IX: A Protected Dataplane Operating System for High Throughput and Low Latency

Adam Belay, George Prekas, Samuel Grossman, Ana Klimovic, Christos Kozyrakis, Edouard Bugnion
HW is fast

64-byte TCP Echo:

- HW Limit
- Linux
- IX

Microseconds

Requests per Second

Millions

0 10 20 30 40 50 60

20 40 60 80 100
HW is fast, but SW is a Bottleneck

64-byte TCP Echo:

- HW Limit
- Linux
- IX

4.8x Gap

8.8x Gap

Requests per Second
IX Closes the SW Performance Gap

64-byte TCP Echo:

- **Microseconds**
  - HW Limit
  - Linux
  - IX

- **Requests per Second**
  - Millions

[Bar charts showing performance metrics for different system architectures.]
Two Contributions

#1: Protection and direct HW access through virtualization

#2: Execution model for low latency and high throughput
Why is SW Slow?

Complex Interface

Code Paths Convoluted by Interrupts and Scheduling

Created by: Arnout Vandecappelle
http://www.linuxfoundation.org/collaborate/workgroups/networking/kernel_flow
Problem: 1980s Software Architecture

• Berkeley sockets, designed for CPU time sharing
• Today’s large-scale datacenter workloads:

  **Hardware: Dense Multicore + 10 GbE (soon 40)**
  - API scalability critical!
  - Gap between compute and RAM -> Cache behavior matters
  - Packet inter-arrival times of 50 ns

  **Scale out access patterns**
  - Fan-in -> Large connection counts, high request rates
  - Fan-out -> Tail latency matters!
Conventional Wisdom

• Bypass the kernel
  – Move TCP to user-space (Onload, mTCP, Sandstorm)
  – Move TCP to hardware (TOE)

• Avoid the connection scalability bottleneck
  – Use datagrams instead of connections (DIY congestion management)
  – Use proxies at the expense of latency

• Replace classic Ethernet
  – Use a lossless fabric (Infiniband)
  – Offload memory access (rDMA)

• Common thread: Give up on systems software
Our Approach

• Bypass the kernel
  – Move TCP to user space (Onload, mTCP, Sandstorm)
  – Move TCP to hardware (TOE)

• Avoid the connection scalability bottleneck
  – Use datagrams instead of connections (DIY congestion)
  – Use proxies at the expense of latency

• Replace classic Ethernet
  – Use a lossless fabric (Infiniband)
  – Offload memory access (rDMA)

• Tackle the problem head on...

Robust Protection Between App and Netstack
Connection Scalability
Commodity 10Gb Ethernet
Separation of Control and Data Plane

- Userspace
  - CP
  - DP
  - DP

- Kernelspace
  - Host
    - Kernel
  - C
    - C
    - C
    - C
    - C
    - C
Separation of Control and Data Plane

Userspace

Kernelspace

Host Kernel

CP

DP

DP

RX
TX

RX
TX

RX
TX

RX
TX

C

C

C

C

C

C
Separation of Control and Data Plane

IX CP

IX DP

IX DP

Ring 3

Guest Ring 0

Host Kernel

RX TX

RX TX

RX TX

RX TX

C

C

C

C

C

C

C
Separation of Control and Data Plane

Ring 3
IX CP

Guest Ring 0
IX DP
IX DP

Host Ring 0
Linux kernel
Dune
RX TX
RX TX
RX TX
RX TX

C C C C C
Separation of Control and Data Plane

Ring 3
IX CP

Guest
Ring 0
IX
libIX
Memcached
HTTPd

Host
Ring 0
Linux kernel
Dune
RX
TX
RX
TX
RX
TX
RX
TX
C
C
C
C
C
C
The IX Execution Pipeline

Event-Driven App

libIX

RX

TX

TCP/IP

FIFO

RX

TX

Ring 3

Event Conditions

Guest Ring 0

TCP/IP

Ring 3

Batched Syscalls

TCP/IP

Timer

RX

TX

1

2

3

4

5

6
Design (1): Run to Completion

Improves Data-Cache Locality
Removes Scheduling Unpredictably
Design (2): Adaptive Batching

Event-Driven App

LibIX

Event Conditions

Batched Syscalls

TCP/IP

RX FIFO

Adaptive Batch Calculation

Improves Instruction-Cache Locality and Prefetching
See the Paper for more Details

• Design (3): Flow consistent hashing
  – Synchronization & coherence free operation
• Design (4): Native zero-copy API
  – Flow control exposed to application
• Libix: Libevent-like event-based programming
• IX prototype implementation
  – Dune, DPDK, LWIP, ~40K SLOC of kernel code
Evaluation

- Comparison IX to Linux and mTCP [NSDI ’14]
- TCP microbenchmarks and Memcached

![Diagram showing network configuration with IX, 10GbE Switch, and 1x10GbE to 4x10GbE connections with L3+L4 bond, and 25 Linux hosts.](image)
TCP Netpipe

- ½ Bandwidth @ 20 KB
- ½ Bandwidth @ 135 KB
- 5.7 us ½ RTT

Goodput (Gbps) vs Message Size (KB)

- IX-IX
- Linux-Linux
- mTCP-mTCP
TCP Echo: Multicore Scalability for Short Connections

![Graph showing the scalability of different modes of operation over the number of CPU cores.](image-url)

- **IX 10GbE**
- **IX 4x10GbE**
- **Linux 10GbE**
- **Linux 4x10GbE**
- **mTCP 10GbE**

**Saturates 1x10GbE**

- The graph demonstrates the scalability of different network configurations with increasing CPU cores.
- The IX 10GbE shows a steady increase in messages per second across all CPU core counts.
- The Linux 10GbE configuration reaches saturation at around 2 CPU cores.
- The IX 4x10GbE and mTCP 10GbE configurations show similar trends but with higher capacity.

**Notes:**
- The x-axis represents the number of CPU cores.
- The y-axis represents messages/sec (x 10^6).
- The graph highlights the performance differences and saturation points across various network configurations.
Connection Scalability

Messages/sec (x 10^6)

Connection Count (log scale)

IX-40Gbps
IX-10Gbps
Linux-40Gbps
Linux-10Gbps

~10,000 Connections Limited by L3
Memcached over TCP

Latency (µs)

Latency% 6x%Less%Tail%Latency% With%IX%clients%

3.6x More RPS

2x Less Tail Latency

USR: Throughput (RPS x 10³)

IX (p99)  IX (avg)

Linux (p99)  Linux (avg)
IX Conclusion

• A protected dataplane OS for datacenter applications with an event-driven model and demanding connection scalability requirements
• Efficient access to HW, without sacrificing security, through virtualization
• High throughput and low latency enabled by a dataplane execution model