Two optimizations discussed in this talk:

1) Optimizing Miniflow Extract
2) Protocol Aware Hashing
• **What is “miniflow extract”?**
  • Parses packet, extracts metadata, builds miniflow
    • Represents all metadata OVS uses
    • Used in datapath for classifying packets/rules etc

• **Designing Optimized Code**
  • Optimizing MFEX with AVX512 SIMD Instructions
  • Scaling both Traffic Profiles and CPU ISA

• **Validation and Unit testing**
  • Automated approach to ensure correctness
Using the Optimized MFEX

$ dpif-netdev/miniflow-parser-set scalar  # default
$ dpif-netdev/miniflow-parser-set avx512    # only works if CPU ISA is available
$ dpif-netdev/miniflow-parser-set autovalidator  # validate implementations are ==
MFEX Basics: The AVX512 Shuffle

Packet:

- 0
- 0
- 0
- 0x8
- 0x0
- 0
- 0
- 0
- 0x45
- 0
- 0
- 0
- 0

Mask:

- 0
- 0
- 0
- 0xFF
- 0xFF
- 0
- 0
- 0
- 0
- 0
- 0
- 0
- 0

Hit values:

- -
- -
- 0x8
- 0x0
- -
- -
- 0x45
- -
- -
- -
- -
- -
- -

Protocol mask:

- 0
- 0
- 0
- 0xFF
- 0xFF
- 0
- 0
- 0
- 0
- 0
- 0
- 0
- 0

Protocol values:

- -
- -
- 0x8
- 0x0
- -
- -
- 0x45
- -
- -
- -
- -
- -
- -
MFEX Basics: The AVX512 Shuffle

Packet:

0 0 0 0x08 0x00 0 0 0x45 0

Pattern:

0 0 0 0xFF 0xFF 0 0 0xFF 0

Hit Values:

- - - 0x08 - - 0x00 - - 0x45 -

Shuffle:

Miniflow:

FF FF FF FF FF 7 FF FF FF FF FF FF FF FF

Miniflow (Result):

FF FF FF FF FF FF FF FF FF

Protocol Mask:

Packet:

0 0 0 0x08 0x00 0 0 0x45 0

Protocol Values:

Shuffle
The AVX512 MFEX: Static Data

- Processing Ether/IPv4/UDP
  - Protocol Specific Data
    - **mask**, **value**, and **shuffle**
  - Extra Metadata
    - Miniflow bits
      - known from Ether/IPv4/UDP
    - DP Packet properties
      - Packet Lengths
      - L2/L3 Offsets
The AVX512 MFEX: Static Data

- Processing Ether/IPv4/UDP
  - Protocol Specific Data
    - `mask`, `value`, and `shuffle`
  - Extra Metadata
    - Miniflow bits
      - known from Ether/IPv4/UDP
    - DP Packet properties
      - Packet Lengths
      - L2/L3 Offsets
mfex_generic(int profile) { ... }

Performance & Scaling AVX512 MFEX

PROTOCOL MASK
PROTOCOL VALUES
MINIFLOW
SHUFFLE

Eth/IP/UDP

Eth/IP/TCP

Eth/.../.../...
Performance & Scaling AVX512 MFEX

mfex_generic(int profile) { ... }

mfex_ip_udp() {
    mfex_generic(0)
}

mfex_ip_tcp() {
    mfex_generic(1)
}
Performance & Scaling AVX512 MFEX

mfex_generic(profile) { ... }

Generic Function DOES NOT execute on datapath!

mfex_ip_udp()  
Eth/IP/UDP

mfex_ip_tcp()  
Eth/IP/TCP

Inlined version DOES execute on datapath
- Specialized to profile!
- Specialized to CPU ISA!
Validation And Testing

- **Autovalidator Compares Miniflows**
  - Ensure Miniflow is always same
  - Confidence in all implementations

- **Fuzz testing**
  - Try make the autovalidator fail!

```bash
$ dpif-netdev/miniflow-parser-set autovalidator
```
Hashing: Overview

• What hashing is done? Software RSS
  • L3 IP src/dst
  • L4 UDP/TCP src/dst ports

• When does this need to be done?
  • Guest to Network (southbound vhost)

• What is Protocol Aware Hashing?
  • Re-use MFEX profile knowledge
  • AVX512 “knowledge” to Optimize Hashing
Protocol Aware Hashing

- **Known Packet Data Layout**
  - From AVX512 Profile Match
  - Load Hash data *directly* from packet
    (not from miniflow like scalar)
Performance Benefits: Overall

Instruction Parallelism
- AVX512 SIMD
- Packet Hashing
MFEX IPv6 & Protocol Aware Hashing

http://patchwork.ozlabs.org/project/openvswitch/list/?series=263303

Reviews & Comments welcomed
! Thank You !

? Questions ?

Kumar Amber  kumar.amber@intel.com
Harry van Haaren  harry.van.haaren@intel.com