CS 839: Design the Next-Generation Database

Lecture 17: Smart NIC

Xiangyao Yu

3/24/2020

Announcements

Feedback on project proposals will be provided this week

Upcoming deadlines

- Paper submission: Apr. 23
- Peer review: Apr. 23 – Apr. 30
- Presentation: Apr. 28 & 30

Discussion Highlights

Active memory without in-order delivery?

- Assign seq number to each packet and resemble at the receiving side

Active Memory vs.Write Behind Logging?

- Both use “force” instead of “no-force”
- Can be combined (single- vs. multi-versioning)
- Keep data in persistent memory in Active Memory

Other examples of increasing computation to reduce network overhead

- Caching
- Data centric computing (moving computation to data)
- Compression and decompression
- Directory-based cache coherence: unicast vs. multicast

Today's Paper

Offloading Distributed Applications onto SmartNICs using iPipe

Ming Liu
University of Washington

Tianyi Cui
University of Washington

Henry Schuh
University of Washington

Arvind Krishnamurthy
University of Washington

Simon Peter
The University of Texas at Austin

Karan Gupta
Nutanix

Abstract

Emerging Multicore SoC SmartNICs, enclosing rich computing resources (e.g., a multicore processor, onboard DRAM, accelerators, programmable DMA engines), hold the potential to offload generic datacenter server tasks. However, it is unclear how to use a SmartNIC efficiently and maximize the offloading benefits, especially for distributed applications. Towards this end, we characterize four commodity SmartNICs and summarize the offloading performance implications from four perspectives: traffic control, computing capability, onboard memory, and host communication.

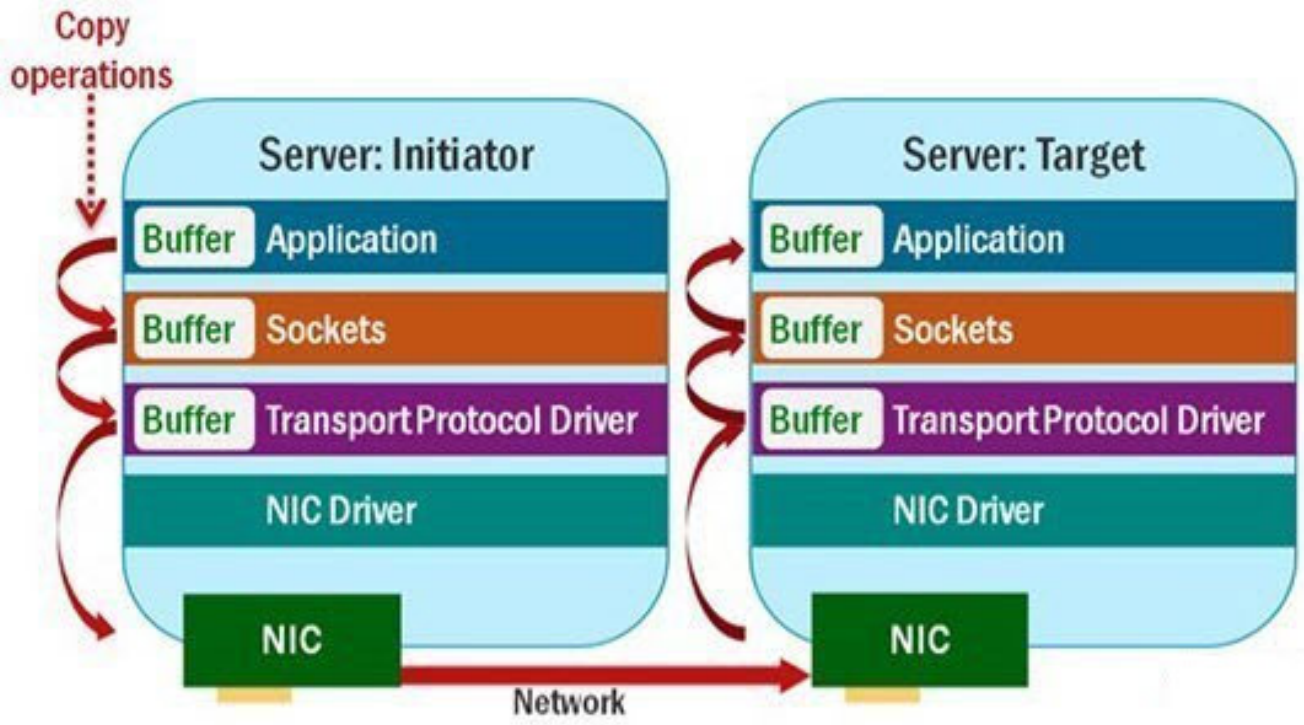
Based on our characterization, we build iPipe, an actor-based

last two years, major network hardware vendors have released different SmartNIC products, such as Mellanox's BlueField [43], Broadcom's Stingray [7], Marvell (Cavium)'s LiquidIO [42], Huawei's IN5500 [24], and Netronome's Agilio [47]. They not only target acceleration of protocol processing (e.g., Open vSwitch [52], TCP offloading, traffic monitoring, and firewall), but also bring a new computing substrate into the data center to expand the server computing capacity at a low cost: SmartNICs usually enclose computing cores with simple microarchitectures that make them cost-effective.

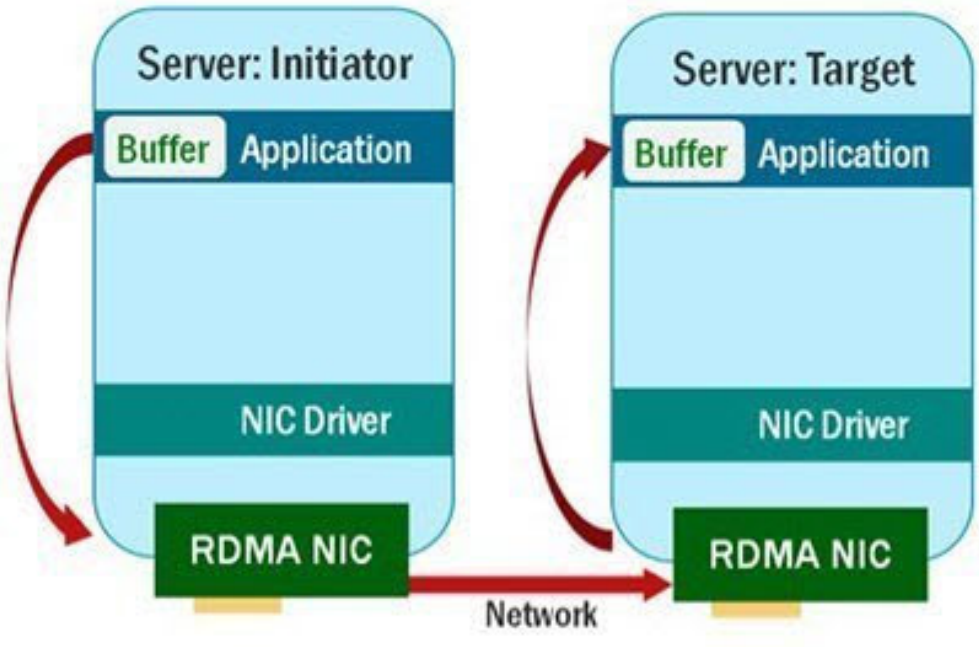
Generally, these SmartNICs comprise a multicore, possibly wimpy, processor (i.e., MIPS/ARM), onboard SRAM/DRAM, packet process-

SIGCOMM 2019

Kernel Bypass

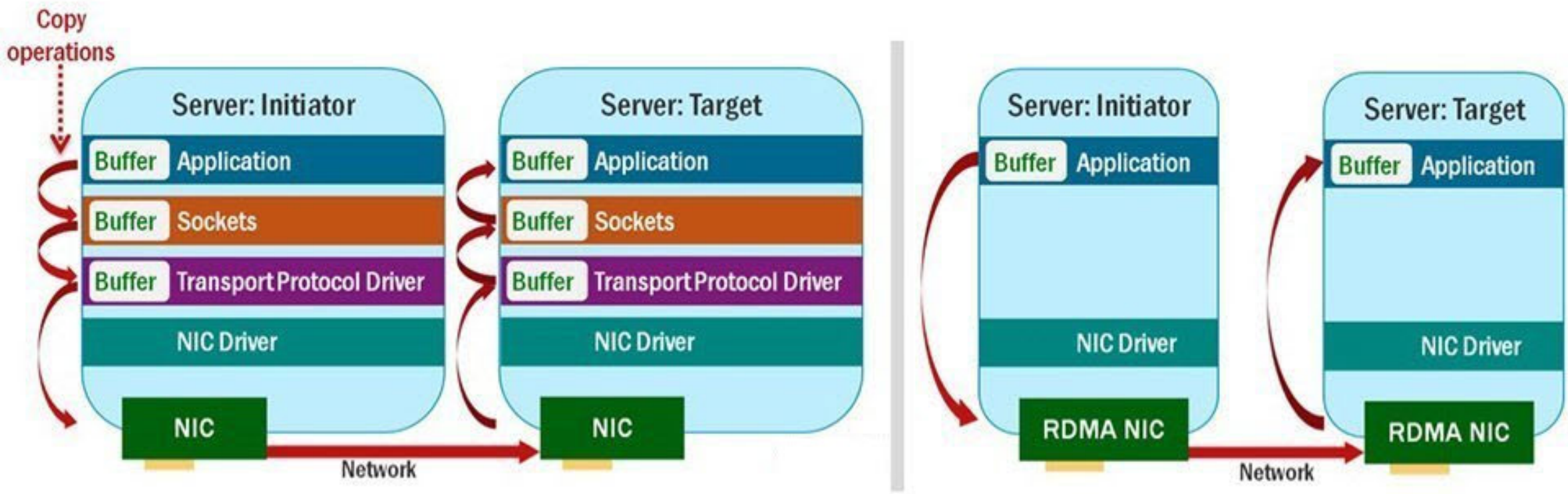


Conventional network stack



Kernel bypass (DPDK and RDMA)

Kernel Bypass



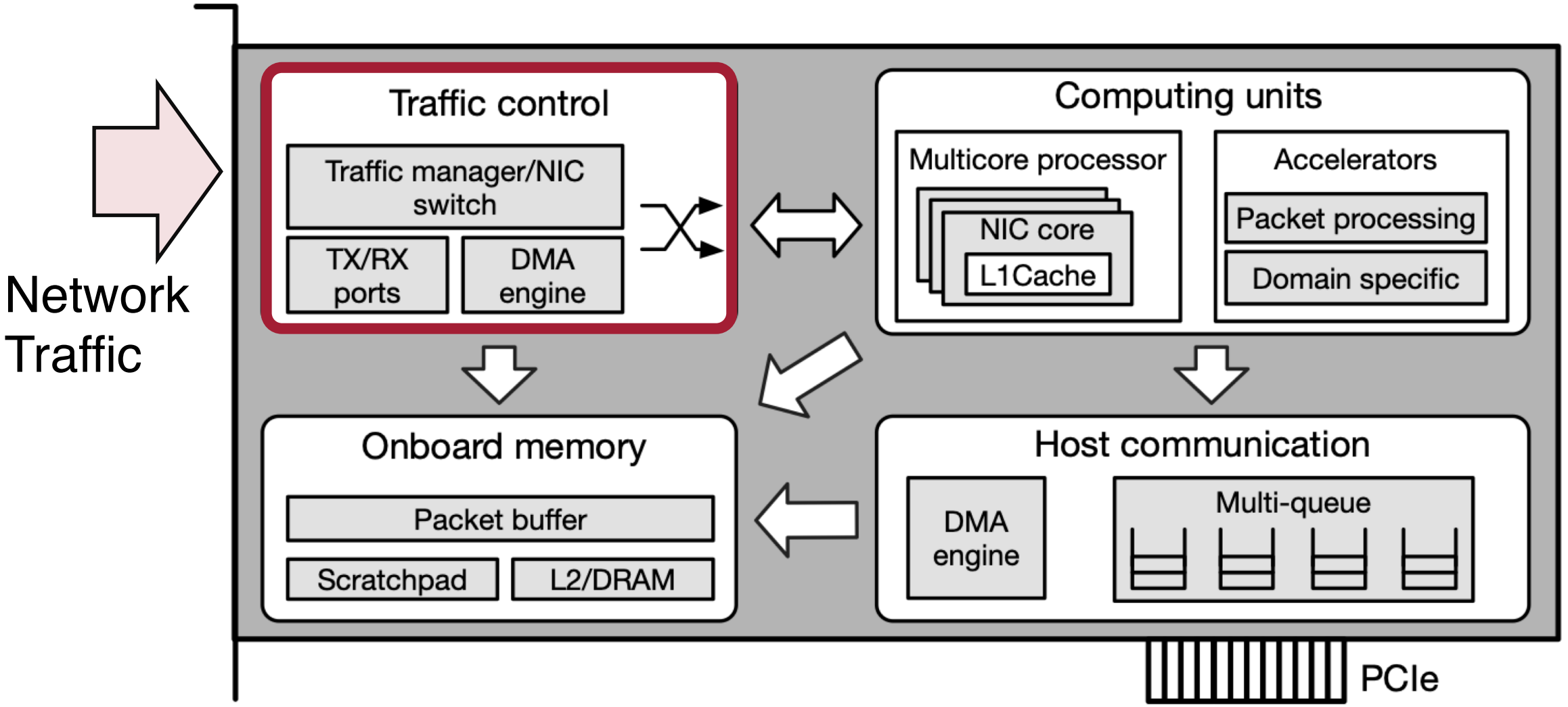
Conventional network stack

Kernel bypass (DPDK and RDMA)

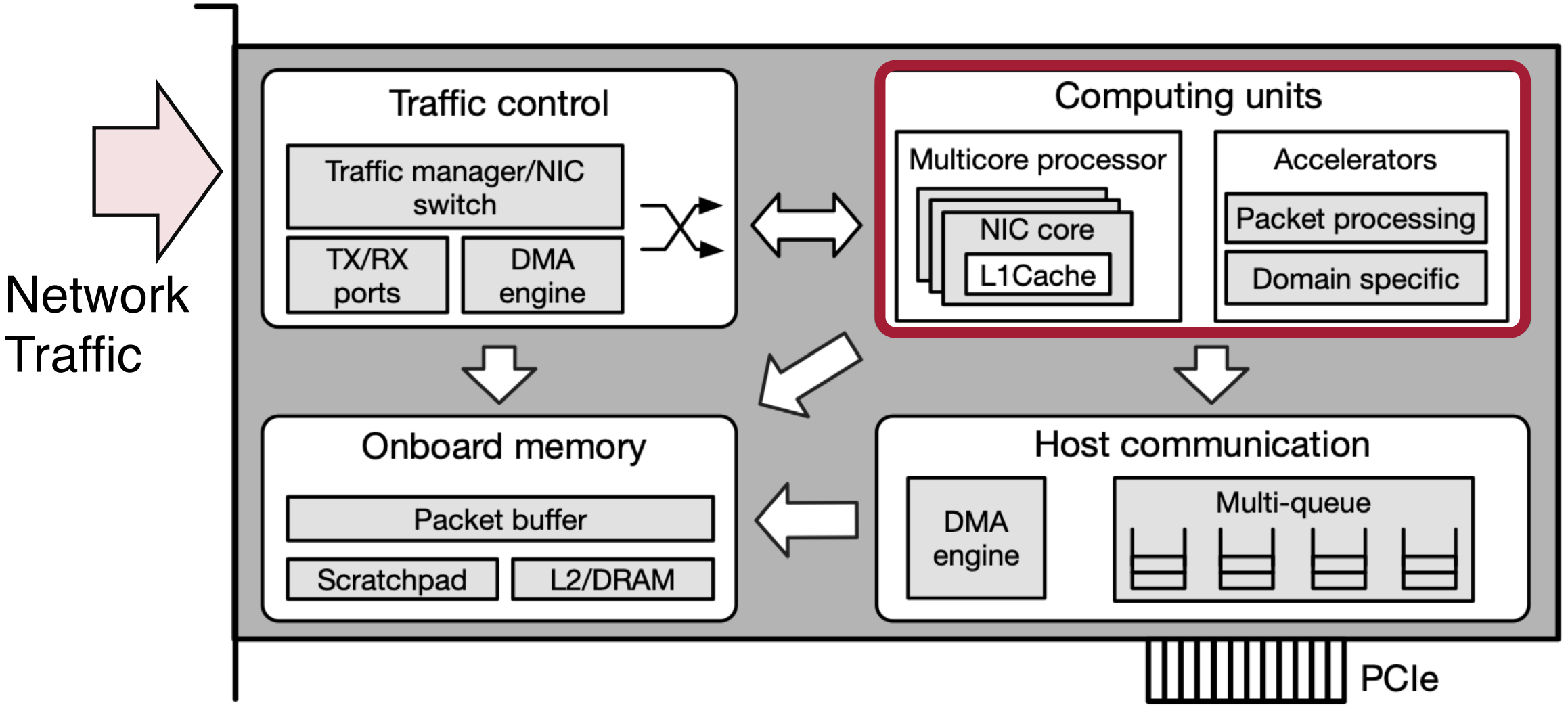
Pushing computation to storage => **Smart SSD**

Pushing computation to network => **Smart NIC**

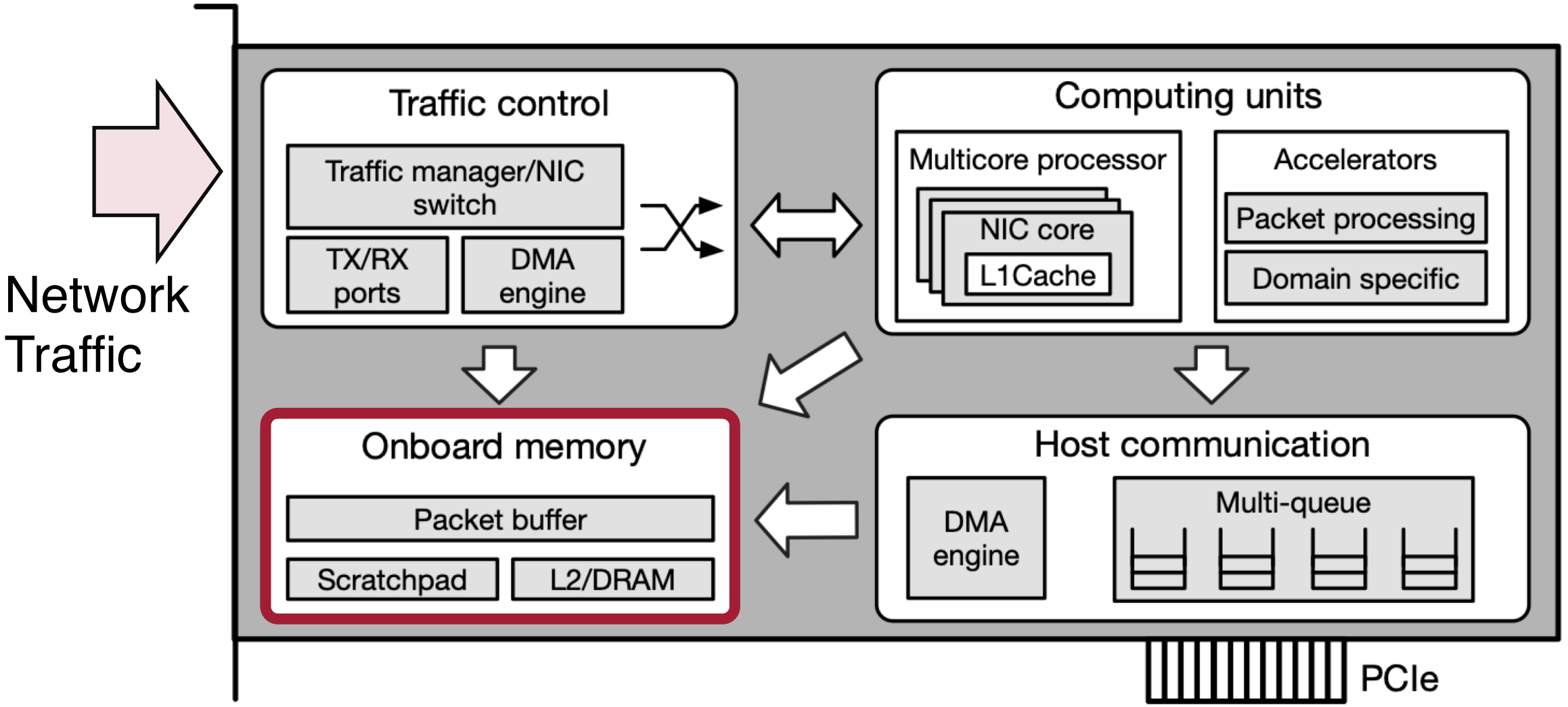
Smart NIC Architecture



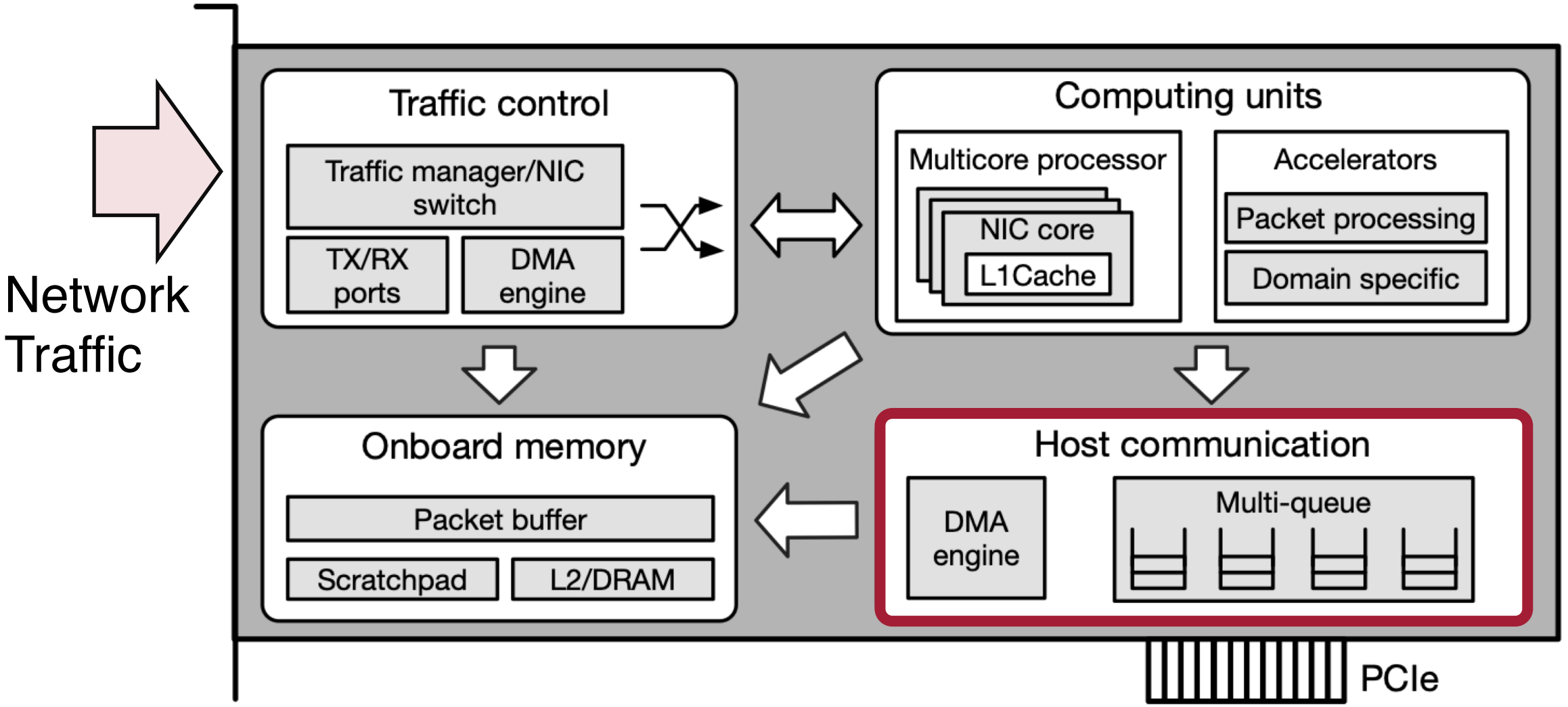
Smart NIC Architecture



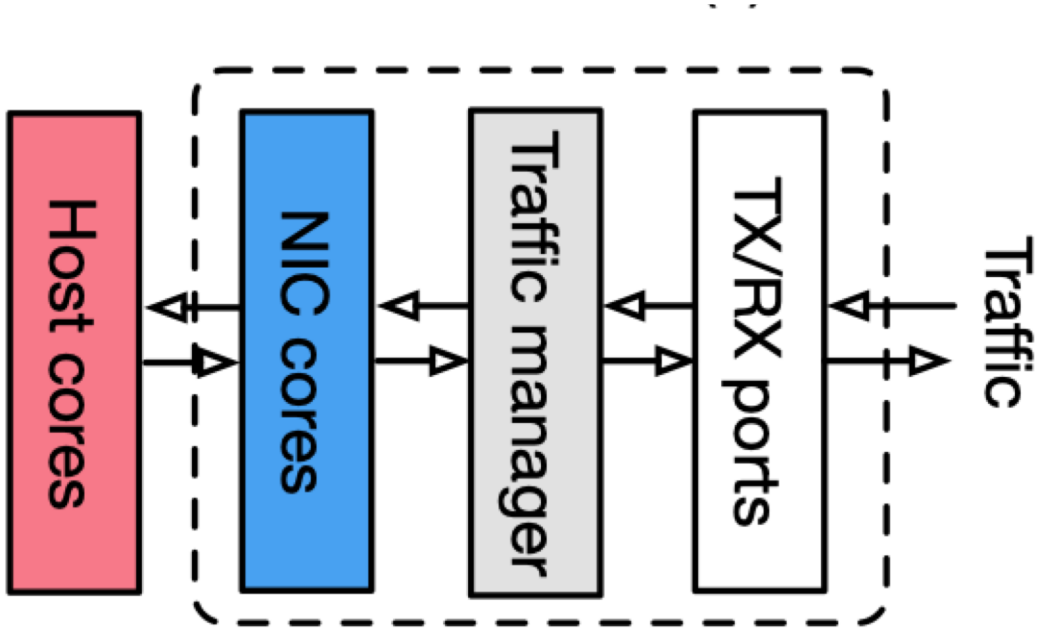
Smart NIC Architecture



Smart NIC Architecture



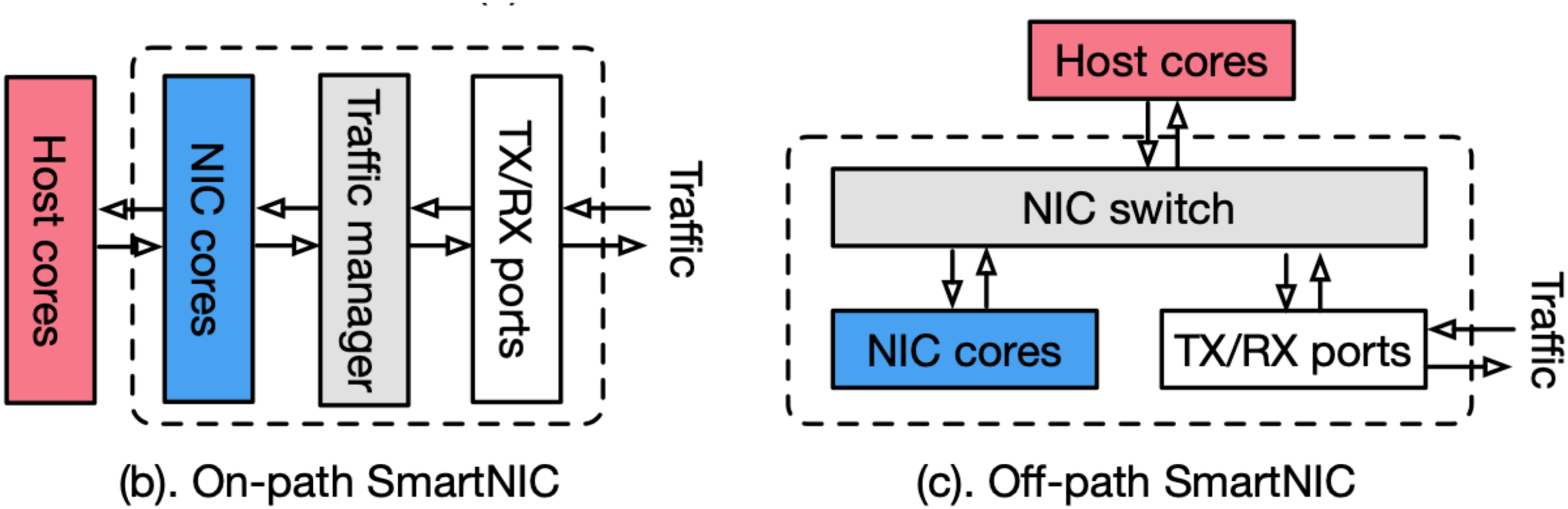
On-path vs. Off-path



(b). On-path SmartNIC

On-path: NIC cores handle all traffic on both send & receive paths

On-path vs. Off-path



(b). On-path SmartNIC

(c). Off-path SmartNIC

On-path: NIC cores handle all traffic on both send & receive paths
Off-path: Host traffic does not consume NIC cores

SmartNIC Specifications

	Vendor	BW	Processor	Deployed SW
LiquidIOII CN2350	Marvell	2X 10GbE	12 cnMIPS core, 1.2GHz	Firmware
LiquidIOII CN2360	Marvell	2X 25GbE	16 cnMIPS core, 1.5GHz	Firmware
BlueField 1M332A	Mellanox	2X 25GbE	8 ARM A72 core, 0.8GHz	Full OS
Stingray PS225	Broadcom	2X 25GbE	8 ARM A72 core, 3.0GHz	Full OS

on-path

off-path

- Low power processor with simple micro-architecture

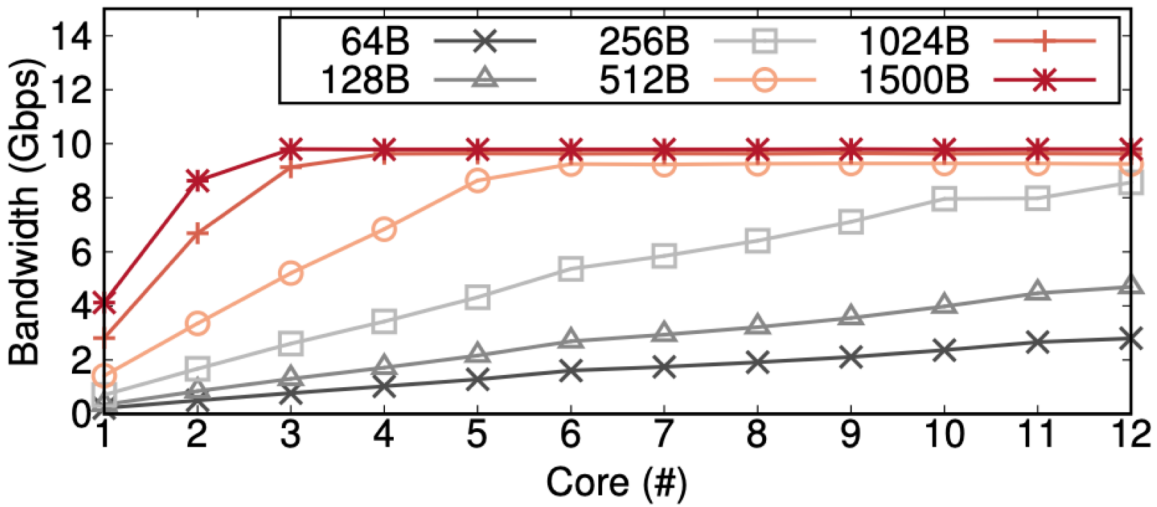
On-Board Memory

	L1 (ns)	L2 (ns)	L3 (ns)	DRAM (ns)
LiquidIOII CNXX	8.3	55.8	N/A	115.0
BlueField 1M332A	5.0	25.6	N/A	132.0
Stingray PS225	1.3	25.1	N/A	85.3
Host Intel server	1.2	6.0	22.4	62.2

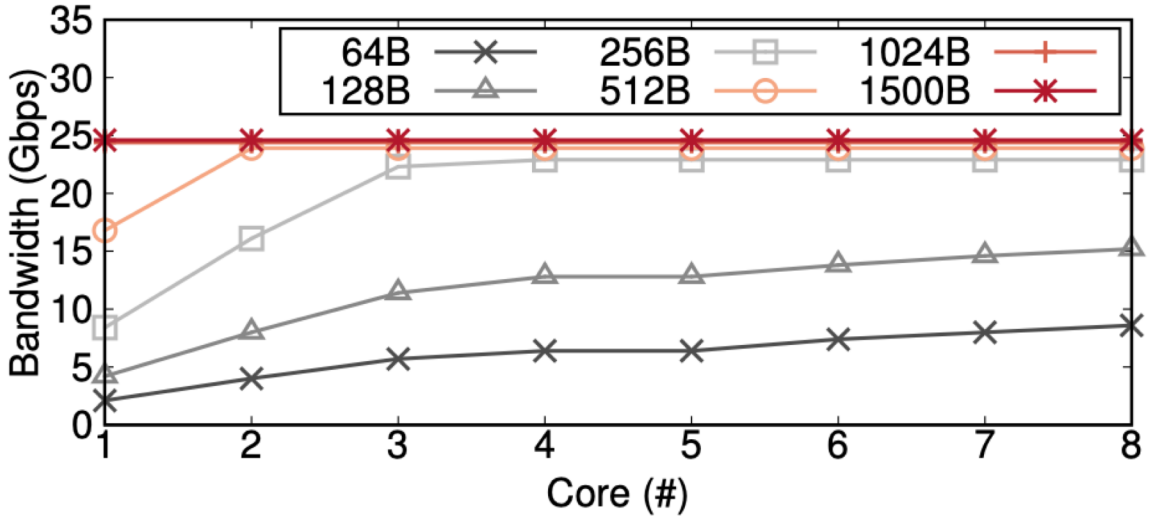
1. Scratchpad/L1
2. Packet Buffer (only for on-path)
   - Onboard SRAM with fast indexing
3. L2 cache
4. NIC local DRAM (4GB – 8GB)
5. Host DRAM (accessed through DMA)

Performance Characterization

Bandwidth vs. Core Count



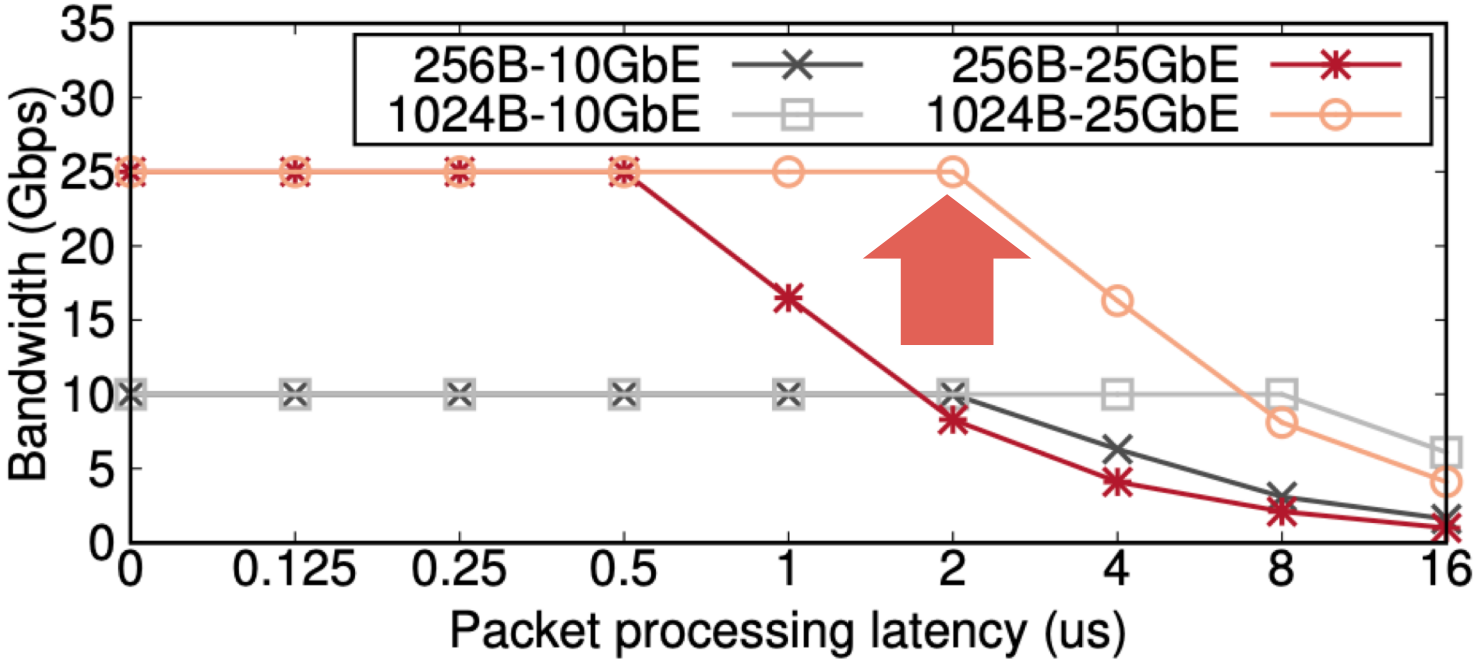
10 GbE LiquidIO II CN2350



25 GbE Stingray PS225

- Echo server
- Packet transmission through a Smart NIC core incurs nontrivial cost
- Packet size distribution impacts availability of computing cycles

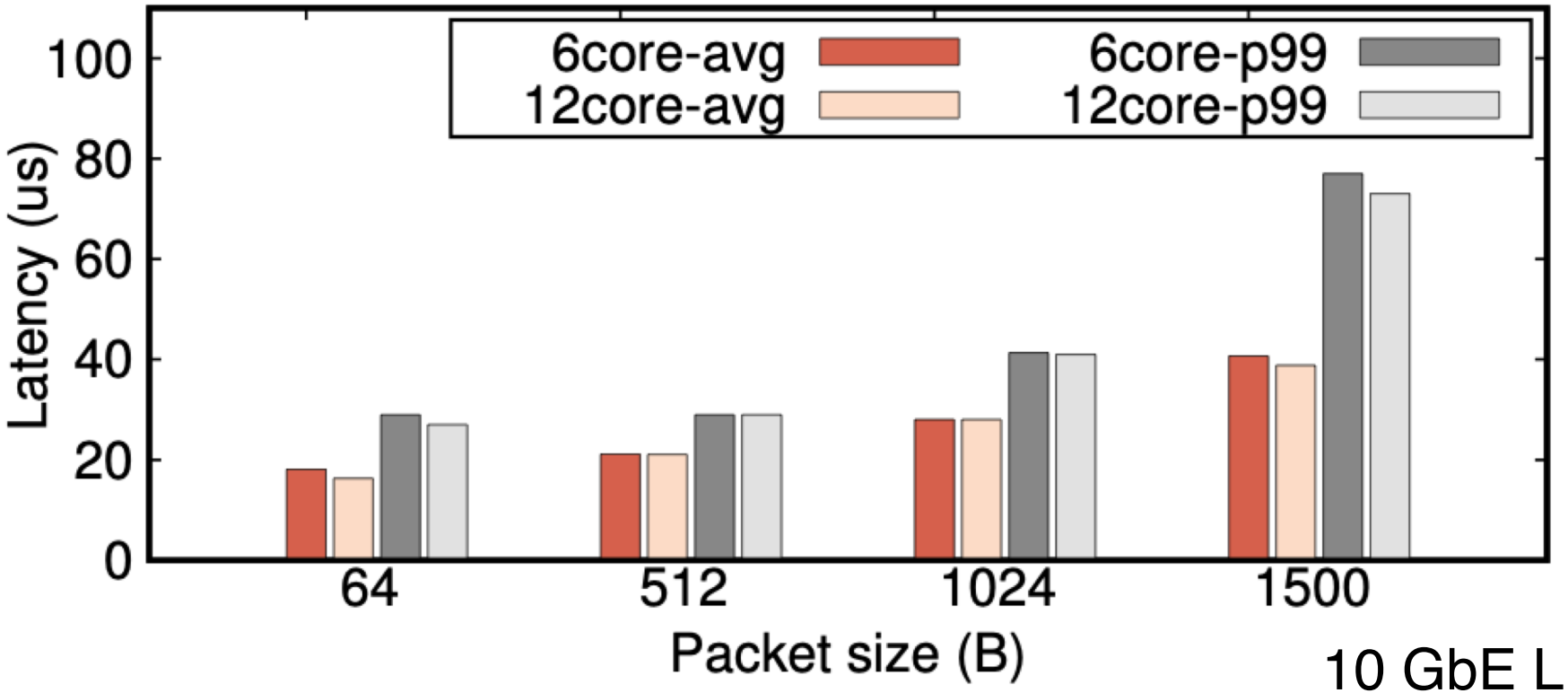
Bandwidth vs. Packet Processing Cost



10 GbE: LiquidIO II CN2350
25 GbE Stingray PS225

- Processing headroom is workload dependent and only allows for execution of tiny tasks

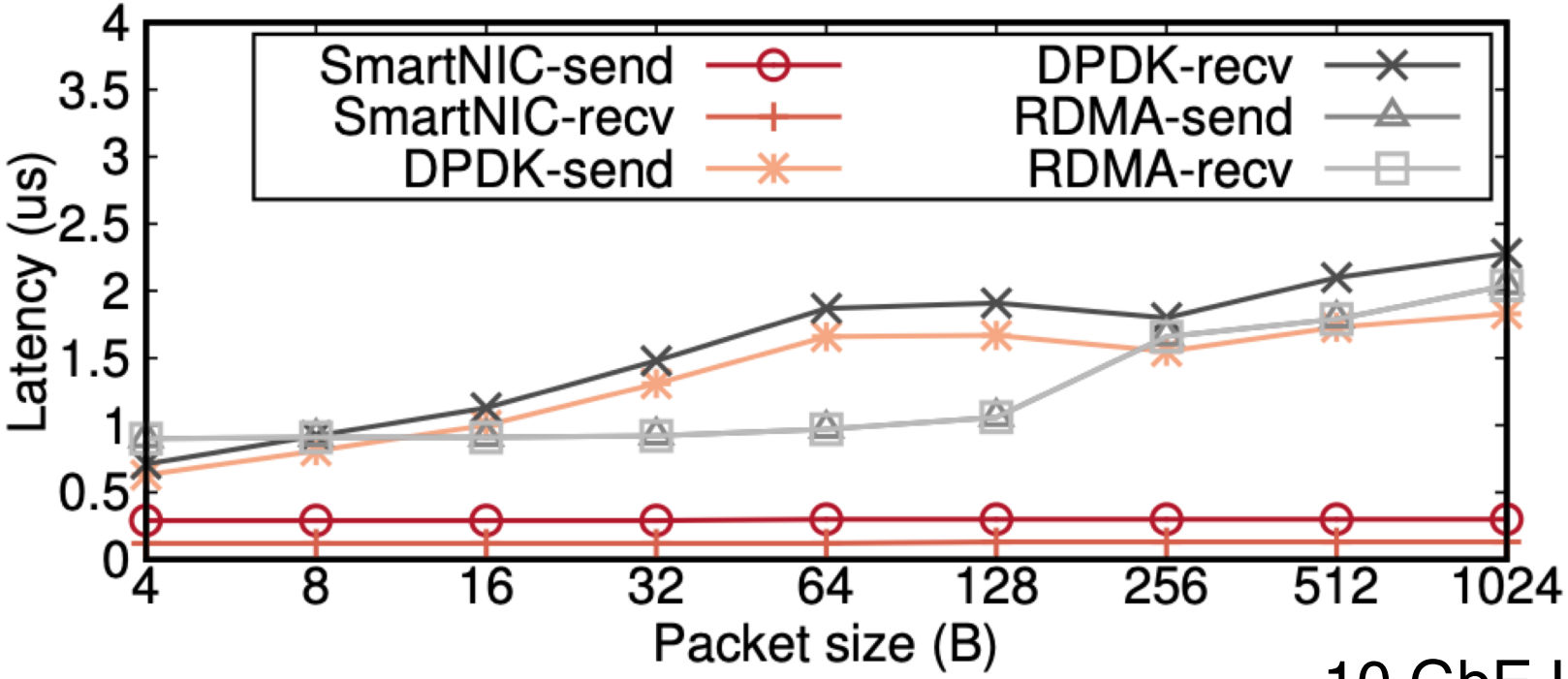
Average and P99 Latency



10 GbE LiquidIO II CN2350

- Achieving maximum throughput using 6 and 12 cores
- Hardware support reduces synchronization overheads

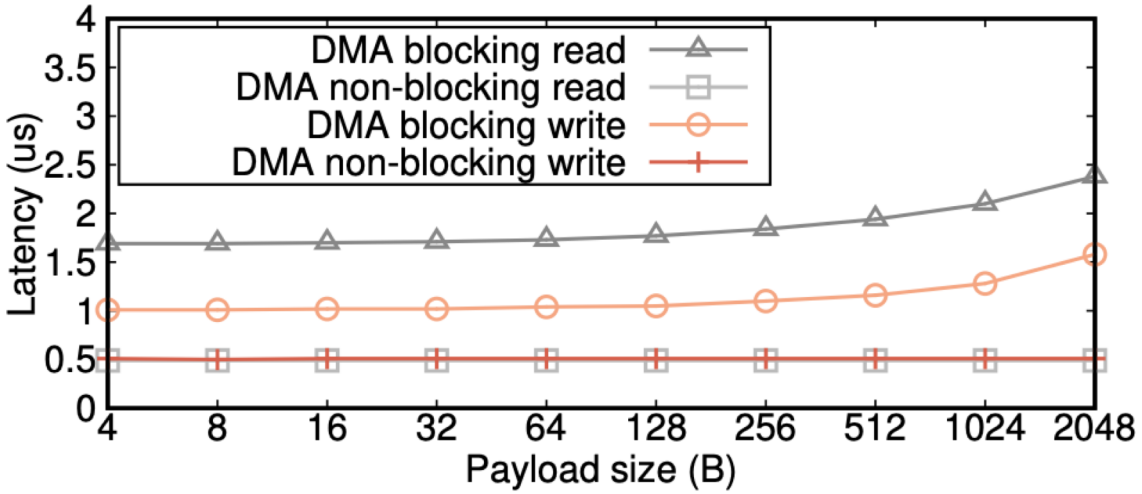
Send/Recv Latency

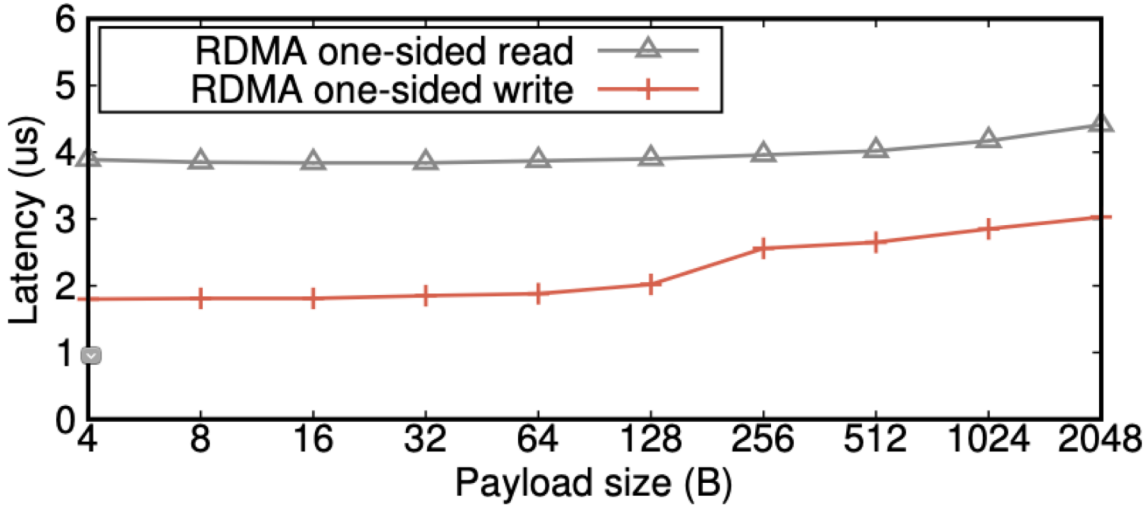


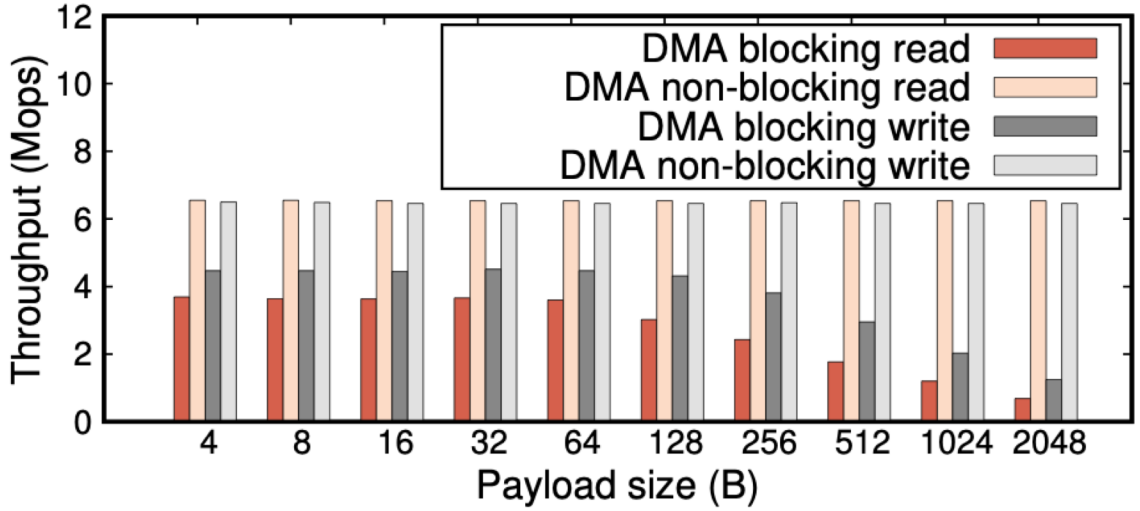
10 GbE LiquidIO II CN2350

- Special accelerators for packet processing
- Send/recv Latency lower than RDMA or DPDK

Host Communication



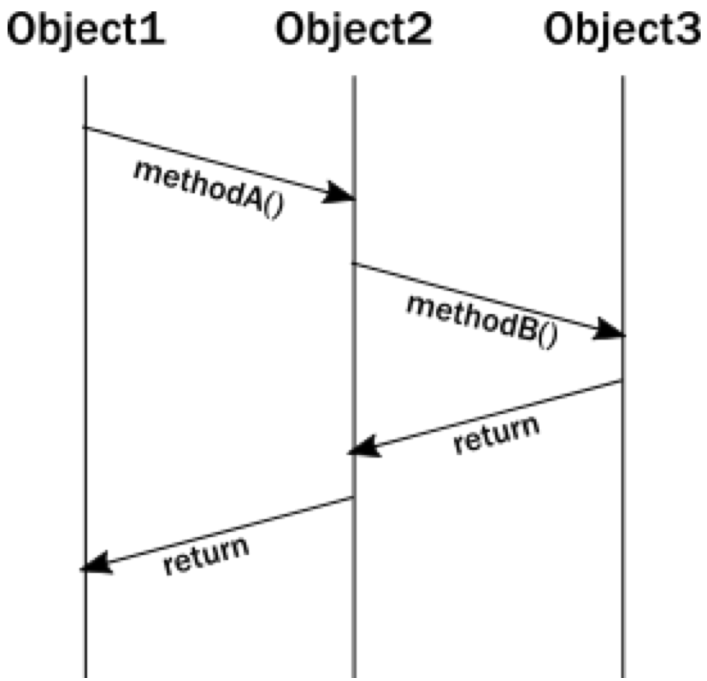




- DMA latency is 10X higher than DRAM latency in host cores
- 1-sided RDMA latency is higher than DMA latency

iPipe Framework

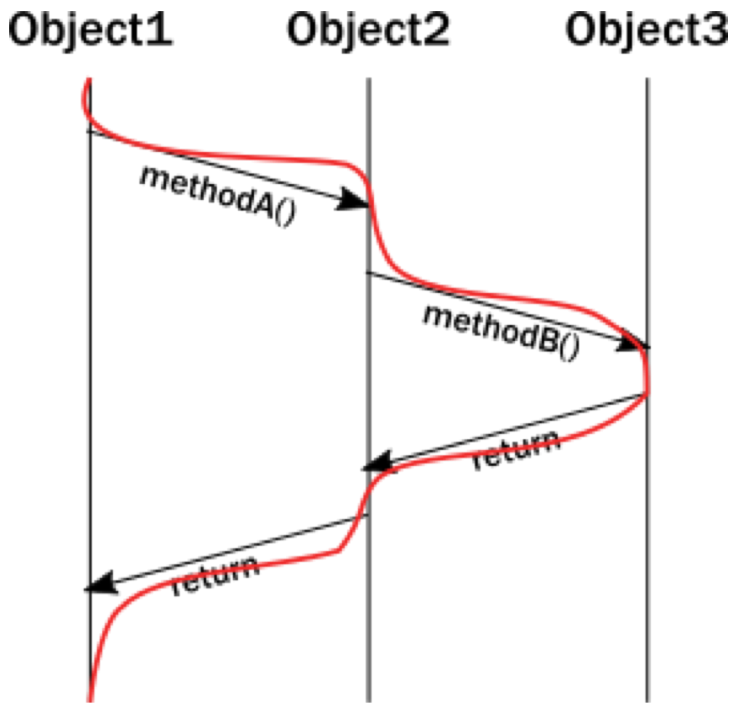
Actor Programming Model



Object-oriented programming

- **Encapsulation**: internal data of an object is not accessible from the outside

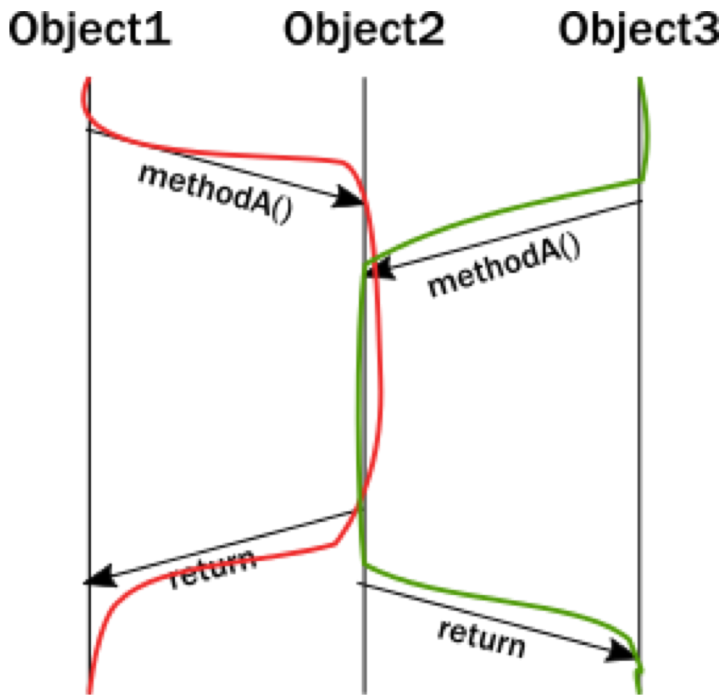
Actor Programming Model



Object-oriented programming

- **Encapsulation**: internal data of an object is not accessible from the outside
- Calls to different objects executed by the same thread

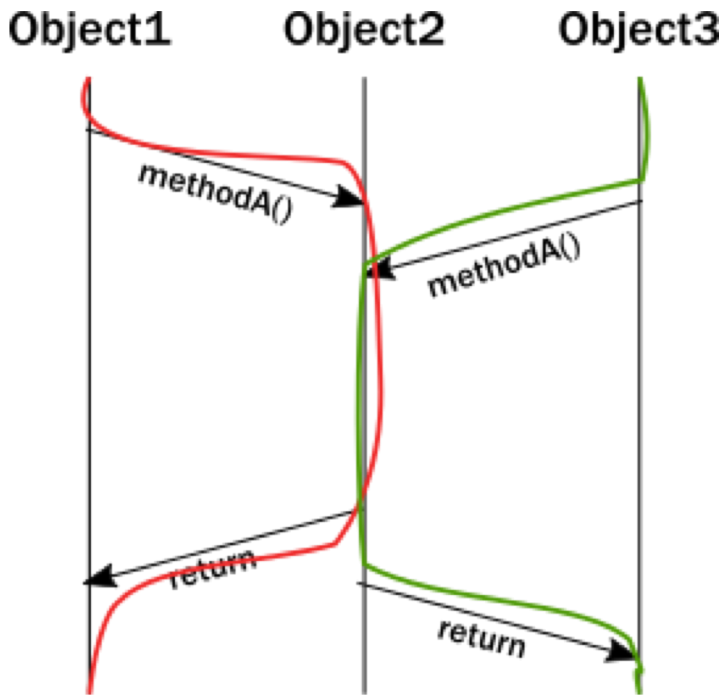
Actor Programming Model

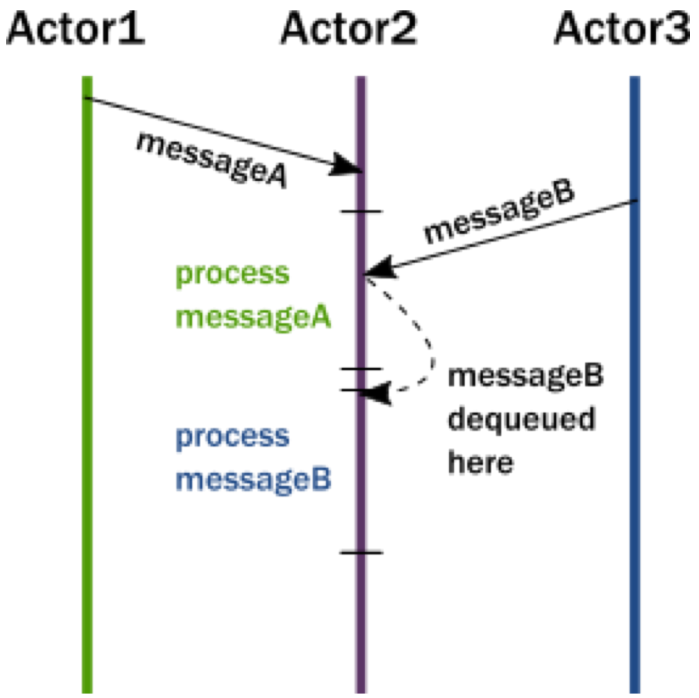


Object-oriented programming

- **Encapsulation**: internal data of an object is not accessible from the outside
- Calls to different objects executed by the same thread
- Must handle concurrent accesses

Actor Programming Model





Object-oriented programming

- **Encapsulation**

Actor programming model

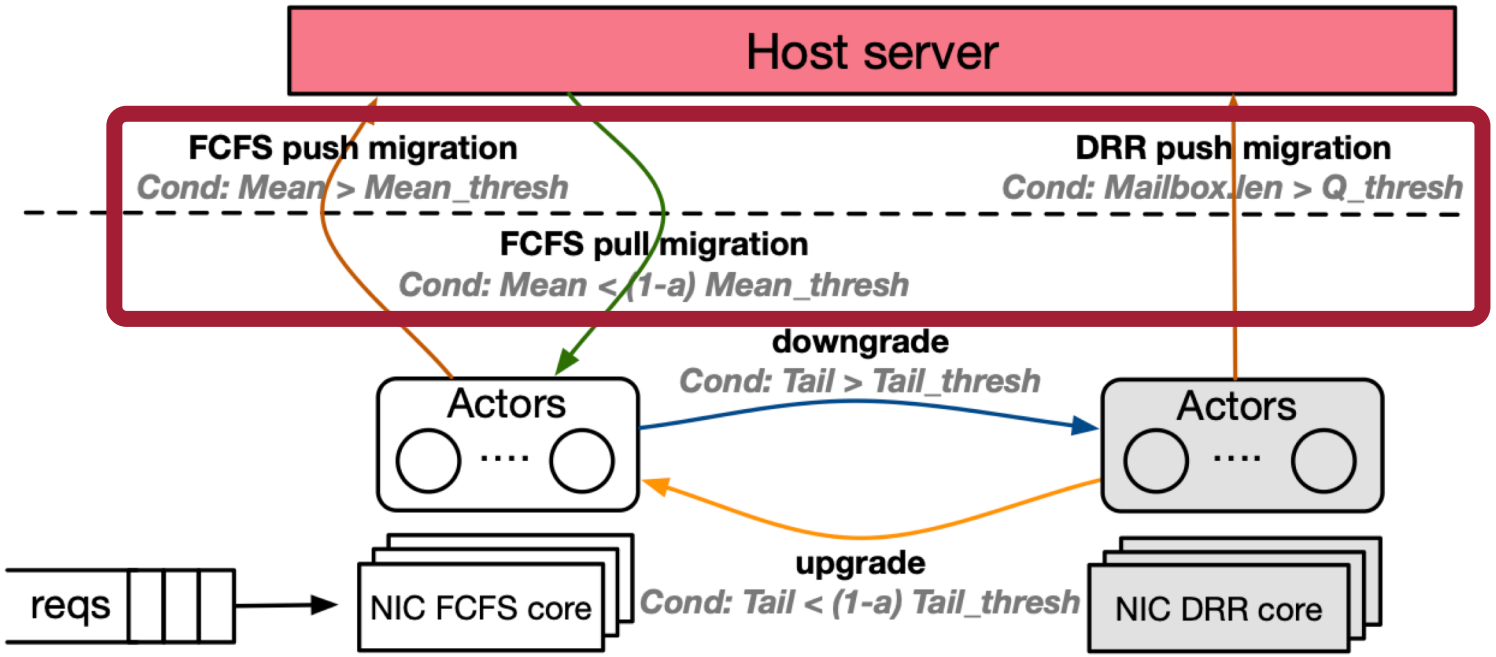
- An Actor has its local private states
- Actors communicate through messages

Advantages of Actor Model

Actor model supports computing heterogeneity and hardware parallelism automatically

Actors have well-defined associated states and can be migrated between the NIC and the host dynamically

iPipe Scheduler



Migration steps

1. Remove from runtime dispatcher
2. Actor finishes execution
3. Moves objects to host
4. Forwards buffered requests to host

Distributed Memory Object (DMO)

iPipe-host object table

Object ID	Actor ID	Start address	Size
1	1	0xfc0000000	1KB
x	x	0xfc0001234	2KB
z	z	0x10f005678	4KB
x	x	0x10f00abcd	8KB

iPipe-NIC object table

Object ID	Actor ID	Start address	Size
0	0	0x10f000000	1KB
~~x~~	~~x~~	~~0x10f001234~~	~~2KB~~
y	y	0x10f005678	4KB
~~x~~	~~x~~	~~0x10f00abcd~~	~~8KB~~

(a). Object migration

Normal SkipList node

```
struct node{
    char key[KEY_LEN];
    char *val;
    struct node *forwards[MAX_LEVEL];
}
```

DMO SkipList node

```
struct node{
    char key[KEY_LEN];
    int val_object;
    int forward_obj_id[MAX_LEVEL];
}
```

(b). Skiplist node implementation in DMO

All pointers replaced by object IDs

Security Isolation

Actor state corruption:

- Problem: Malicious actor manipulating other actors' states
- Solution: Paging mechanism to secure object accesses

Denial of service:

- Problem: An actor occupies a SmartNIC core and violates the service availability of other actors
- Solution: Timeout mechanism

Applications on iPipe

Replicated Key-Value Store

Log-structured merge tree for durable storage

Replication using Multi-Paxos

Actors:

1. Consensus actor
2. LSM Memtable actor
3. LSM SSTable read actor
4. LSM compaction actor

Distributed Transactions

Phase 1: read and lock

Phase 2: validation

Phase 3: log by coordinator

Phase 4: commit

Actors:

1. Coordinator
2. Participant
3. Logging actor

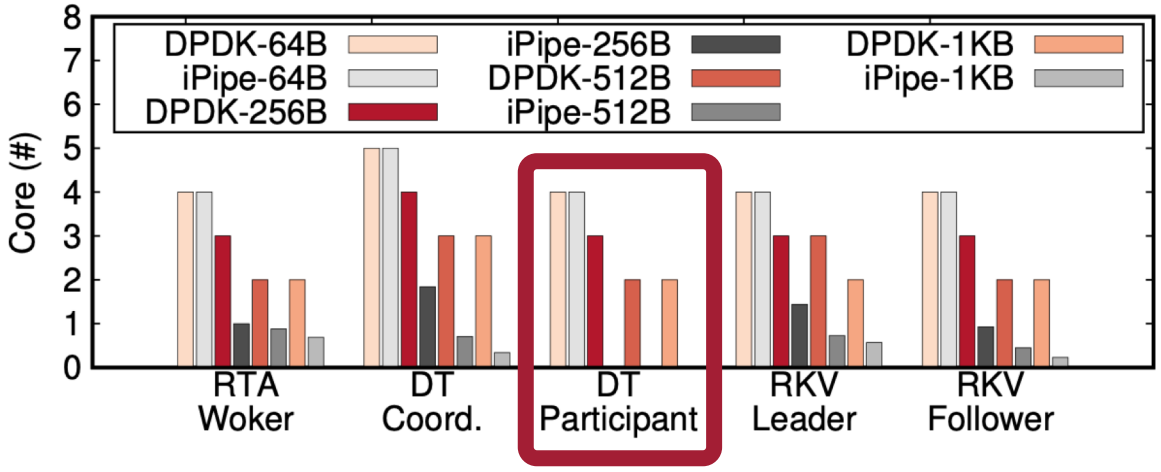
Real-Time Analytics

Analytics over streaming data

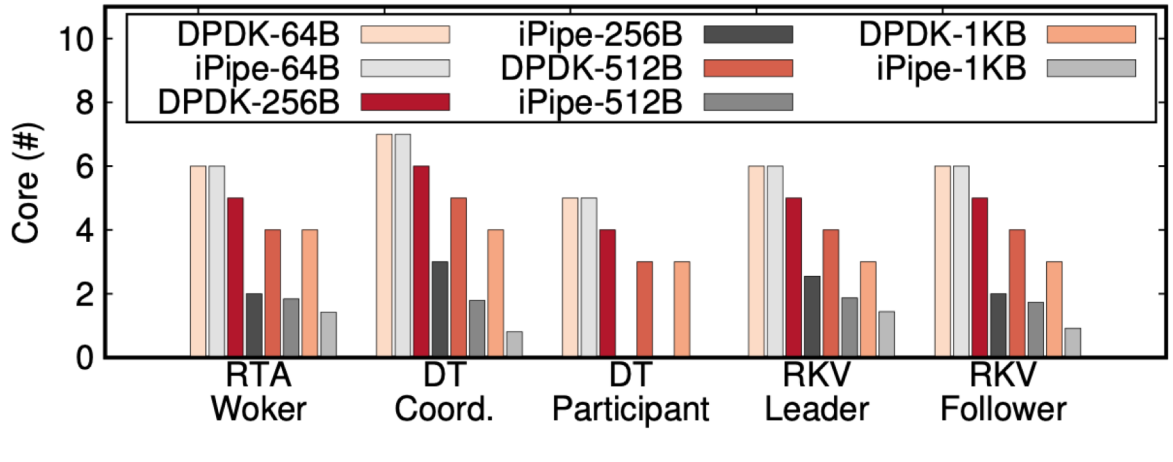
Actors:

1. Filter
2. Counter
   - Sliding winder and periodically emit tuple to the ranker
3. Ranker
   - Sort to report top-n

Evaluation – Busy CPU Cores



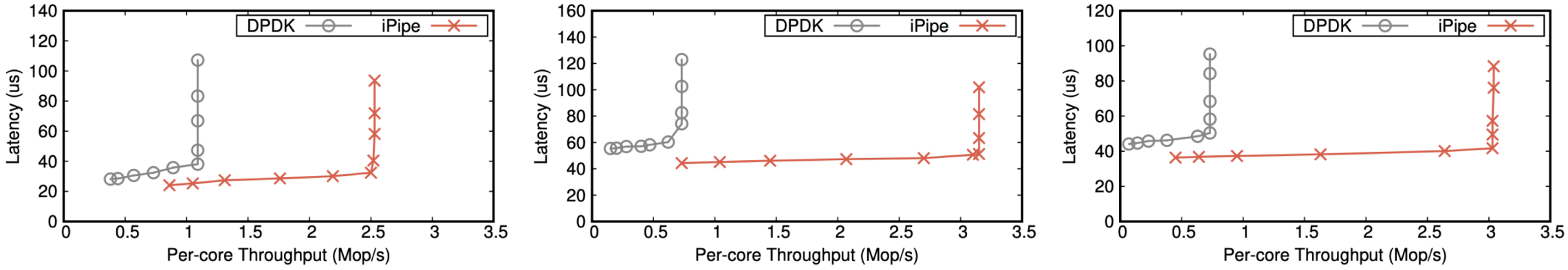
(a) 10GbE w/ LiquidIOII CN2350.



(b) 25GbE w/ LiquidIOII CN2360.

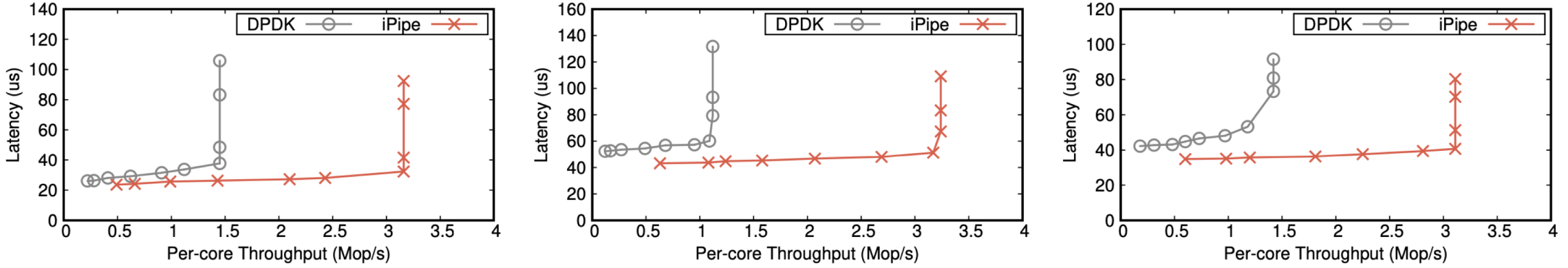
- Host CPU cycles are saved
- Offloading adapts to workload

Evaluation – Latency vs. Throughput



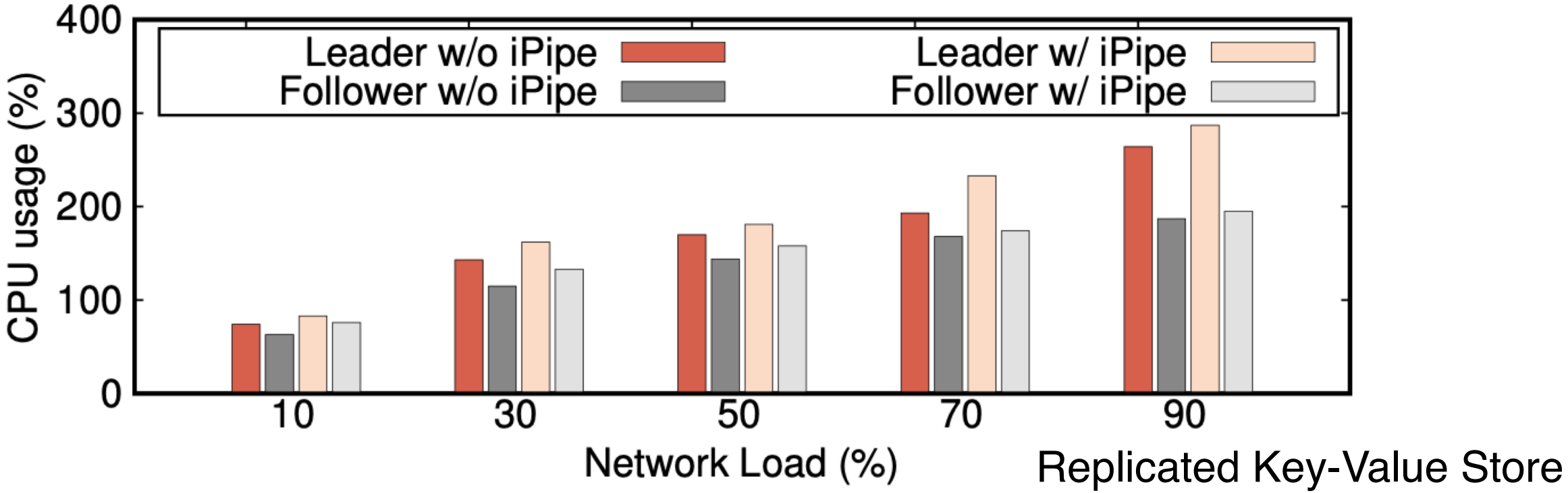
(a) RTA. (b) DT. (c) RKV.

Figure 14: Latency versus per-core throughput for three applications on 10GbE network. Packet size is 512B.



(a) RTA. (b) DT. (c) RKV.

Evaluation – iPipe Overhead



Replicated Key-Value Store

Overhead 1: DMO address translation when accessing objects

Overhead 2: Cost of iPipe scheduler

Smart NIC – Q/A

Actor Model in detail

Compare to RMA based approaches as defined in SNAP (SOSP'19)?

Are SmartNICs widely used nowadays and where?

Can transactional databases benefit from SmartNIC?

Limitation of SmartNIC (cost?)

Side-channel attacks?

Offloading control-intensive complex workloads to SmartNICs a promising path?

Group Discussion

SmartNIC pushes computation to network while SmartSSD pushes computation to storage. What are the main differences in terms of opportunities and challenges between the two technologies?

What database operations should be pushed to SmartNIC? Please discuss OLTP and OLAP separately.

One can consider processors in a Smart NIC as extra heterogeneous cores in a system. What extra benefits do we get by putting these extra cores into the NIC (in contrast to putting them close to storage or CPU)?

Before Next Lecture

Submit discussion summary to https://wisc-cs839-ngdb20.hotcrp.com

- **Deadline: Wednesday 11:59pm**

Next lecture will be given by Dr. Mike Marty from Google