VPP Host Stack
Transport and Session Layers

Florin Coras, Dave Barach, Keith Burns, Dave Wallace
VPP - A Universal Terabit Network Platform
For Native Cloud Network Services

- **Most Efficient on the Planet**
  - EFFICIENCY
    - The most efficient software data plane Packet Processing on the planet

- **Superior Performance**
  - PERFORMANCE
    - FD.io on x86 servers outperforms specialized packet processing HW

- **Flexible and Extensible**
  - SOFTWARE DEFINED NETWORKING
    - Software programmable, extendable and flexible

- **Cloud Native**
  - CLOUD NETWORK SERVICES
    - Foundation for cloud native network services

- **Open Source**
  - LINUX FOUNDATION
    - Open source collaborative project in Linux Foundation

**Breaking the Barrier of Software Defined Network Services**
**1 Terabit Services on a Single Intel® Xeon® Server!**
Motivation: Container networking

FD.io Mini-Summit at KubeCon 2017
Motivation: Container networking

FD.io Mini-Summit at KubeCon 2017
Why not this?

FD.io Mini-Summit at KubeCon 2017
VPP Host Stack

FD.io Mini-Summit at KubeCon 2017
VPP Host Stack: Session Layer

- Maintains per app state and conveys to/from session events
- Allocates and manages sessions/segments/fifos
- Isolates network resources via namespaces
- Session lookup tables (5-tuple) and local/global session rule tables (filters)
- Support for pluggable transport protocols
- Binary/native C API for external/builtin applications
VPP Host Stack: SVM FIFOs

- Allocated within shared memory segments
- Fixed position and size
- Lock free enqueue/dequeue but atomic size increment
- Option to dequeue/peek data
- Support for out-of-order data enqueues
VPP Host Stack: TCP

- Clean-slate implementation
- “Complete” state machine implementation
- Connection management and flow control (window management)
- Timers and retransmission, fast retransmit, SACK
- NewReno congestion control, SACK based fast recovery
- Checksum offloading
- Linux compatibility tested with IWL TCP protocol tester

FD.io Mini-Summit at KubeCon 2017
Comms library (VCL) apps can link against
LD_PRELOAD library for legacy apps
epoll
Application Attachment

attach
bind (server)
connect (client)

App
Binary API
Session
TCP
IP, DPDK
VPP

shm segment
Session Establishment

Client

Binary API

Session

TCP

IP, DPDK

VPP

Server

Binary API

Session

TCP

IP, DPDK

VPP

attach
bind

listen
Session Establishment

Client

Binary API

Session

TCP

IP, DPDK

VPP

Server

Binary API

Session

TCP

IP, DPDK

VPP

attach
connect

open

attach
bind

listen

FD.io Mini-Summit at KubeCon 2017
Session Establishment

Client
- Binary API
  - Session
  - TCP
  - IP, DPDK
- VPP

Server
- Binary API
  - Session
  - TCP
  - IP, DPDK
- VPP
Session Establishment

connect succeeded

handshake

new client
Session Establishment

Session Establishment Diagram:

- **Client**
  - Binary API
  - Session
  - TCP
  - IP, DPDK
  - shm segment
  - rx tx

- **Server**
  - Binary API
  - Session
  - TCP
  - IP, DPDK
  - shm segment
  - rx tx

- **connect reply**
- **accept notify**
- **VPP**

FD.io Mini-Summit at KubeCon 2017
Data Transfer

**Client**
- TCP
- IP, DPDK
- VPP

**Session**
- Binary API

**Client**
- Copy to buffer
- Write

**Server**
- TCP
- IP, DPDK
- VPP

**Session**
- Binary API

**Server**
- Read
- Copy to fifo

**Congestion control**
- Reliable transport

**Flow Diagram**:
- TX write evt
- RX write evt
- RX to TX
- TX to RX
Data Transfer

Not yet part of CSIT but some rough numbers on a E2690: ~200k CPS and ~12Gbps/core!

FD.io Mini-Summit at KubeCon 2017
Redirected Connections (Cut-through)

Client

Binary API

Session

TCP

IP, DPDK

Server

bind

VPP

FD.io Mini-Summit at KubeCon 2017
Redirected Connections (Cut-through)

Client

connect

Binary API

Session

TCP

IP, DPDK

VPP

Server

redirect

FD.io Mini-Summit at KubeCon 2017
Redirected Connections (Cut-through)

Throughput is memory bandwidth constrained: ~120Gbps!
Multi-threading

- Connections/sessions ‘pinned’ to a thread
- Per-thread data structures/state
Namespaces are configured independently and associate applications to network layer resources like interfaces and fib tables.
Features: Session Tables

Request access to global and/or local scope

FD.io Mini-Summit at KubeCon 2017
Features: Session Tables

- Both tables have “rules table” that can be used for filtering
- Local tables are namespace specific and can be used for egress filtering
- Global tables are fib table specific and can be used for ingress filtering
Ongoing work

- Overall integration with k8s
  - Istio/Envoy
- TCP
  - Rx policer/tx pacer
  - TSO
  - New congestion control algorithms
  - PMTU discovery
  - Optimization/hardening/testing
- VCL/LD_PRELOAD
  - Iperf, nginx, wget, curl
Next steps – Get involved

• **Get the Code, Build the Code, Run the Code**
  • Session layer: src/vnet/session
  • TCP: src/vnet/tcp
  • SVM: src/svm
  • VCL: src/vcl

• **Read/Watch the Tutorials**

• **Read/Watch VPP Tutorials**

• **Join the Mailing Lists**
Thank you!

Florin Coras
email: fcoras@cisco.com
irc: florinc