HKG18-106: Introducing OpenDataPlane® Tiger Moth

Bill Fischofer
LNG Tech Lead, ODP Project
Topics for Today

- Brief background on ODP motivation/rationale
- Where we’ve come from
- Today’s ODP Announcement
- Where we’re going
- Q&A
Why OpenDataPlane?

Application Goals

- Write once, run on any Platform
- Take full advantage of available HW acceleration w/o effort

Platform Vendor Goals

- Compete for any Socket
- Fully exploit unique IP without requiring application changes
Platforms of Interest

General Server Platforms
- SW-centric: Cores must touch every packet
- Standardized acceleration (Special CPU instructions, e.g., AES, and/or plug-in cards with PCIe latencies)
- Standardized I/O (mainly PCIe)
- Standard form-factors (mainly rack chassis)
- Leading architectures: x86, Arm (AArch64)
- Mainly cloud environment

Network SoC-based Platforms
- HW-centric: Packets may bypass cores partially or completely
- Innovative and non-standard approaches to acceleration (value-add IP)
- Varying I/O (e.g., integrated)
- Varying form factors (e.g., Smart NIC, embedded devices)
- Leading architectures: Arm (AArch64), custom
- Mainly embedded environment
OpenDataPlane - A Brief History

- Linaro Networking Group founded, ODP project launched
- First preview release(s)
- OpenDataPlane v1.0 released for evaluation
- First Long Term Support (LTS) release - ODP “Monarch”
- Development releases
- Second LTS release - ODP “Tiger Moth”
Target ODP Applications

OpenDataPlane is finding traction in many different markets - for example:

- **Telecommunications**
  - EPC - Evolved Packet Core
  - Cellular Base Stations - pico to macro
  - Cellular Base Station Backhaul - with large encrypted “FAT” pipes
  - vCPE and uCPE - Customer Premise Equipment
  - SD-WAN - Software Defined Wide Area Networks
  - ETSI NFV based solutions - Application does not change with or without acceleration

- **Enterprise and Cloud**
  - Advanced Firewall Appliances and vFirewall in the Cloud
  - Security and VPN Gateways
    - vIPS/IPS (Intrusion Prevention Systems)
    - Malware detection
  - Routers, Application Load Balancers - any packet moving devices

Suitable for applications where low cost/low power/high efficiency of Arm is important, but may also want to run in virtual environments on Arm-based or x86 server class machines
ODP API as an “Innovation Boundary”

For Applications
- Define functional needs without concern for how they are met
- No requirement to be expert in platform architecture or internals
- Permit portability across all conforming implementations with at most a recompile
- Defines an ABI that permits application binary portability within an Instruction Set Architecture (ISA)

For SoC Vendors
- Implementation internals not exposed across API boundary
- APIs may be realized in HW, SW, or any combination without impacting applications
- Vendors not constrained by a fixed SW-centric implementation model
- Permits unique platform value to be exposed without requiring any application changes
Today - Second LTS Release - ODP “Tiger Moth”

- Production Grade Data Plane with Major New Features
  - IPsec Lookaside and Inline support - up to 3X Performance
  - New Crypto Ciphers and Hashes
  - Updated Classifier & Scheduler and more…

- Spans both Arm and x86 architectures
  - Fully leverages native HW acceleration on SoC platforms
  - Leverages DPDK on SW-centric platforms

- LNG reference implementation available 1Q18
- Arm-based SoC native implementations available later this year
- Foundation for future development and direction
API Specification defines the ODP “innovation boundary”

Reference Implementation is a pure-SW implementation of ODP, and a starting point for SoC native implementations. Platform-neutral and can run anywhere that has a Linux kernel.

Validation Test Suite ensures that all implementations conform to the ODP API specification

SoC Native Implementations offer best-in-class performance with full HW offload/acceleration
ODP “Tiger Moth” New APIs and Features

Classification
- RED and back pressure support, hashing

Crypto
- New cipher and authentication modes
- SHA2-hashes (256, 384, 512)
- Packet-oriented ops (with data ranges)
- Cryptographic and True randoms

Events
- New types and subtypes
- Multi-free, filtering and conversion

Initialization
- Selectable features
- Application resource configuration

IPsec
- Lookaside and inline offload support
- Traffic Flow Confidentiality (TFC) support

Packets
- Packet references
- Per-packet checksum and protocol controls

PktIO Support
- Checksum controls
- Integrated hashing, parsing, and IPsec

Scheduler
- Sequential and asynchronous ordered locks

Shared Memory
- Shared VAs, External communication
ODP “Tiger Moth” Packet Processing Overview

Application need only program its own threads (yellow boxes). Everything else is provided by ODP.

Inline IPsec flows show core bypass on RX and TX paths
Application touches packet 3 times:
- Receive encrypted packet
- Pass encrypted packet to IPsec engine, receive decrypted result
- Process decrypted packet (work application wants to do)
- Pass result packet to IPsec engine, receive encrypted result
- Transmit encrypted packet
Application touches packet only once:
- Receive encrypted packet, which is decrypted as part of RX path
- Process decrypted packet (work application wants to do)
- Transmit result packet, which is encrypted as part of TX path

3x lookaside performance
SW-Centric Platforms and ODP

Important class of platforms of interest, particularly in Cloud environments

DPDK has matured considerably over past 2 years
- Focused on server-class platforms
- Available on Arm-based as well as x86 platforms

How does ODP fit in? Two options:
- Develop an independent production-grade SW implementation of ODP
- Leverage existing DPDK investment for SW-centric platforms
Assessment

- Telecom applications can be very complex: Application may spend 98% of per-packet cycles in application and only 2% in framework, so framework cycle differences not very important.
- Lots of tunnels. Simple RSS is not enough. ODP brings transparent framework that can leverage HW or implement receiver+worker threads in software
- ETSI NFV requirement: Application does not change with or without acceleration

We decided to extend DPDK with ODP to cover the largest range of situations and rely on native ODP implementations on SoCs for the best performance.
Cross Platform Support - Arm & x86

ODP Apps

Write Once - Accelerate Anywhere

ODP API

Arm-based SoC Native ODP Implementation

Optimal Performance

Arm-based SoC Platforms

ODP Extensions

Adds missing DPDK features

DPDK

SW-Centric Arm-based or x86 Platform

Good Performance

Binary compatibility across platforms sharing same Instruction Set Architecture (ISA).
Recompile to switch to different ISA.
ODP 2018 Directions

Future support to include:

● New APIs
  ○ Compression
  ○ Enhanced Tunneling
  ○ Additional 3GPP Crypto Algorithms

● Further Arm-based SoC optimizations

● Further ODP Extension enhancements for DPDK
  ○ Allow better mapping of ODP features/APIs onto DPDK APIs for server/Cloud Arm-based and x86 implementations
  ○ Future DPDK improvements pass through to ODP with minimal incremental investment
Summary

OpenDataPlane is Ready for Production Use

● Spans Arm and x86 Networking Communities
  ○ Optimized for Arm-based SoCs, but also runs on Arm-based and x86 Servers
  ○ Native implementations available - but can also use DPDK

● Tiger Moth Long Term Support (LTS) release
  ○ Will be supported for at least 24 months

● Availability
  ○ Reference Implementation available 1Q18
  ○ Arm-based SoC native implementations available later this year

● Write once, accelerate anywhere
  ○ Allows applications to be written once and be assured of best performance wherever they run
Thank You

#HKG18

HKG18 keynotes and videos on: connect.linaro.org
For further information: www.linaro.org