I/O is faster than the OS

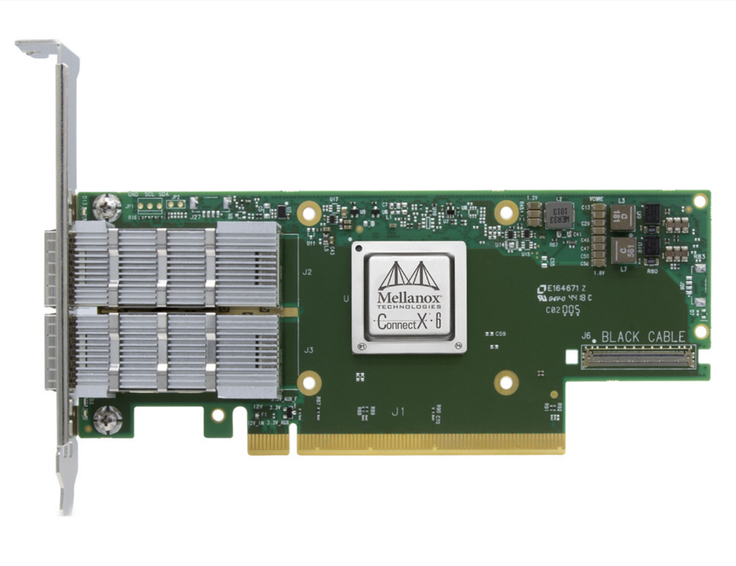
July 22, 2020

Nicolai Fischer

Friedrich-Alexander-Universität Erlangen-Nürnberg

Introduction

Introduction



[2]

Introduction

- I/O Devices are getting faster (200GbE NICs)
- CPU core counts increase, per core performance does not
- New smarter hardware can perform previous kernel tasks
- Legacy OS abstractions do no longer scale

Introduction

Legacy POSIX Abstractions

Legacy POSIX Abstractions

Portable Operating System Interface

- In development since over 30 years
- Common API between different UNIX/UNIX-like OSes
- Foundation for many software projects

Legacy POSIX Abstractions

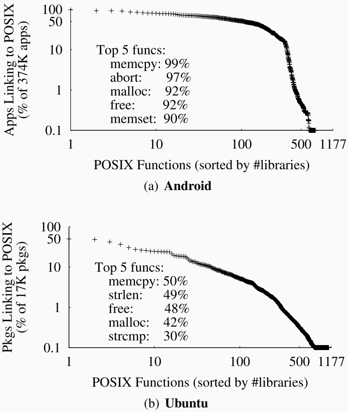
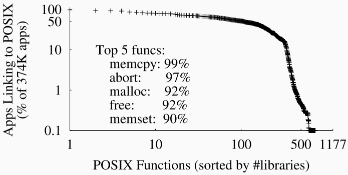


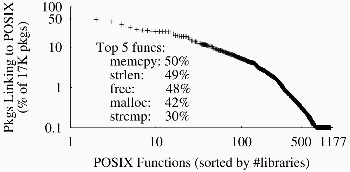
Figure 1: POSIX function linkage (logscale both axis). Static analysis of (a) 374,463 Android apps with native libs and (b) 17,989 Ubuntu packages. Only a fraction of POSIX functions are ever linked.

[1]

Legacy POSIX Abstractions



(a) **Android**



(b) **Ubuntu**

Figure 1: POSIX function linkage (logscale both axis). Static analysis of (a) 374,463 Android apps with native libs and (b) 17,989 Ubuntu packages. Only a fraction of POSIX functions are ever linked.

[1]

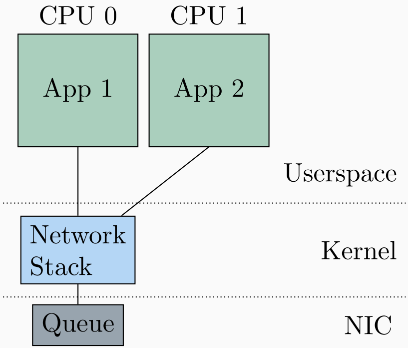
POSIX usage in Ubuntu, Mac OS and Android

- API not fully implemented
- Lack of proper GPU and async I/O support
- System-call intensive API
- Security risks

Network Stack

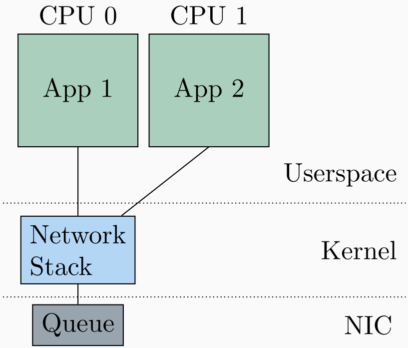
Legacy Network Stack on Linux

Legacy Network Stack on Linux



- Kernel multiplexes devices
- Packets pass through kernel network stack

Legacy Network Stack on Linux



- Kernel multiplexes devices
- Packets pass through kernel network stack

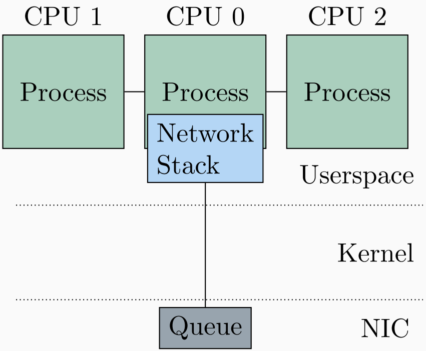
Problems

- Per-packet memory allocation
- Copy between kernel and userspace
- Single CPU core is not fast enough

Network Stack

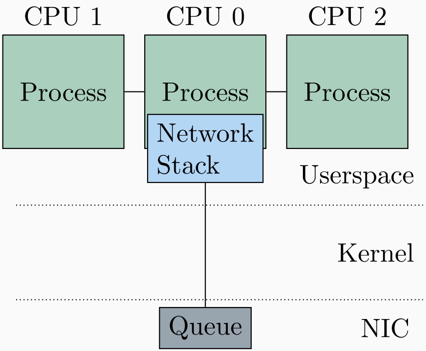
Kernel bypass with DPDK

Kernel bypass with DPDK



- Circumvent kernel entirely
- Exclusive access to hardware
- Dedicate CPU cores to network processing
- Low latency

Kernel bypass with DPDK



- Circumvent kernel entirely
- Exclusive access to hardware
- Dedicate CPU cores to network processing
- Low latency

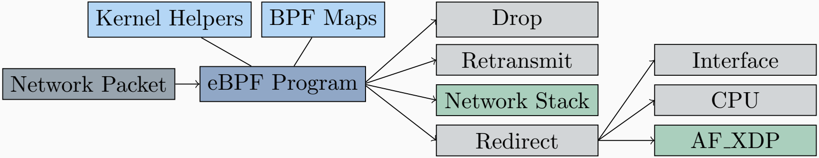
Downsides

- Devices are unavailable to remaining system
- Userspace drivers
- Tighter coupling to specific hardware

Network Stack

eXpress Data Path

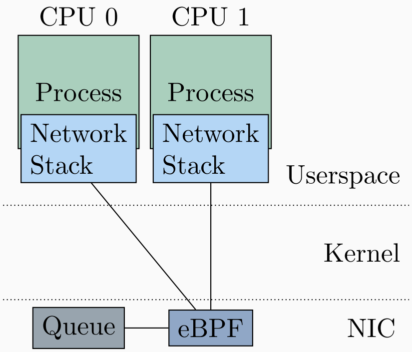
XDP on Linux



eBPF Programs

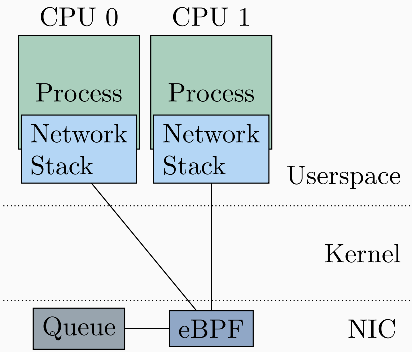
- Extended Berkely Packet Filter
- Access to the entire network packet and metadata
- Exit code determines route of packet
- Stateless between executions

XDP with Hardware Offload



- Kernel removed from the data-plane
- Zerocopy into userspace
- Device usable for legacy applications

XDP with Hardware Offload



- Kernel removed from the data-plane
- Zerocopy into userspace
- Device usable for legacy applications

Caveats

- Special hardware required (Smart NICs)
- Driver support needed

Parakernel

Partition Devices

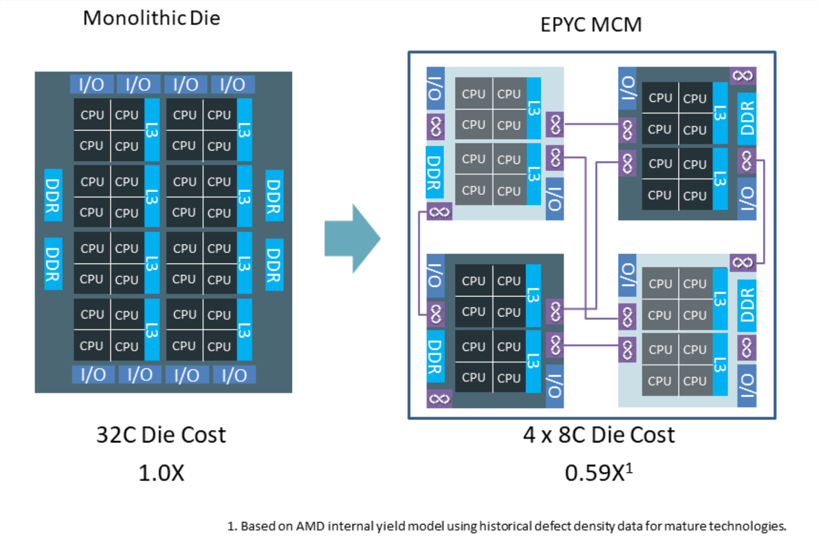
Partition Devices

- Eliminate kernel from data-plane
- Partition I/O devices that support it (eBPF)
- Only multiplex legacy hardware (SATA, Timers)

Parakernel

Multikernel Architecture

Multikernel Architechture



[3]

Multikernel Architechture

Multikernel Architecture

- Inspired by distributed systems
- Relatively independent OS instances on each CPU core
- Global state gets explicitly replicated
- Message passing instead of shared memory
- Highly scalable design

Parakernel

Eliminate Legacy Abstractions

Eliminate Legacy Abstractions

Asynchronous Kernel API

- No kernel threads, only Processes
- No blocking system-calls
- Application controlled concurrency (Coroutines, Fibers)
- POSIX compatibility through userspace libraries

Conclusion

Conclusion

Parakernel

- Very scalable
- Likely more secure
- Async design pattern supported by libraries, managed runtimes and modern languages
- POSIX compatibility in userspace

References

References

Atlidakis, V., Andrus, J., Geambasu, R., Mitropoulos, D., and Nieh, J.
POSIX abstractions in modern operating systems: the old, the new, and the missing.
In *Proceedings of the Eleventh European Conference on Computer Systems* (London, United Kingdom, Apr. 2016), EuroSys '16, Association for Computing Machinery, pp. 1–17.

NVIDIA Corporation.
ConnectX®-6 EN Single/Dual-Port Adapter Supporting 200Gb/s Ethernet, 2020.
Library Catalog: www.mellanox.com.

TIRAS Research, and AMD.
AMD Optimizes EPYC Memory with NUMA.
Tech. rep., 2018.

Appendix

Zero-copy Architecture

Zero-copy Architecture

Motivation

- Avoid copying data between user and kernelspace

Zero-copy Archticture

- Share buffers between user and kernelspace
- Devices read/write data directly from/into them
- *O_DIRECT* flag in Linux

O_DIRECT

- Needs filesystem support
- Buffer alignment dependent on filesystem
- Circumvents filesystem cache

Appendix

io_uring

io_uring

Goals

- Zero-copy disk I/O
- Reduce context switches
- asynchronous API

io_uring

- Ringbuffers shared between kernel and userspace
- Queue multiple I/O operations
- Use system-call to execute operations
- Alternative polling mode without system-calls

Appendix

Context-Switches are Expensive

Context-Switches are Expensive

Meltdown

- Trick CPU into executing specific instructions out-of-order/speculative
- Raise an exception
- CPU fails to wipe changed state correctly
- Use cache side-channels to extract arbitrary data

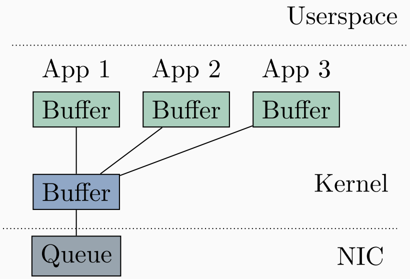
Context Switches

- More exploits: Spectre, Fallout, etc
- Software mitigation slow 2% to 11%
- Hardware solution partially available, but also expensive
- Context switches have to be avoided whenever possible

Appendix

Multi-Queue Devices

Multiplexing



I/O Devices

- Ring buffer of DMA-descriptors
- Write data into DRAM
- Interrupt informs OS of new data
- Kernel multiplexes device for applications

Multiplexing

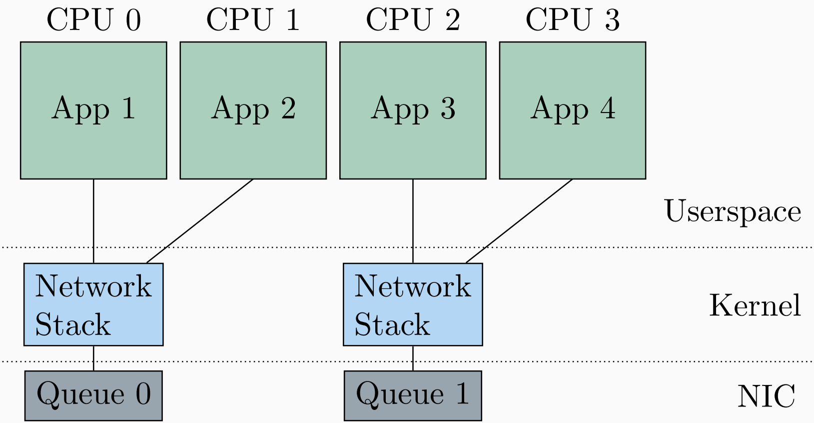
Problems

- One CPU core is not fast enough
- Copying data in memory is too slow

Optimizations

- Write data directly into LLC
- Multi-Queue Devices
  - Up to 1535 queue pairs on Intel X710
  - Up to 65535 queue pairs in NVMe specification

Multi-Queue NICs



- Queues are processed by multiple CPU cores