Data plane acceleration on ARM architecture – An OSS update

Song Zhu Song.Zhu@arm.com

12/10/2018

© 2018 Arm Limited

Agenda

- Introduction
- DPDK on ARM
- FD.io/VPP on ARM
- OVS Datapath Offload and SmartNICs
- Other Projects

© 2018 Arm Limited
arm

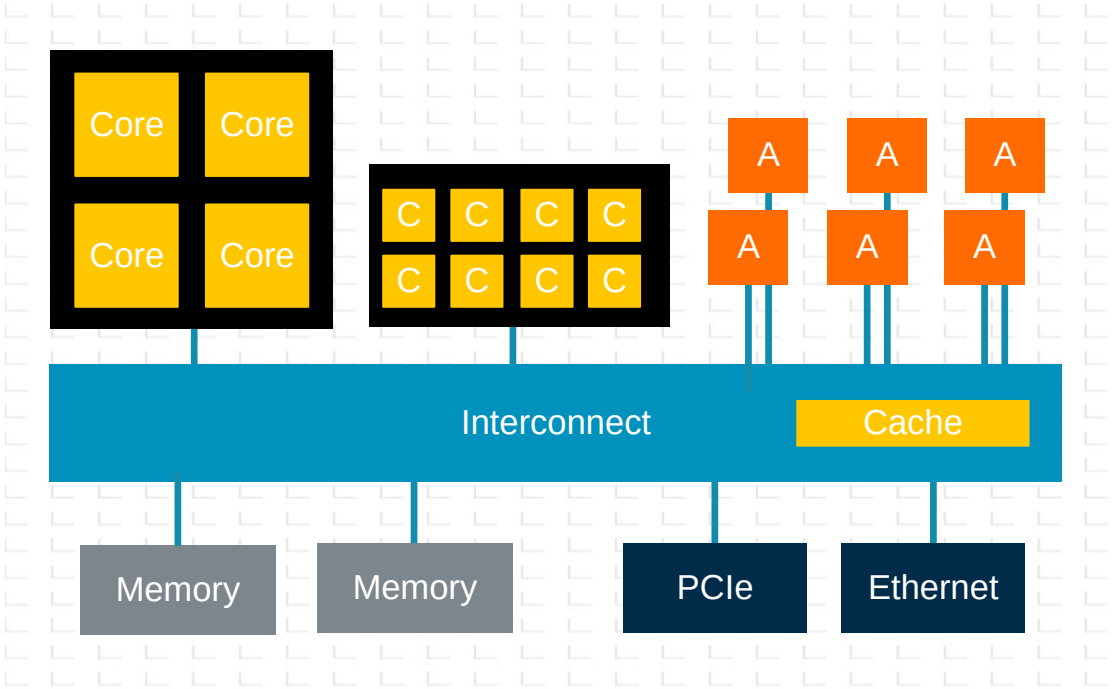
Introduction

© 2018 Arm Limited

arm

ARM SoC Diversity

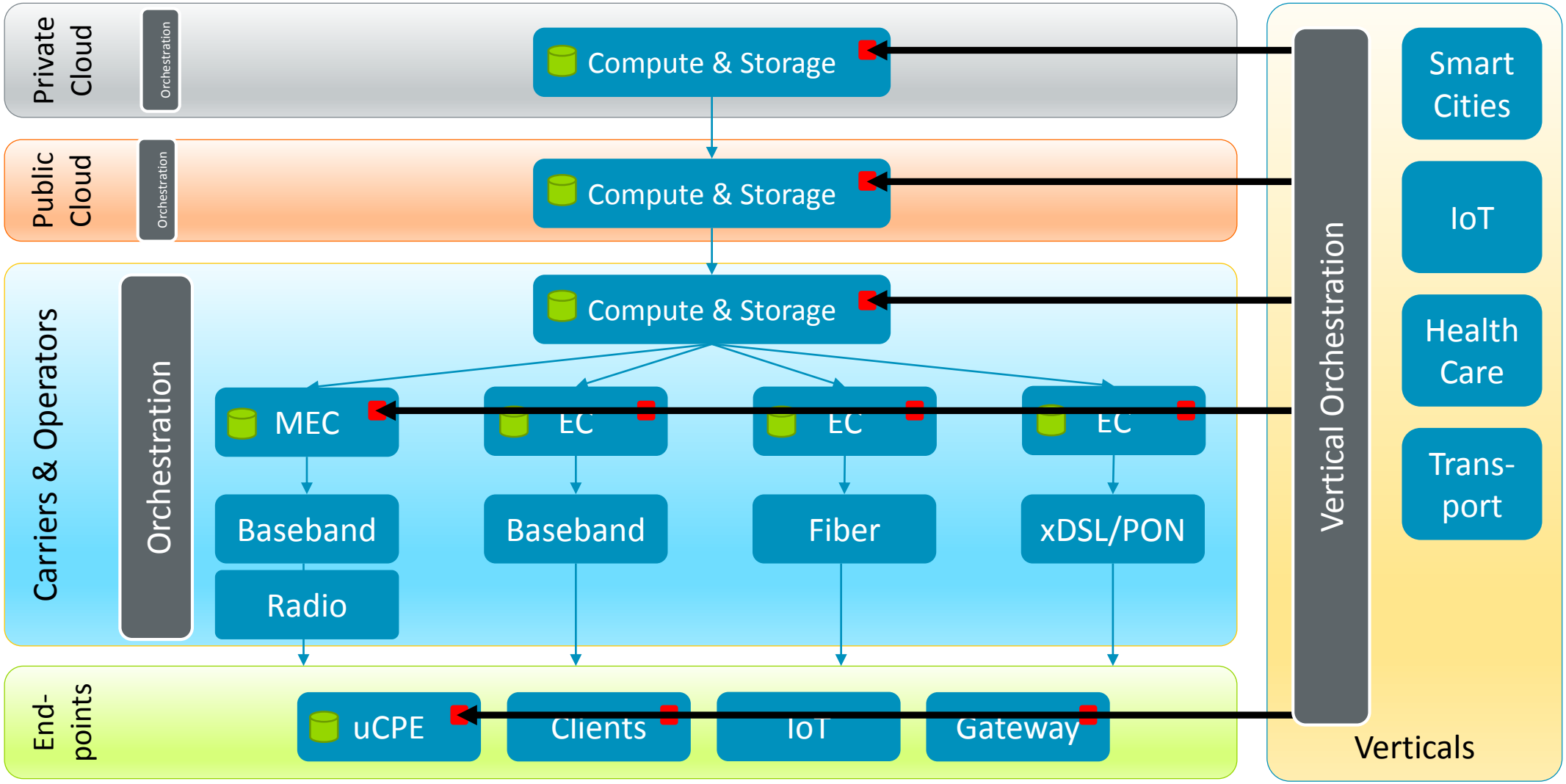
- Control plane + Data plane
  - Heterogeneous processing
- High speed IO
  - Ethernet, PCIe, CCIX …
- Acceleration
  - Scheduling, DPI, Crypto, TCAM, DSP …



(ARM SoC Diversity)

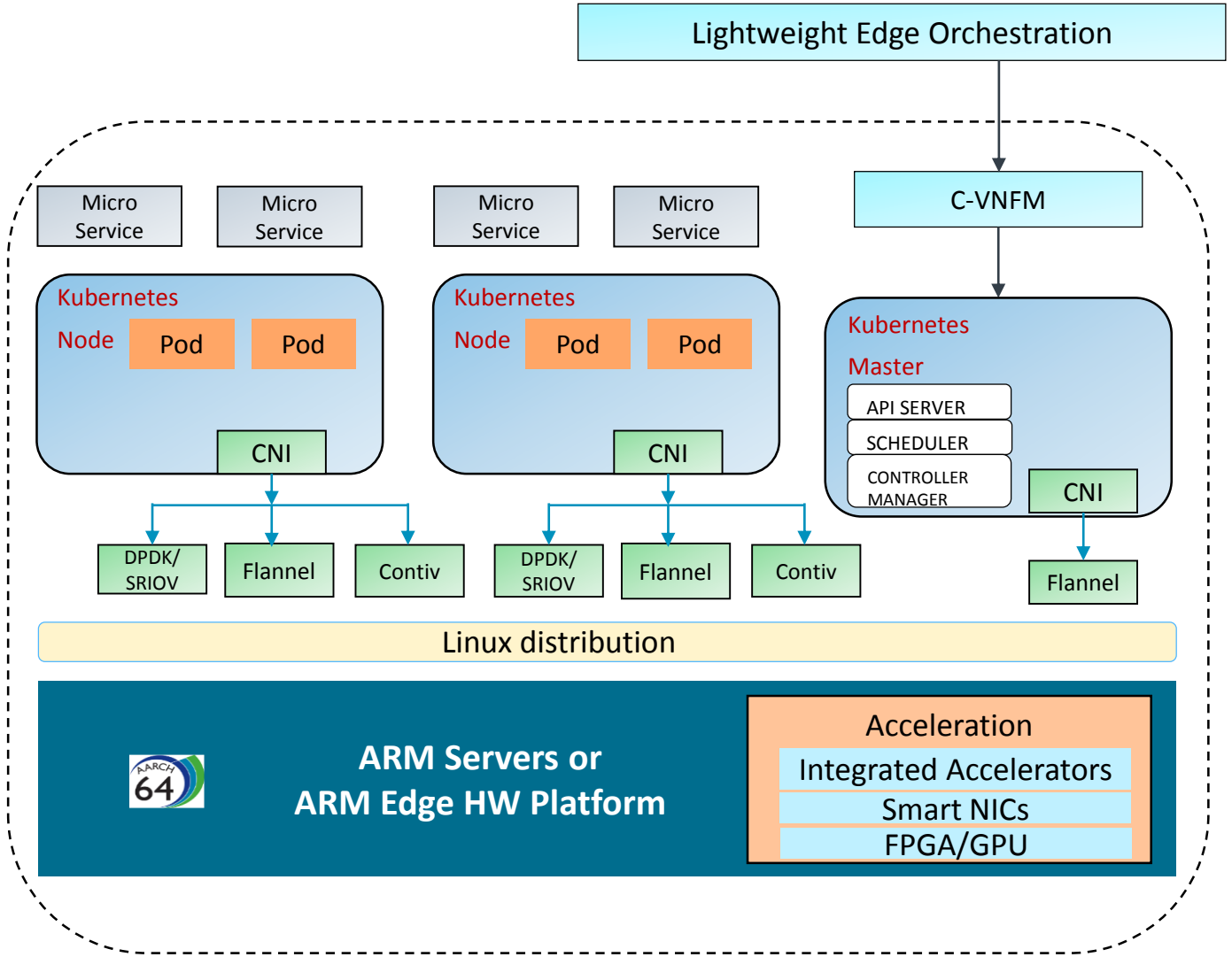
© 2018 Arm Limited
arm

Emerging Networking Architecture



© 2018 Arm Limited
arm

Container-based Edge Reference Stack



kubernetes

docker

.io

DPDK
DATA PLANE DEVELOPMENT KIT

© 2018 Arm Limited

arm

Our Goals in OSS Community

- **Enablement and optimization of data plane software on ARM**
  - Fast packet processing libraries – DPDK
  - Virtual switching – VPP/OVS

- **Edge Reference Stack design on ARM**
  - Container-based solutions
  - Integration of data plane acceleration

© 2018 Arm Limited
arm

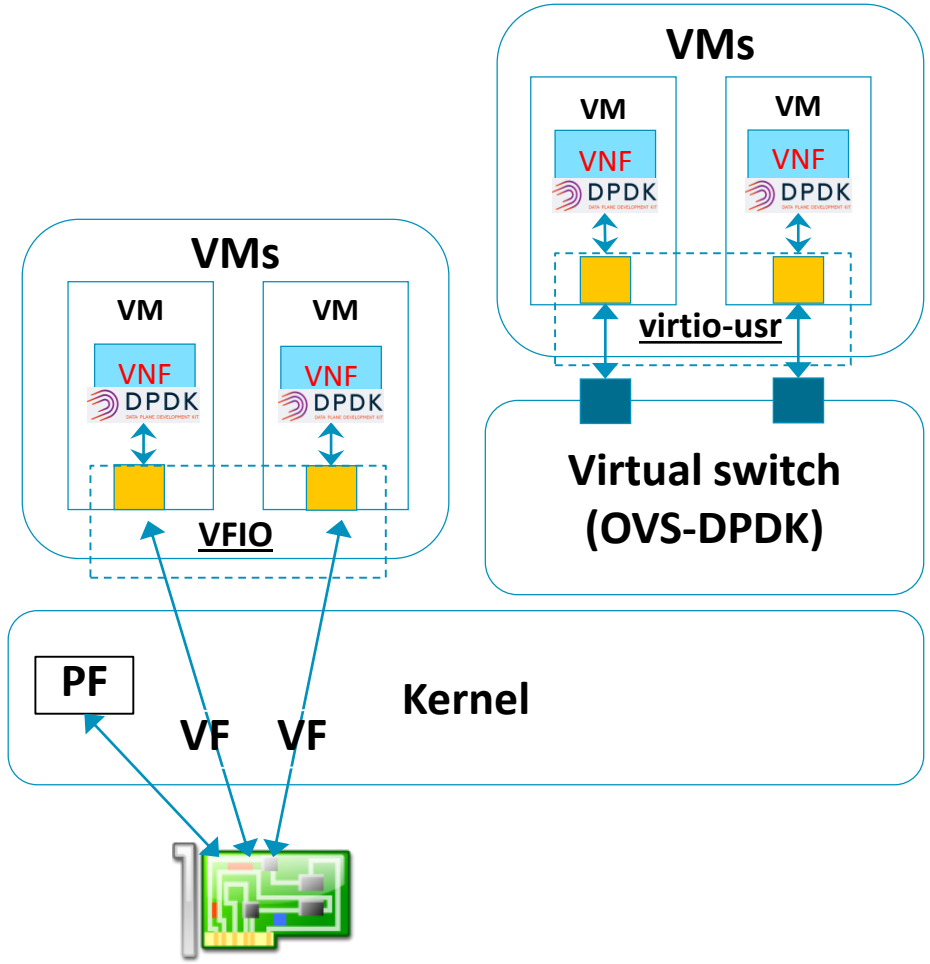
DPDK on ARM

© 2018 Arm Limited

arm

DPDK on ARM Status

- Multiple active members on ARM platforms
  - DPDK on ARM maintainer
- ARM platform porting & optimization
  - Functional verification / enabling
  - DPDK performance tuning on ARM platforms
  - DPDK dts and CI
- Architecture Support enablement for ARM SoCs
- DPDK Use cases verifications
  - OVS/VPP+DPDK, VFIO/virtio, DPDK in container etc.

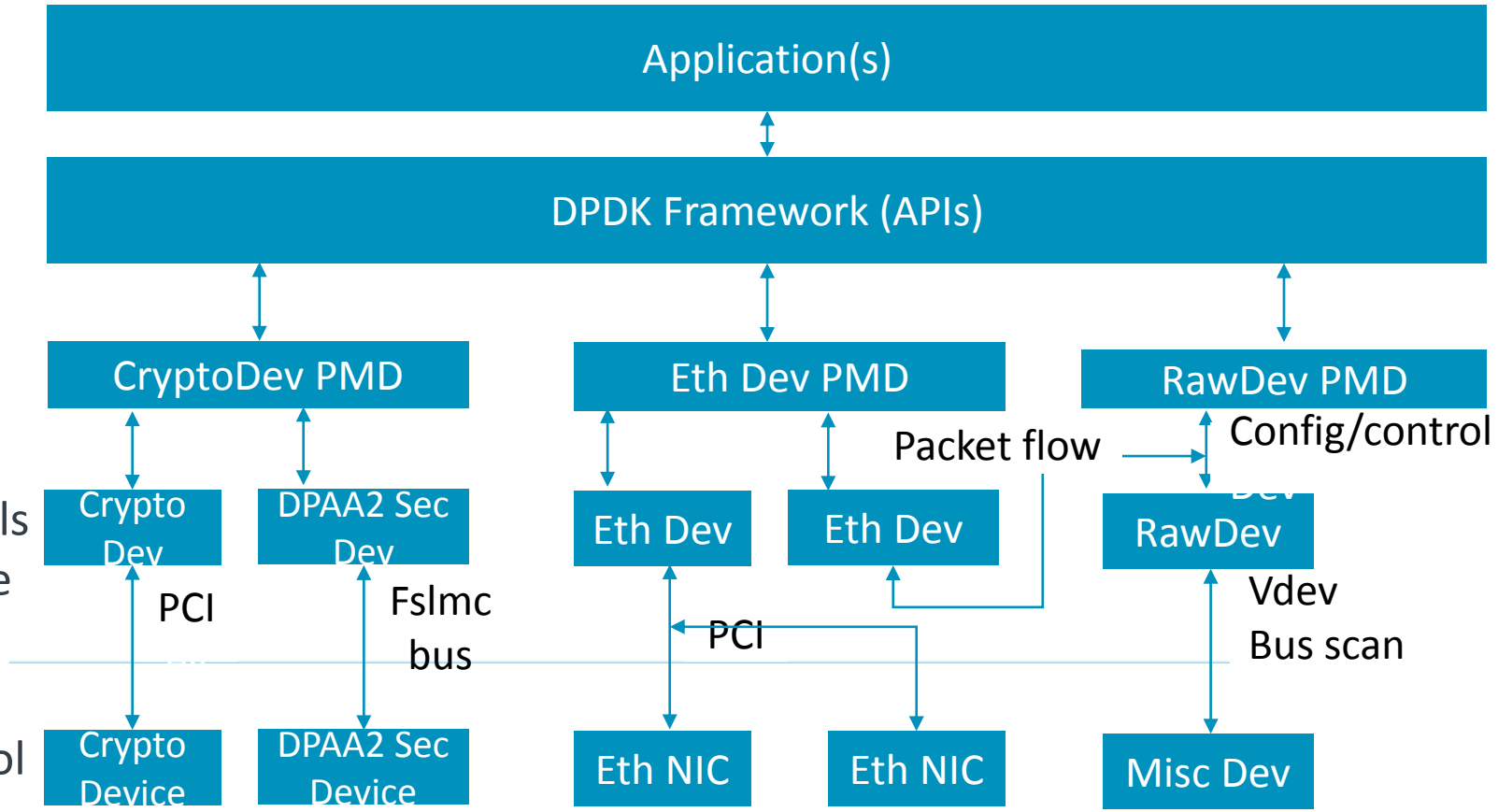


(DPDK use cases)

© 2018 Arm Limited
arm

DPDK virtual device support - vdev

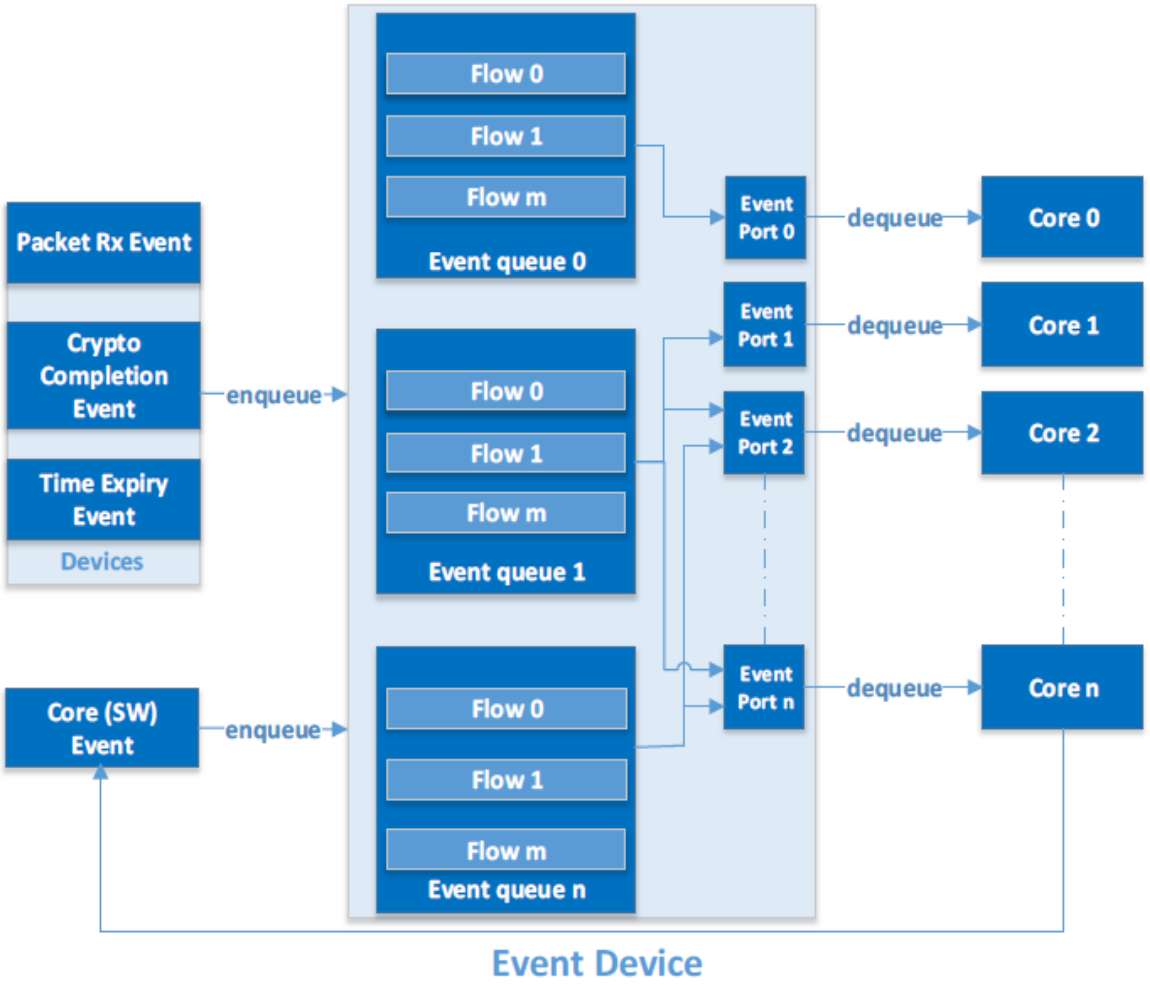
- x86 servers mostly use PCI-E NICs, the config/data flows go through PCI bus.
- ARM platforms support PCI-E NICs, and also support on-chip NICs, as well as other on-chip accelerators and HW engines
- vdev bus and raw device models were introduced to support the OPS for on-chip devices, including: start/stop/configuration/control OPS to applications



(DPDK device model)

© 2018 Arm Limited
arm

DPDK Eventdev Framework

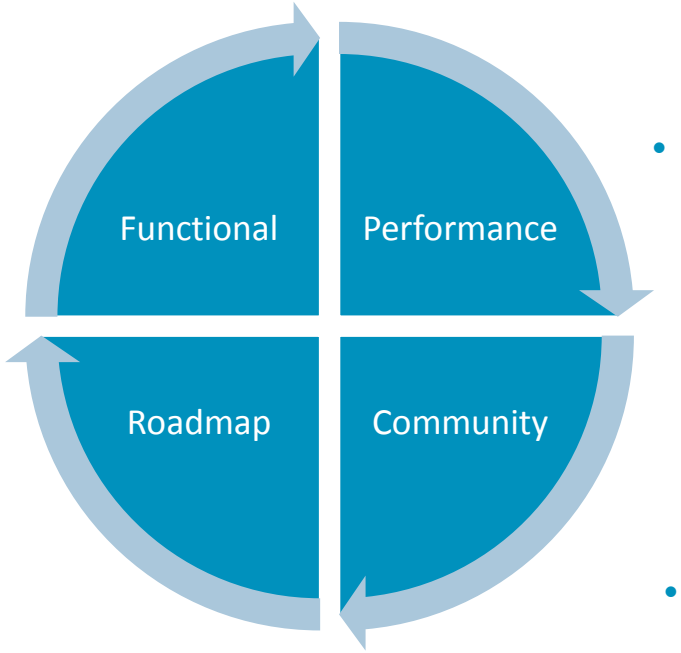


- Leverage HW Scheduler from ARM platforms
- Flow based event pipelining
- Queue based event pipelining
- Supports schedule types per flow:
  - ATOMIC
  - ORDERED
  - PARALLEL
- Event scheduling QoS based on event queue priority

© 2018 Arm Limited
arm

Future plan

- Use case & Virtualization scenarios investigation
- Architectural enhancements
- CI setup with selected platforms



- Continue with the bare-metal performance investigation & optimization, e.g. PMD, Crypto drivers, etc
- Investigate performance in virtualization & multi-core scenarios

- Define DPDK on ARM roadmap with partners

- Documentation:
  - Startup guide for ARM64
  - Optimization app notes

© 2018 Arm Limited
arm

FD.io/VPP on ARM

© 2018 Arm Limited

arm

FD.io/VPP on ARM Status

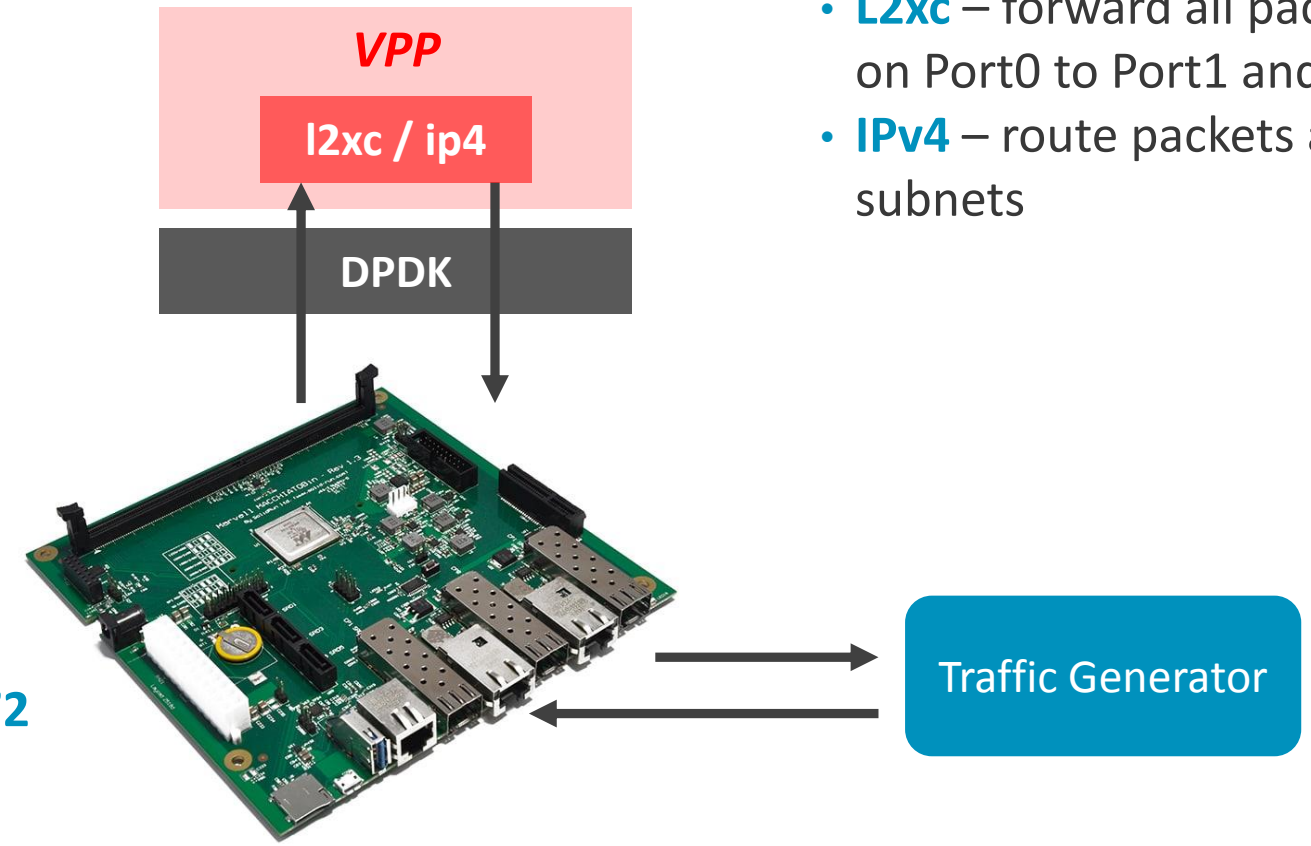
- Lead/guide the ARM community
  - Collaboration between ARM partners on FD.io/VPP
  - Set up VPP/AArch64 wiki page on FD.io for collaboration

- Enable VPP on ARM: 3-step strategy
  - Fix build, unit test, and packaging issues
  - Integrate ARM platforms into upstream CSIT (CI)
  - Performance benchmarking and tuning

- ARM code implementation
  - Vectorization in Packet Processing using Neon Intrinsic.
  - Architecture specific loop unrolling.
  - Arch specific Function Dispatching.

© 2018 Arm Limited

Starting from a Simple Use Case



- **L2xc** – forward all packets received on Port0 to Port1 and vice versa
- **IPv4** – route packets across IPv4 subnets

© 2018 Arm Limited
arm

Summary

64B packet – single flow – single core

Observations

- Most hotspots are memory accesses
- Software-defined data placement consumes processing cycles
- Unintentionally ordering memory accesses can slow the system down
- Compiler may fuse loops which alters memory access pattern from original program order

Further Directions

- RFC2544 testing
- Multicore scaling
- PMU data
- Cache stashing
- Compiler and C library versions
- Other platforms

© 2018 Arm Limited
arm

Recent Contributions

Change – Patch Set	Status
Vectorization Changes (Adding NEON Instructions in ip4 forwarding path)	Upstream - Merged
Arch specific Loop Unrolling – Dual/Quad Loop and Arch specific Function Dispatching	Upstream - Merged
Auto tools to Cmake: Marvell, Mellanox	Upstream - Merged
Marvell PP2 device	Upstream - Merged
gcc compilation issues	Upstream - Merged
clang compilation issues	Upstream - Merged
Other Build related issues	Upstream - Merged
Memory Ordering Changes	Upstream under review

- Software Efforts
  - Vectorization in Packet Processing using Neon Intrinsic.
  - Architecture specific loop unrolling
  - Arch specific Function Dispatching.

- Performance Benchmarking and Analysis
  - Marvell MacchiattoBin (Cortex A72 core)
  - Qualcomm Centriq (Falkor Core)
  - Cavium ThunderX2 (Vulcan Core)

© 2018 Arm Limited
arm

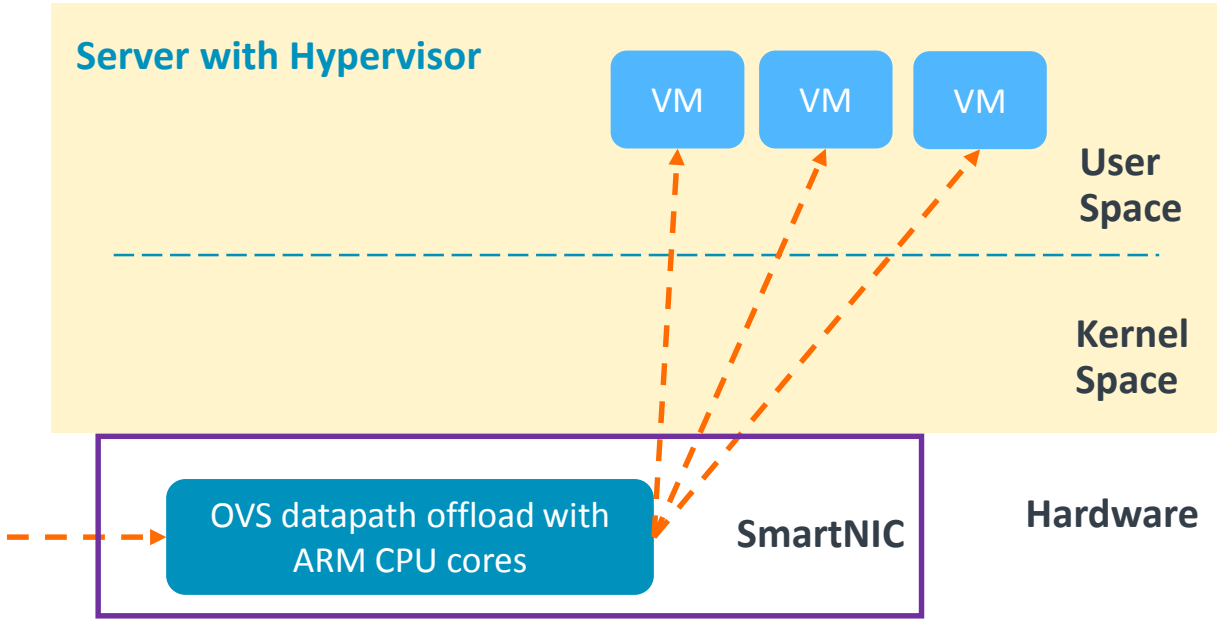
OVS Datapath Offload and SmartNICs

© 2018 Arm Limited

arm

OVS Datapath Offload

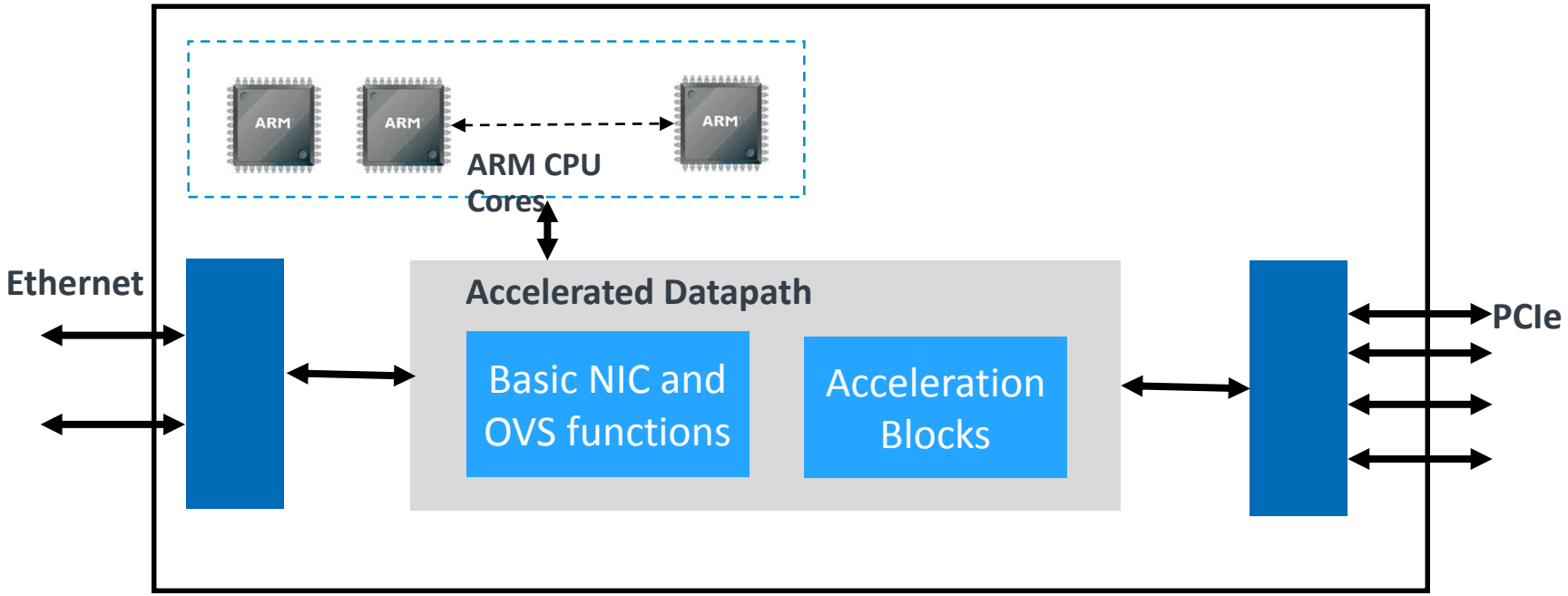
- Preserve host CPU cycles for applications
- Combination of ARM cores + accelerators
  - Most flexible and cost effective solution for networking “pre/post processing”
- Examples
  - SmartNICs: Marvell, Mellanox, Broadcom…



(OVS dataplane offload with SmartNICs)

© 2018 Arm Limited
arm

A SmartNIC Platform with ARM CPU Cores



(A SmartNIC Platform)

- Dataplane IO Interface Standardization
- HW Acceleration
- Platform standardization
- vSwitch Portability

© 2018 Arm Limited
arm

Other Projects

© 2018 Arm Limited

arm

Other Projects

- eBPF/XDP network fast path
- AF_XDP
- VirtIO 1.1
- Network protocol stack

© 2018 Arm Limited

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

arm

© 2018 Arm Limited

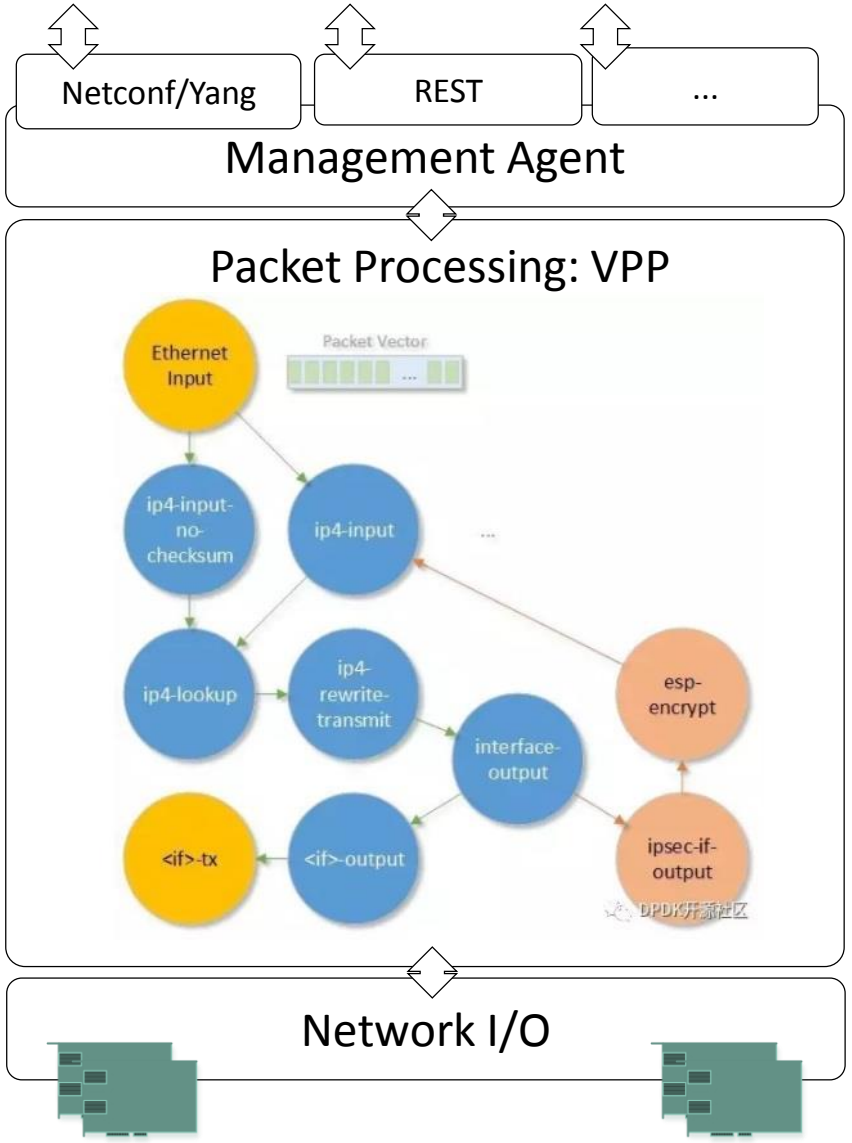
DPDK Overview

- Data Plane Development Kit
  - A set of libraries and drivers for **fast packet processing**
  - Runs mostly in Linux **userland**
  - The first supported CPU was Intel x86 and it is now extended to IBM POWER and **ARM**.
- Major members and contributors
  - Several active members on ARM platforms
  - ARM is one of the golden members

© 2018 Arm Limited

FD.io/VPP (Vector Packet Processing)

- User Space software platform providing switch/router functionalities
- Aiming to run on commodity CPUs
- Cisco developed it from 2002 and open sourced it in FD.io (Linux Foundation) on Feb 2016
- Leverage DPDK, XDP, netmap… as fast I/O
- Batch packet processing - more efficient iCache utilization
- Packet processing graph: modular, flexible, and extensible



© 2018 Arm Limited
arm

Identifying hotspots

First access to packet data

```
b0 = vlib_get_buffer (vm, bi0);
b1 = vlib_get_buffer (vm, bi1);

error0 = error1 = ETHERNET_ERROR_NONE;
e0 = vlib_buffer_get_current (b0);
type0 = clib_net_to_host_u16 (e0->type);
e1 = vlib_buffer_get_current (b1);
type1 = clib_net_to_host_u16 (e1->type);

/* Speed-path for the untagged case */
if (PREDICT_TRUE (variant == ETHERNET_INPU
                  && !ethernet_frame_is_an
```

```
 0.44   prfm   pldl1keep, [x0]
        add    x0, x25,
59.48   ldrh   w15, [x12,#12]
 0.03   prfm   pldl1keep, [x1]
 0.45   str    x0, [x29,#592]
        sub    w0, w28, #0x2
        mov    w1, w15
 0.05   str    w0, [x29,#576]
        rev16  w1, w1
10.05   ldrh   w0, [x11,#12]
        mov    v0.h[0], w1
        rev16  w0, w0
        mov    v1.h[0], w0
```

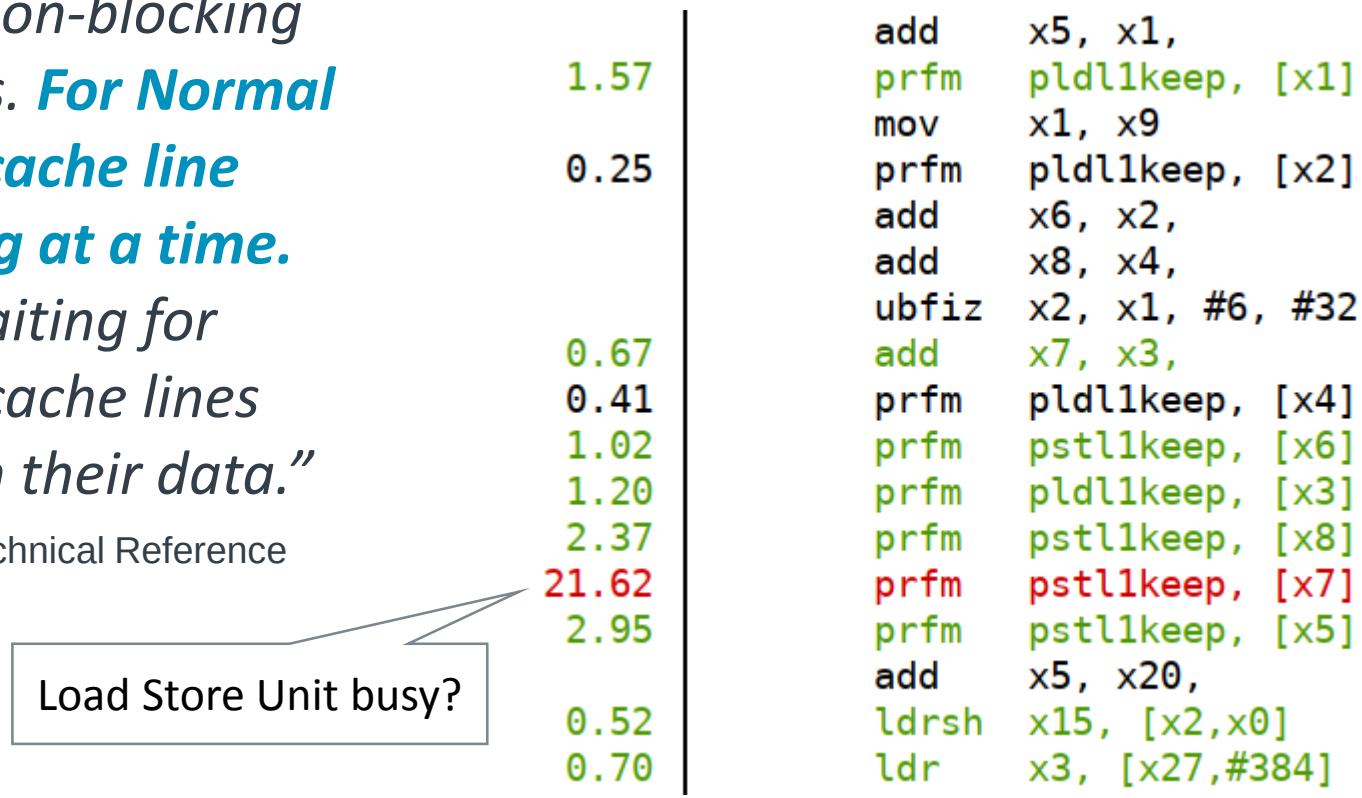
Why is memory access the hotspot?

© 2018 Arm Limited
arm

Avoiding bottlenecks

- *"The L1 memory system is non-blocking and supports hit-under-miss.* ***For Normal memory, up to six 64-byte cache line requests can be outstanding at a time.*** *While those requests are waiting for memory, loads to different cache lines can hit the cache and return their data."*
- ARM® Cortex®-A72 MPCore Processor Technical Reference Manual

Prefetching combined with loop unrolling is demanding!



© 2018 Arm Limited
arm