SSICLOPS

Part1: PASTE: Network Stacks Must Integrate with NVMM Abstractions
Part2: Report on ACM HotNets 2016

Michio Honda (NEC Laboratories Europe)

With acknowledge to Lars Eggert and Douglas Santry

IIJ-II Seminar

December 26th, Tokyo, Japan

*work done

Motivation

SSICLOPS

- Non-Volatile Main Memories (NVMMs)
  - Persistent
  - Byte-addressable
  - Low latency
    - 10s-1000s of ns

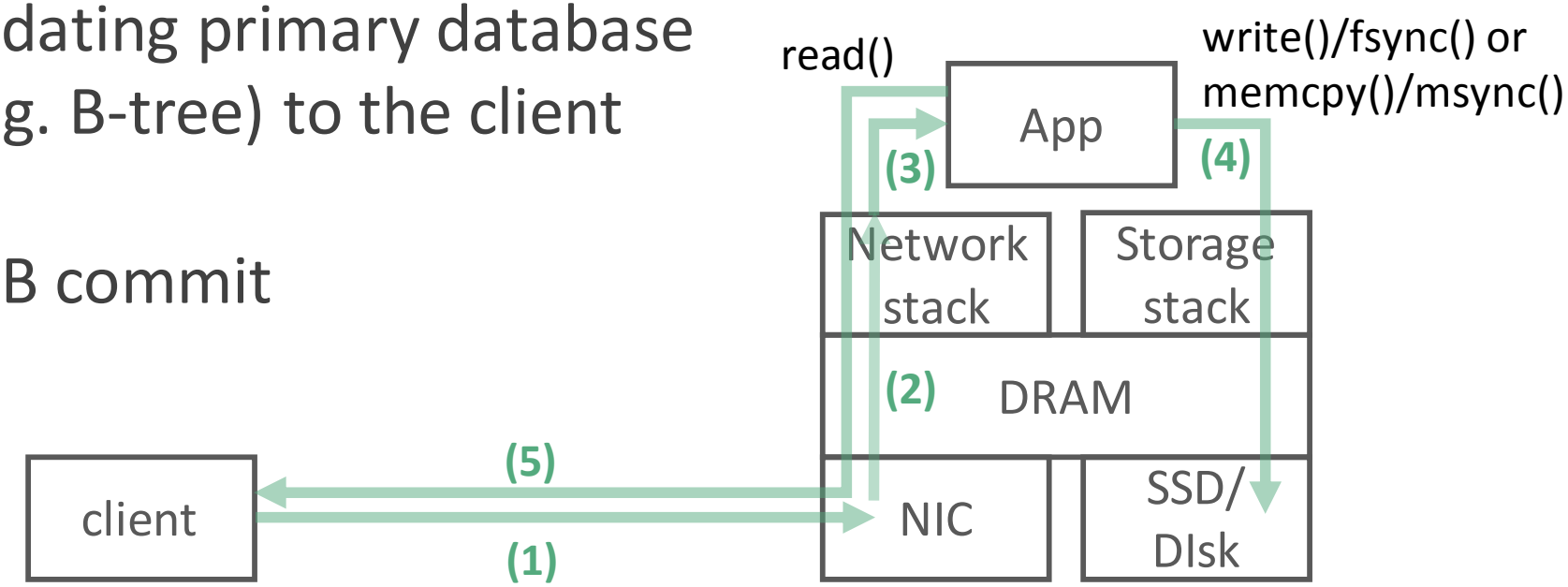
https://www.hpe.com/us/en/servers/persistent-memory.html

- Shift from ***block***- to ***byte***-granularity persistency
  - OS abstractions
    - ***Direct*** access to mmap()-ed ***files***
  - Data structures
    - Filesystems and databases

What are implications for networking?

Case Study: Write-Ahead Logging

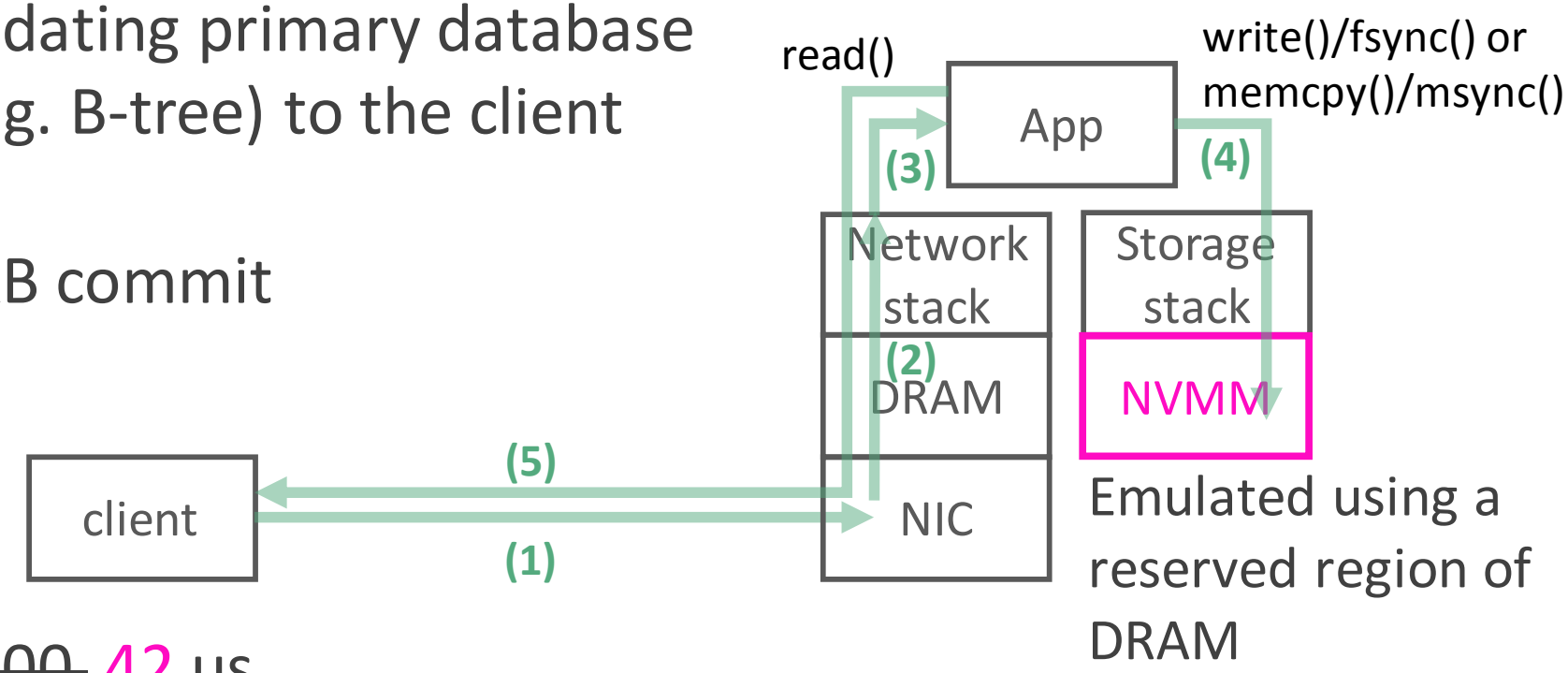
- Persist client's request prior to acknowledgment
- Durably store data into a log file to mask overhead of updating primary database (e.g. B-tree) to the client



- 1KB commit
- 2030 us
  - Networking takes 40 us

Case Study: Write-Ahead Logging

- Persist client's request prior to acknowledgment
- Durably store data into a log file to mask overhead of updating primary database (e.g. B-tree) to the client

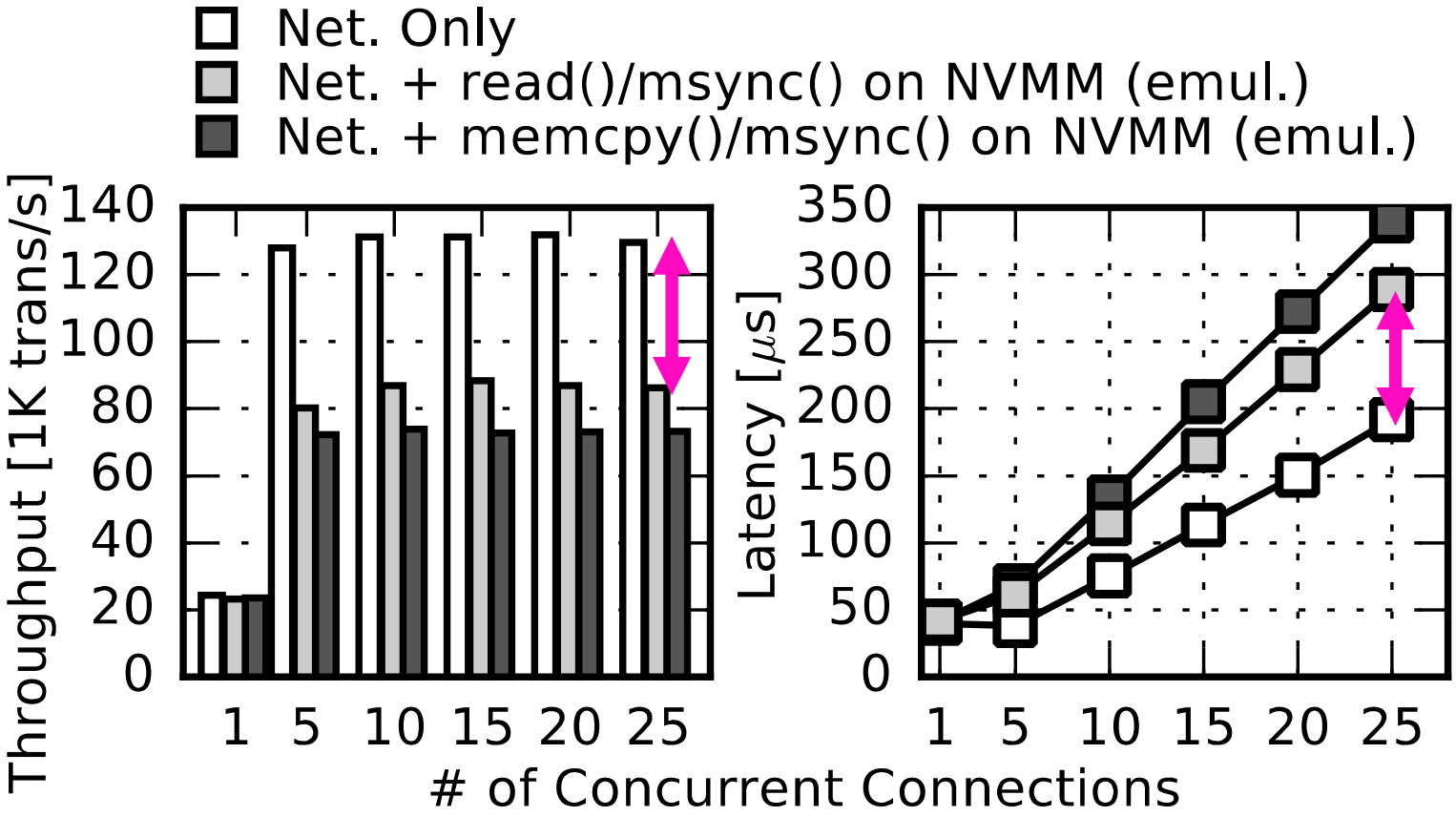


- 1KB commit
- ~~2000~~ 42 us
  - Networking takes 40 us
- This 2 us is not small

Case Study: Write-Ahead Logging

SSICLOPS

- Parallel requests are serialized on each core

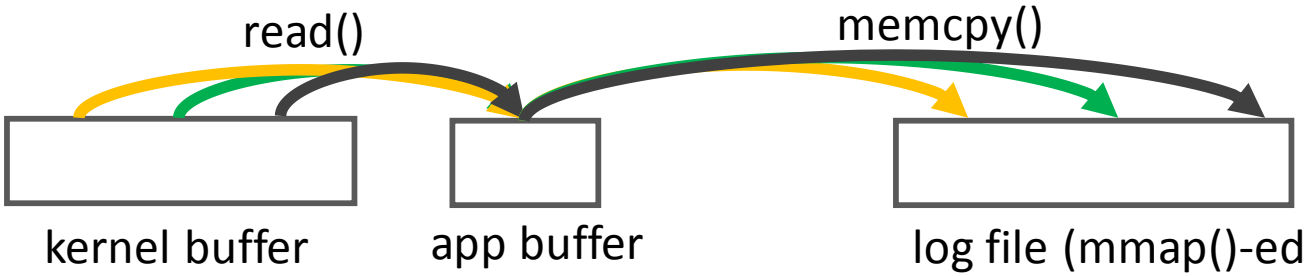


33 % throughput decrease, 50 % latency increase

Data Copies Matter

SSICLOPS

- **Cache Misses**
  - Copy to tmp buffer (e.g., read()) is cheap
  - **Logging always happens to a different destination**



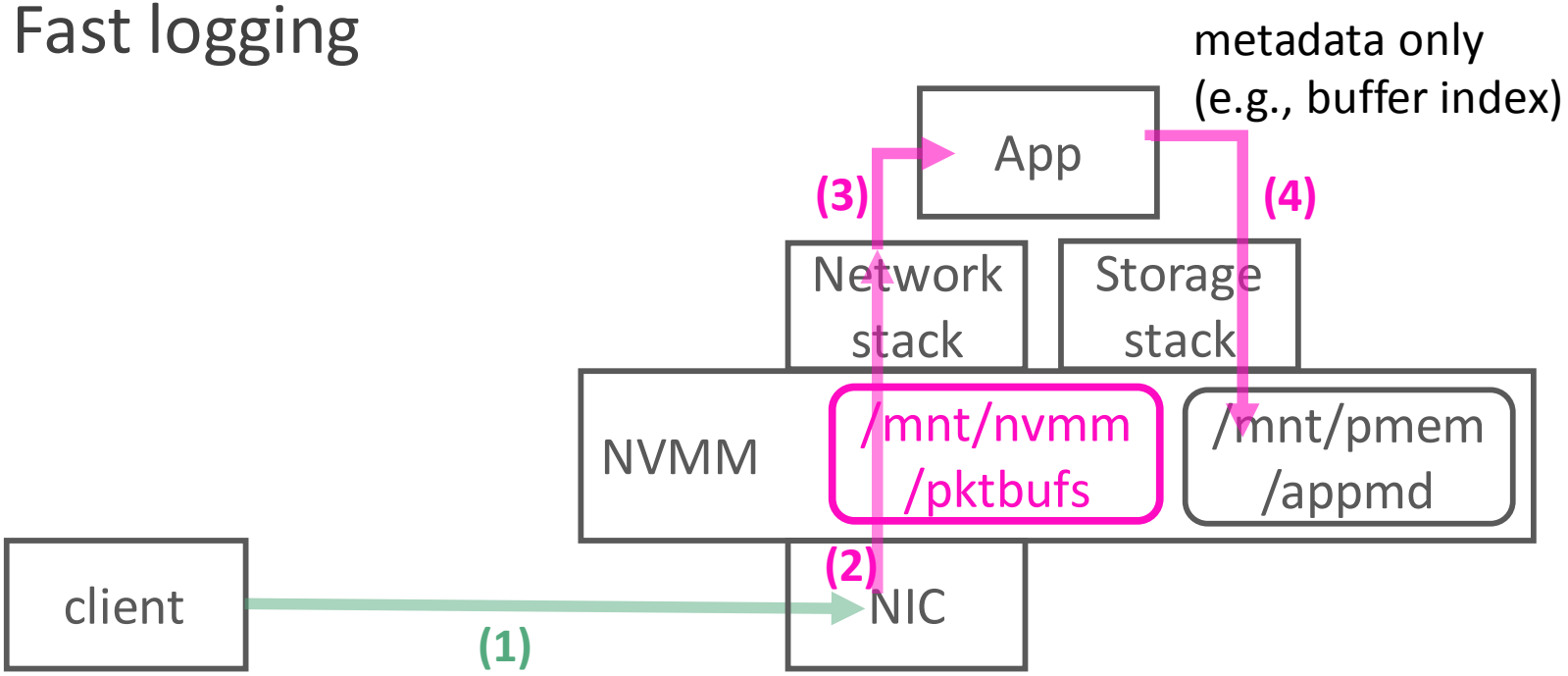
	Overall cache misses	Largest Contributor
Networking only	0.0004 %	net_rx_action() (84%)
Networking + NVMM (read() + memcpy() + msync())	4.4121 %	memcpy() (98%)
Networking + NVMM (read()+msync())	8.3451 %	sys_read() (99 %)

We must avoid data copy for logging!

Packet Store (PASTE) Overview

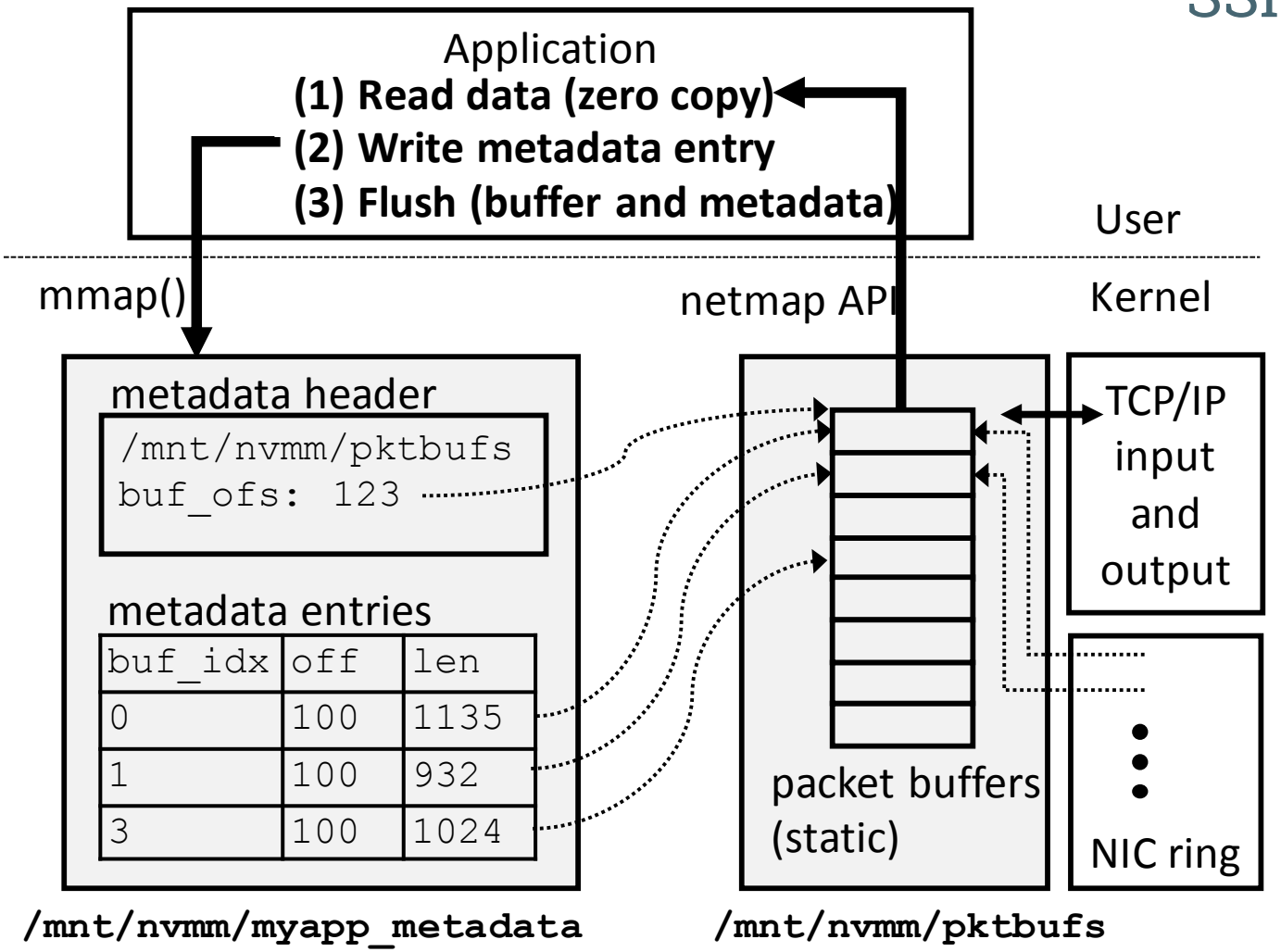
SSICLOPS

- Static packet buffers on a named NVMM region
  - DMA to NVMM
- Zero-copy APIs
- Fast logging



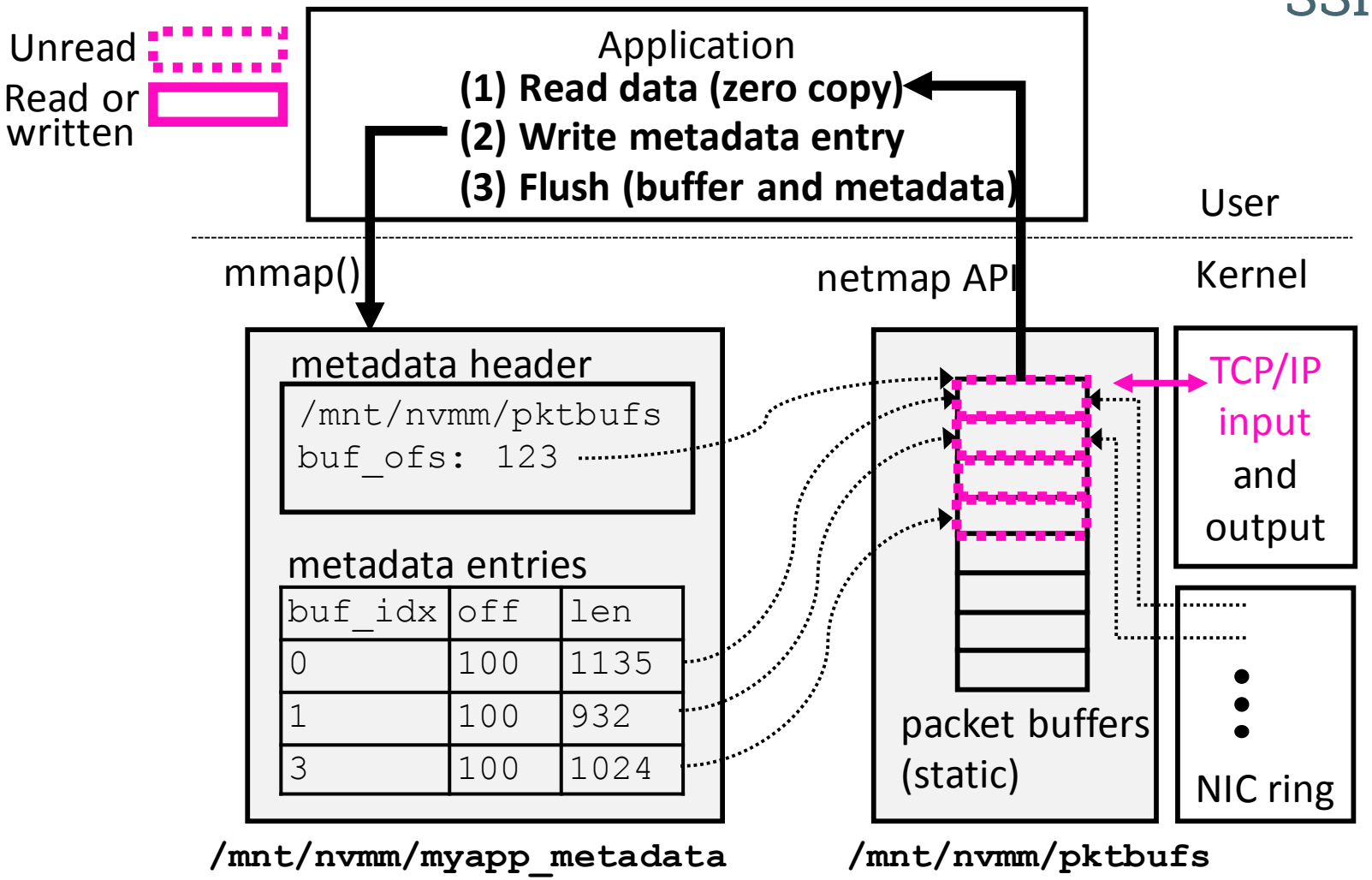
Fast Logging with PASTE

SSICLOPS



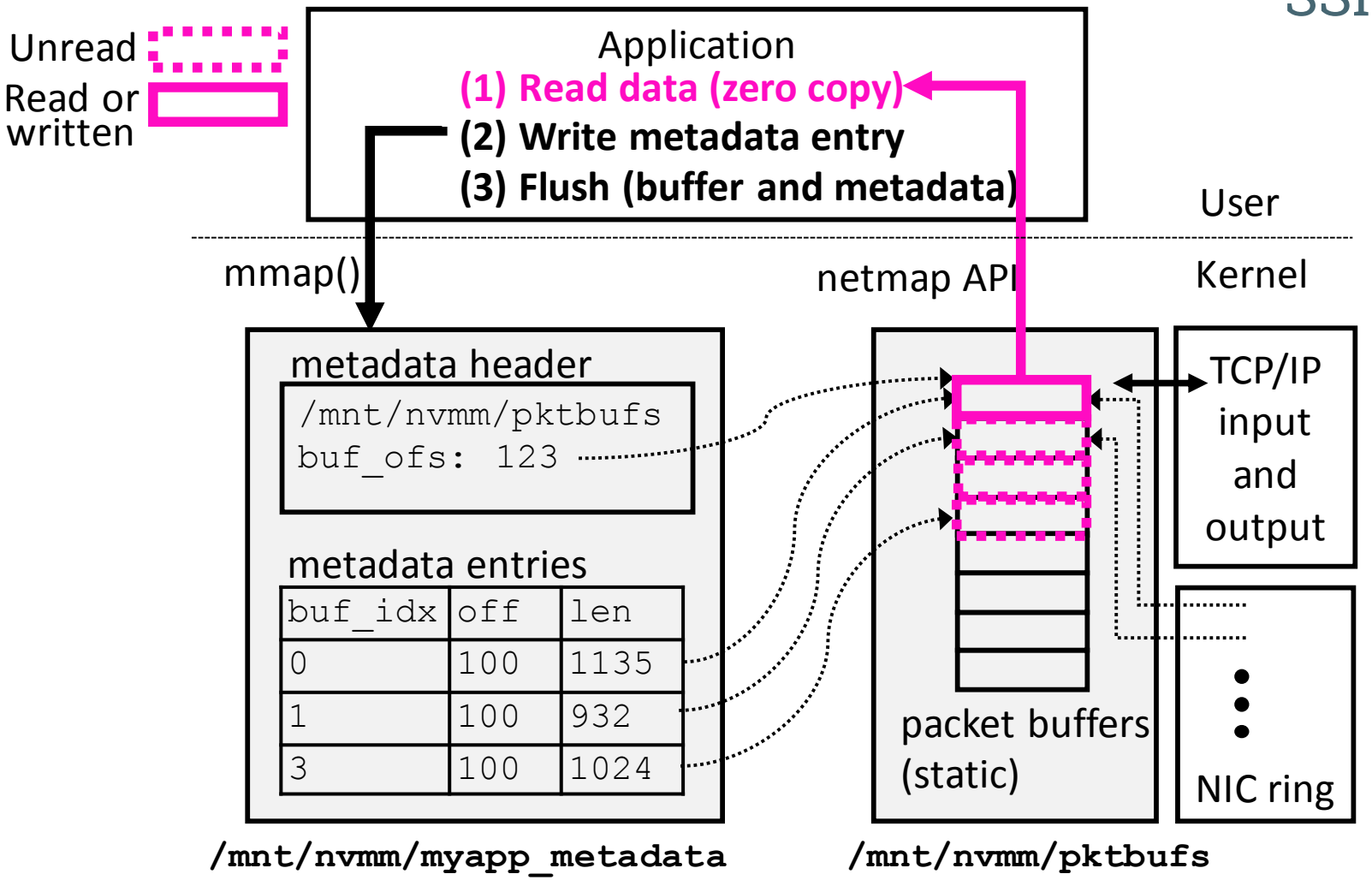
Fast Logging with PASTE

SSICLOPS



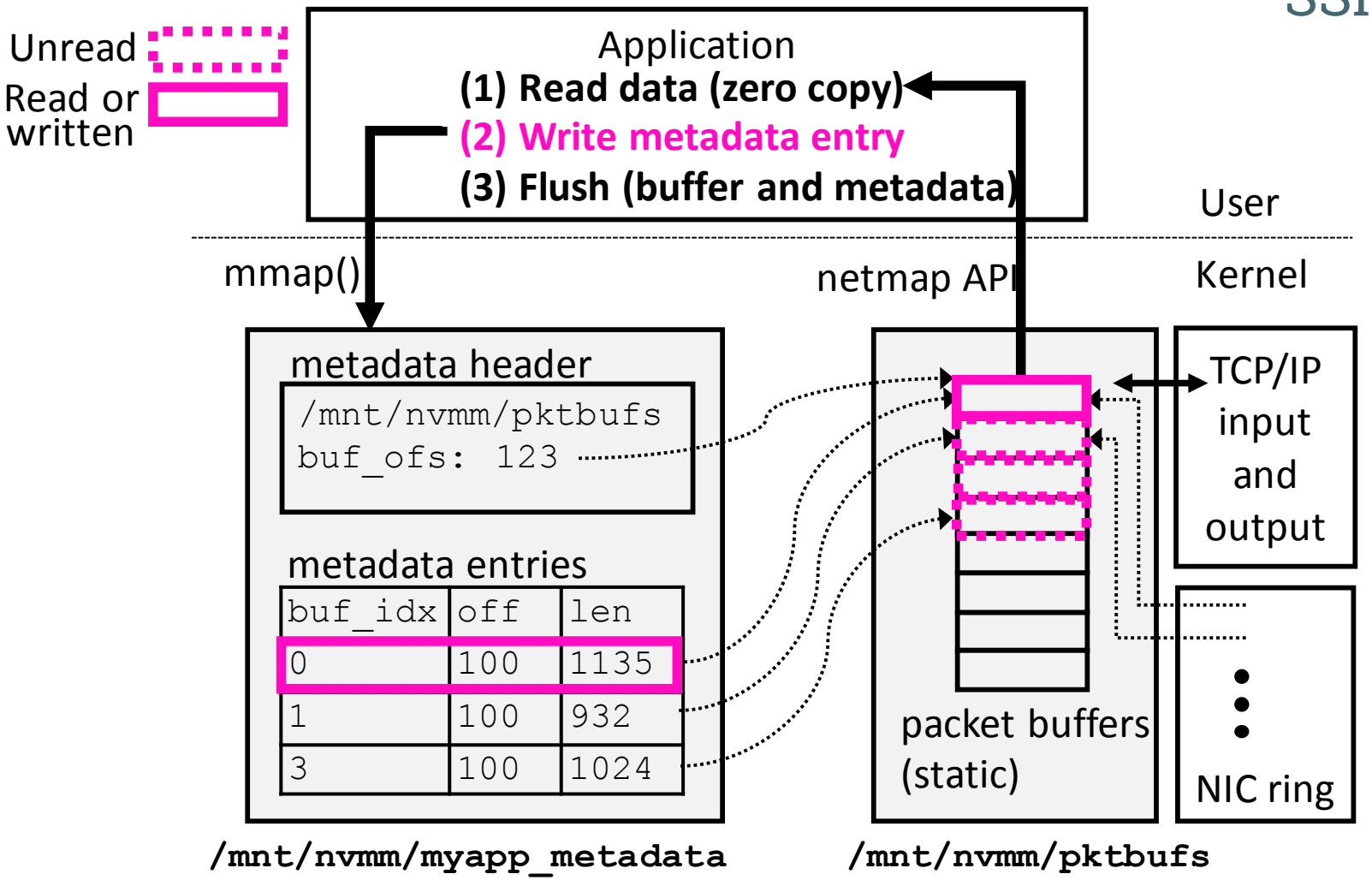
Fast Logging with PASTE

SSICLOPS



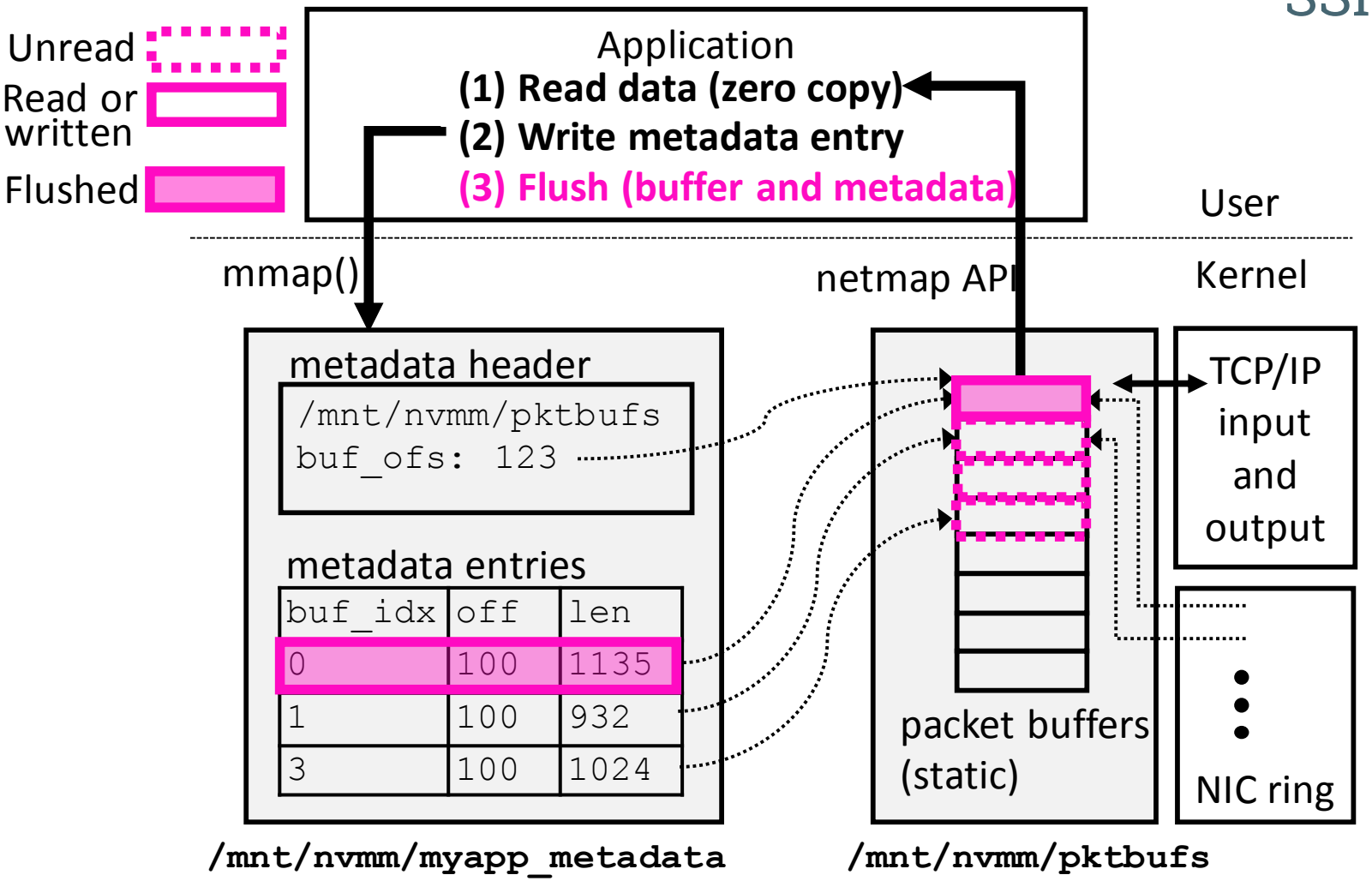
Fast Logging with PASTE

SSICLOPS



Fast Logging with PASTE

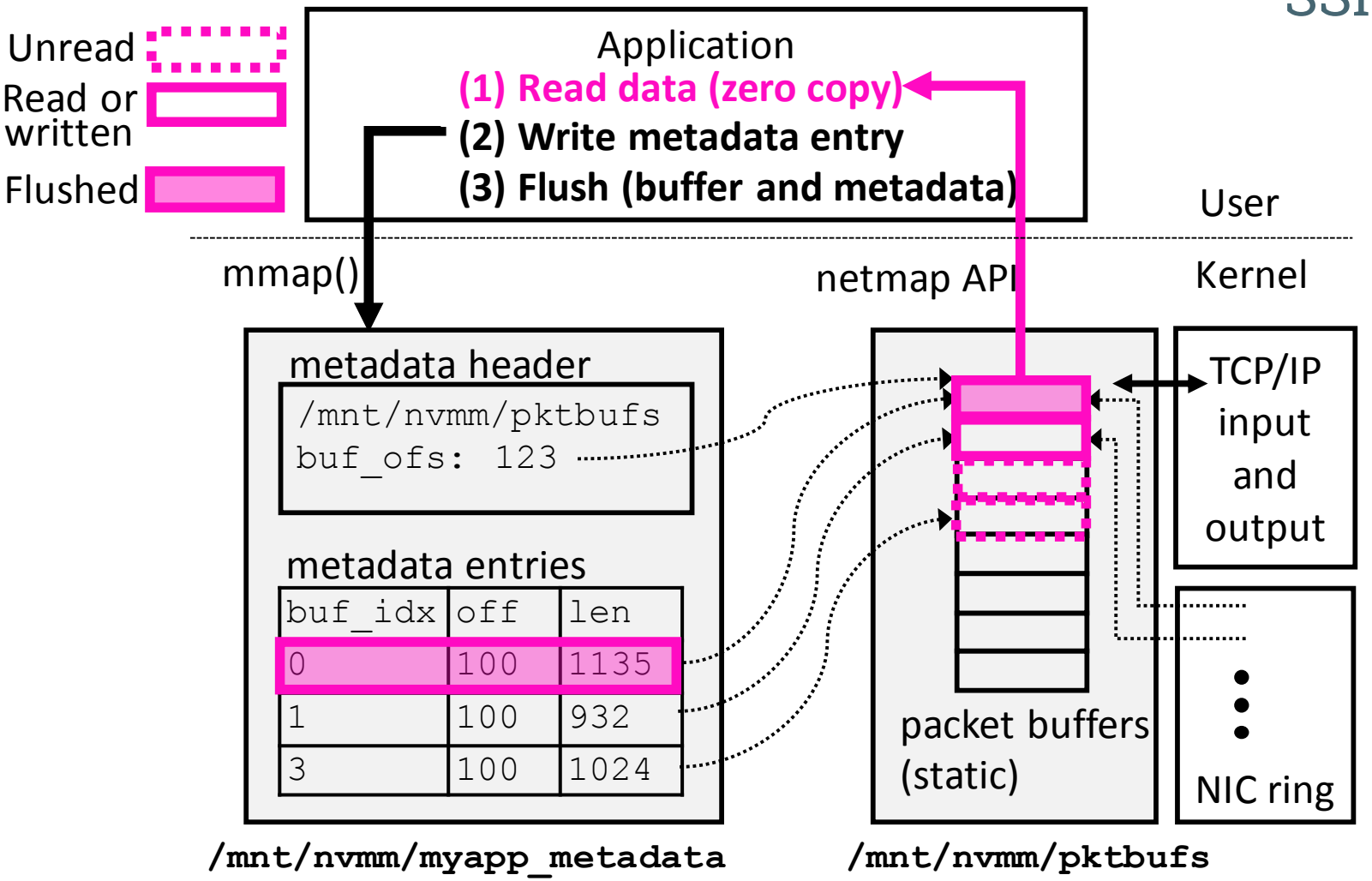
SSICLOPS



DMA is performed to L3 cache (DDIO)

Fast Logging with PASTE

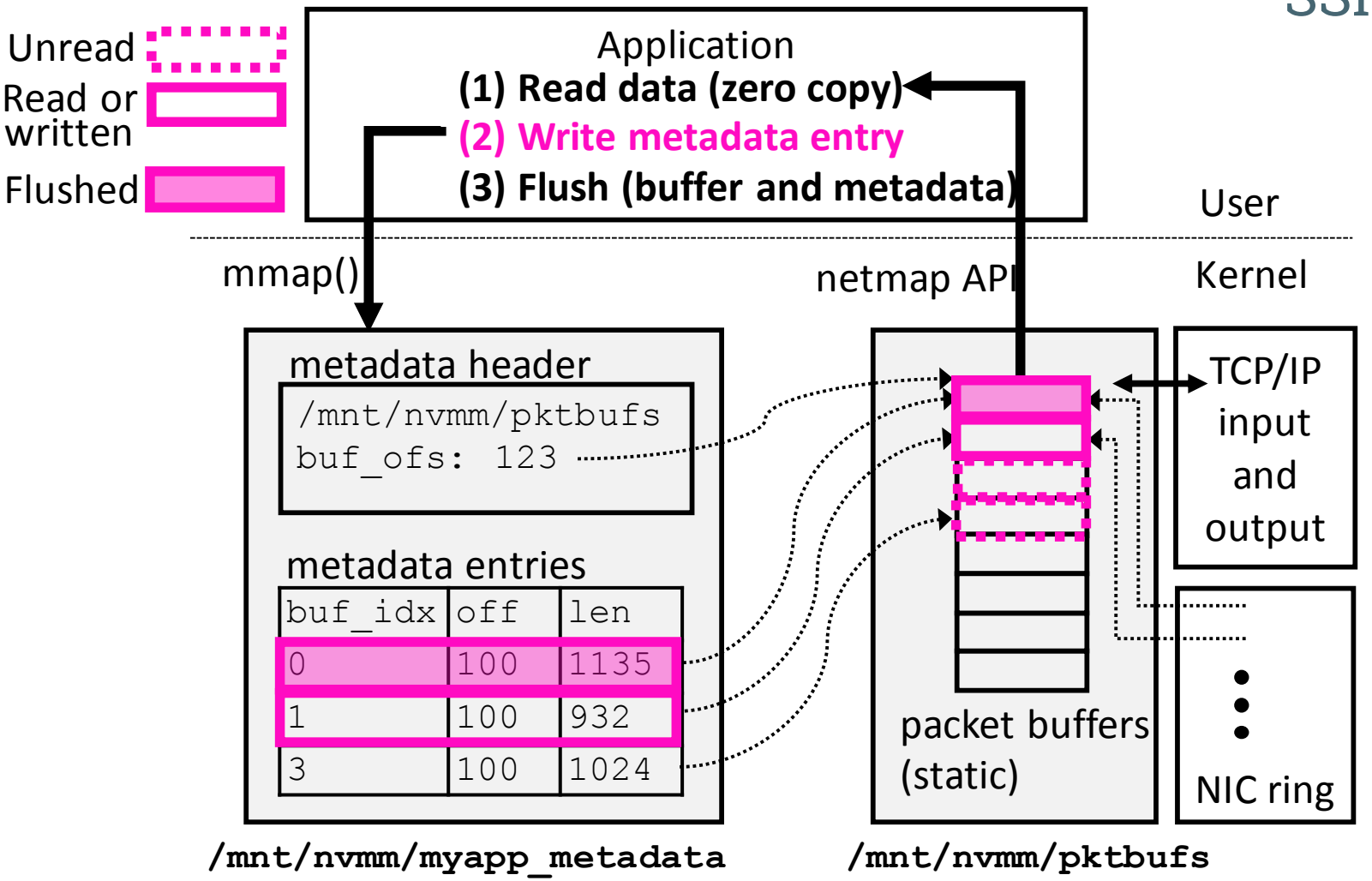
SSICLOPS



DMA is performed to L3 cache (DDIO)

Fast Logging with PASTE

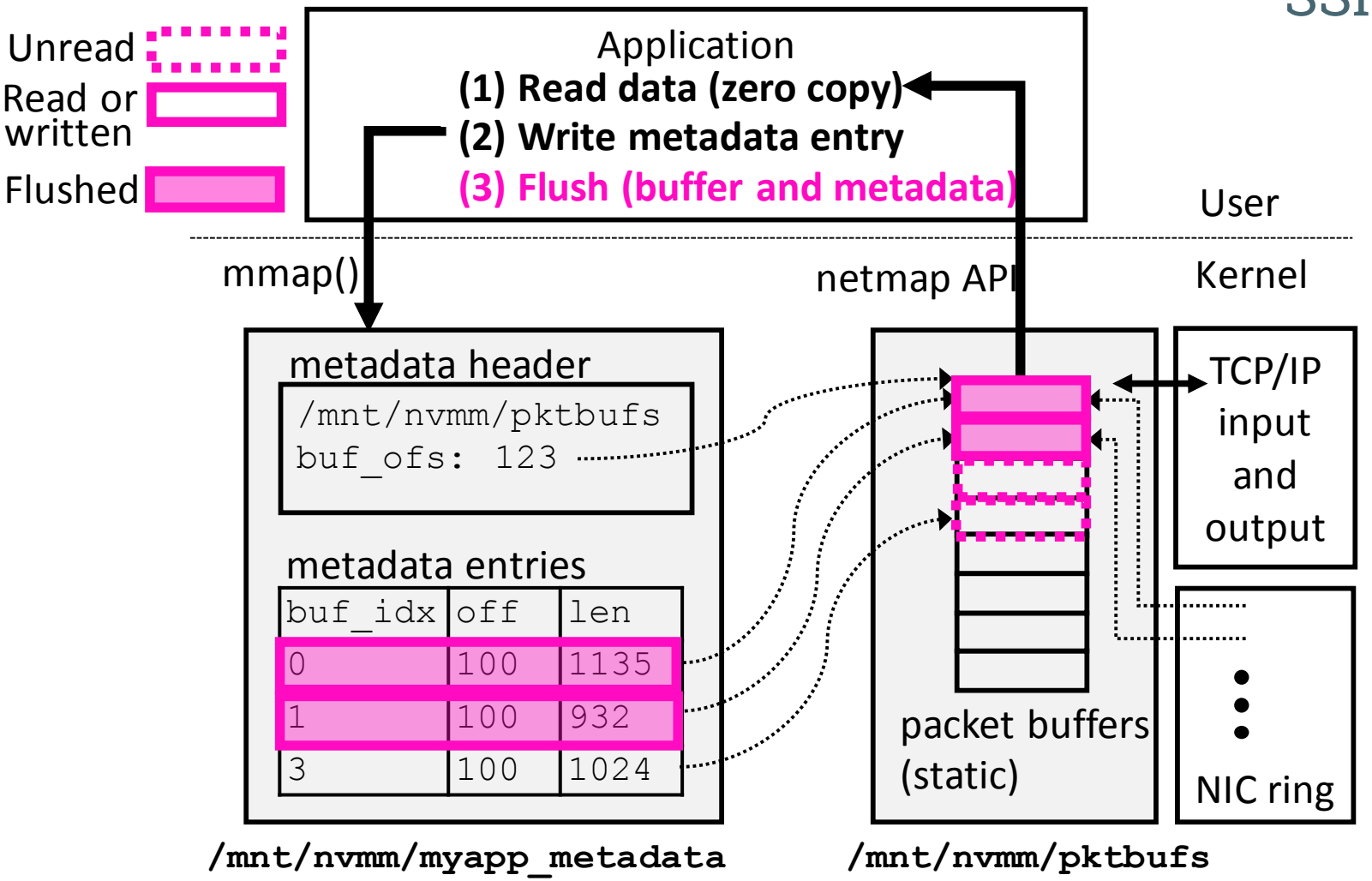
SSICLOPS



DMA is performed to L3 cache (DDIO)

Fast Logging with PASTE

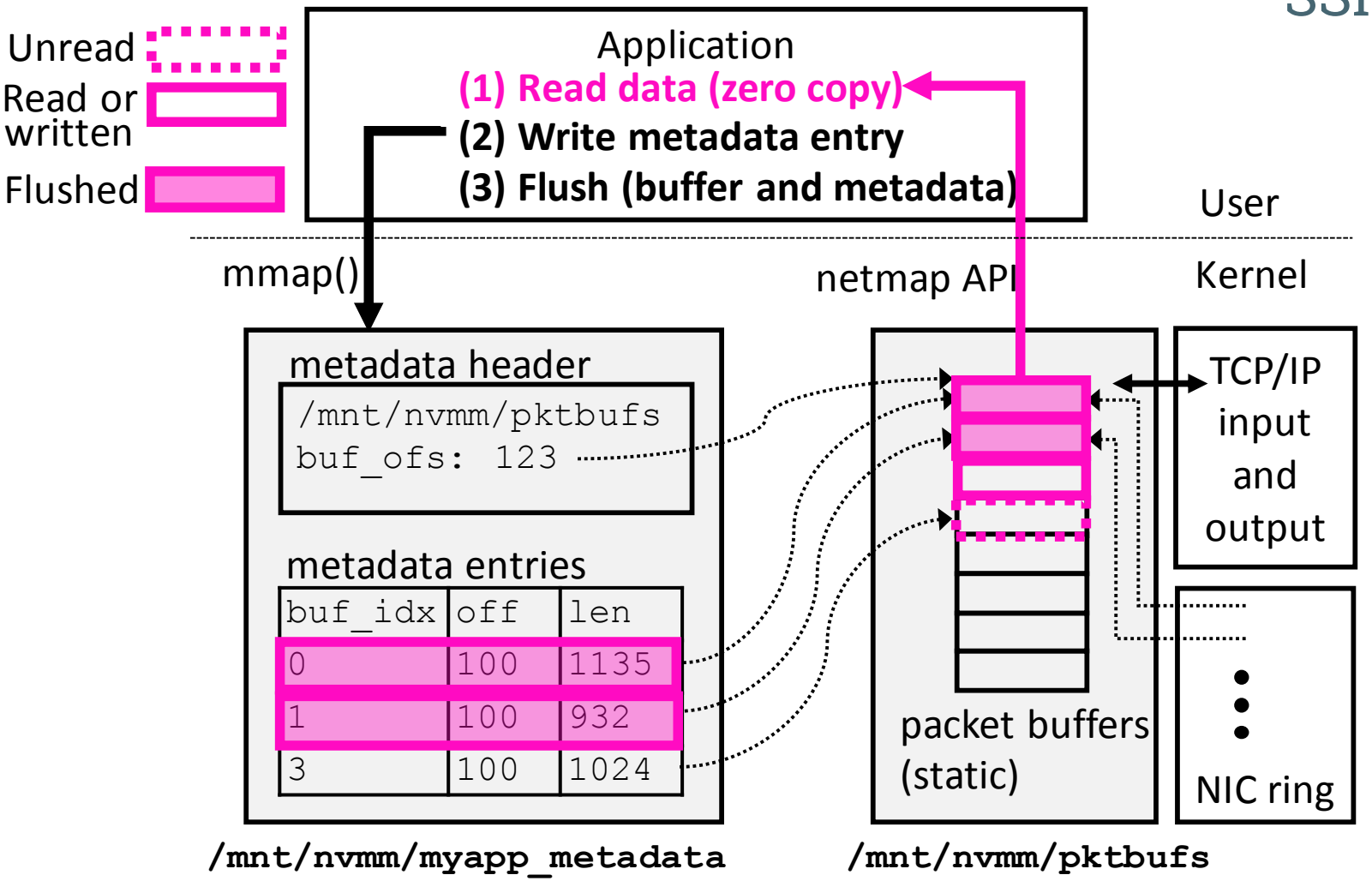
SSICLOPS



DMA is performed to L3 cache (DDIO)

Fast Logging with PASTE

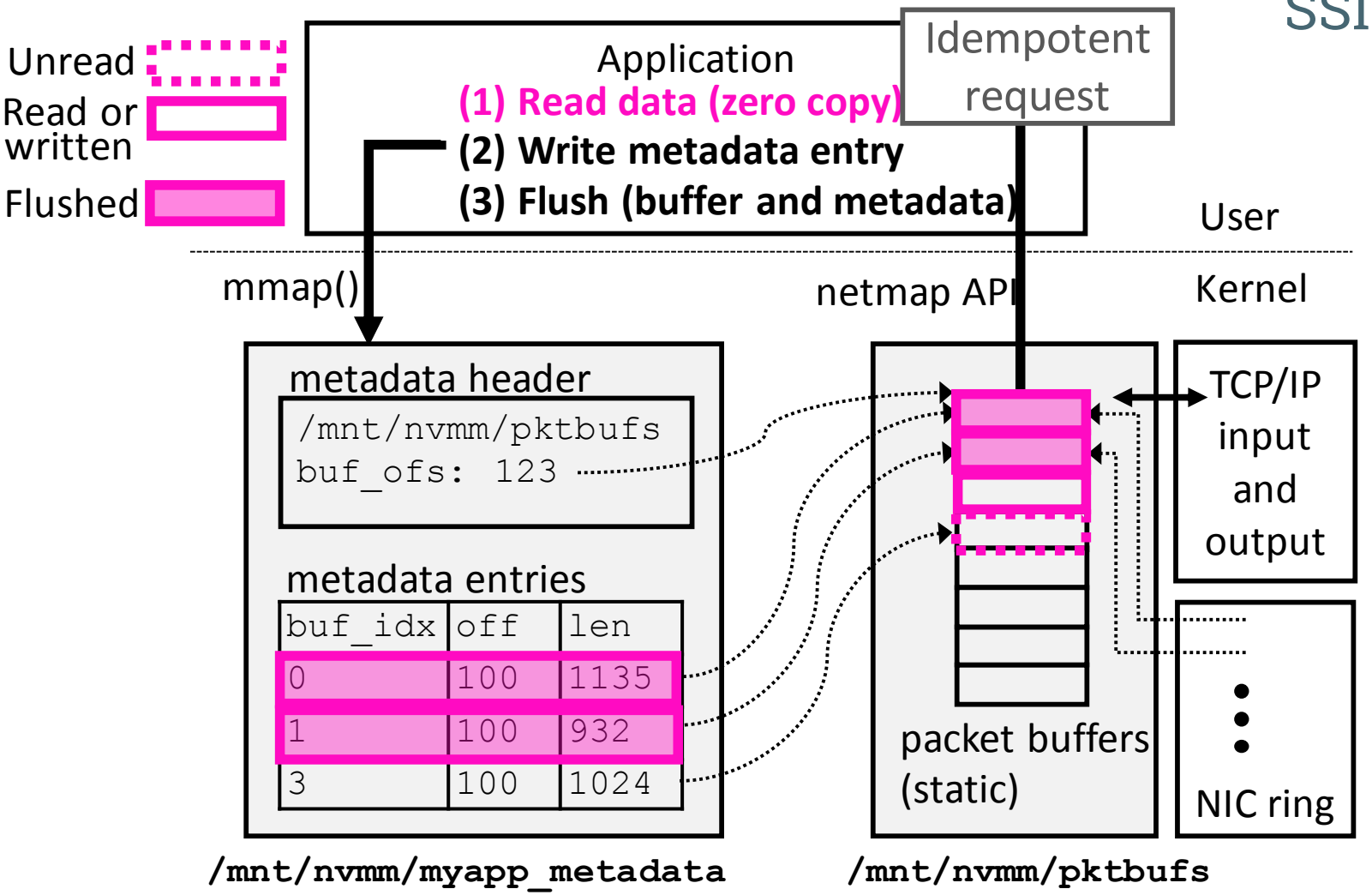
SSICLOPS



DMA is performed to L3 cache (DDIO)

Fast Logging with PASTE

SSICLOPS

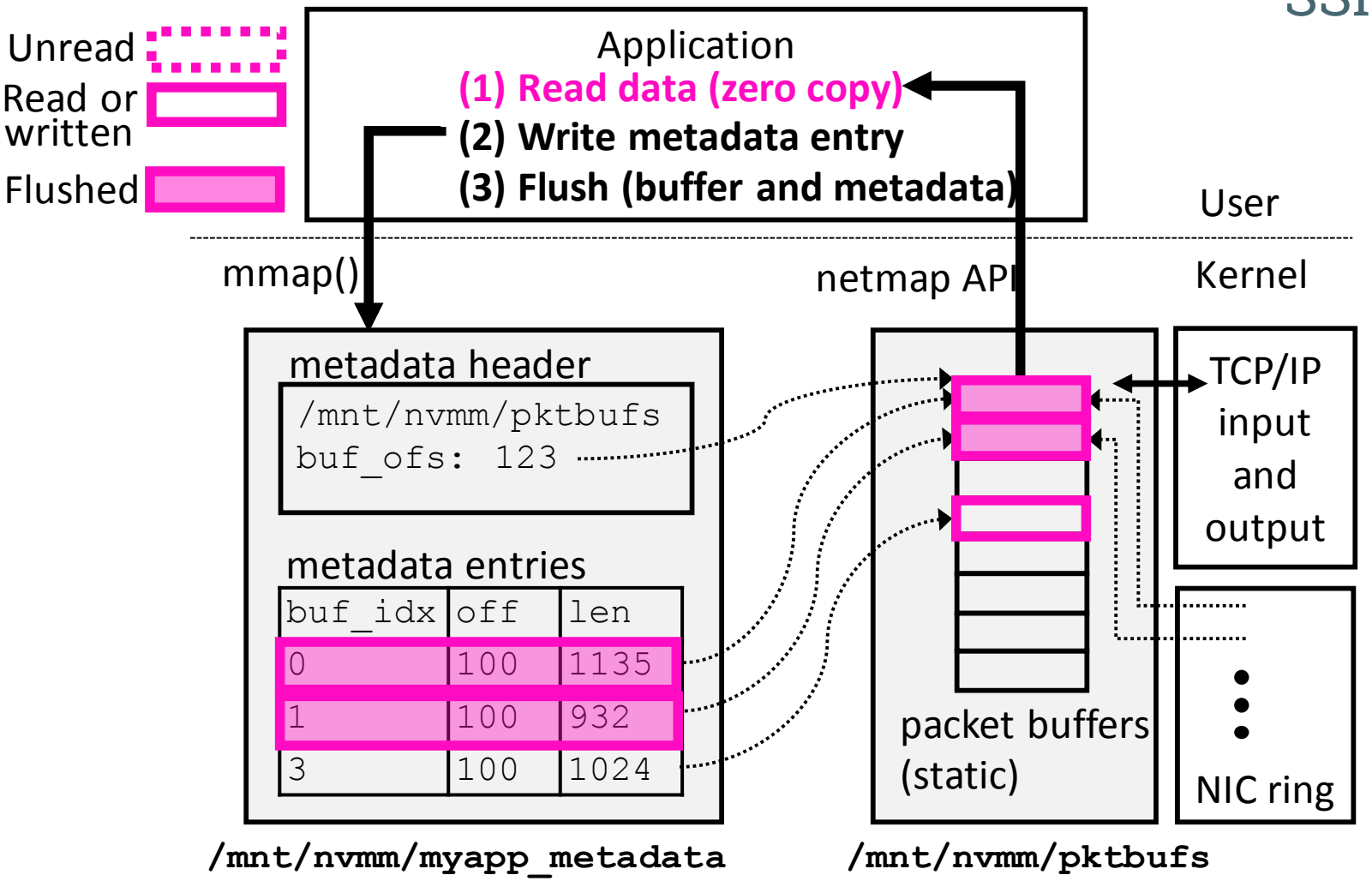


DMA is performed to L3 cache (DDIO)

Unnecessary data is not flushed to DIMM

Fast Logging with PASTE

SSICLOPS

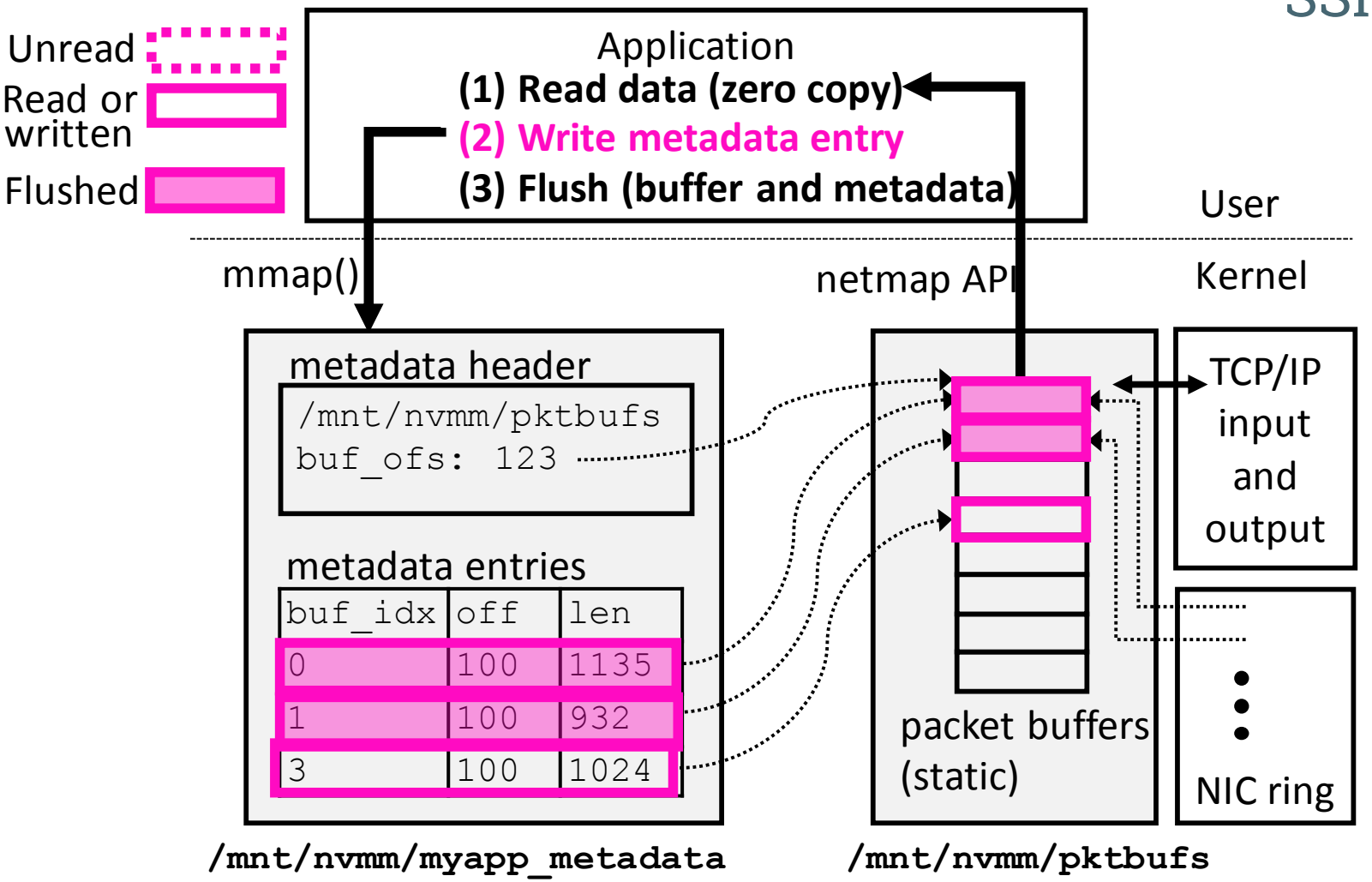


DMA is performed to L3 cache (DDIO)

Unnecessary data is not flushed to DIMM

Fast Logging with PASTE

SSICLOPS

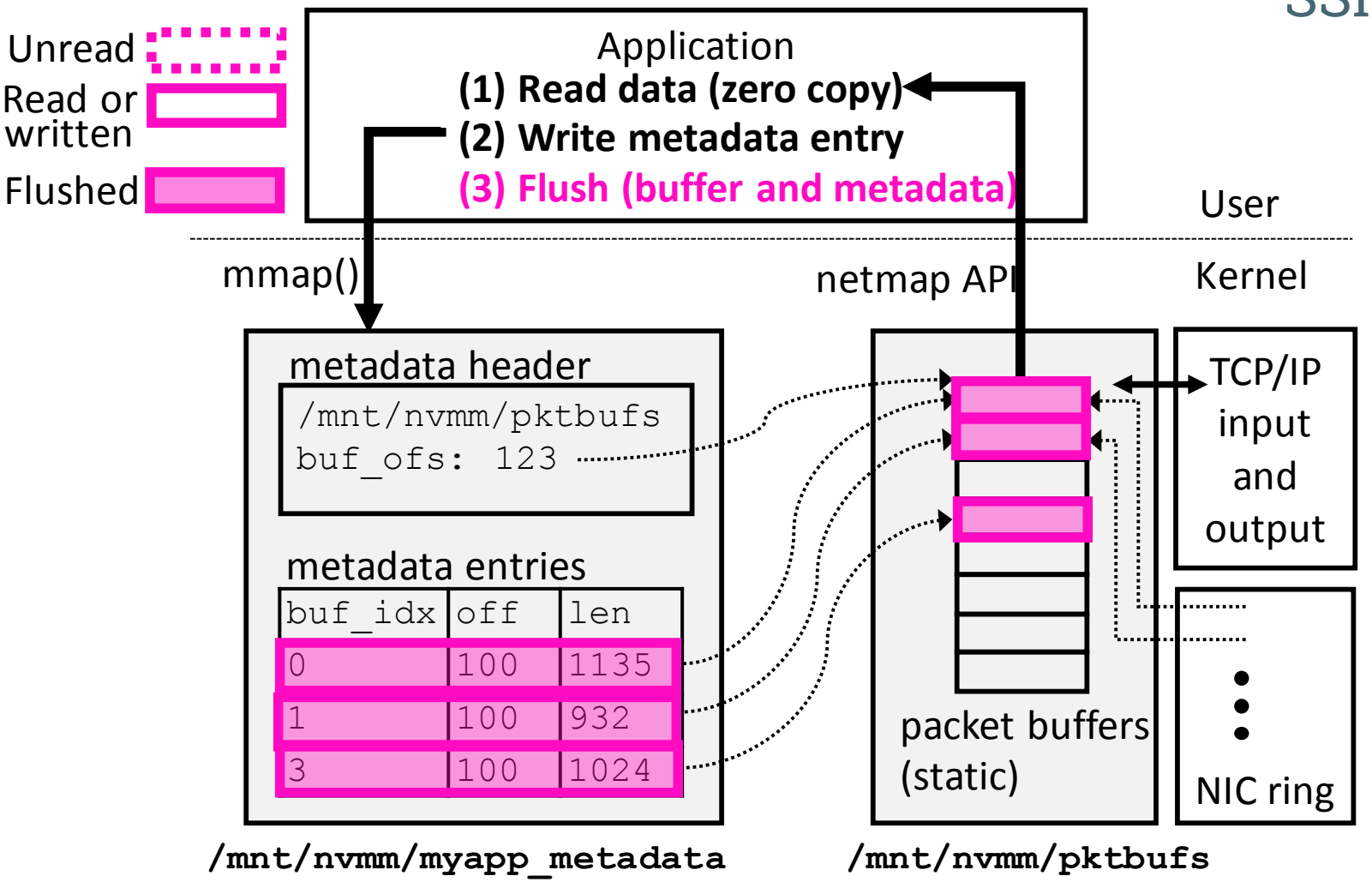


DMA is performed to L3 cache (DDIO)

Unnecessary data is not flushed to DIMM

Fast Logging with PASTE

SSICLOPS



DMA is performed to L3 cache (DDIO)

Unnecessary data is not flushed to DIMM

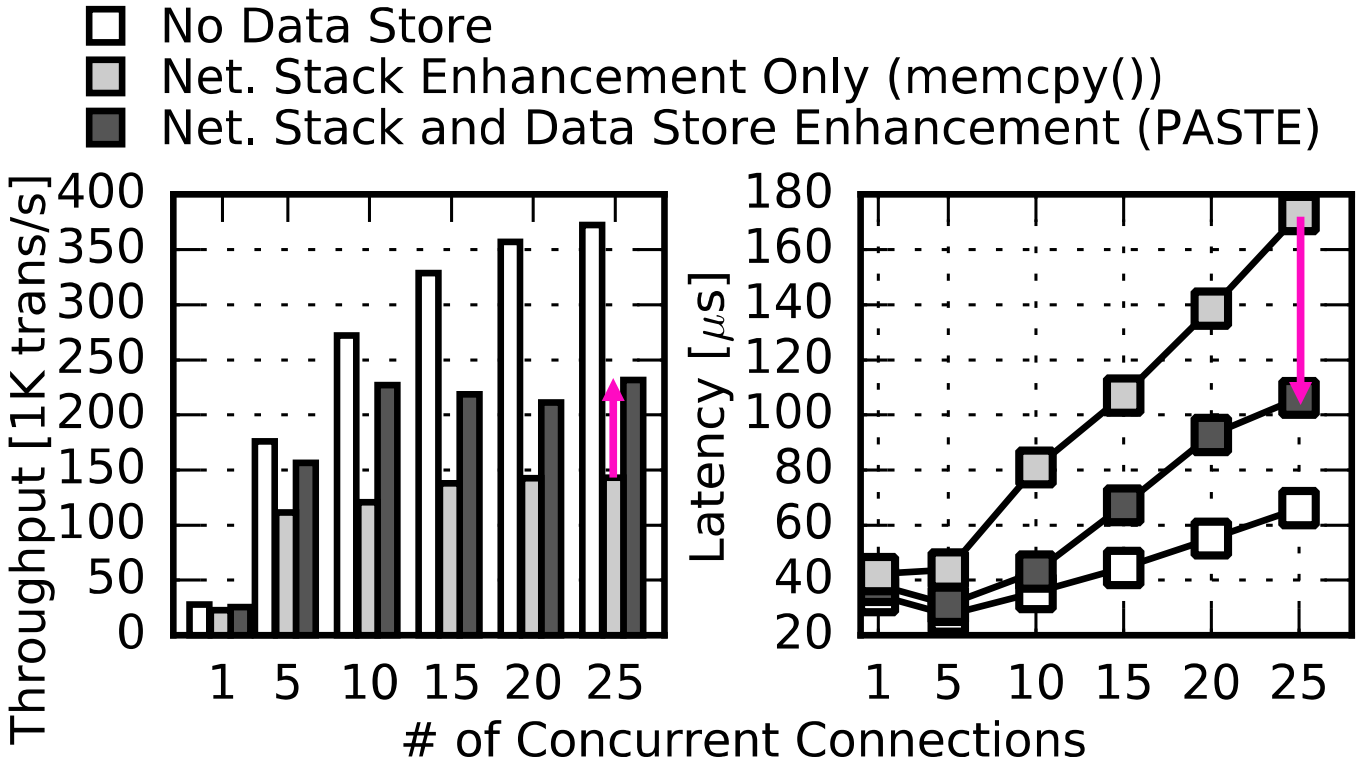
Implementation

- Extension to netmap memory allocator
  - Claim packet buffers from a given file backed by NVMM
    - e.g., pkt-gen -i eth1@/mnt/pmem/bufs -f rx
- Server app using the netmap API can easily implement logging

Preliminary Results

SSICLOPS

- Implementation
  - Extend the netmap framework
    - Stackmap for TCP/IP



10-88 % throughput increase, 9-46 % latency reduction

Related Work

- Enhanced network stacks
  - MegaPipe (OSDI'12), Stackmap (ATC'16), Fastsocket (ASPLOS'16)
  - IX and Arrakis (OSDI'14), mTCP (NSDI'13), Sandstorm (SIGCOMM'14), MICA (NSDI'14)

No NVMM aware

- NVMM filesystems
  - BPFS (SOSP'09), NOVA (FAST'15)
- NVMM databases
  - NVWAL (ASPLOS'15), REWIND (VLDB'15), NV-Tree (FAST'15)

No networking aware

Conclusion

PASTE: Fast logging with named packet buffers on NVMM and zero-copy API

- Implications
  - Network stacks are now a bottleneck for durably storing data
  - Improving network and storage stacks in isolation is not enough
  - We need new stacks design

HotNets 2016 Reports

- ACM Workshop on Hot Topics in Networks
  - Focus on new ideas and future directions in networking
- November 9-10 2016 @ Atlanta
- ~90 attendees (Invitation only)
  - 1 author per paper
  - Invited people in the community
  - Lottery

Submission and Reviews

- 30 papers (out of 108 submitted)
- 3-6 reviews
  - 48 papers on PC discussion
- We submitted 2 papers
  - 1 rejected (received 4 reviews)
    - 3 weak rejects + 1 weak accept
  - 1 accepted (received 6 reviews)
    - 1 accept + 4 weak accepts + 1 weak reject

Workshop Format

- Format was awesome
  - Very friendly – like a university internal workshop
  - Young people can be active
  - Senior people (incl. TPCs) comment and/or raise discussion

Workshop Topics

- ISPs
  - Frontier networks, Traffic engineering using MPTCP, monitoring
- Resource allocation
  - ML for cluster scheduling etc, blockchain for the Internet (BGP, DNS)
- Container networking
  - RDMA-based interfaces
- Social networks and clouds
  - SaaS, recommendation etc
- Datacenters
  - Topology, deadlockes in RDMA networks, debugging
- Mobile
  - Low-energy consumption network stack, network personalization
- Network monitoring and analysis
  - Using programmable switches
- Wireless (MIT)
- NF modeling verification
- DDoS