Smart (Programmable) NICs

ECE/CS598HPN

Radhika Mittal
FlexTOE: Flexible TCP Offload with Fine-Grained Parallelism

Rajath Shashidhara, Tim Stamler, Antoine Kaufmann, Simon Peter

NSDI’22

Some of the content has been taken from Rajath’s NSDI talk.
Motivation

- Software network stacks (including kernel bypass) have high CPU overhead.
  - Specifically evaluated Linux TCP and TAS
Motivation

• Existing TCP offload engines have limited flexibility (slow upgrade cycles).
  – Specifically evaluated Chelsio Terminator TOE
Motivation

• Software network stacks (including kernel bypass) have high CPU overhead.

• Existing TCP offload engines have limited flexibility (slow upgrade cycles).

• How to get both flexibility and performance?
FlexTOE

- Flexible TCP offload on SoC-based smartNICs with network processors.
Key challenges

• SoC based SmartNICs have large number of wimpy cores with limited memories.
  – Parallel architecture geared towards stateless offloads.

• TCP connections require stateful sequential (in-order) processing.
Netronome Agilio (NFP-4000)
Netronome Agilio (NFP-4000)

Each FPC island has 12 FPCs (flow processing cores).
- Each FPC is an independent 32 bit core at 800MHz.
- Each core supports up to 8 hardware threads.
- Lacks support for floating point operations or timers.

Small amount of memory.
Netronome Agilio (NFP-4000)

Each FPC island has 12 FPCs (flow processing cores).
- Each FPC is an independent 32 bit core at 800MHz.
- Each core supports up to 8 hardware threads.
- Lacks support for floating point operations or timers.

Small amount of memory.
NVIDIA (Mellanox) BlueField DPU

ARM cores

Bluefield DPU 2:
8 64bit cores
(upto 2GHz)

Bluefield DPU 3:
16 64bit cores
(upto 3GHz)
released in April 2021

Figure from: https://www.storagereview.com/news/nvidia-bluefield-2-dpu-delivers-record-setting-performance
NVIDIA (Mellanox) BlueField DPU

Programmable datapath accelerator

16 cores, 256 threads for massive parallelism

Programmed through NVIDIA’s DOCA interface.

Figure from: https://www.storagereview.com/news/nvidia-bluefield-2-dpu-delivers-record-setting-performance
NVIDIA (Mellanox) BlueField DPU

- 8MB L2 cache
- 16MB L3 cache
- 16GB on-board RAM (DDR)

Figure from: https://www.storagereview.com/news/nvidia-bluefield-2-dpu-delivers-record-setting-performance
Key challenges

• SoC based SmartNICs have large number of wimpy cores with limited memories.
  – Parallel architecture geared towards stateless offloads.

• TCP connections require stateful sequential (in-order) processing.
FlexTOE’s approach

- Decouple control plane from datapath.

- Modularity: fine-grained modules keep private state and communicate explicitly

- Fine-grained parallelism: Modules may be replicated, sharded, execute out-of-order

- One-shot data-path offload: Payload is never buffered on the NIC
FlexTOE’s Offload Architecture

SmartNIC
- Data-path
  - Segment Generation & Transmission
  - Loss Detection & Recovery
  - Payload Transfer
  - Application Notification
  - Flow Scheduling

Host
- Control-plane
  - Application Interface Mgmt.
  - Connection Control
  - Congestion Policy

Applications
- Application / libTOE
- POSIX Sockets
FlexTOE’s Offload Architecture

- **Data-path**: per-packet transport logic for established connections
FlexTOE’s Offload Architecture

- **Control-plane**: policy, management and infrequent recovery code-paths
• **libTOE library**: provides POSIX sockets to the application with kernel-bypass
FlexTOE's on-NIC datapath

Baseline
FlexTOE’s on-NIC datapath
FlexTOE’s on-NIC datapath
FlexTOE’s on-NIC datapath
FlexTOE’s on-NIC datapath
Example: Transmit (Tx)

TX Seg #1: Alloc Head Steer Seq Pos Payload TX

enters the pipeline first

TX Seg #2: Alloc Head Steer Seq Pos Payload TX

Time

Modules
- Pre
- Proto
- Post
- DMA
Example: Transmit (Tx)

TX Seg #1
- Alloc
- Head
- Steer
- Seq
- Pos
- Payload
- TX

assign sequence number

TX Seg #2
- Alloc
- Head
- Steer
- Seq
- Pos
- Payload
- TX

Time

Modules
- Pre
- Proto
- Post
- DMA
Example: Transmit (Tx)

Parallel TCP Processing Example: Transmit (TX)

Time

<table>
<thead>
<tr>
<th>TX Seg #1</th>
<th>Alloc</th>
<th>Head</th>
<th>Steer</th>
<th>Seq</th>
<th>Pos</th>
<th>Payload</th>
<th>TX</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX Seg #2</td>
<td>Alloc</td>
<td>Head</td>
<td>Steer</td>
<td>Seq</td>
<td>Pos</td>
<td>Payload</td>
<td>TX</td>
</tr>
</tbody>
</table>

Modules

- Pre
- Proto
- Post
- DMA
Example: Transmit (Tx)

TX Seg #1: Alloc, Head, Steer, Seq, Pos

TX Seg #2: Alloc, Head, Steer, Seq, Pos

Payload, TX

Time

Modules:
- Pre
- Proto
- Post
- DMA
Example: Transmit (Tx)

TCP requires processing in-order for loss detection

but ...

Data-parallel modules have varying processing times and may reorder segments

Tx Seg #1

Time

Tx Seg #2

out-of-order transmit
Example: Transmit (Tx)

Assign sequence number on data-path ingress → reorder segments on egress

TX Seg #1

Alloc  Head  Steer  Seq  Pos  Payload  TX

TX Seg #2

Alloc  Head  Steer  Seq  Pos  Payload  TX

Time

Modules
Pre  Proto  Post  DMA

reorder before transmit
Other design and implementation aspects

- No buffering in NIC, but not zero-copy.
  - Send and receive buffers maintained in libTOE (POSIX-compliant).
- Transmissions triggered when app sends more data or when data is acked.
- On-NIC datapath takes care of retransmission due to duplicate acks.
- On-NIC datapath also generates acks (with ECN bits or timestamp information).
- On-NIC datapath collects relevant stats and reports them to on-host control plane (used for congestion control).
- On-host control plane handles rate/window adjustment (congestion control logic) and retransmissions due to timeouts.
- On-NIC datapath enforces per-flow rates using timing wheel (Carousel).
- Build specialized caches at different levels based NFP-4000’s memory architecture.
Timing Wheel in Carousel

Time slots from “now” till “horizon”. All packets in the “now” slot get dequeued. $O(1)$ insertion and deletion.
Other design and implementation aspects

- No buffering in NIC, but not zero-copy.
  - Send and receive buffers maintained in libTOE (POSIX-compliant).
- Transmissions triggered when app sends more data or when data is acked.
- On-NIC datapath takes care of retransmission due to duplicate acks.
- On-NIC datapath also generates acks (with ECN bits or timestamp information).
- On-NIC datapath collects relevant stats and reports them to on-host control plane (used for congestion control).
- On-host control plane handles rate/window adjustment (congestion control logic) and retransmissions due to timeouts.
- On-NIC datapath enforces per-flow rates using timing wheel (Carousel).
- Build specialized caches at different levels based NFP-4000’s memory architecture.
Enabling flexibility

- Support for XDP (eXpress Data Path) modules implemented in eBPF.
eBPF

• Allows running verified code supplied by a user-space application in the kernel.
• eBPF programs are invoked when certain hooks are triggered (e.g. a system call, a network event, etc).

Source: https://www.infoq.com/articles/gentle-linux-ebpf-introduction/
XDP

- Uses eBPF to provide support for bare-metal processing of raw packets at the lowest point in the stack.

Source: https://www.iovisor.org/technology/xdp
Enabling flexibility

• Support for XDP (eXpress Data Path) modules implemented in eBPF.

• Use it to implement common datacenter features
  – Tracing, statistics, profiling
  – Connection firewalling
  – VLAN encapsulation/decapsulation
  – TCPDump.
Evaluation: tail latency

Memcached latency distribution across different stack combinations

FlexTOE achieves the lowest median and tail latencies

FlexTOE Server

Client: Linux, Chelsio, TAS, FlexTOE

Latency (us)

0 50 100 150

50p 99p 99.99p
Evaluation: throughput

Memcached throughput, varying number of server cores

**FlexTOE saves up to 81% CPU cycles versus Chelsio and 50% versus TAS**

![Graph showing throughput comparison](image)

Even though latency difference is small between FlexTOE and TAS, why do we see a more significant throughput improvement?
Evaluation: factor analysis

Throughput (Gbps):
- Baseline: 0.07
- Pipelining: 3.64
- Intra-module parallelism: 8.2
- Replicated Pre/Post: 11.09
- Replicated Pipeline: 22.69

Latency 99.99p (us):
- Baseline: 6929
- Pipelining: 684
- Intra-module parallelism: 148
- Replicated Pre/Post: 106
- Replicated Pipeline: 58

Improvement factors:
- Baseline to Pipelining: 286x
- Baseline to Replicated Pipeline: 120x
Evaluation: on BlueField

FlexTOE on Bluefield

Speedup = improvement in throughput
Your thoughts?

• What did you like about the paper?

• What were its limitations?
What are some other applications of smart NICs?
Other applications of SmartNICs

• Offloading distributed applications
  – iPipe, SIGCOMM’19
• Caching for key-value stores
  – IncBricks, ASPLOS’17
• Load balancing / request steering
  – RPCValet, ASPLOS’19
  – A Case for Informed Request Scheduling at the NIC, HotNets’19
• Remote memory calls, HotNets’20
• Network functions (using FPGA-based smartNICs)
  – ClickNP, SIGCOMM’16
  – FlowBlaze, NSDI’19
• …..