



arm

# The Arm Architecture for Exascale HPC

EXALAT - Lattice Field Theory at the Exascale

Dr. Olly Perks - Principal HPC Engineer

[Olly.Perks@arm.com](mailto:Olly.Perks@arm.com)

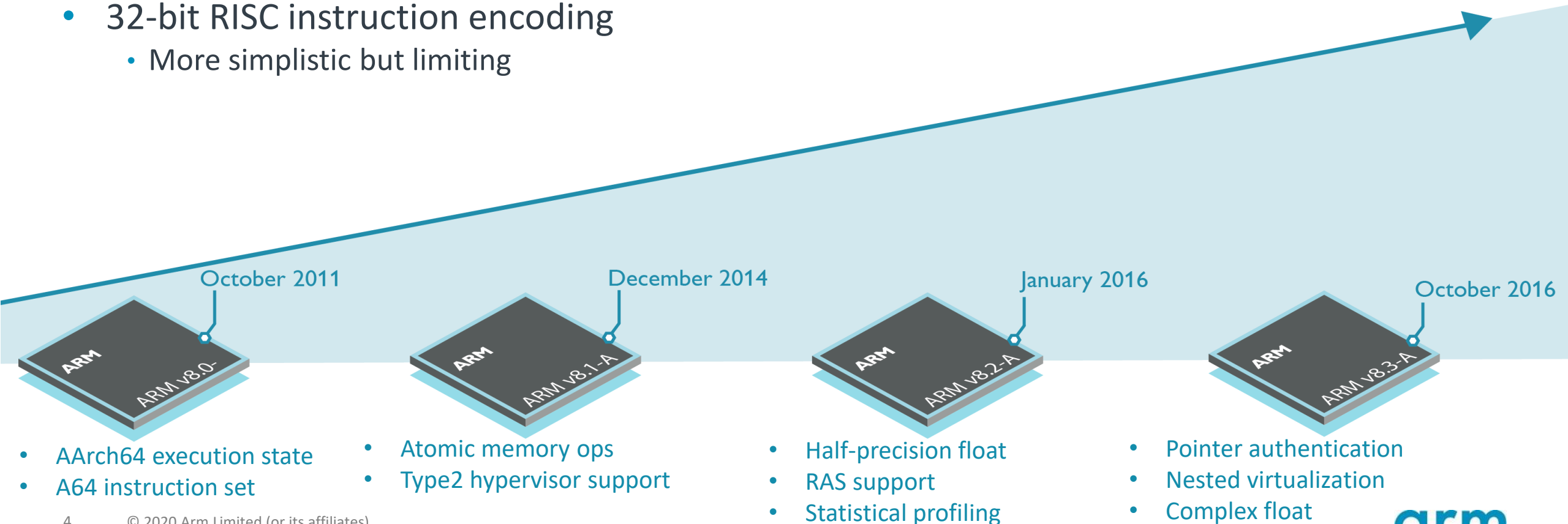
17<sup>th</sup> June 2020



# Arm and our role in HPC

# What is Arm?

- Arm designs IP (such as the Arm ISA)
  - We do not manufacture hardware
- 32-bit RISC instruction encoding
  - More simplistic but limiting



# Why Arm?

Especially for Infrastructure / HPC / Scientific Computing / ML?



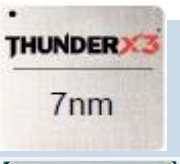













## Hardware

- Flexibility: Allow vendors to differentiate
  - Speed and cost of development
- Provide different licensing
  - Core - Reference design (A53/A72/N1)
  - Architecture - Design your own (TX2, A64FX)
- Other hardware components
  - NoCs, GPUs, memory controllers
  - “Building blocks” design
- Architecture validation for correctness

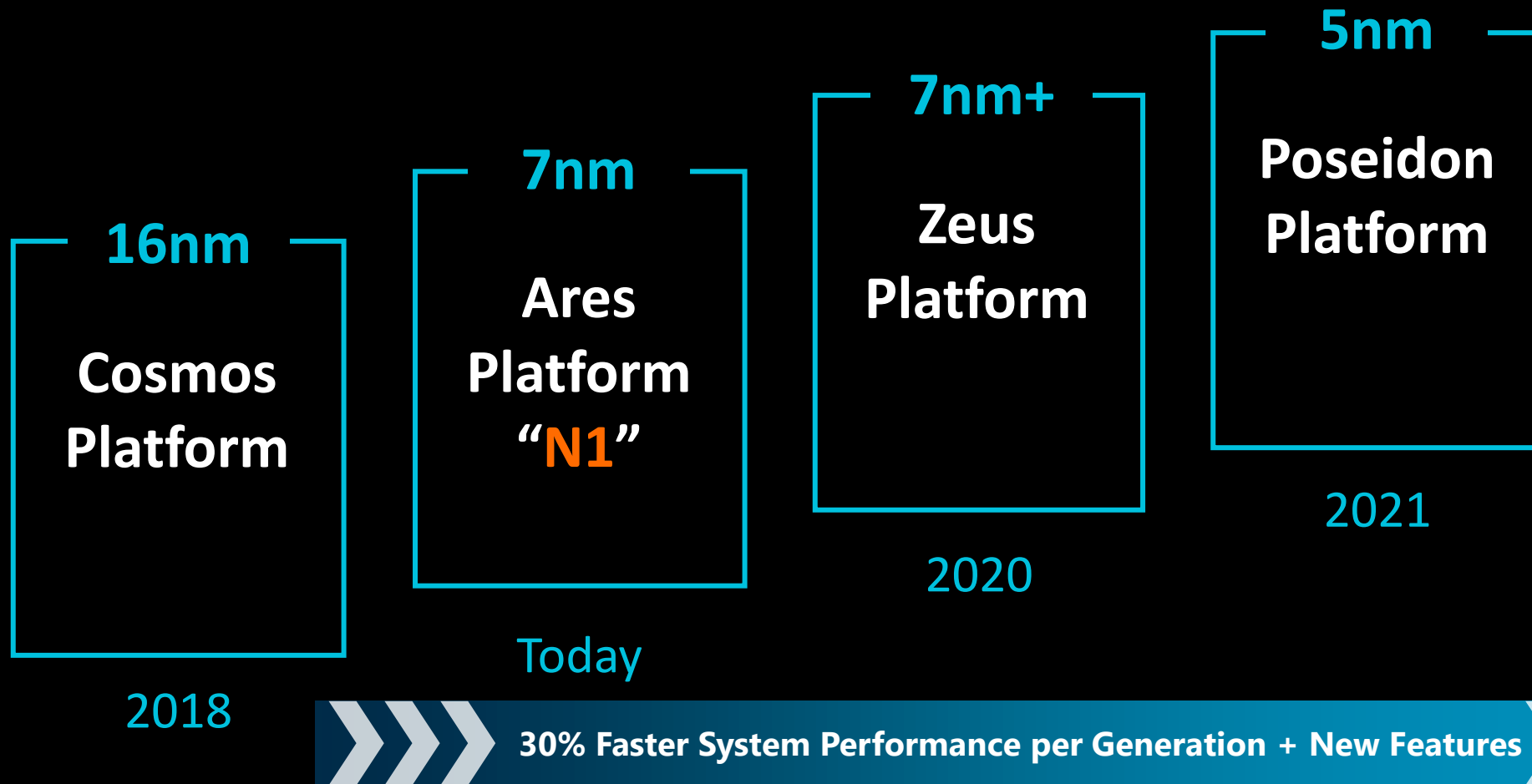
## Software

- All based on the same instruction set
  - Commonality between hardware
  - Reuse of software
- Comprehensive software ecosystem
  - Operating systems, compilers, libraries, tools
  - Not just vendor - third party too
- Large community
  - Everything from Android to HPC

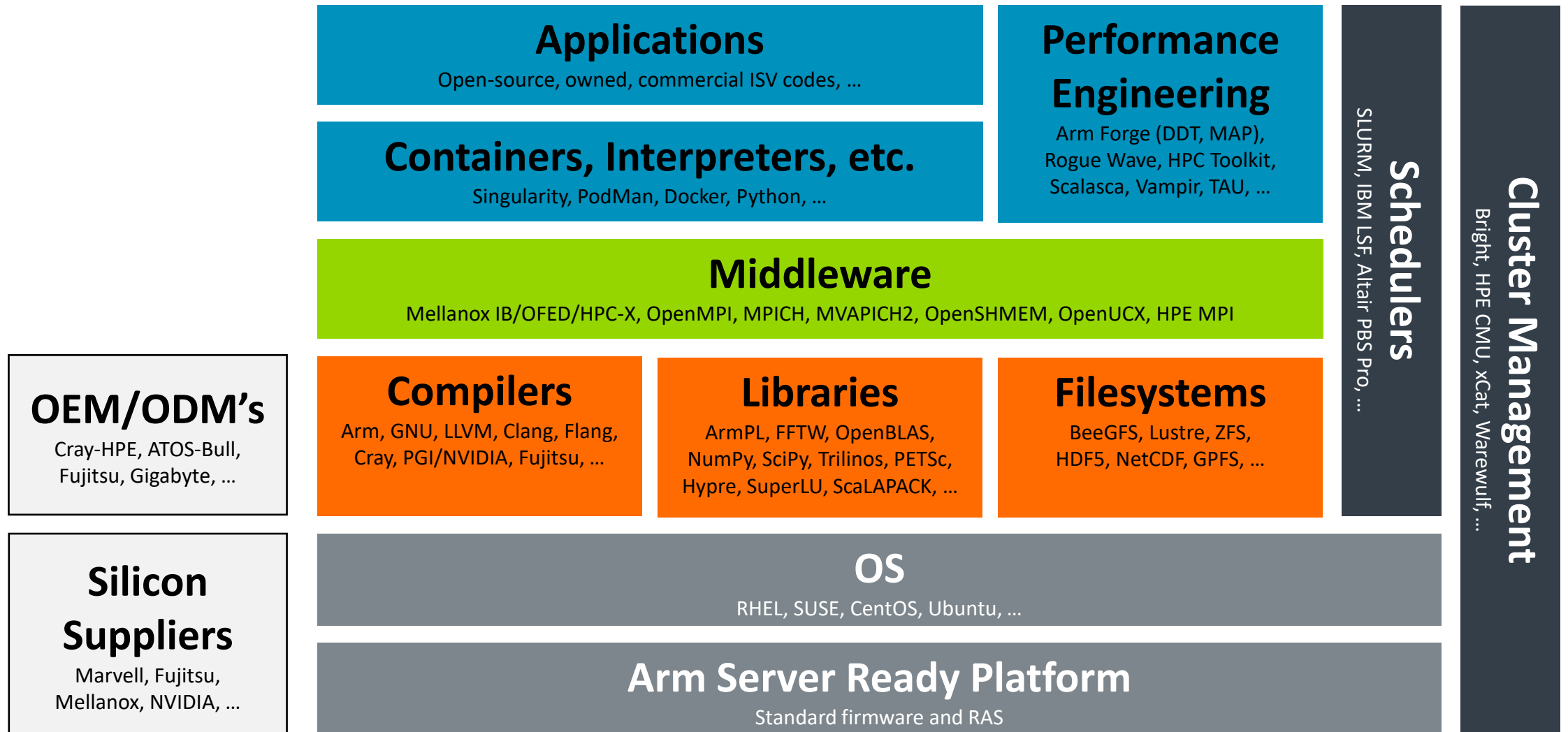
# Variation in the Processor Market

Marvell (Cavium)						
Ampere (X-Gene)						
Fujitsu						
Huawei (HiSilicon)						
Amazon (Annapurna)						
EPI / SiPearl						
Other						

# Each generation brings faster performance and new infrastructure specific features

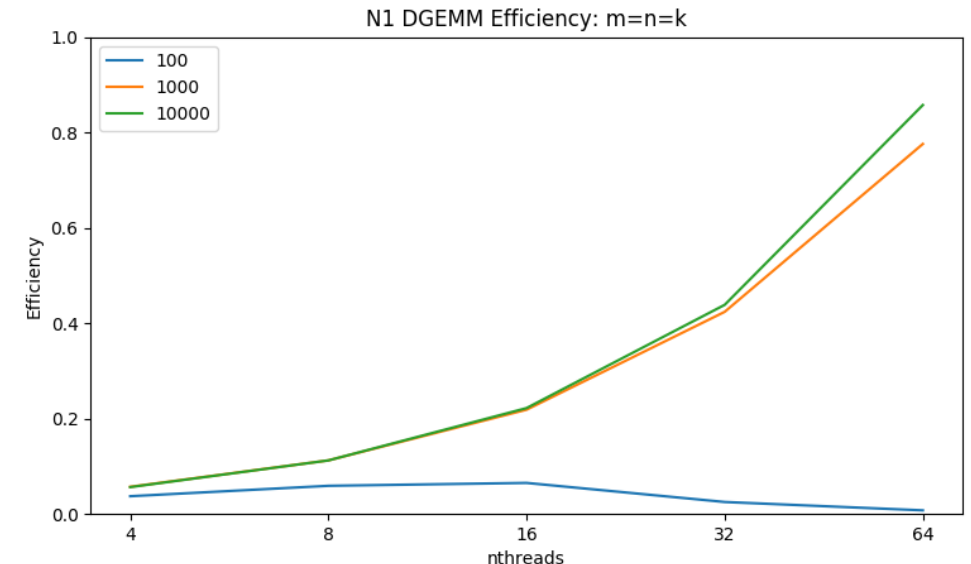


# Not Just Hardware



# Accelerated Maths Libraries

- Arm produce a set of accelerated maths routines
  - Microarchitecture tuned for each Arm core
  - BLAS, LAPACK, FFT (Standard interface)
  - Tuned math calls
    - Transcendentals (libm) + string functions
  - Sparse operations
    - SpMV / SpMM
  - Available for GCC and Arm compiler
- Open source maths libraries available
  - OpenBLAS, BLIS, SLEEFP
- Other vendor maths libraries also available
  - Cray (libsci), Fujitsu (SSL2)

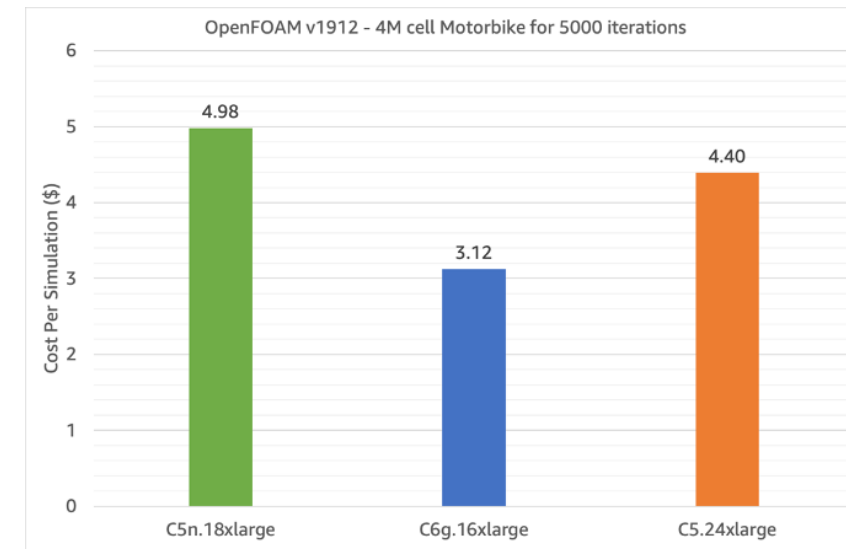
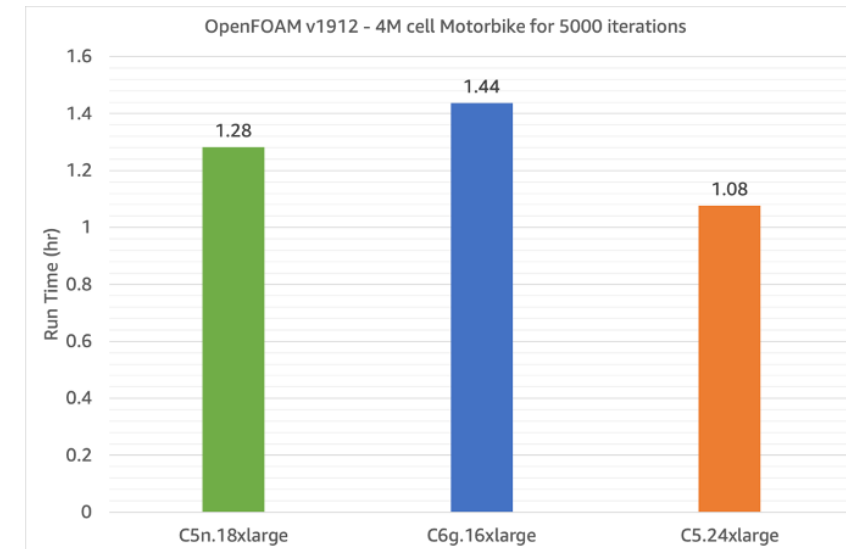


DGEMM Performance on the  
Neoverse N1 for different matrix sizes  
Max efficiency 85.7%



# Arm HPC in the Cloud - AWS Graviton2

- Perfect example of Arm model
- AWS designed and built their own processor
  - Based on an Arm - N1 core license
  - With additional custom IP
  - Optimise for cloud environment (e.g. power, cost)
- Specs:
  - 64-core socket, @2.5 GHz (single socket nodes)
  - 8x DDR4-3200 memory channels
    - 128 GB (C6g), 256 GB (M6g), 512 GB (R6g)
- Case study: OpenFOAM on C6g
  - <https://aws.amazon.com/blogs/compute/c6g-openfoam-better-price-performance/>
  - Vs Skylake: 12% slower, but 37% lower \$/solution
  - OpenFOAM v1912, GCC 9.2, Open MPI 4.0.3, UCX 1.8



arm

Arm and Exascale

# Arm Based Processors for Exascale systems

- Arm technology can be a great fit for exascale system design
  - Customisation and configuration for energy and performance efficiency
- Exascale really isn't just about the processors (FLOPs are easy, performance is hard)
- Key technology 'necessary' for Exascale
  - Processors + vector units (e.g. AVX-512, SVE)
  - Memory subsystems (e.g. High Bandwidth Memory)
  - High performance networks (e.g. InfiniBand, TOFU, Slingshot)
  - Accelerators, Filesystems, Middleware, Compilers, .....
- Two key Arm-based case studies
  - Fugaku / A64FX
  - EPI



# Quick Introduction to SVE

# SVE: Scalable Vector Extension

- **SVE is Vector Length Agnostic (VLA)**
  - Vector Length (VL) is a hardware implementation choice from 128 up to 2048 bits.
  - New programming model allows software to scale dynamically to available vector length.
  - No need to define a new ISA, rewrite or recompile for new vector lengths.
- **SVE is not an extension of Advanced SIMD (*aka* Neon)**
  - A separate, optional extension with a new set of instruction encodings.
  - Initial focus is HPC and general-purpose server, not media/image processing.
- **SVE begins to tackle traditional barriers to auto-vectorization**
  - Software-managed speculative vectorization allows uncounted loops to be vectorized.
  - In-vector serialised inner loop permits outer loop vectorization in spite of dependencies.

# How can you program when the vector length is unknown?

SVE provides features to enable VLA programming from the assembly level and up

	1	2	3	4
+	5	5	5	5
<i>pred</i>	1	0	1	0
=	6	2	8	4

## Per-lane predication

Operations work on individual lanes under control of a predicate register.

<b>for (i = 0; i &lt; n; ++i)</b>				
<i>INDEX i</i>	n-2	n-1	n	n+1
<i>CMPLT n</i>	1	1	0	0

## Predicate-driven loop control and management

Eliminate scalar loop heads and tails by processing partial vectors.

	1	2		
+	1	2	0	0
<i>pred</i>	1	1	0	0

## Vector partitioning & software-managed speculation

First Faulting Load instructions allow memory accesses to cross into invalid pages.

# Vectorizing A Scalar Loop With ACLE

`a[:] = 2.0 * a[:]`

```
for (int i=0; i < N; ++i) {  
    a[i] = 2.0 * a[i];  
}
```

## 128-bit NEON vectorization

```
int i;  
  
// vector loop  
for (i=0; (i<N-3) && (N&~3); i+=4) {  
    float32x4_t va = vld1q_f32(&a[i]);  
    va = vmulq_n_f32(va, 2.0);  
    vst1q_f32(&a[i], va)  
}  
  
// drain loop  
for (; i < N; ++i)  
    a[i] = 2.0 * a[i];
```

## SVE vectorization

```
for (int i = 0 ; i < N; i += svcntw() )  
{  
    svbool_t Pg = svwhilelt_b32(i, N);  
    svfloat32_t va = svld1(Pg, &a[i]);  
    va = svmul_x(Pg, va, 2.0);  
    svst1(Pg, &a[i], va);  
}
```

# SVE Compiler Support



Compiler	Assembly / Disassembly	Inline Assembly	ACLE	Auto- vectorization	Math Libraries
Arm Compiler for HPC	SVE + SVE2	SVE + SVE2	SVE + SVE2	SVE+ SVE2	SVE
LLVM/Clang	SVE + SVE2	SVE + SVE2	SVE + SVE2 in LLVM 10	SVE + SVE2 in LLVM 11	
GNU	SVE + SVE2	SVE + SVE2	SVE + SVE2 in GNU 10	SVE now SVE2 in GNU10	

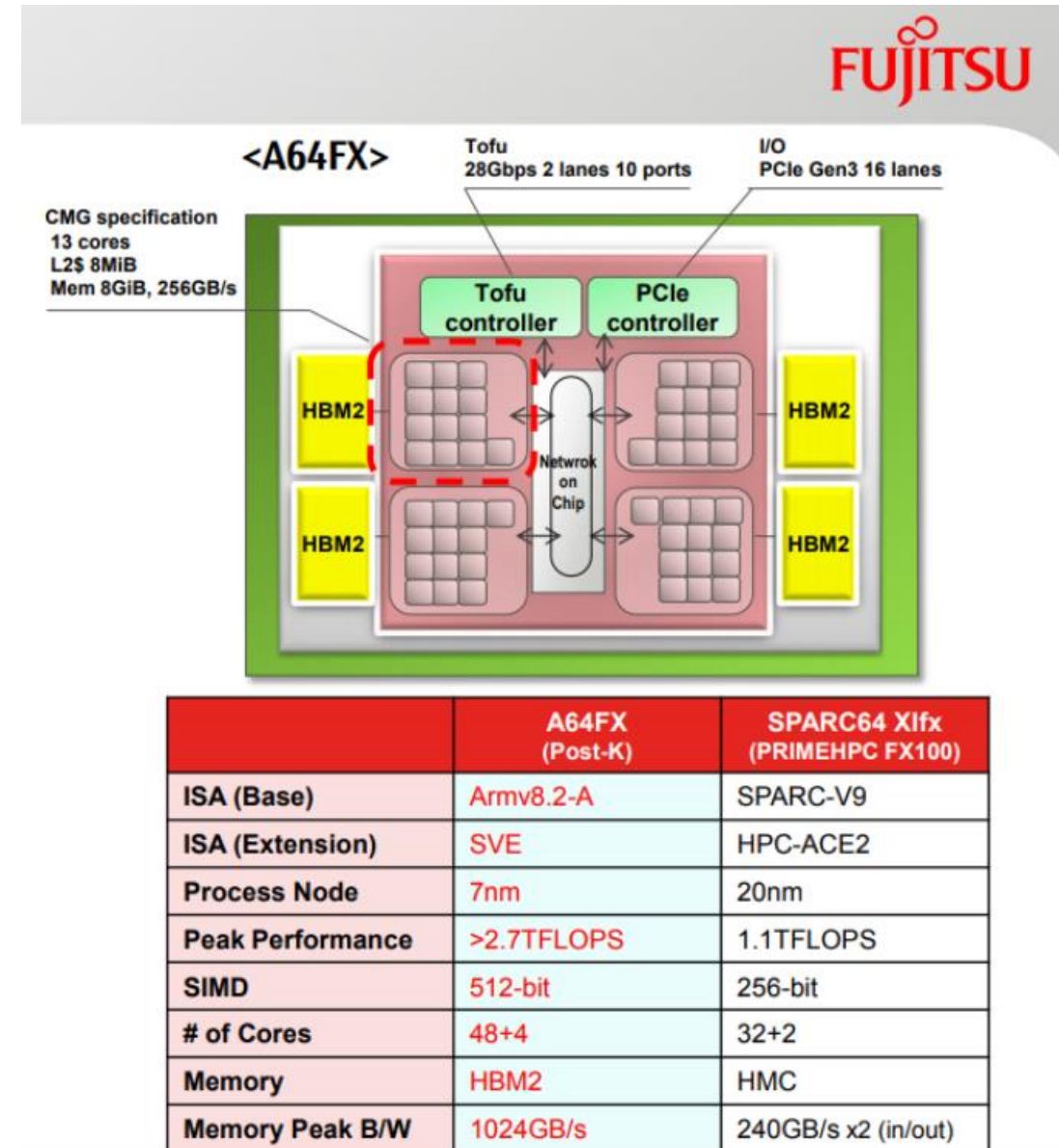




# Exascale Case Study: Fugaku Supercomputer

# Fujitsu A64FX

- Arm Architecture license
  - Built to replace the SPARC64 VIIIfx (in K-Computer)
  - Nearly 10 years of collaboration with Arm for SVE
    - RIKEN + Fujitsu + Arm
- Based around 4 CMGs (Core Memory Group)
  - Essentially a NUMA node
  - 12 cores (+1 Operating system core) (48+4 / socket)
    - 2x 512-bit SVE
    - 1.8-2.2 GHz
    - 2.7 - 3.3 TFLOPS / socket
  - 1 stack of 8 GB HBM2 (~1 TB/s bandwidth / socket)
  - TOFU or InfiniBand
- General purpose CPU with GPU like performance



All Rights Reserved. Copyright © FUJITSU LIMITED 2018

# Fugaku Supercomputer

- Biggest Arm based deployment
  - Due to be announced next week at Top500
  - Twitter pre-announcement at 0.537 Exaflops
  - 7.6M Arm cores (no accelerators)
- Energy Consumption
  - Designed to be low
  - ~150 W / node \*1
- Delivered early to assist with COVID-19
  - National and international projects
  - Open Science
  - AI/ML/DL

📌 Pinned Tweet



**Satoshi Matsuoka** @ProfMatsuoka · 15 May

We just announced the Fugaku config at a press briefing. The whole system is 158,976 A64FX nodes, and running at 2.2Ghz peak performances for 64bitFP/32bitFP/16bitFP/8bitINT are 537Peta/1.07Exa/2.15Exa/4.30Exa (FL)ops respectively, as well as the total memory BW 163 PetaByte/s.



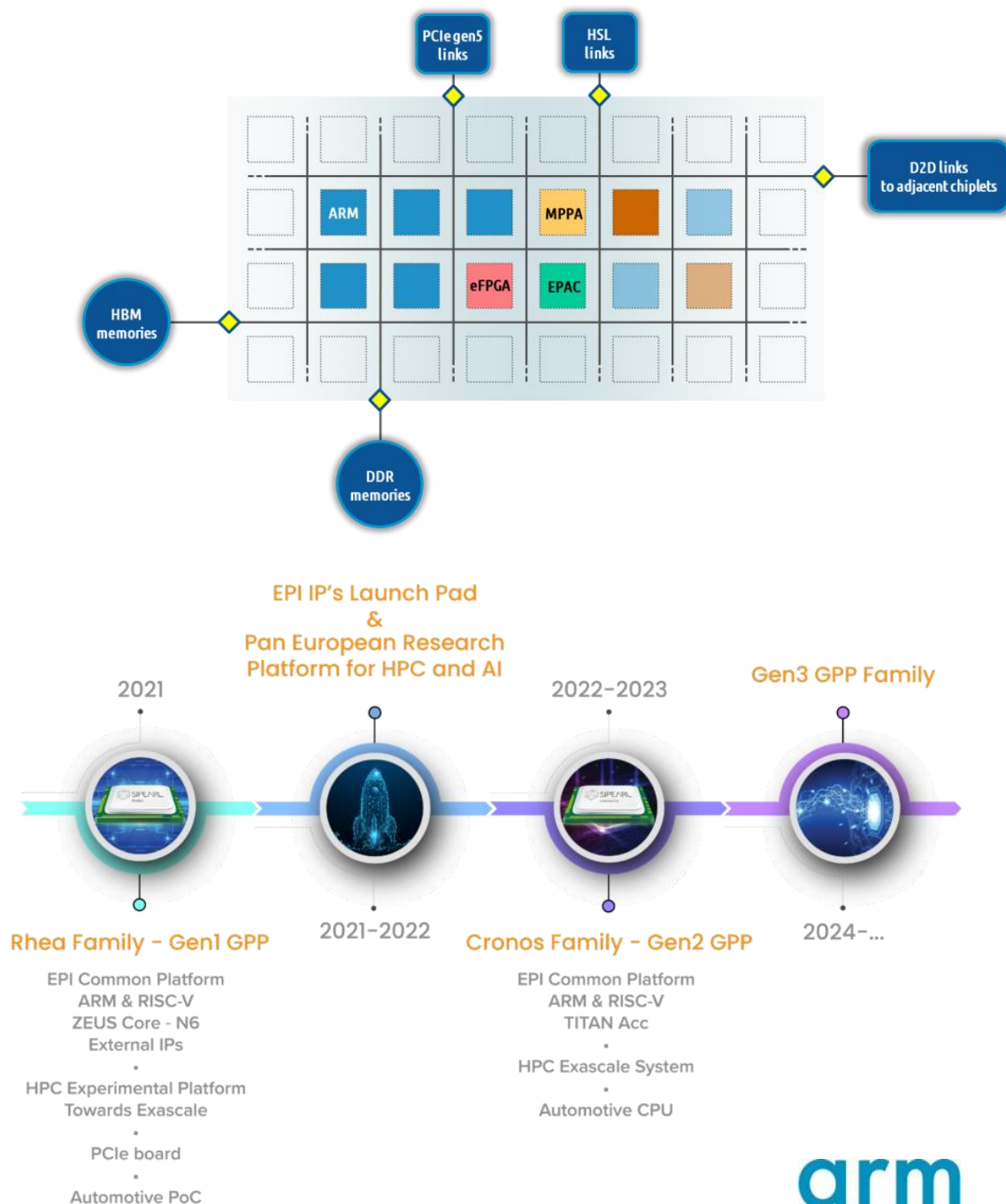
Rank	TOP500		System	Cores	Rmax (TFlop/s)	Power (kW)	Power Efficiency (GFlops/watts)
	Rank	Rank					
1	159		<b>A64FX prototype</b> - Fujitsu A64FX, Fujitsu A64FX 48C 2GHz, Tofu interconnect D, Fujitsu Fujitsu Numazu Plant Japan	36,864	1,999.5	118	16.876



# Exascale Case Study: EPI and SiPearl

# EPI: SiPearl Rhea (1<sup>st</sup> Gen)

- New initiative as part of EuroHPC
  - Drive for European technology for HPC
- Mixture of new technologies
  - Arm general purpose cores (Zeus - N2)
  - Accelerators: RISC-V, FPGA
  - Memory: DDR 4/5, HBM
  - Connectivity: PCIe G5, CCIX
- Targeted for key European markets
  - Automotive
  - HPC
  - AI / ML



arm

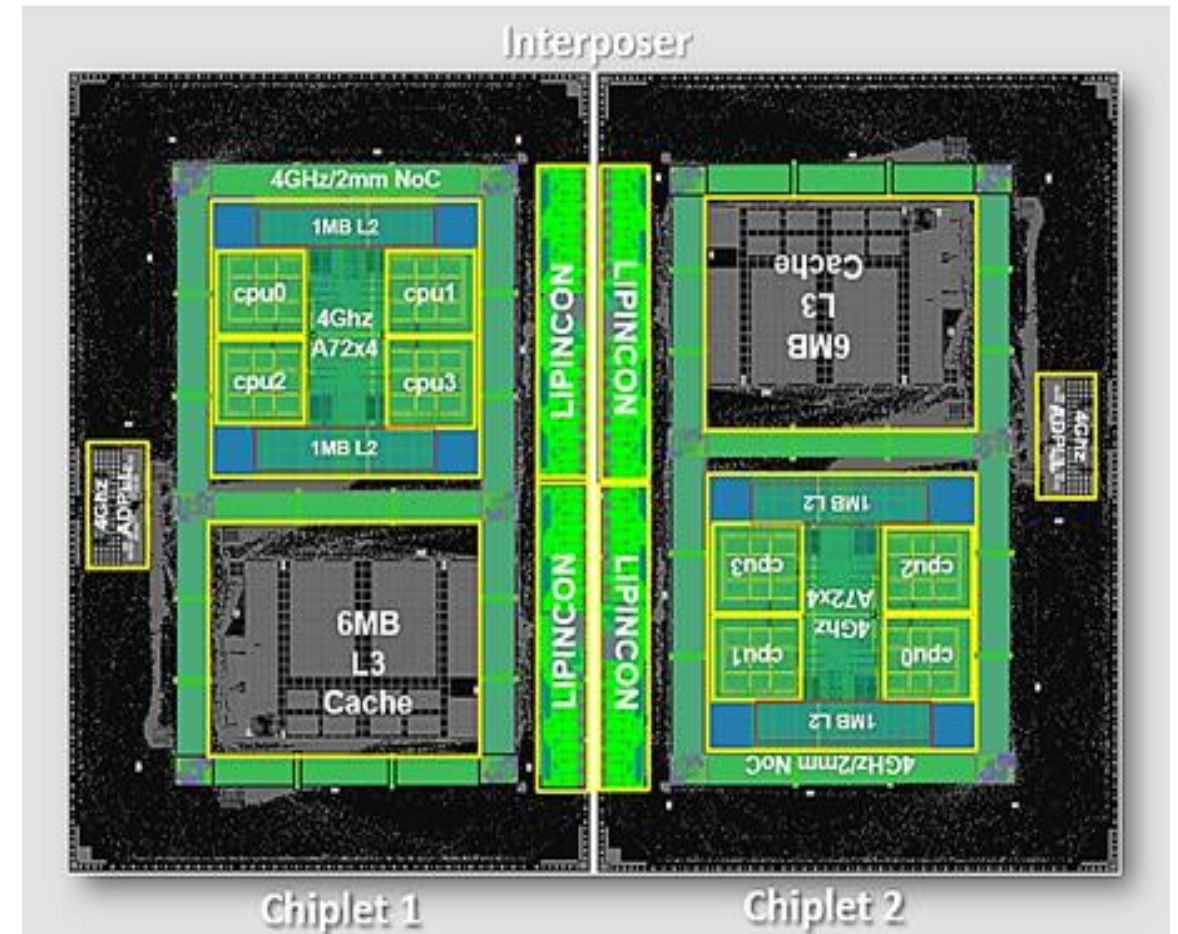
# The Next Steps



# Chiplet Demonstration

<https://www.arm.com/company/news/2019/09/arm-and-tsmc>

- Proof-of-concept produced in April 2019
- Dual-chiplet 7nm CoWoS
  - Chip-on-Wafer-on-Substrate
- Each chiplet contains four Arm Cortex<sup>®</sup>-A72 processors
  - on-die interconnect mesh bus.



# FMMLA: High Performance Matrix Multiplication

- Added to Armv8.6
  - NEON and SVE instructions
  - FMMLA instructions for FP (SVE)

FMMLA <Zda>.S, <Zn>.S, <Zm>.S

FMMLA <Zda>.D, <Zn>.D, <Zm>.D

- 2x2 matrix multiplication
  - Works on multiple of 'vector granules'
  - 2x2xFP32 = 128-bit granules
  - Assumes vector length is multiple
- May require layout transformations
  - Outer loop to minimise cost
- Accelerated libraries

0	1
2	3

 += 

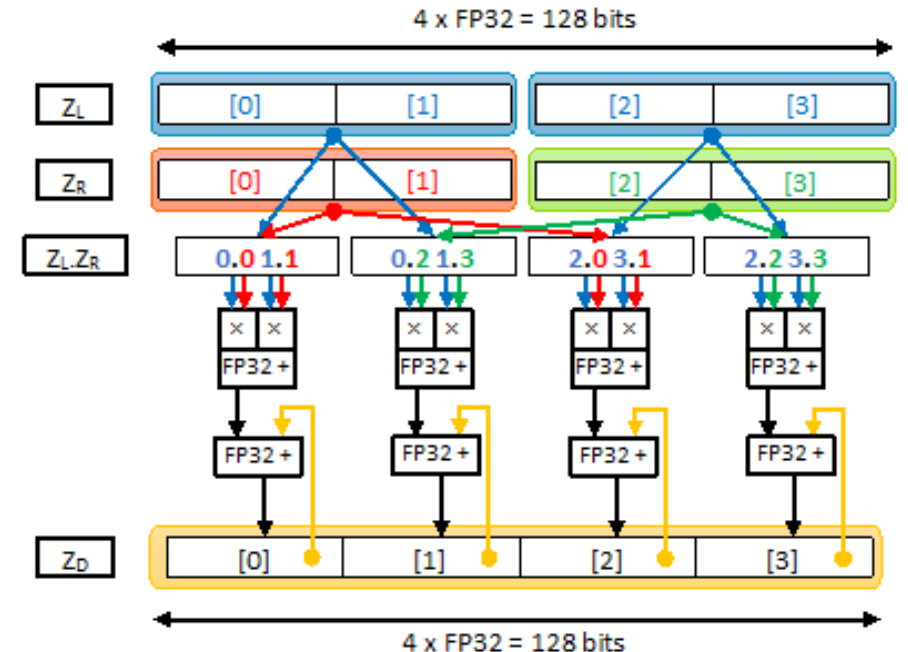
0	1
2	3

 X 

0	1
2	3

Dest (D)                  Left (L)                  Right (R)  
2x2xFP32                  2x2xFP32                  2x2xFP32

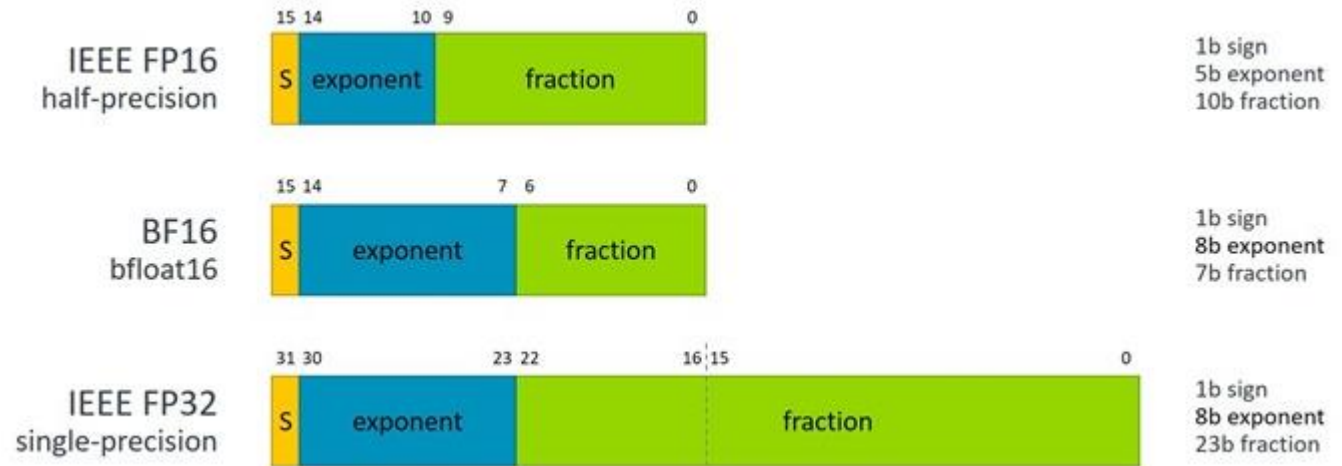
$$\begin{aligned} D[0] &+= (L[0] * R[0]) + (L[1] * R[1]) \\ D[1] &+= (L[0] * R[2]) + (L[1] * R[3]) \\ D[2] &+= (L[2] * R[0]) + (L[3] * R[1]) \\ D[3] &+= (L[2] * R[2]) + (L[3] * R[3]) \end{aligned}$$





# New Data Type Support: BFloat16

- New addition to Armv8-A
  - Adds support for BF16
- Instructions for NEON and SVE
  - Including:
    - **BFDOT**: Dot Product (1x2)x(2x1)
    - **BFMMLA**: Mat Multiply (2x4)x(4x2)
- Significant performance gains
  - ML training and inference workloads
- Supported in Arm libraries
  - Arm NN and Arm Compute Libraries



# Conclusion

- Exascale is about far more than just the processor technology
- But Arm provides a great foundation on which to design Exascale systems
- Robust hardware and software ecosystem
  - Coupled with world class performance
- Lots more exciting features to come



The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

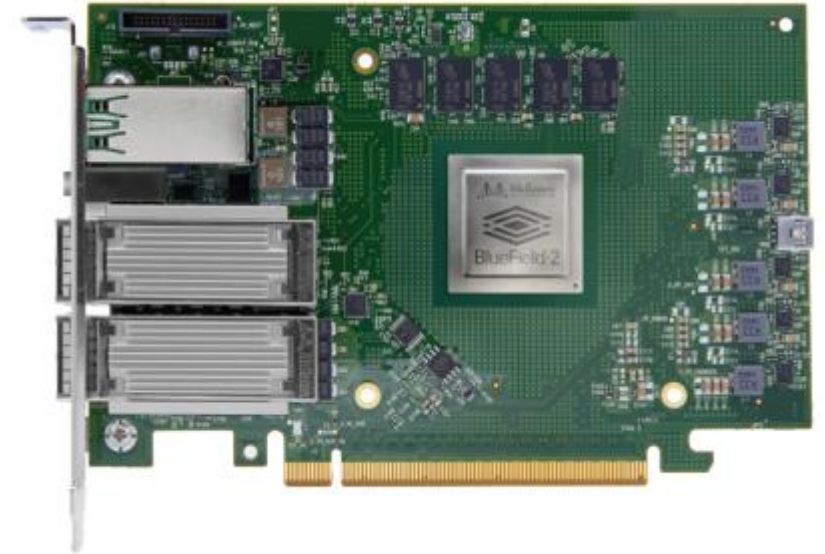
[www.arm.com/company/policies/trademarks](http://www.arm.com/company/policies/trademarks)

arm

Backup: More exciting  
new things

# NVIDIA Mellanox BlueField-2 SmartNIC

- Smart NICs are going to play a significant role in new systems
- BlueField-2 Integrates an IB NIC with Arm cores
  - 200 Gb/s InfiniBand
  - 8x Arm A72 cores
  - 8/16 GB DDR4
- NIC operates as an offload node
  - Runs Ubuntu
  - Host MPI ranks, map network storage, burst buffer, ...



# Scalable Vector Extensions V2 (SVE2)

SVE for non HPC markets



Built on SVE



Improved scalability



Vectorization of  
more workloads

- Built on the SVE foundation.
  - Scalable vectors with hardware choice from 128 to 2048 bits.
  - Vector-length agnostic programming for “write once, run anywhere”.
  - Tackles some obstacles to compiler auto-vectorisation.
- Scaling single-thread performance to exploit long vectors.
  - SVE2 adds NEON™-style fixed-point DSP/multimedia plus other new features.
  - Performance parity and beyond with classic NEON DSP/media SIMD.
  - Tackles further obstacles to compiler auto-vectorization.
- Enables vectorization of a wider range of applications than SVE.
  - Multiple use cases in Client, Edge, Server and HPC.
    - DSP, Codecs/filters, Computer vision, Photography, Game physics, AR/VR,
    - Networking, Baseband, Database, Cryptography, Genomics, Web serving.
  - Improves competitiveness of Arm-based CPU vs proprietary solutions.
  - Reduces s/w development time and effort.

# SVE2 Instructions Add:

## What's new

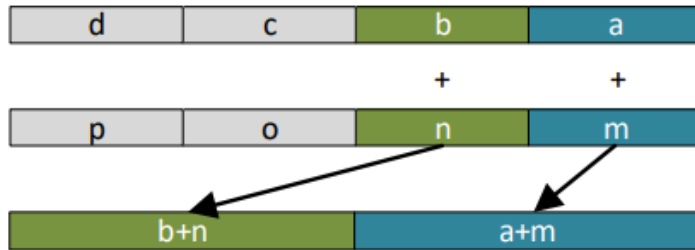
- Thorough support for fixed-point DSP arithmetic
  - (traditional Neon DSP/Media processing, complex numbers arithmetic for LTE)
- Multi-precision arithmetic
  - (bignum, crypto)
- Non-temporal gather/scatter
  - (HPC, sort)
- Enhanced permute and bitwise permute instructions
  - (CV, FIR, FFT, LTE, ML, genomics, cryptanalysis)
- Histogram acceleration support
  - (CV, HPC, sort)
- String processing acceleration support
  - (parsers)
- (optional) Cryptography support instructions for AES, SM4, SHA standards
  - (encryption)

# Example: Widening and Narrowing

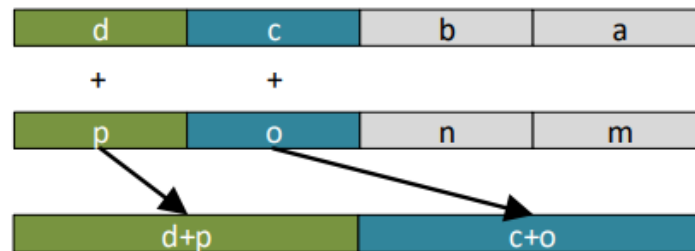
## NEON vs SVE2

### NEON

UADDL Vd.2D, Vn.2S, Vm.2S



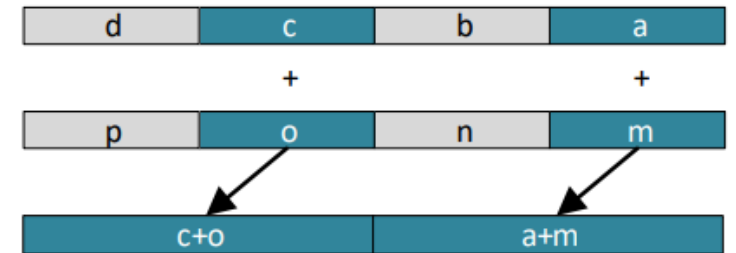
UADDL2 Vd.2D, Vn.4S, Vm.4S



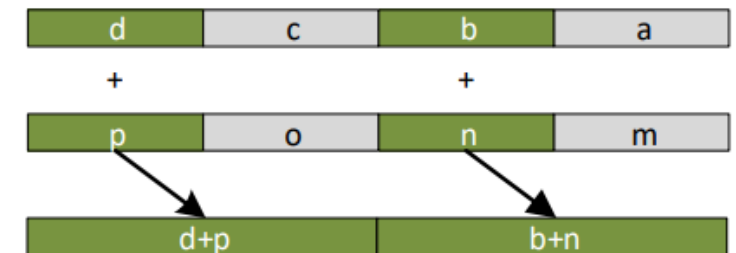
- NEON uses high/low half of vector
- Expensive for large vector lengths
  - >> 128-bit
- SVE2 uses odd/even half of vector
- Bottom and top
- Happens 'in-lane'

### SVE2

UADDLB Zd.D, Zn.S, Zm.S



UADDLT Zd.D, Zn.S, Zm.S





# Transactional Memory Extension (TME)

Scalable Thread-Level Parallelism (TLP) for multi-threaded applications



Hardware Transactional  
Memory



Improved scalability



Simpler software design

- Hardware Transactional Memory (HTM) for the Arm architecture.
  - Improved competitiveness with other architectures that support HTM.
  - Strong isolation between threads.
  - Failure atomicity.
- Scaling multi-thread performance to exploit many-core designs.
  - Database.
  - Network dataplane.
  - Dynamic web serving.
- Simplifies software design for massively multi-threaded code.
  - Supports Transactional Lock Elision (TLE) for existing locking code.
  - Low-level concurrent access to shared data is easier to write and debug.