Can SmartNICs help accelerate distributed systems?

Arvind Krishnamurthy
Univ. of Washington

Programmable NICs

- Renewed interest in NICs that allow for customized per-packet processing
- Many NICs equipped with multicores or packet processing pipelines
  - E.g., Mellanox BlueField, Marvell LiquidIO, Pensando, Fungible, Intel IPU, etc.
- Primarily used to accelerate networking & storage
  - Supports offloading of fixed functions used in protocols

Can we use programmable NICs to accelerate general distributed applications?

Outline

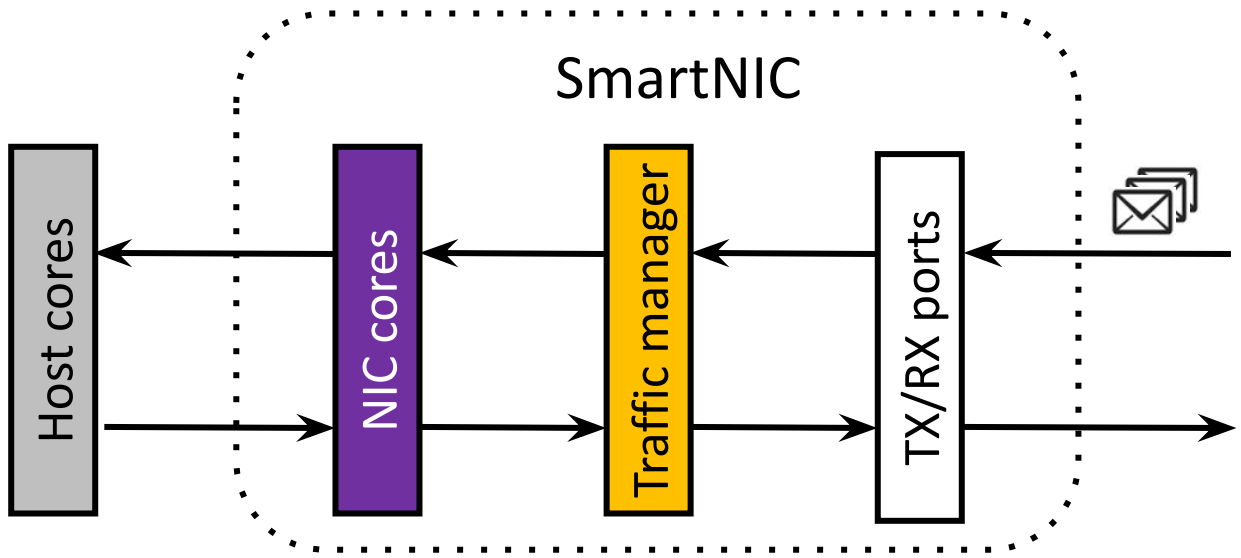
- Hardware model
- Case studies of accelerating distributed systems
- Opportunities & challenges
- Discussion & future agenda

Diversity of Compute Elements

- PISA packet processing on the NIC
  - E.g., Pensando, Netronome
- General purpose computation on packets (multi-core, FPGA, etc.)
  - Further classified into:
    - On-path computing
    - Off-path computing

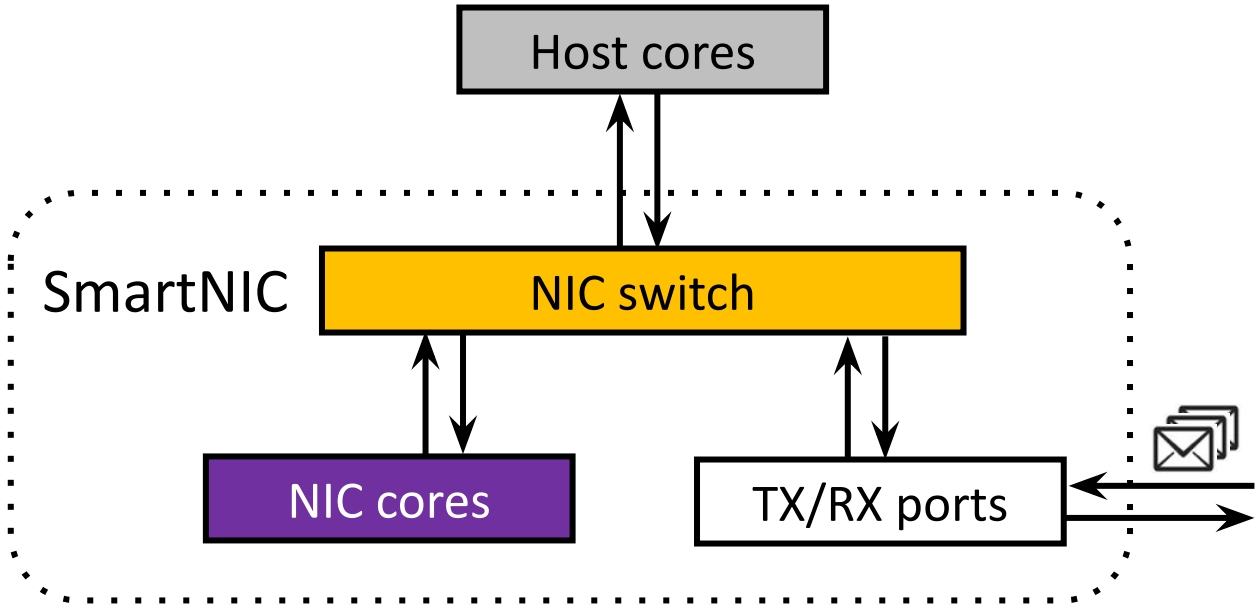
On-path SmartNICs

NIC cores handle all traffic on both the send & receive paths



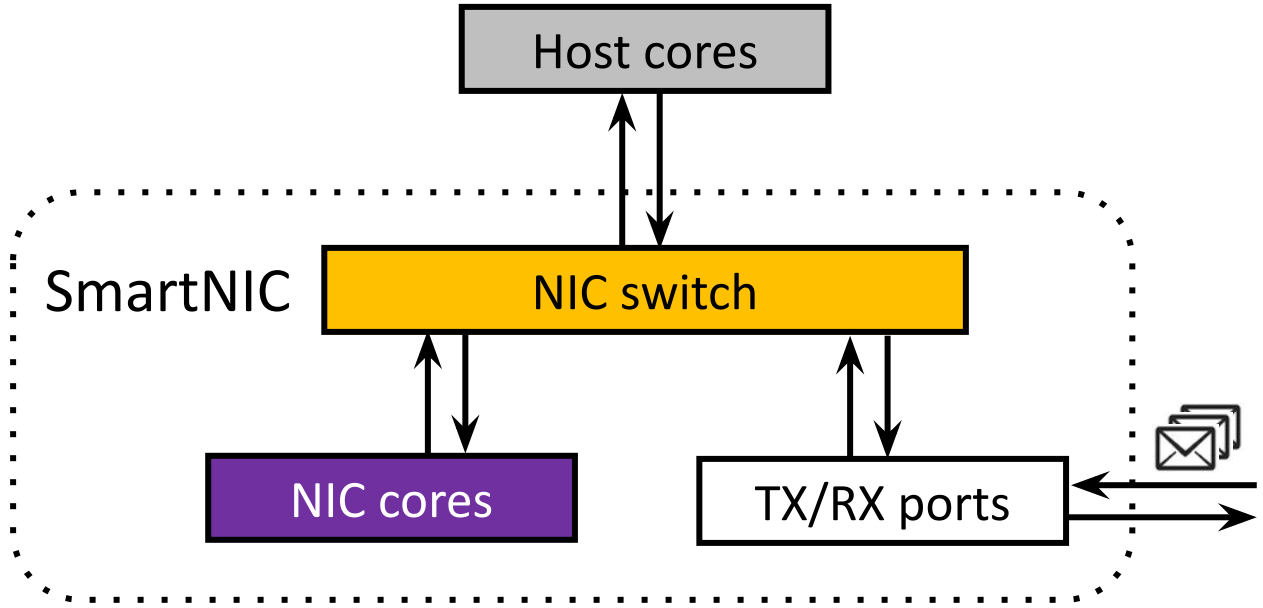
Tight integration of computing and communication

Off-path SmartNICs



- Programmable NIC switch (P4-like) enables targeted delivery

Off-path SmartNICs



- Programmable NIC switch (P4-like) enables targeted delivery
- Host traffic does not consume NIC cores
- Communication support is less integrated

Hardware Convergence

- Many off-path SmartNICs are now embracing on-path compute cores (e.g., Mellanox BlueField 3)
- On-path SmartNICs are also including programmable switching capabilities (e.g., Marvell Octeon 10)
- Question: how do we make use of these diverse computing elements on the SmartNIC?

Use case studies to drive the design of offloading frameworks and new NIC designs

#1: Deploying Load Balancers on SmartNICs

- Load balancers are a crucial part of datacenters (e.g., L4 and L7)
  - Now also being used within service meshes
- Traditional networking application that seems appropriate for SmartNICs
  - NIC cores can perform general-purpose computing, while programmable NIC switch can perform specialized packet processing at line rates
  - If successful, can form a cheap (low-cost) load-balancing substrate within datacenters

Challenges & Opportunities

- NIC cores can perform general computation but aren't powerful
- Characterized the performance of Nginx running on host and the NIC

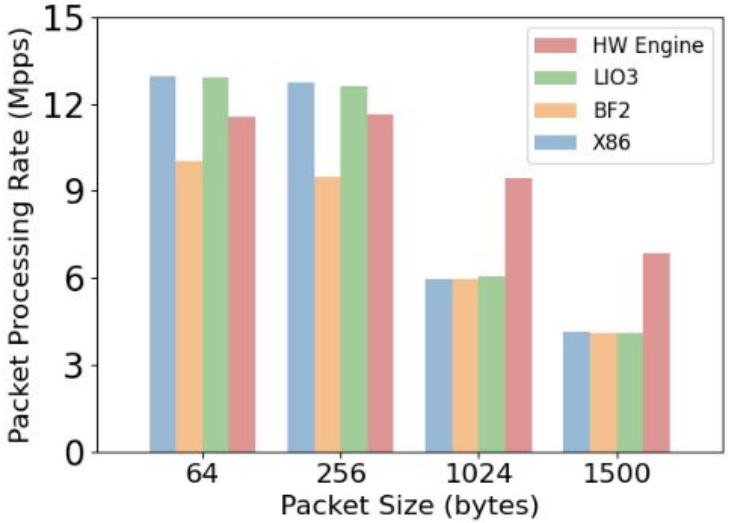
Response size		LIO3	BlueField-2	x86
1KB	1 Core	0.17	0.12	0.31
	8 Core	1.01	0.51	2.36
1MB	1 Core	5.08	3.07	10.84
	8 Core	4.92	10.47	40.86

Nginx Performance across different platforms (Gbps)

NIC-side computing is 2-8x slower executing traditional server-based apps

Challenges & Opportunities

- Disparity in packet processing is much less, especially if we use hardware packet processing engine (i.e., NIC switch)
- Characterized the performance of MAC-Swap on host and the NIC



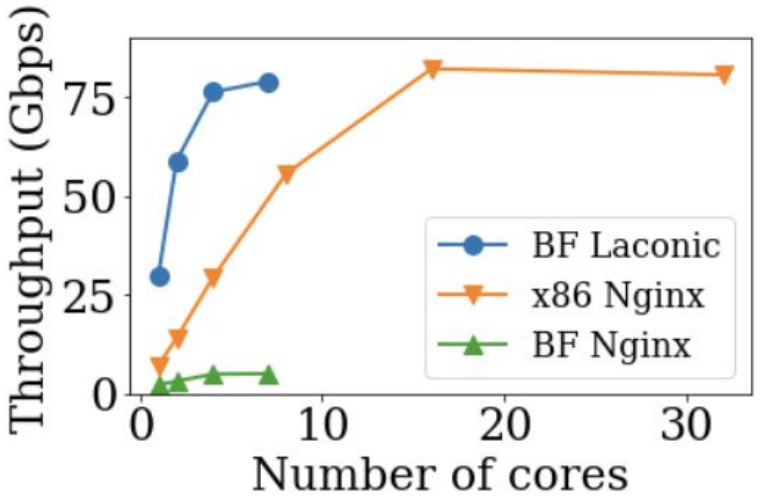
NIC-side packet processing can be faster than CPU due to the hardware engines

Laconic: Streamlined Load Balancers for SmartNICs

- Designed an L7 load balancer that is streamlined for a SmartNIC
- Key ideas:
  - SmartNIC does not run a full transport stack, but merely runs a lightweight packet rewriting/forwarding engine
  - Inspects and processes only the “control” messages (e.g., resource requests that have to be vetted for policy and routed appropriately)
  - Bulk of the communication are simple packet rewrites; relies on the end-hosts to handle losses & congestion control
  - Simple packet rewrites can be accelerated using the hardware flow engine

Performance Benefits on BlueField-2

- Real-world workloads can benefit from Laconic



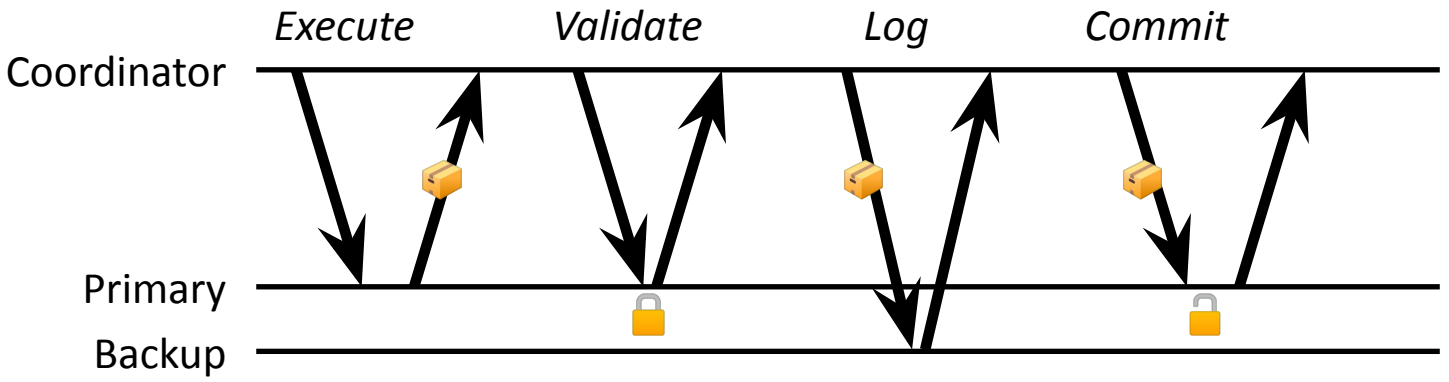
- Performance currently limited by updates to hardware flow engine

Batch Size	Insert Latency	Delete Latency
1	305.40 us	57.49 us
2	100.48 us	24.48 us
8	38.72 us	19.42 us
16	25.39 us	18.08 us

#2: Distributed Transactions in the Datacenter

Our target: distributed ACID transactions on a replicated, in-memory database

Common approach is **Optimistic Concurrency Control + replication**



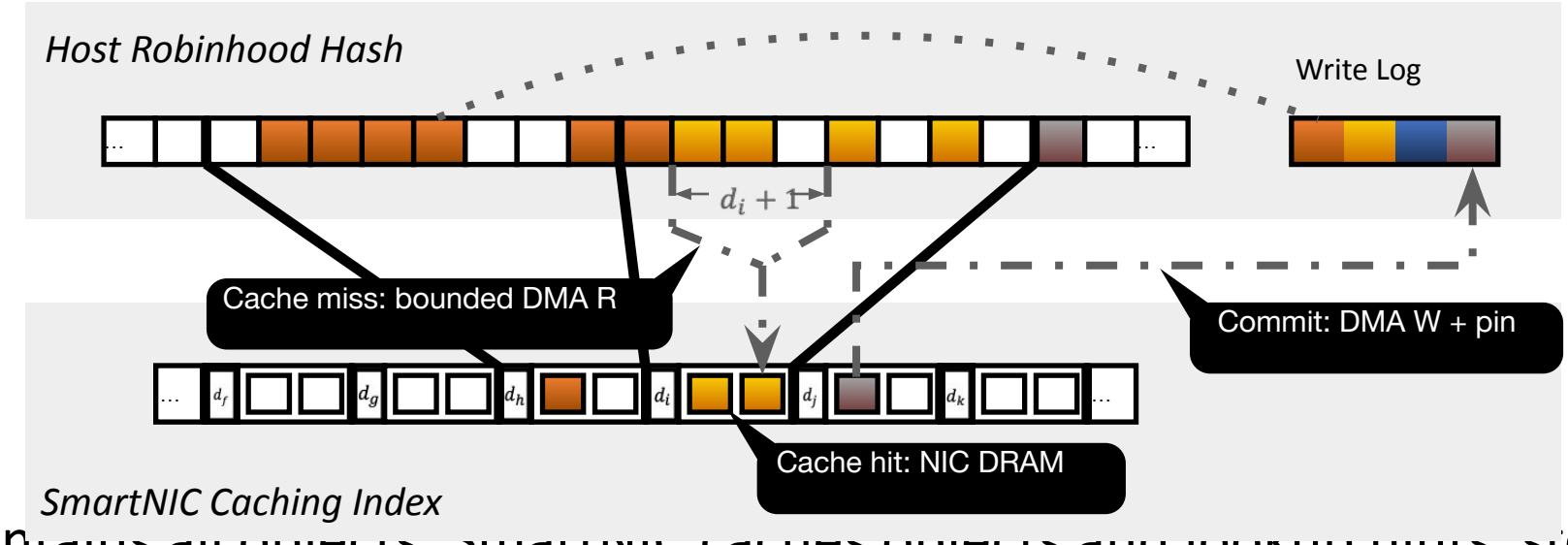
Viability depends on efficient remote operations → hardware acceleration

Xenic

Distributed transactions accelerated with on-path SmartNICs

1. Co-designed data store, spread across NIC + host DRAM
   - Minimize lookup overhead, utilizing NIC's on-board memory
2. SmartNIC function shipping
   - Offload transaction logic to avoid PCIe crossings
3. Multi-hop OCC protocols
   - Reduce communication with optimized message patterns

Xenic: Robinhood Data Store



Host DRAM contains all objects, SmartNIC caches objects and lookup hints, stores locks

Critical path accesses: NIC memory hit or DMA read, DMA log write

- Lookup hints limit DMA cost for cache misses
- OCC + pinning ensure NIC/host consistency

Xenic: SmartNIC Function Shipping

SmartNIC cores act as a function shipping target

Shipping execution can reduce overhead, depending on application-level computation and state requirements

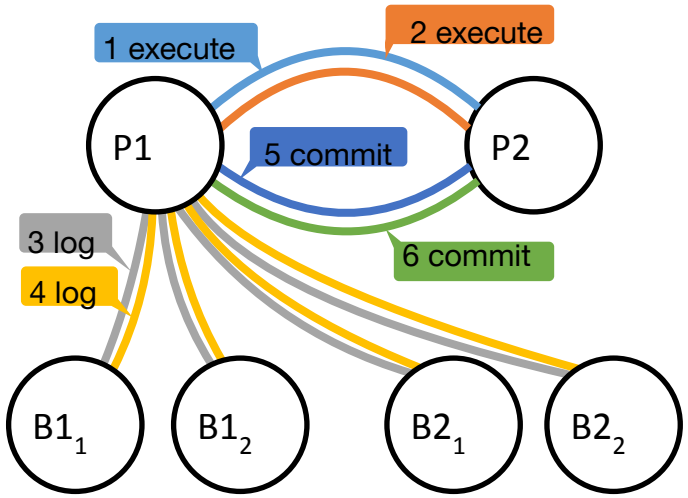
```
int smallbank_exec(reads, writes, AMOUNT) {
        writes[0].val = reads[0].val + AMOUNT;
        writes[1].val = reads[1].val - AMOUNT;
        return START_COMMIT;
}
```

```
fn = smallbank_exec, AMOUNT = 5
```

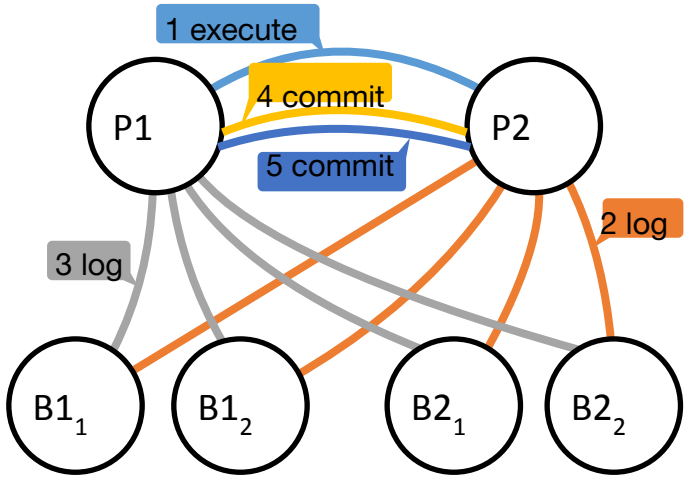
Xenic: Multi-hop OCC Protocols

Xenic also ships execution to remote SmartNICs

Multi-hop NIC-to-NIC communication increases network efficiency



4 → 3 message delays to commit

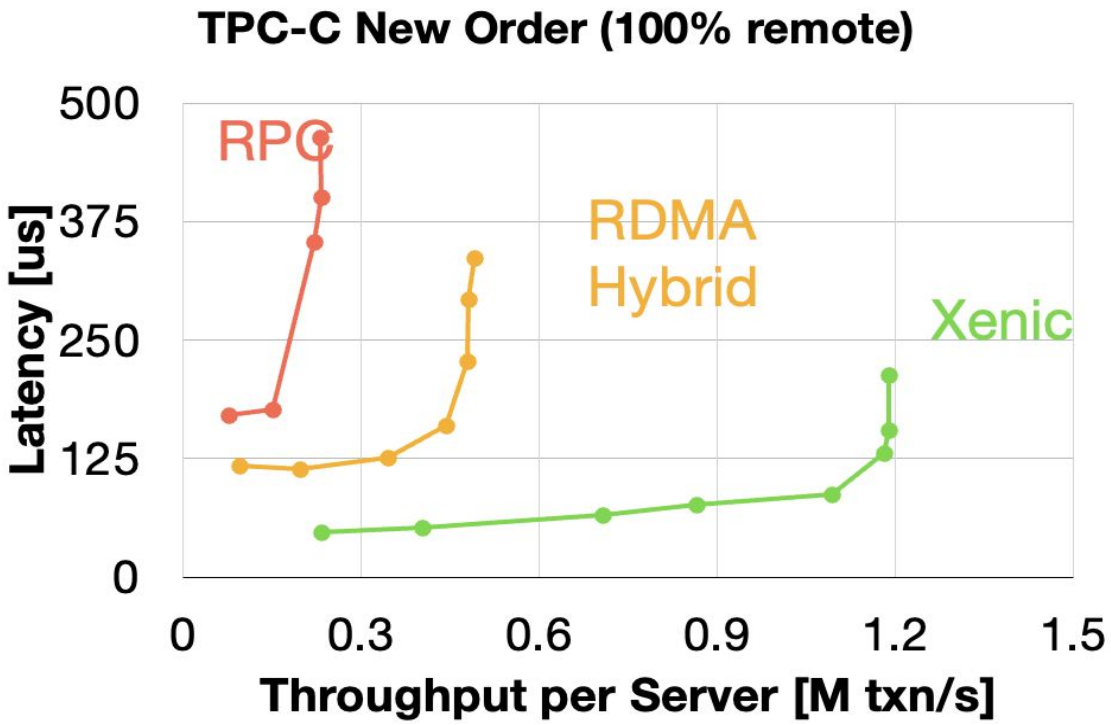


Local write (P1) + remote write (P2)
Execution at coordinator P1

Local write (P1) + remote write (P2)
Shipped to remote primary P2

Evaluation: Benchmark Results

Xenic (2x50GbE LiquidIO 3) **versus RDMA systems** (100GbE CX5)



Better latency & throughput than RPC, RDMA, hybrid designs

SmartNIC Opportunities & Challenges

- Offload CPU operations to SmartNIC, but
  - NIC cores are wimpy
- Perform stateful operations on SmartNIC, but
  - Need to keep state consistent with CPU
  - Limited memory capacity & bandwidth on NIC
- On-path cores: Efficient NIC-to-NIC communication, but
  - Software packet processing means latency overheads
- Off-path cores: Avoids traffic sent to host cores, but
  - Lack of tight integration with the host cores or the NIC pipeline

Discussion & Future Agenda

- Application programmability of SmartNICs is now viable
  - Can offload or accelerate end-host computations
  - Many opportunities but also challenges

- Many interesting research directions:
  - Hardware features that can aid performance & functionality
  - Systems support for shared state & adaptive execution
  - Programming support for application-specific tasks; support both general-purpose computation & packet processing logic

Thank you!