advantage of learnings from a uniquely collaborative ecosystem

1,690+
licenses, growing by 100+ every year

500
licensees
Industry leaders and high-growth start-ups; chip companies and OEMs

155+bn
Arm-based chips shipped to-date

25+bn
Arm-based chips shipped in 2019

arm

My personal ”MPI” retrospective at Arm

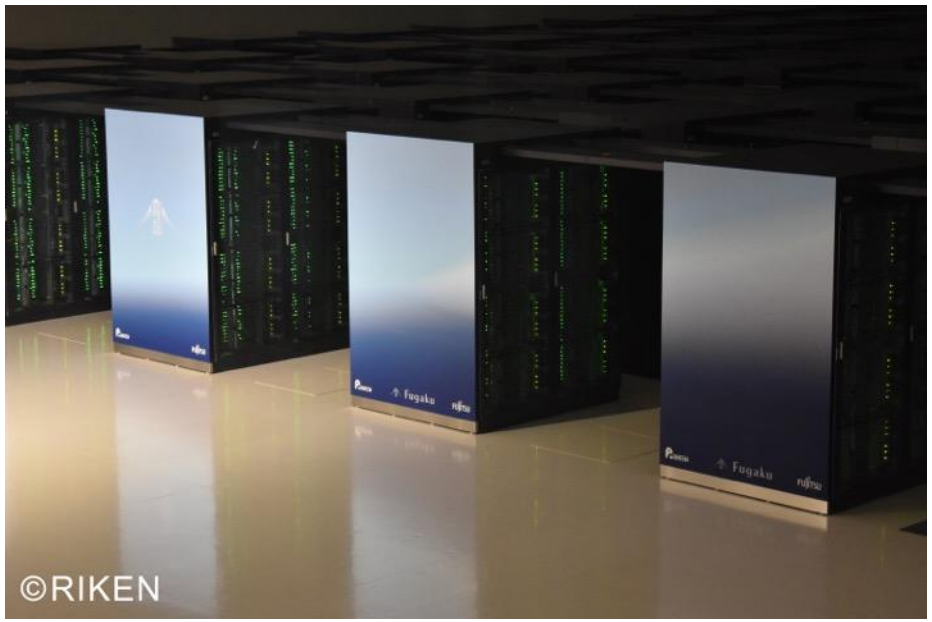
My first MPI development platform at Arm, 2016

arm

Fujitsu's Fugaku: Fastest Supercomputer in the World

Top place in 4 categories:

Top500	@ 416 Pflop/s
HPCG	@ 13.4 Pflop/s
HPL-AI	@ 1.42 Eflop/s
Graph 500	@ 70980 GTEPS




TOP500 CERTIFICATE

The List.

Supercomputer Fugaku - A64FX 48C 2.2GHz, Tofu interconnect D

RIKEN Center for Computational Science, Japan

is ranked

No. 1

among the World's TOP500 Supercomputers

with 415.53 Pflop/s Linpack Performance

in the 55th TOP500 List published at the ISC 2020 Digital Conference on June 22nd, 2020.

Congratulations from the TOP500 Editors

Erich Strohmaier, NERSC/Berkeley Lab
Jack Dongarra, University of Tennessee
Horst Simon, NERSC/Berkeley Lab
Martin Meuer, Prometeus

arm

Arm Neoverse Momentum in Servers & HPC

Arm Neoverse Announced

VMware demonstrated ESXi on 64-bit ARM

Neoverse N1 & E1 Platform announced

AWS announced 2nd Generation Arm-based Graviton2 Server CPU

Marvell Announced 96-core ThunderX3 Server SoC

Fujitsu Fugaku Riken #1 on Top500

Huawei released Kunpeng 920 CPU and TaiShan server platform

Nvidia CUDA stack to Arm Platforms

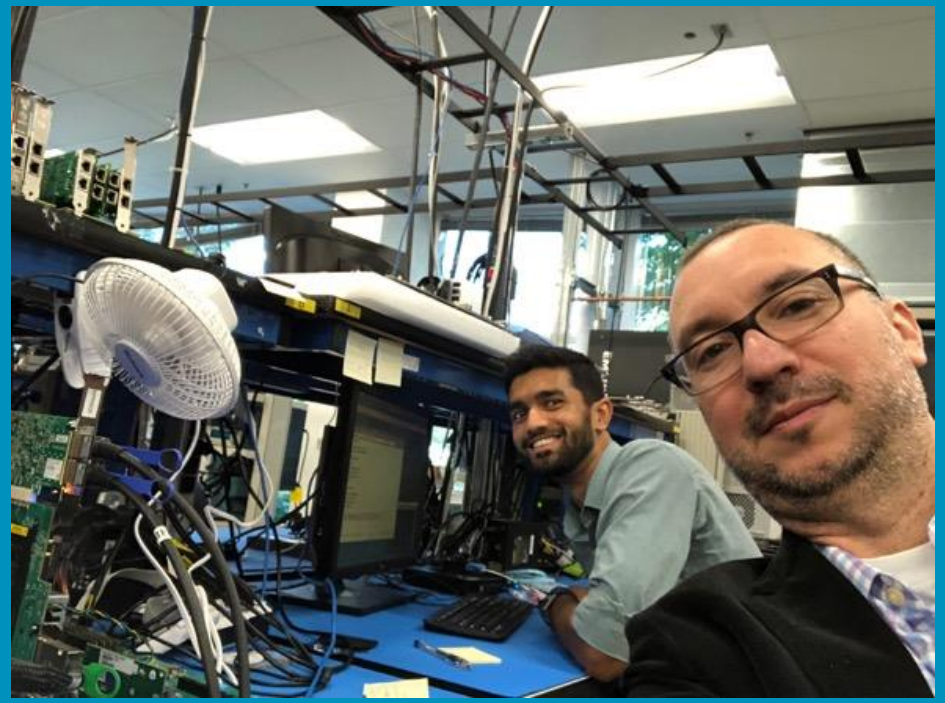
Marvell's ThunderX2 Solution for Microsoft Azure Development

Ampere announced industry's 1st 80-Core Server SoC (128 Altra Max)

EPI Zeus License

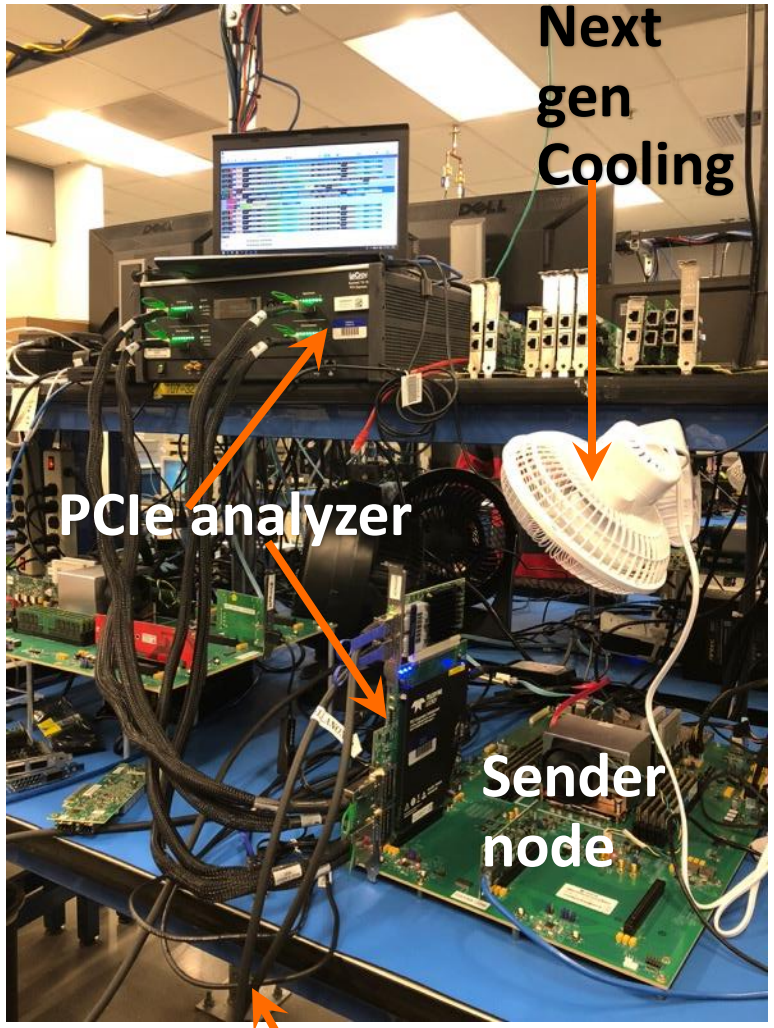
arm

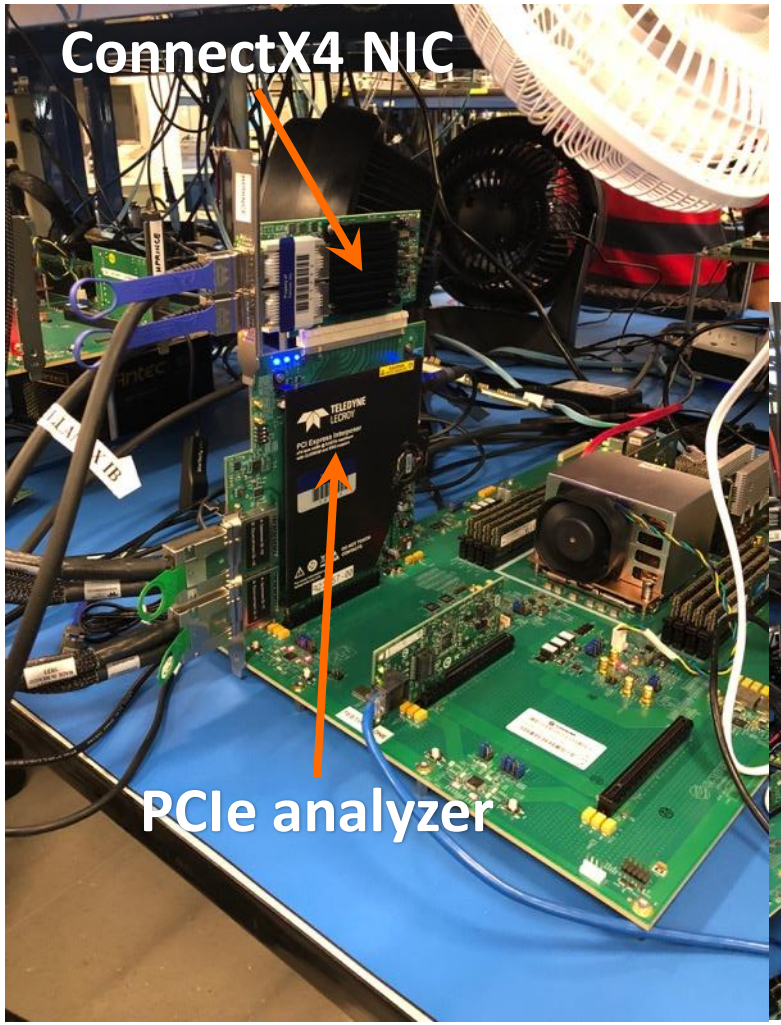
Breaking Band: A Breakdown of High-performance Commination on Arm



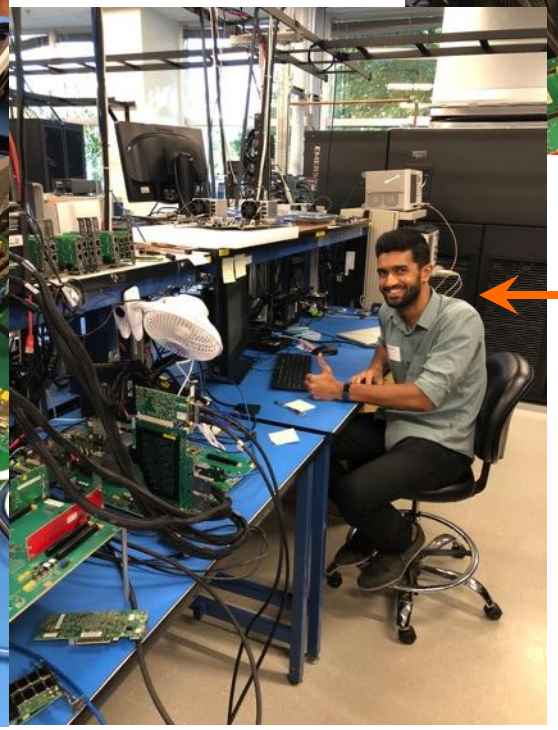
Zambre, R., Grodowitz, M., Chandramowlishwaran, A. and Shamis, P., 2019, August. Breaking Band: A Breakdown of High-performance Communication. In *Proceedings of the 48th International Conference on Parallel Processing*

The ”Breaking” part of the paper …





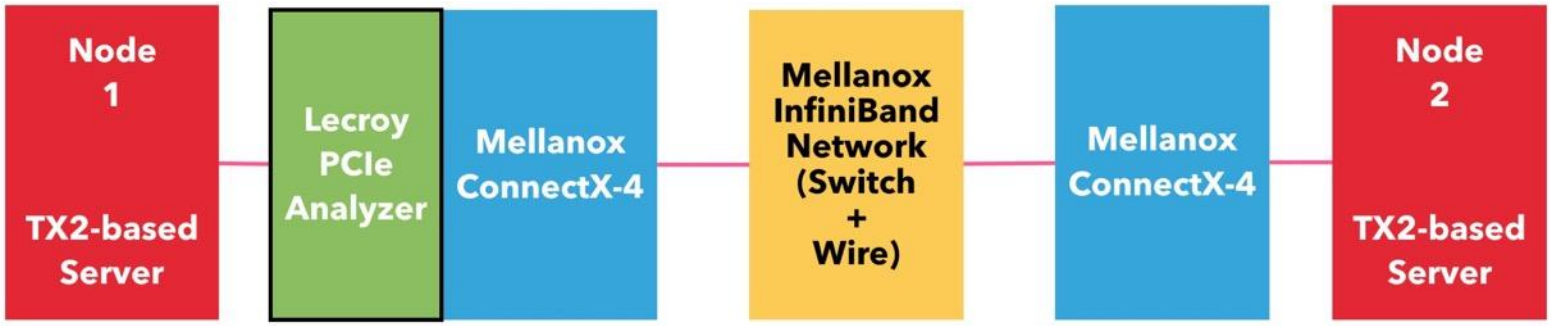
PCIe trace viewer



Rohit is trying not to break anything

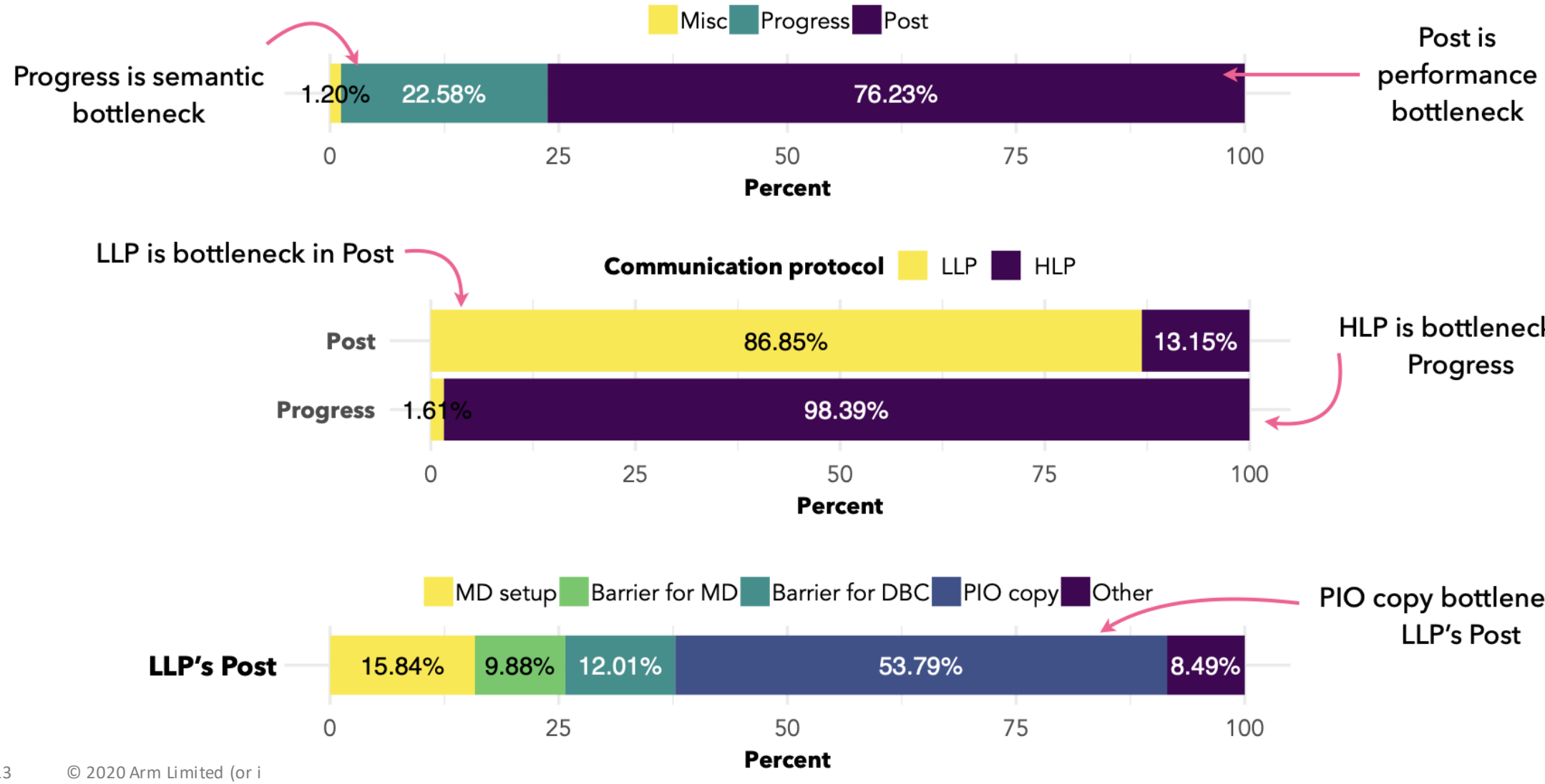
arm

Latency – Getting to the Bottom of It

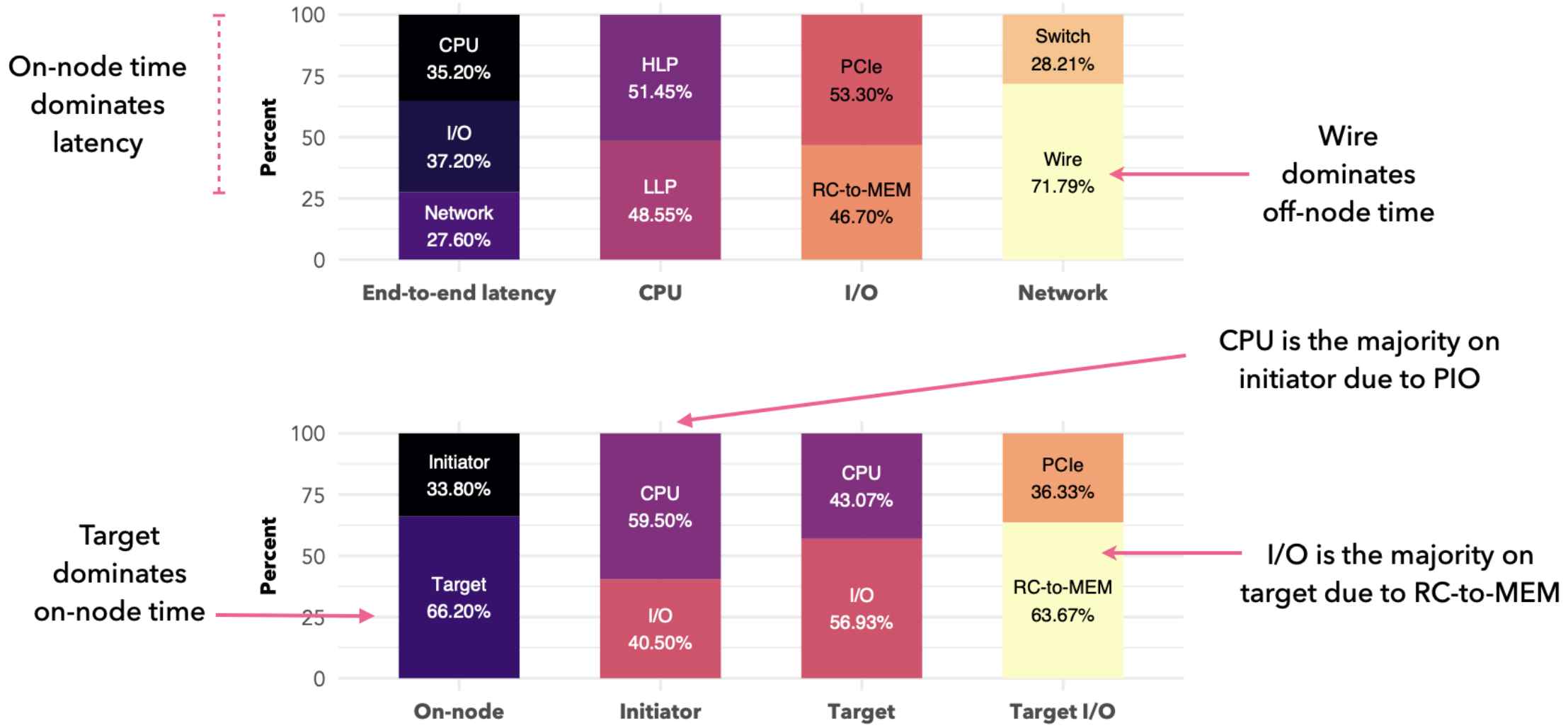


Link Tra			TLP		MWr(64)	Length	RequesterID	Tag	Address	1st BE	Last BE	Data	VC ID	ExplicitACK	Metrics	# Packets	Time Delta	Time Stamp
4143180	R→	8.0 x16	3645	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268156		2	264.000 ns	0001 . 429 405 854 2 s
4143184	R→	8.0 x16	3646	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268161		2	260.000 ns	0001 . 429 406 118 2 s
4143186	R→	8.0 x16	3647	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268166		2	317.000 ns	0001 . 429 406 378 2 s
4143187	R→	8.0 x16	3648	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268169		2	258.000 ns	0001 . 429 406 695 2 s
4143188	R→	8.0 x16	3649	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268173		2	264.000 ns	0001 . 429 406 953 2 s
4143193	R→	8.0 x16	3650	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268179		2	270.250 ns	0001 . 429 407 217 2 s
4143196	R→	8.0 x16	3653	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268189		2	335.000 ns	0001 . 429 407 487 5 s
4143199	R→	8.0 x16	3656	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268193		2	262.000 ns	0001 . 429 407 822 5 s
4143203	R→	8.0 x16	3657	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268198		2	264.000 ns	0001 . 429 408 084 5 s
4143205	R→	8.0 x16	3658	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268204		2	319.000 ns	0001 . 429 408 348 5 s
4143206	R→	8.0 x16	3659	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268207		2	264.000 ns	0001 . 429 408 667 5 s
4143207	R→	8.0 x16	3660	Mem	011:00000	16	000:00:0	0	00000040:00026B00	1111	1111	16 dwords	0	Packet #8268209		2	258.000 ns	0001 . 429 408 931 5 s
4143210	R→	8.0 x16	3661	Mem	011:00000	16	000:00:0	0	00000040:00026A00	1111	1111	16 dwords	0	Packet #8268215		2	264.000 ns	0001 . 429 409 189 5 s

Send / Post Flow



Software/Hardware/Network Latency Breakdown



arm

Key Contributions

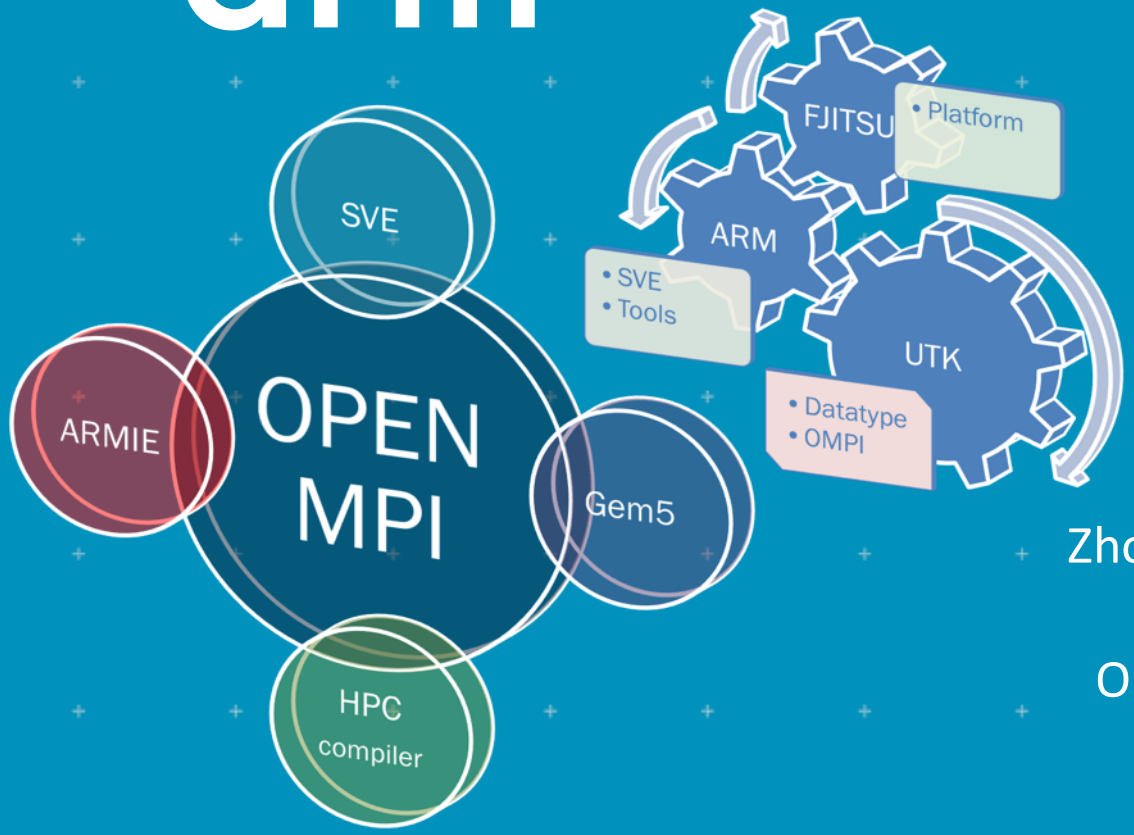
- Our models explain observed performance within 5% margin error.
- Breakdown explain where, why, and how much time is spent providing key insights.
- Breakdown helps researches and developers guide their optimization efforts.

"When you can measure what you are speaking about,
and express it in numbers, you know something about it" — Lord Kelvin

Special thanks to
Giri Chukkapalli, and Ham Prince from Marvell Technology Group,
Yossi Itigin from Mellanox Technologies, and
Pavan Balaji from Argonne National Laboratory

arm

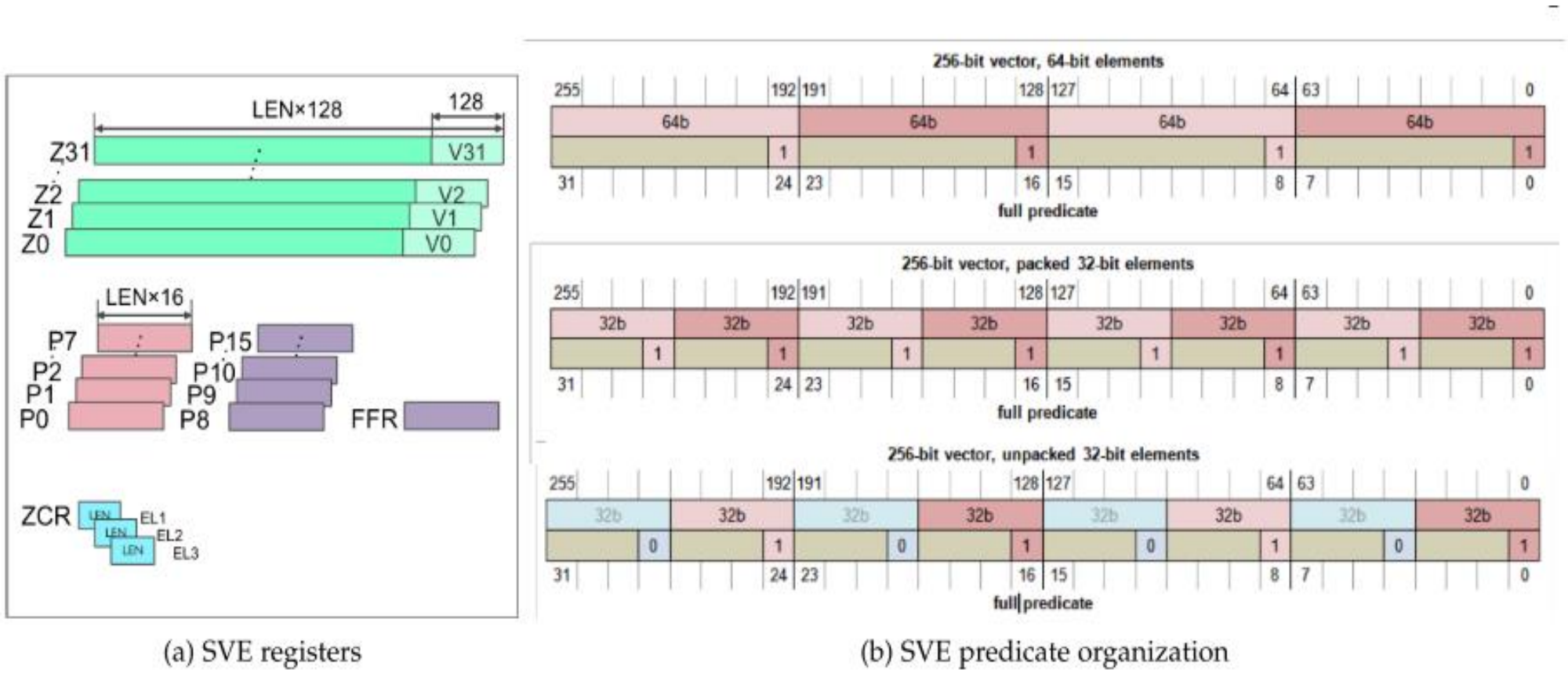
arm



Using Arm Scalable Vector Extension to Optimize Open MPI

Zhong D, Shamis P, Cao Q, Bosilca G, Sumimoto S, Miura K, Dongarra J. Using Arm Scalable Vector Extension to Optimize OPEN MPI. In2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID) 2020

Scalable Vector Extensions



(a) SVE registers

(b) SVE predicate organization

Fig. 1: SVE architectural state - vector registers (Z0–Z31), predicate registers (P0–P15), the first-fault register (FFR) and exception-level specific control registers (ZCR_EL1–ZCR_EL3)

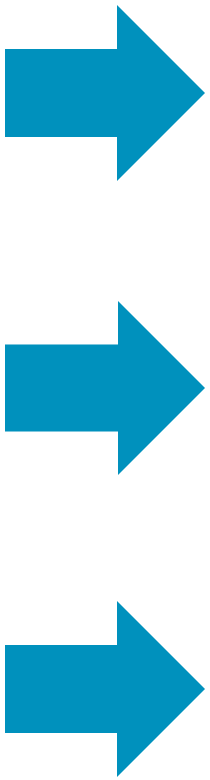
SVE Applications for MPI

MPI Wants….

MPI pack and unpack contiguous buffer

MPI pack and unpack non-contiguous buffer

MPI reduction operation



And SVE can do …

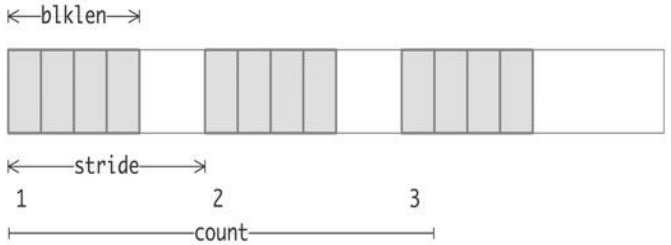
Predicated multiple vector contiguous load and store

Rich addressing mode that enables non-linear data access that can deal with non-contiguous data

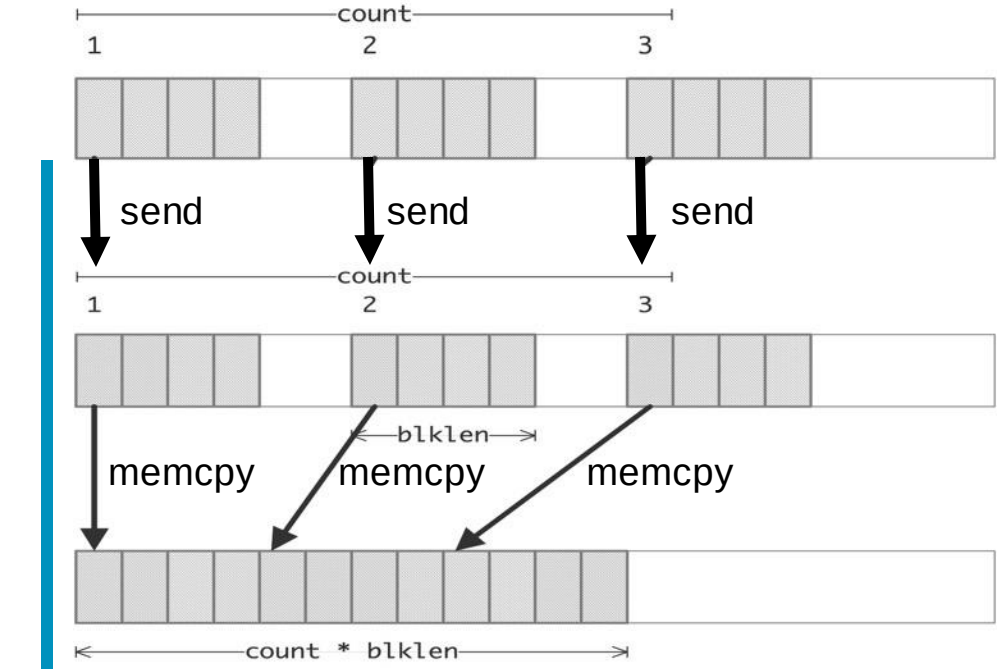
Horizontal reduction operations which applies to more types of reducible loop carried dependencies including both logical, integer and floating-point reductions;

MPI - Gather load example

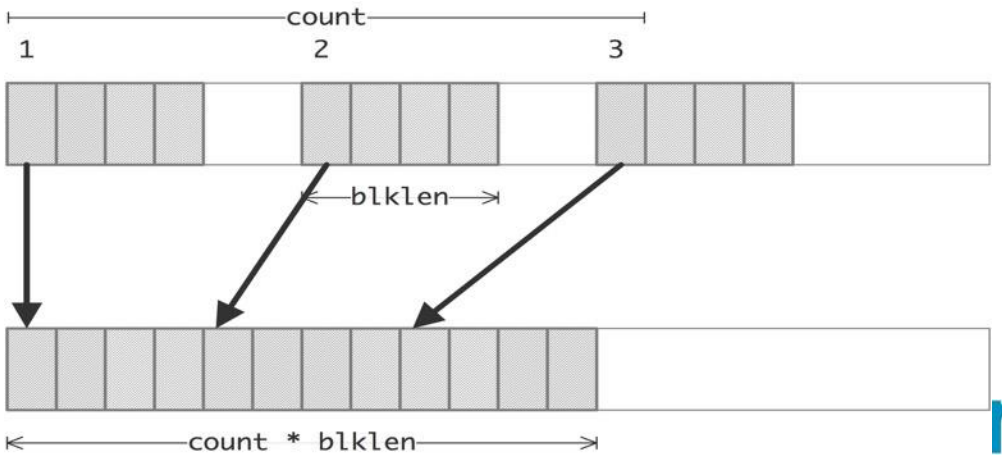
Gather Load /Scatter Store



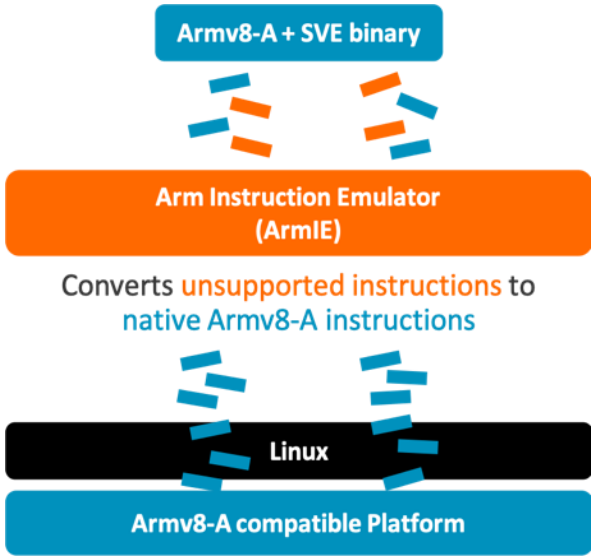
Typical Implementation



Vector Implementation



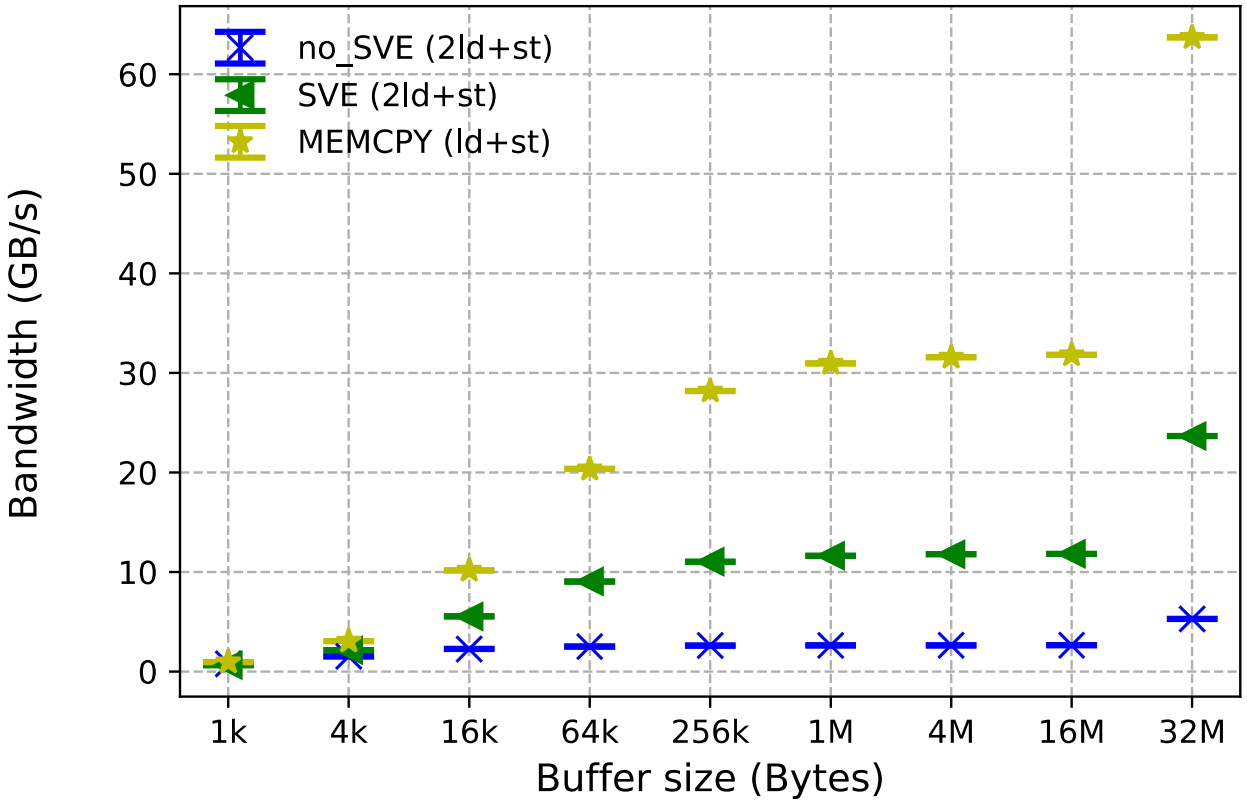
The methodology of development and evaluation



arm

MPI Reduction with SVE on Fujitsu A64FX (pre-production)

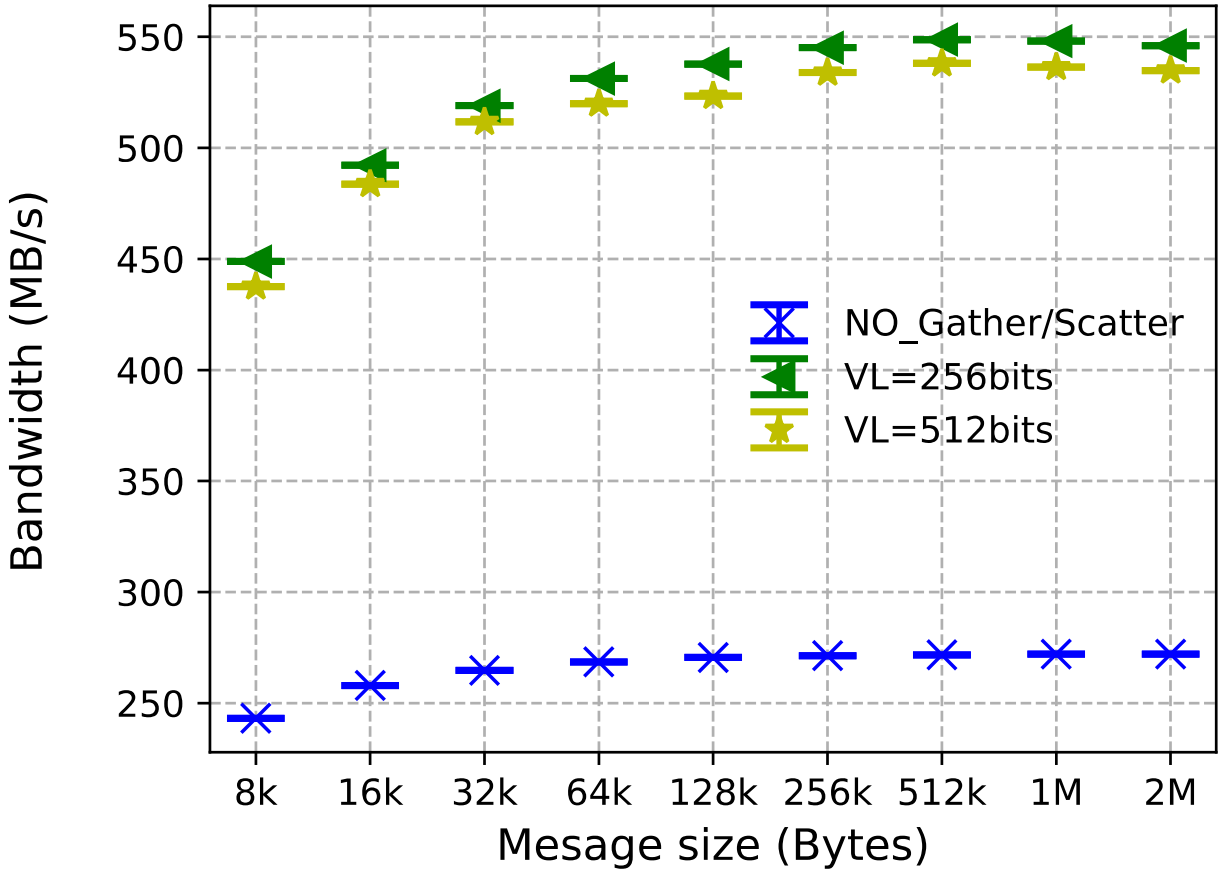
- Our results demonstrate that with SVE-enabled operation it is 4× faster than element-wise operation.
- We also compare MPI operation performance together with memcpy() which indicates the peak memory bandwidth.
- For MPI reduction operation it needs 2loads + 1store + computation , for memcpy() it only needs 1load+1store.



arm

MPI Scatter/Gather with SVE on Fujitsu A64FX (pre-production)

- Pack and unpack using gather/scatter feature with different vector length for non-contiguous buffer.
- The green and yellow line indicates the performance using vector length 256 bits and 512 bits respectively with our gather and scatter strategy.
- Compared to the blue line which is not using gather scatter feature. We can see that that optimized algorithm is 2× faster.



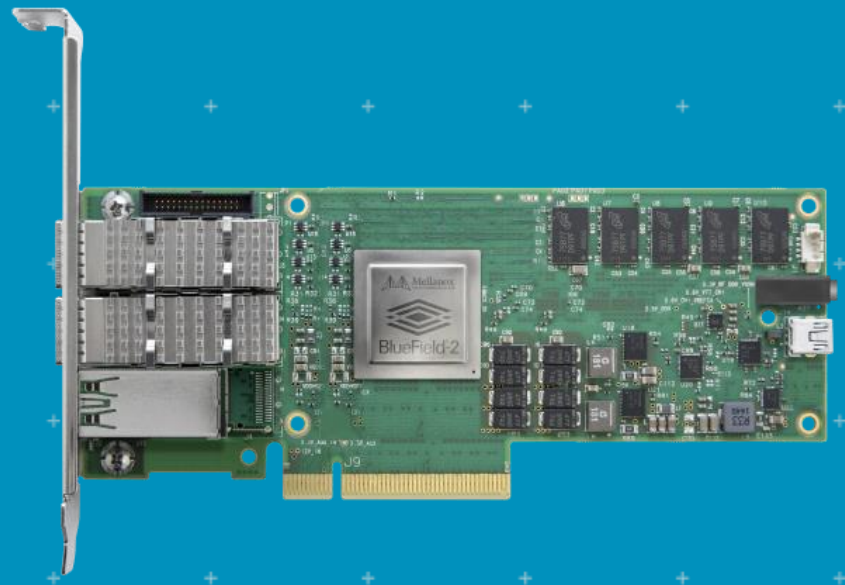
arm

Key Contributions

- ARMIE study
  - Reduced the instruction counts from 16% to 10X respectively depending on the vector instruction length.
  - From VL = 128 to VL = 2048 bits we decreased the instruction count from 50% to 30 ×.
- Fujitsu's A64FX processor
  - Scatter/Gather pack and unpack 2 × faster using new SVE implementation
  - MPI Local reduce with VL = 512 SVE based reduction operation is 4 × faster
- Our analysis and implementation of OPEN MPI optimization provides useful insights and guidelines on how Arm SVE vector ISA can be used in actual high-performance computing platforms and software to improve the efficiency of network stack

arm

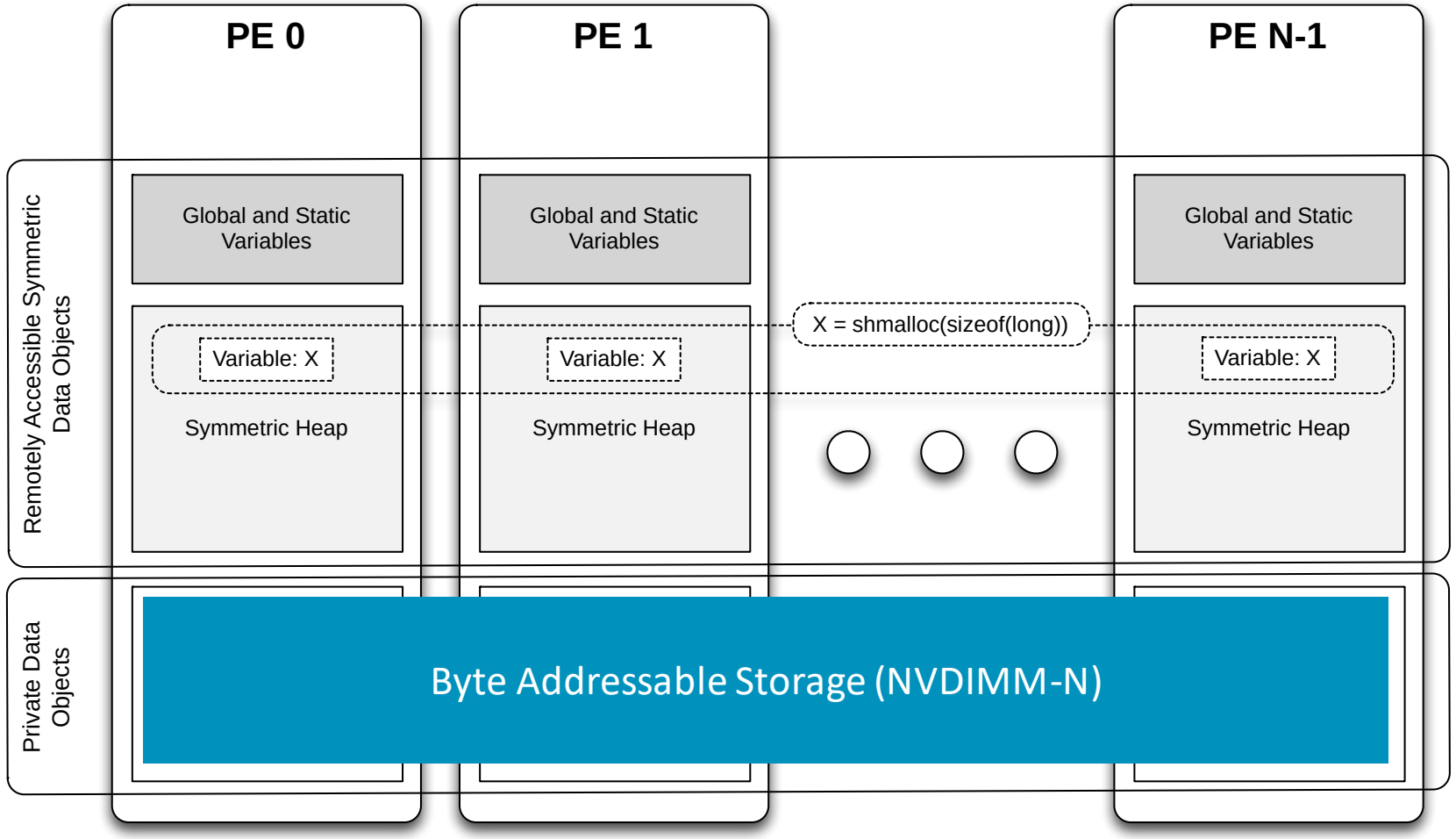
OpenSHMEM I/O Extensions for Fine-grained Access to Persistent Memory Storage



Megan Grodowitz, Pavel Shamis, and Steve Poole

SMC2020

PGAS/OpenSHMEM as a Storage Programming Model

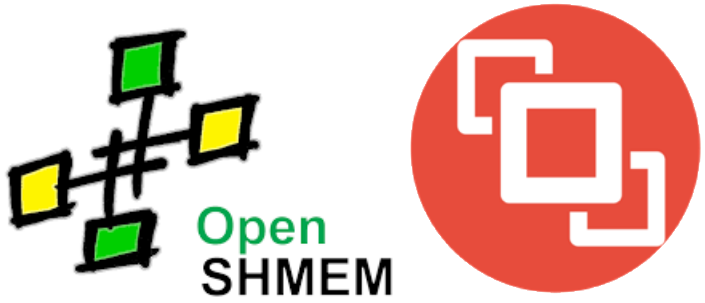


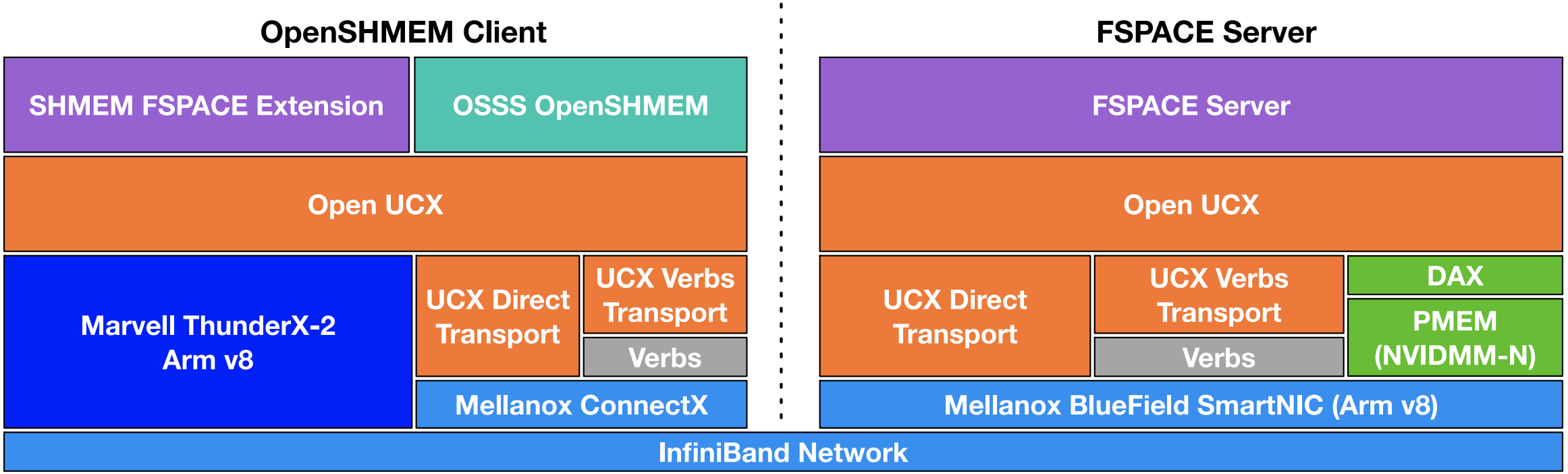
arm

Software & Hardware Stack Overview

Multiple innovations in OpenSHMEM/PGAS API

- Client-Server API
- Dynamic remotely attachable ”spaces” API
- Multi-level persistency API





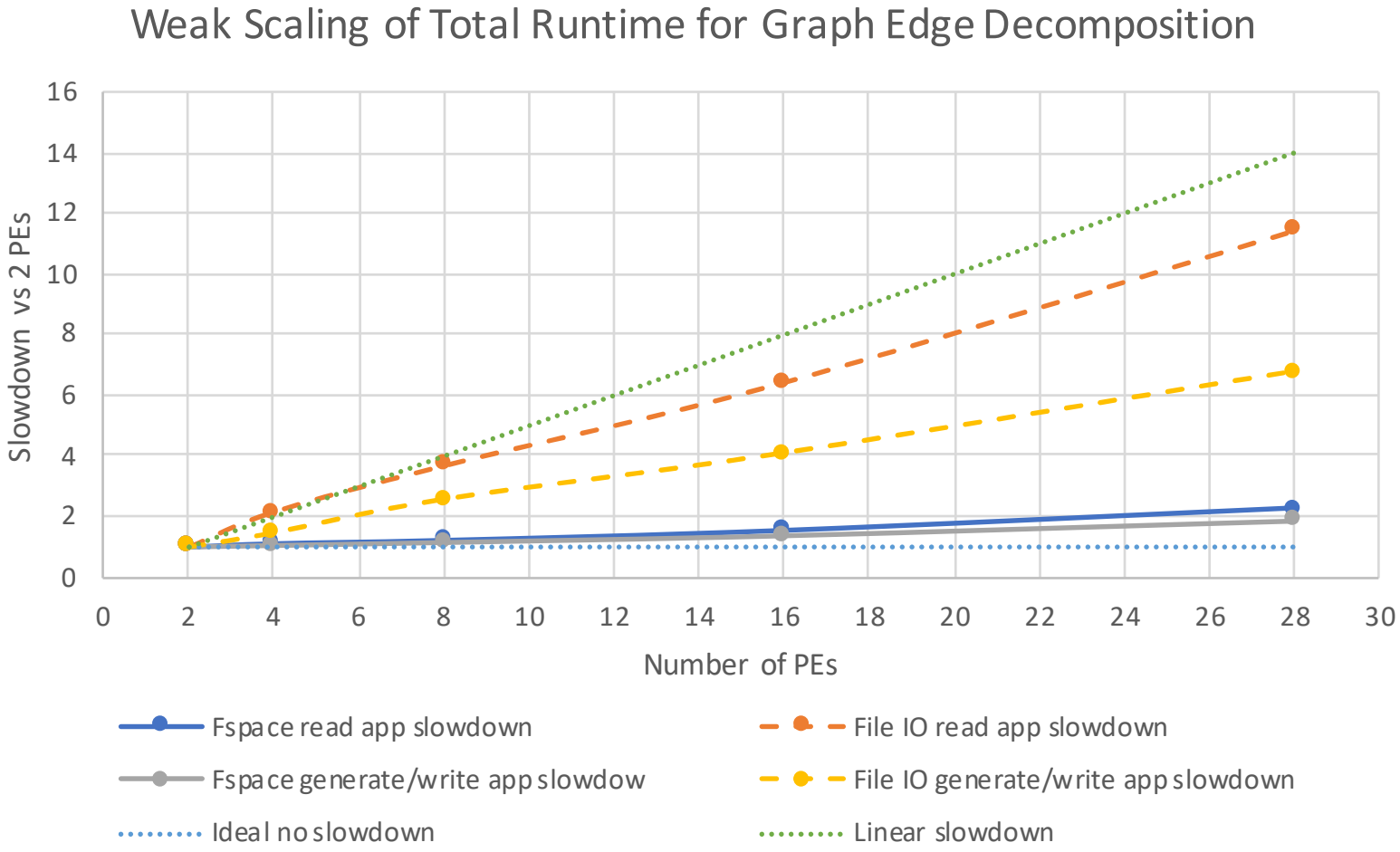
arm

Graph-update Workflow Benchmark

- Platform
  - Mellanox Bluefield SmartNIC Storage server
  - ThunderX2 28 core Client
- Observations
  - Nearly linear scaling
  - I/O dominates the workload

POSIX File I/O on NFS degrades linearly as number of processes increase. All access in parallel to non-overlapping file regions. Read app (App1) also writes back after sort so performance degrades worse for App1 as expected.

Fspace File I/O over same network fabric shows only small performance degradation as number of processes increases.



Key Contributions

- OpenSNAPI research software
- Multiple Innovations in OpenSHMEM/PGAS API
  - OSSS SHMEM functional implementation
- OpenSNAPI storage server
  - Implemented on Arm based Mellanox Bluefield
  - Implemented and evaluated with NVIDIMM-N (not emulation !)
- Open source: https://github.com/openucx/shmem-opensnapi
- More details announced at SMC-2020

Acknowledgments
United States Department of Defense and Los Alamos National Laboratory for their continued support of this project.

Gilad Shainer and Wang Wong from Mellanox Technologies for providing us BlueField development platform and enabling NVDIMM support in BIOS.

arm

Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
ধন্যবাদ
תודה