Network Acceleration
XDF 2018

Ravi Sunkavalli
VP, DC & IP Design Solutions
Agenda

> Networking Industry Trends
> Network Acceleration Platform & Usecases
> Key enabling IP, Tools & Technology
Data rates increasing rapidly

CPU Performance not scaling

Virtualization & Bare Metal

High Feature Velocity
SmartNICs in Hyperscale Cloud

- Amazon Nitro, Microsoft Azure SmartNIC in every server (>2 million deployed)
- Use cases evolving, need for reconfigurable acceleration
- Cloud service providers want new efficiencies with the decline of Moore’s Law
- Storage and compute acceleration in the NIC
Virtualization and BareMetal trend favors FPGAs

Microsoft Azure SmartNIC

- FPGA enables support for >50Gbps networking
- Microsoft leadership for FPGA adoption
- Hyperscale deployment (>1M)

Aren’t FPGAs much bigger than ASICs?
Not really, within 2x to 3x with targeted features

Aren’t FPGAs very expensive?
Not really, compared to CPUs, DRAM and Flash

Aren’t FPGAs hard to program?
Not really, hw/sw codesign provides hw performance with sw flexibility

Can FPGAs be deployed at hyperscale?
Yes, >1m servers with FPGA SmartNICs deployed

Isn’t my code locked in to a single FPGA vendor?
No, Shell-role model allows portability
Agenda

> Networking Industry Trends

> Network Acceleration Platform & Use cases

> Key enabling IP, Tools & Technology
Network Acceleration Platform

Host Application Framework

- Network Offload (VPN, FW, SLB, Sec, Meter, SL)
- Compute Offload (ML, Database etc)
- Storage Offload (RDMA, Comp etc)

Network SW Stack
Compute SW Stack
Storage SW Stack

PCle (Gen4) / CCIX

PCIe Gen3/4

QDMA

FPGA Network Acceleration Platform

10G NIC
Match-Action (SDNet-P4)
QoS TM Meter
Crypto (IPsec/DTLS)
Vswitch Offload (OVS/GFT)
RDMA (RoCE)
Compute Acceleration Kernels

To ToR
## Typical Usecases

### Basic Networking
- Basic NIC Offloads (LSO, TSO, RSS, Checksum)
- Tunneling Offloads (VXLAN, NVGRE)
- SR-IOV, MAC/VLAN Filtering
- QoS/Buffering

### Network Acceleration
- P4/SDNet Programmable Data Plane
- OVS Acceleration
- Datapath Encryption/Decryption/DPI
- In-Band Telemetry and Monitoring

### Storage Acceleration
- NVMe, NVMe-oF, RoCEv2, NVMe-oTCP
- Storage Compression, Encryption, Dedup (RAID)
- Database Key Value Store and Analytics
- SQL Query Acceleration

### Compute Acceleration
- Machine Learning – Random Forest, K-means
- Video Encode, Decode, Scaling
- Deep Learning – CNN Inference
- HPC/Scientific

© Copyright 2018 Xilinx
Usecase - OVS Offload with ML

**Problem:** Malware detection on encrypted traffic being switched by OVS

**Solution:**
- Redirect encrypted traffic to separate ML block for scanning
- Feed header fields, packet length, inter-packet gap, packet rate, etc. into ML engine
- Use trained decision trees to flag and tag or redirect suspicious traffic
Agenda

Networking Industry Trends

Network Acceleration Platform & Usecases

Key enabling IP, Tools & Technology

- QDMA
- SDAccel Streaming Platform
- SDNet-P4
- Typical SoftNIC
PCIe QDMA IP

Converged Host Interface
- Stream, Memory Mapped, Network, Storage

Configurable
- Upto Gen3x16 support

2048 Queue sets
- H2C, C2H, CMPT

SR-IOV Support
- 4PF, 252VF

Customization with User Logic
- Traffic Management, Virtual Switch, RDMA, NVMe-EP
PCIe QDMA IP Performance

Leading edge Small Packet Performance exceeding 95Mpps

* Forwarding performance >98Mpps for 64B packets with 8 Queues

https://www.xilinx.com/products/intellectual-property/pcie-qdma.html#overview
https://www.xilinx.com/support/answers/34536.html
SDAccel Streaming Platform

> Streaming Platform EA planned for 2018.3
> Kernels can Rd/Wrt streaming data from Host or to each other
> Stream Traffic Manager enables multiple streaming kernels
Why SDNet?
Parser in P4: Data Center Usecase Example

```c
parser MyParser(packet_in packet,
    out headers hdr,
    inout metadata meta,
    inout standard_metadata_t
smeta) {

    state start {
        transition parse_eth;
    }

    state parse_eth {
        packet.extract(hdr.eth);
        meta.eth_dmac = hdr.eth.dmac;
        transition select(hdr.eth.type) {
            VLAN_TYPE : parse_vlan;
            IPV4_TYPE : parse_ipv4;
            IPV6_TYPE : parse_ipv6;
            default   : accept;
        }
    }

    state parse_vlan {
        packet.extract(hdr.vlan.next);
        //meta.vlan_tpid = hdr.vlan.last.tpid;
        transition select(hdr.vlan.last.tpid) {
            VLAN_TYPE : parse_vlan;
            IPV4_TYPE : parse_ipv4;
            IPV6_TYPE : parse_ipv6;
            default   : accept;
        }
    }

    state parse_ipv4 {
        packet.extract(hdr.ipv4);
        packet.extract(hdr.ipv4opt, (((bit<32>)hdr.ipv4.hdr_len - 5) * 32));
        verify(hdr.ipv4.version != 4, error.IpVersionNotSupported);
        meta.ipv4_src = hdr.ipv4.src;
        transition select(hdr.ipv4.protocol) {
            TCP_PROT : parse_tcp;
            UDP_PROT : parse_udp;
            GRE_PROT : parse_gre;
            default   : accept;
        }
    }

    state parse_gre {
        packet.extract(hdr.gre);
        packet.extract(hdr.greopt, (((bit<32>)hdr.gre.c_flag * 32) +
            ((bit<32>)hdr.gre.r_flag * 32) +
            ((bit<32>)hdr.gre.e_flag * 32) +
            ((bit<32>)hdr.gre.s_flag * 32));
        meta.gre_proto = hdr.gre.proto_type;
        transition select(hdr.gre.proto_type) {
            NVGRE_TYPE : parse_nvgre;
            default   : accept;
        }
    }
}
```

P4 enables networking design to be completely programmable, top-to-bottom, soup-to-nuts*

* P4.org
100Gbps Packet Processing with SDNet
Resource Utilization for standard five tuple example

**Tool** | **Language** | **Table size** | **k LUTs** | **k FFs** | **BRAM** | **URAM** | **Rate**
--- | --- | --- | --- | --- | --- | --- | ---
SDNet 2018.1 | SDNet – PX | 64k EM (64Kx104x16) | 19.1 | 23.7 | 27 | 56 | 100 Gb/sec
SDNet-P4 2018.3 | SDNet - P4 (estimate) | 64k EM (64Kx104x16) | 8K | 10K | 27 | 56 | 100 Gb/sec
SDNet-P4 2018.3 | SDNet - P4 (estimate) | Parser only | 1.1K | 2.5K | - | - | 100 Gb/sec

**Legend**
- Packet
- Tuple

**Diagram**
- **IN**
- **parser**
- **mytable**
- **editor**
- **OUT**

- **Extracts standard five-tuple**
- **Looks up five-tuple as exact match to retrieve a partial VLAN header**
- **If there was a hit, inserts VLAN header. If there is already a VLAN header, then it’s added as a Q-in-Q header**
Xilinx Algorithmic Lookup Features

- Exact matching for MAC addresses and flows
- Multi-field wild card matching for ACLs
- Longest prefix matching for IP routes
- Priority decoding for multiple matches
- Ternary and semi Ternary masking
- TCAM capacity of each BRAM = 20Kb, URAM = 160Kb

Longest prefix search using ternary matching

<table>
<thead>
<tr>
<th>IP Prefix (route)</th>
<th>Key</th>
<th>Mask</th>
<th>Prio</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.1/32</td>
<td>0xA01 0101</td>
<td>0xFFFF FFFF</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>10.1.1.0/28</td>
<td>0xA01 0100</td>
<td>0xFFFF FFF0</td>
<td>28</td>
<td>2</td>
</tr>
<tr>
<td>10.1.0.24/24</td>
<td>0xA01 0100</td>
<td>0xFFFF FF00</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>10.1.0.0/16</td>
<td>0xA01 0000</td>
<td>0xFFFF 0000</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>10.0.0.0/8</td>
<td>0xA00 0000</td>
<td>0xFFF0 0000</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>
## SDNet Algorithmic Lookup Solution

### BRAM/URAM
- 600 MSPS
- 20M/Sec inline and 100K/Sec SW updates
- 40-60 clock cycle latency (600 MHz)
- EM/LPM/ACL : SDNet 2018.3

### DDR
- External DRAM based
- Millions of ACL/LPM/EM entries
- Integrated with Xilinx DDR controller IPs
- EM: SDNet 2019.1
  - LPM/ACL: SDNet 2019.2

### HBM
- Up to 8GB on chip DRAM for lookups
- Millions of ACL/LPM/EM
- HBM as result memory
- EM/LPM/ACL : SDNet 2019.2
**Xilinx SDNet-P4 v2018.3**

*Scheduled for release in November 2018*

- **Verification Environment**
  - Top level Verilog wrapper
  - Verilog Engines (Encrypted)
  - System Verilog Testbench
  - Lookup Engine C++ Drivers
  - High level C++ Testbench

- **Native P4 compilation**

- **Xilinx SDNet-P4 compiler**

- **Vivado IPI**

- **FPGA image for any hardware**
  - SmartNIC, XBB boards,
  - Customer Specific hardware

- **P4 Runtime drivers**

  Scheduled for 2019.1 release
## SDNet Roadmap

<table>
<thead>
<tr>
<th>SDNet 2017.3</th>
<th>SDNet 2018.1</th>
<th>SDNet 2018.3</th>
<th>SDNet 2019.1</th>
<th>SDNet 2019.2</th>
</tr>
</thead>
</table>
| • PX Support  
  • System Builder  
  • VU9P SLRD (VCU118) | • Minor release  
  • **200Gbps** throughput with five tuple design on Ultrascale+ devices | • P4 as primary development language  
  • New URAM-based Search IPs for  
  • EM,  
  • LPM  
  • TCAM  
  • 2x-3x better Resource and Latency Improvements | • DRAM support for EM Search IP  
  • New P4 runtime APIs | • Search IPs with DRAM (HBM and DDR) for ACLs and LPM  
  • EA Support for **7nm** devices |
Typical Soft NIC Feature Summary

- PCIe QDMA
- FPGA SDNet-P4 softNIC
- 50G/100G Ethernet

> SDNet-P4 enables NIC datapath
> Ethernet
  >> 100GbE/50GbE/25GbE
> PCIe
  >> Gen3x16, Gen3x8
> OS support
  >> Windows
  >> Linux & DPDK
  >> VMWare

> Stateless NIC offloads
> Packet flow steering/Filtering
> Tunneling offloads
> Server Virtualization
> Server Manageability
> Remote Boot
Summary
Summary

- FPGAs enable Adaptable Network Acceleration
- Hardware Performance with Software-level programmability
- Customized solutions for 100Gbps and beyond
- Rich set of IP, Tools and Boards
Adaptable. Intelligent.