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
Introduction
ARM SoC Diversity

- Control plane + Data plane
  - Heterogeneous processing

- High speed IO
  - Ethernet, PCIe, CCIX ...

- Acceleration
  - Scheduling, DPI, Crypto, TCAM, DSP ...

(ARM SoC Diversity)
Emerging Networking Architecture

Private Cloud
- Orchestration
- Compute & Storage

Public Cloud
- Orchestration
- Compute & Storage

Carriers & Operators
- Orchestration
- MEC
  - Baseband
  - Radio
- EC
  - Baseband
  - Fiber
  - xDSL/PON

End-points
- uCPE
  - Clients
  - IoT
  - Gateway

Verticals
- Smart Cities
- IoT
- Health Care
- Transport
Container-based Edge Reference Stack

Lightweight Edge Orchestration

C-VNFM

Kubernetes Node
Pod
Pod
CNI
DPDK/SRIOV
Flannel
Contiv
Kubernetes Node
Pod
Pod
CNI
DPDK/SRIOV
Flannel
Contiv
Kubernetes Master
API SERVER
SCHEDULER
CONTROLLER MANAGER
CNI
Flannel

Linux distribution

ARM Servers or ARM Edge HW Platform

Acceleration
Integrated Accelerators
Smart NICs
FPGA/GPU

ARM Limited

© 2018 Arm Limited
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
DPDK on 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 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 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
Future plan

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

- Define DPDK on ARM roadmap with partners

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

- Documentation:
  - Startup guide for ARM64
  - Optimization app notes
FD.io/VPP on 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.
Starting from a Simple Use Case

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

![Diagram of ARM Cortex A72 Processors](image)

Traffic Generator
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
## Recent Contributions

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

- **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)
OVS Datapath Offload and SmartNICs
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)
A SmartNIC Platform with ARM CPU Cores

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

(A SmartNIC Platform)
Other Projects
Other Projects

• eBPF/XDP network fast path

• AF_XDP

• VirtIO 1.1

• Network protocol stack
Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
תודה
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
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
Identifying hotspots

First access to packet data

```c
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
```}

Why is memory access the hotspot?
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.**”


```
add     x5, x1, 1.57
prfm    pldl1keep, [x1] 0.25
mov     x1, x9 0.67
prfm    pldl1keep, [x2] 0.41
add     x6, x2, 1.02
add     x8, x4, 1.20
ubfiz   x2, x1, #6, #32 2.37
add     x7, x3, 21.62
prfm    pldl1keep, [x4] 2.95
prfm    pstl1keep, [x6]
prfm    pldl1keep, [x3]
prfm    pstl1keep, [x8]
prfm    pstl1keep, [x7]
prfm    pstl1keep, [x5]
add     x5, x20, 0.52
ldrsh   x15, [x2,x0] 0.70
ldr     x3, [x27,#384]
```