# An Overview of ML and AI on Arm Based HPC Systems for Weather and Climate Applications

Joint IS-ENES3/ESiWACE2 Virtual Workshop on New Opportunities for ML and AI in Weather and Climate Modelling

Phil Ridley

phil.ridley@arm.com

18th March 2021

# Agenda

- Containers
- ML and AI
  - Processor developments
  - Community support
  - Libraries and Applications
- ISA Developments

# Containers

# Arm enables containerization through standardization

## Ensuring standard interfaces work on Arm enables multiple technologies

### Approach

Standards bodies define the interfaces between layers

Example: Open Containers Initiative (OCI)

Arm works with the standards bodies to see that Arm is well supported

Example: Arm supporting containers drives Aarch64 support in Docker

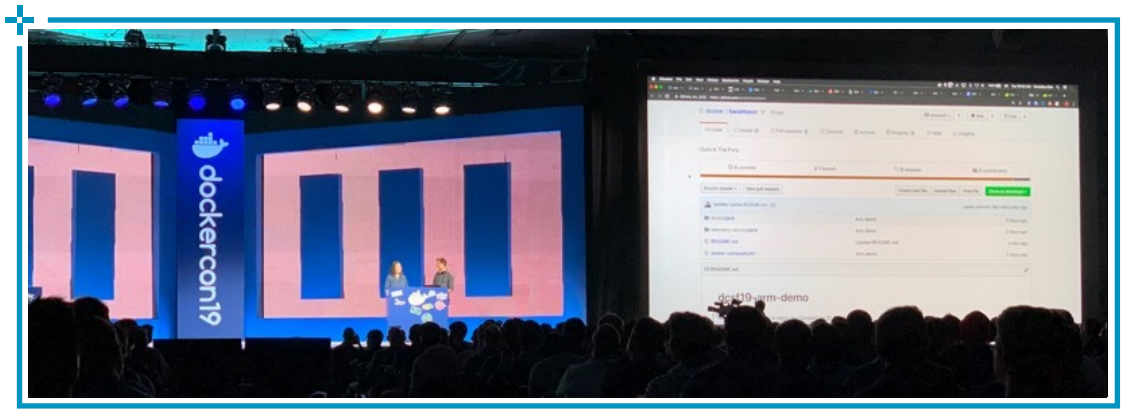
All the "level-set" work is done.

Arm is now focusing on enabling leading technologies




arm

# Arm & Docker Partner to Deliver Frictionless Cloud-Native Software Development

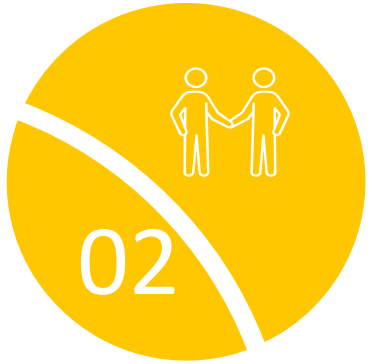

01 Initial phase is focused on Integration of Arm capabilities into Docker Desktop Community to enable a seamless developer environment

02 Docker Enterprise Engine for Amazon EC2 A1 instances

03 Additional work will address end-to-end management of full product life cycle; unified development environments for heterogeneous compute and scaling cloud-native benefits to consolidate edge workloads

# Docker on Arm

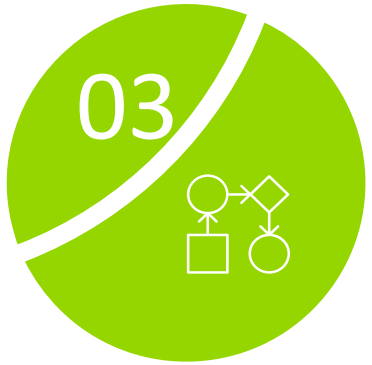
Docker Desktop is the de facto standard Cloud Native development platform for containerized applications


This partnership makes it easier for millions of developers already using Docker to develop containers on Arm

arm docker

**5,386,145** base images on Docker Hub
**51,460** Arm images
**46,167** Arm64 images
Official **166** Docker images
Arm support **118** out of **166** official images


No changes needed to Docker tooling & processes in order to start building for Arm




docker × arm

# Machine Learning

# Machine Learning

TensorFlow

- Arm actively involved

Deepbench

- Arm actively involved


Torch

- Community maintained


Mahout

- Available via Apache Bigtop

Weka

- Community maintained

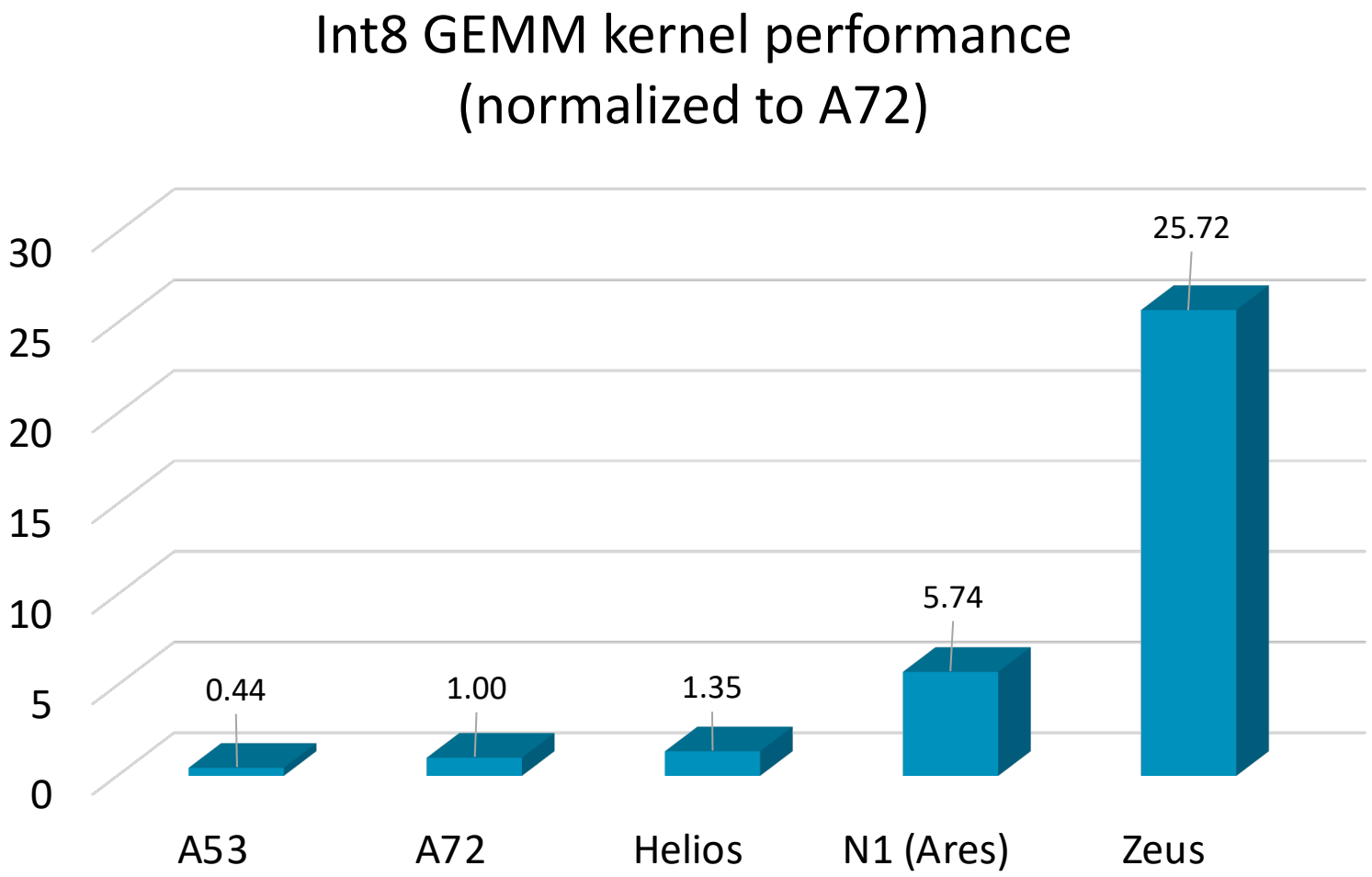

Caffe

- Community maintained


Theano (EOL)

- Community maintained

# Increasing ML performance over CPU generations




**A72**
2x ML performance improvement over Cortex-A53

**Helios**
>3x ML performance improvement over Cortex-A53 (First Multi-threaded CPU)

**N1**
>5x ML performance improvement over Cortex-A72 (PPA leadership & ML enhancements)

**Zeus**
>25x ML performance improvement over Cortext-A72 (Breakthrough ML performance)



arm

# Arm Neoverse Platform Roadmap

In planning

| IP Date | Production | Now | 2021 | 2022+ |
|---|---|---|---|---|
| **Platform features** | 7nm<br>PCIe Gen4, DDR4, HBM2<br>CCIX 1.0 | 7/5nm<br>PCIe Gen5, DDR5, HBM2e<br>CCIX 1.1 | 5nm<br>PCIe Gen5, DDR5, HBM3<br>CCIX 2.0, CXL 2.0 | 5/3nm<br>PCIe Gen5/6, DDR5, HBM3<br>CCIX next, CXL next |
| **V-series**<br>Max, ST Perf, ML | | **V1 Platform (Zeus)**<br>N1+50% ST perf. uplift<br>SVE 2x256b, bFloat16 | | |
| **N-series**<br>Perf/W, Scale out, 5G | **N1 Platform**<br>A72+60% ST perf. uplift | | **N2 Platform (Perseus)**<br>N1+40% ST perf. uplift<br>SVE 2x128b, bFloat16 | |
| **E-series**<br>Data efficiency | **E1 Platform**<br>A53+2.7x throughput | | | |

**Poseidon Generation Platforms**

+30% infra WL perf.
ML/Vector uplift
Greater core density

# On-CPU ML processing

## Primary drivers for on-CPU ML

ML is evolving – design optimization space for accelerators not sufficiently converged

Flexibility

ML programming looks like CPU programming as much as possible

Ease of programming

Sub-optimal to integrate specialized accelerators for intermittent ML processing

ML processing requirements

## Features enhancing ML performance on Arm CPUs

Arch
- Dot product instructions (v8.0 – v8.4)

Arch
- Matrix-multiply-and-accumulate instructions (*MMLA*) (v8.6)

Micro Arch
- SVE vector length

Arch
- Bfloat16 support (v8.6)

# On-CPU Machine Learning

## Easy to use, high performing ML software stack on Aarch64 using ML-specific CPU features

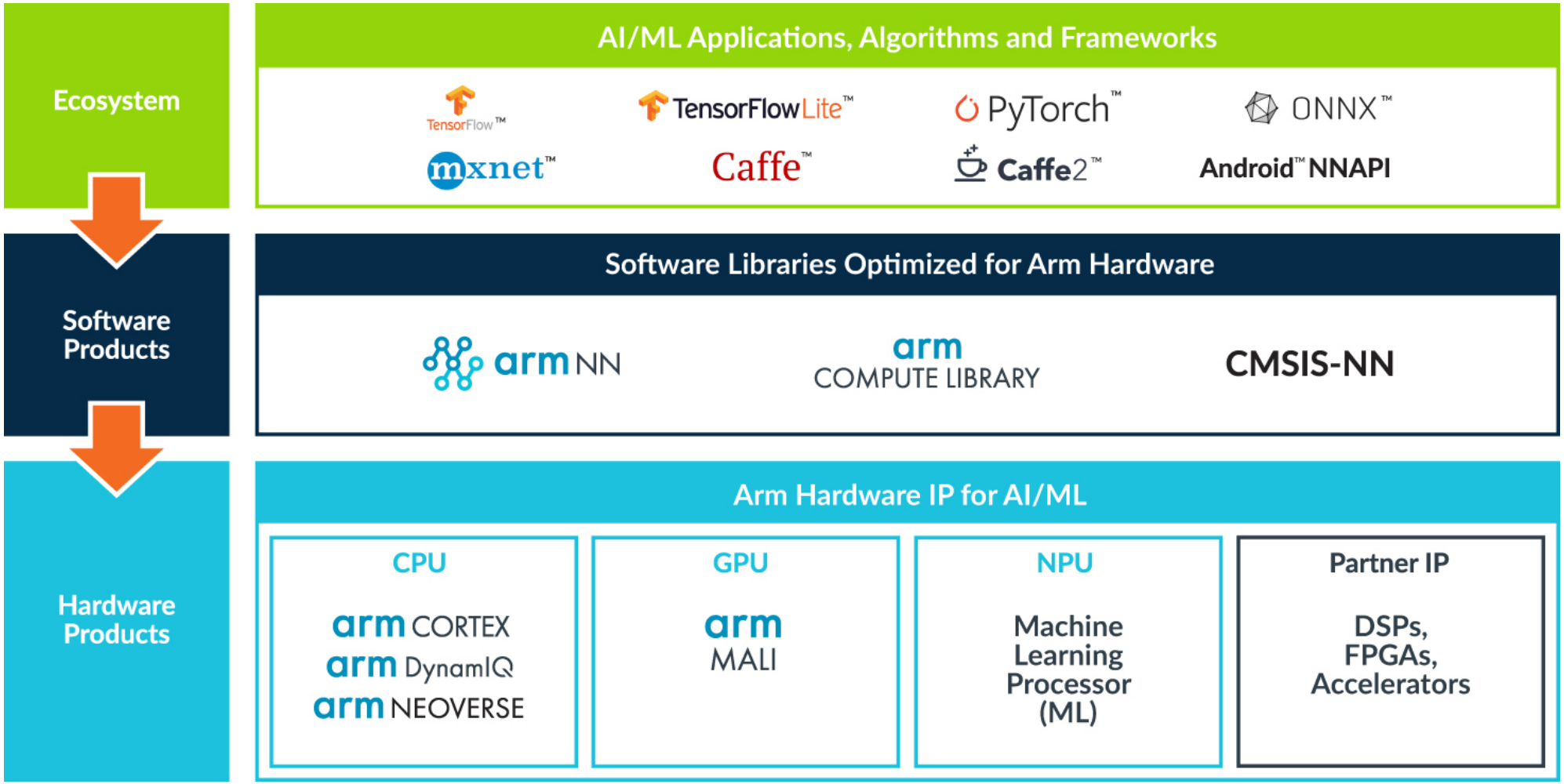
| Easy to use | Wide variety of inference and training workloads | Using Arm architecture features | On the latest Aarch64 hardware |
|---|---|---|---|
| Container images and Python Packages | Image classification | Large core count | Arm Neoverse N1, Zeus, Poseidon |
| Popular ML frameworks support Arm as a first-class citizen | Object detection | INT8, Bfloat16, FP16, FP32 | Marvell ThunderX2 |
| | | SVE / SVE2 | Fujitsu A64FX |
| | | Matrix Multiplier Extension | |



arm

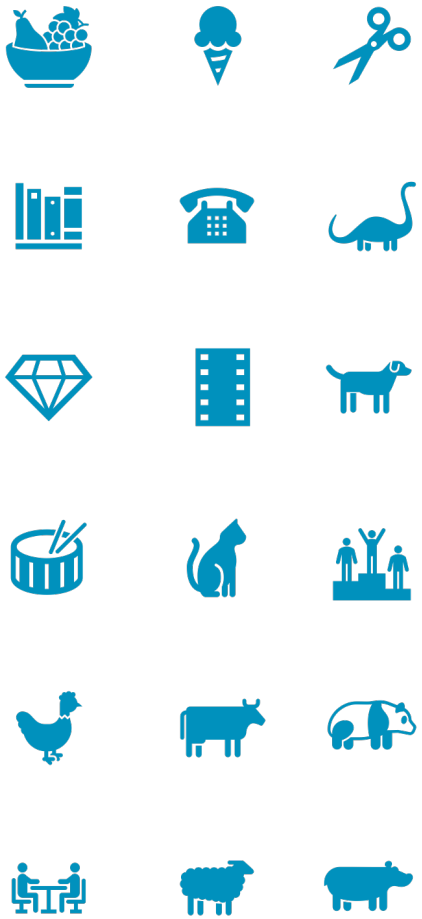
# Machine Learning and Artificial Intelligence

# Machine Learning and Artificial Intelligence

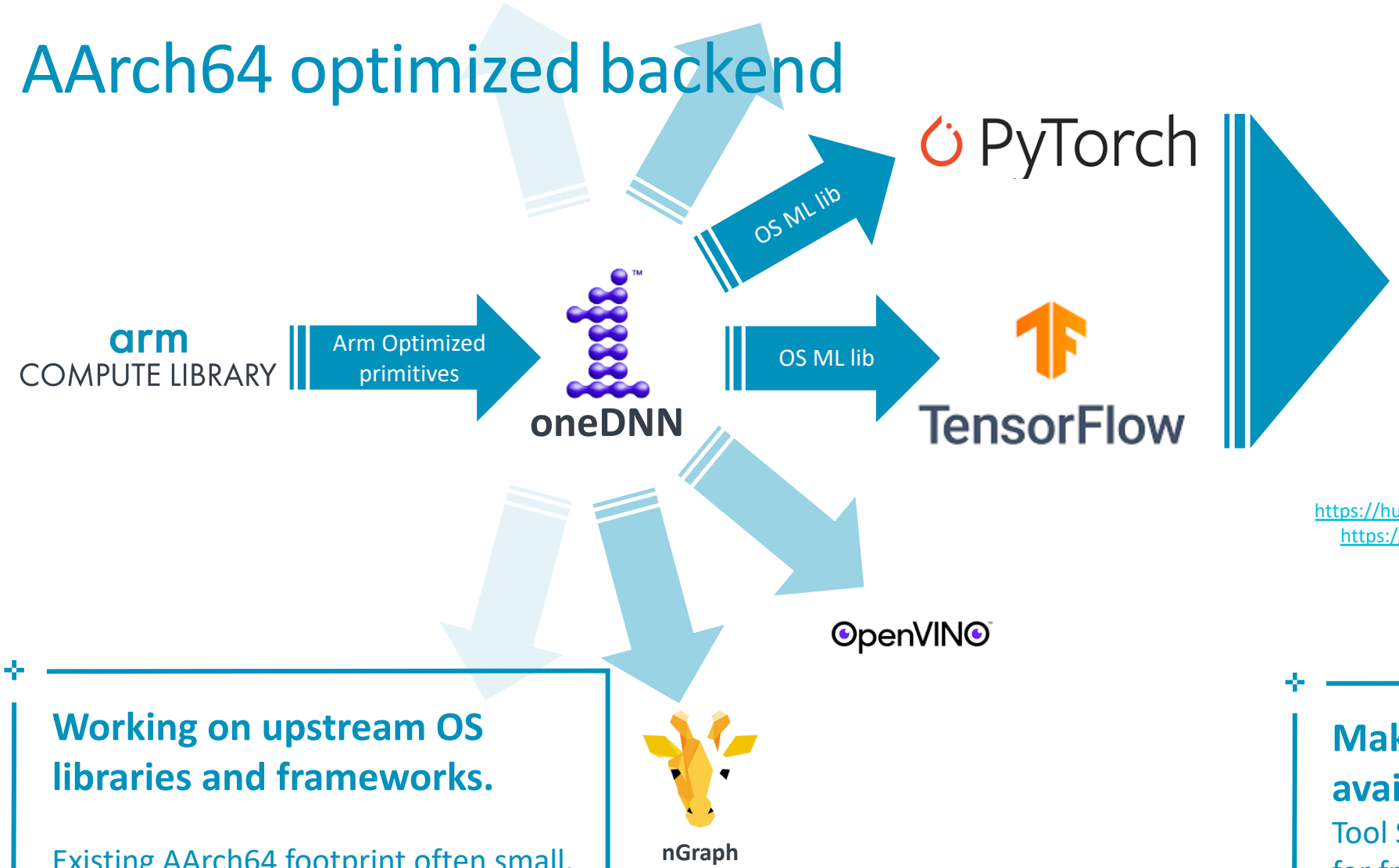


# ML Frameworks on server-class Aarch64 platforms

- Recent effort to enable server-scale on-CPU ML workloads on AArch64
- Build guides for key frameworks available:
  - Tensorflow - https://gitlab.com/arm-hpc/packages/wikis/packages/tensorflow
  - PyTorch - https://gitlab.com/arm-hpc/packages/wikis/packages/pytorch
  - MXNET - https://gitlab.com/arm-hpc/packages/wikis/packages/mxnet
  - And guides for key dependencies: CPython; NumPy etc.
- Currently focusing on inference problems
- ML Perf (https://mlperf.org) for realistic workloads.



# AArch64 optimized backend




Arm Software Tool Solutions

https://github.com/ARM-software/Tool-Solutions

Linaro

https://hub.docker.com/r/linaro/tensorflow-arm-neoverse-n1/
https://hub.docker.com/r/linaro/pytorch-arm-neoverse-n1/

**Working on upstream OS libraries and frameworks.**

Existing AArch64 footprint often small, or edge-focused.

**Making AArch64 builds available.**

Tool Solutions repo. with build scripts for frameworks, dependencies and docker images



arm

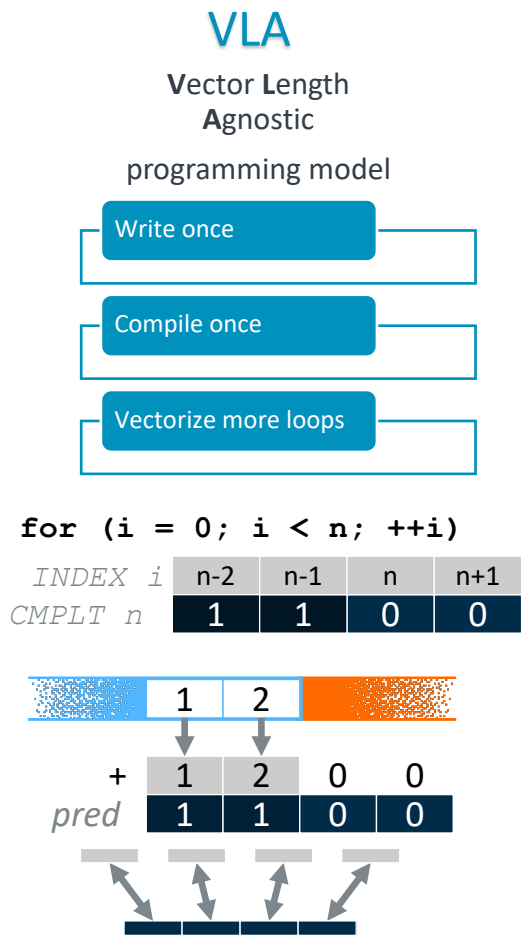
# Arm Compute Library (ACL)

- A collection of **optimized low-level machine learning functions** which leverage **Neon** (or SVE) on CPU and provide acceleration on Mali GPU through OpenCL.
- **Flexible design** and allows developers to source machine learning functions individually or use as part of complex pipelines to accelerate their algorithms and applications.
- Enabled by **Arm microarchitecture optimizations**, the Arm Compute Library provides superior performance to other OSS alternatives and support for new Arm technologies
- Open-source software, available under a permissive MIT license.
- The library provides
  - Over **100 machine learning functions** for CPU and GPU
  - **Multiple convolution algorithms** (GEMM, Winograd, FFT, and Direct)
  - Implemented behind NEConvolutionLayer, kernel selection determined by the size of the kernel; number of input/output feature map; memory footprint.
  - We access ACL at a function level, bypassing the NEConvolutionLayer interface – this pulls kernel selection into oneDNN's existing mechanism, and allows tuning for our specific targets and problems
  - Support for **multiple data types**

# ISA Developments

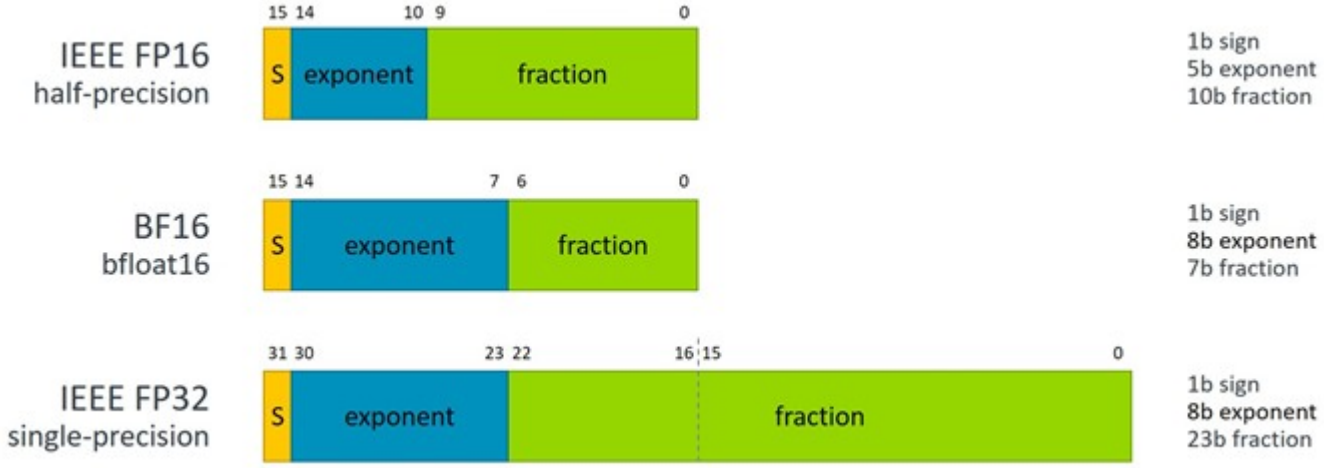
# **Scalable** Vector Extension

- SVE enables Vector Length Agnostic (VLA) programming
- VLA enables portability, scalability, and optimization
- Predicates control which operations affect which vector lanes
  - Predicates are not bitmasks
  - You can think of them as dynamically resizing the vector registers
- The actual vector length is set by the CPU architect
  - Any multiple of 128 bits up to 2048 bits
  - May be dynamically reduced by the OS or hypervisor
- SVE was designed for HPC and can vectorize complex structures
- Many open source and commercial tools currently support SVE



# New Data Type Support: BFloat16

- New addition to Armv8.6-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






arm

# FMMLA: High Performance Matrix Multiplication

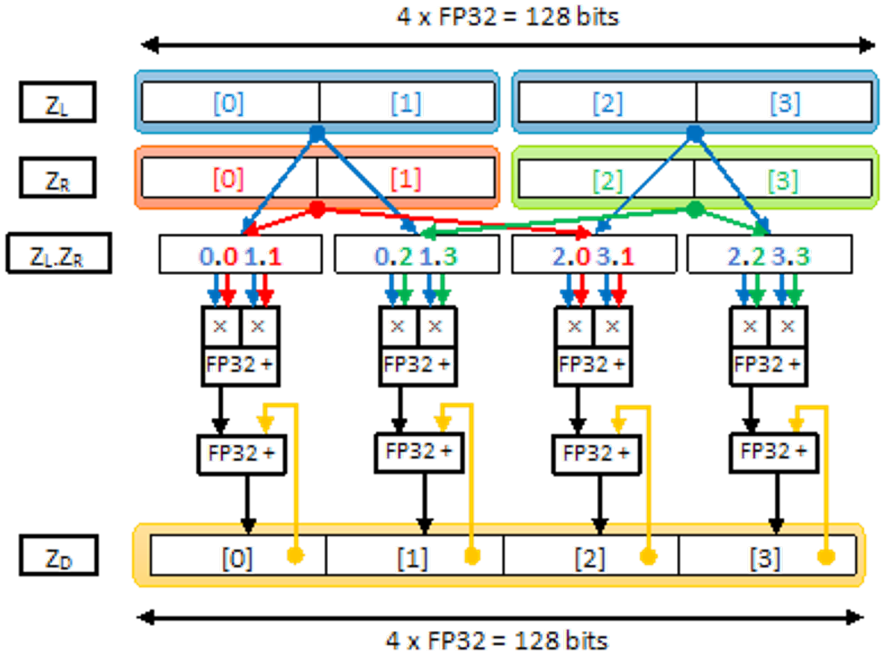
- Added to Armv8.6
  - NEON support for INT and BF16
  - FMMLA instructions for FP (SVE)

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 avoid cost
- Will accelerate maths libraries


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])






arm

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