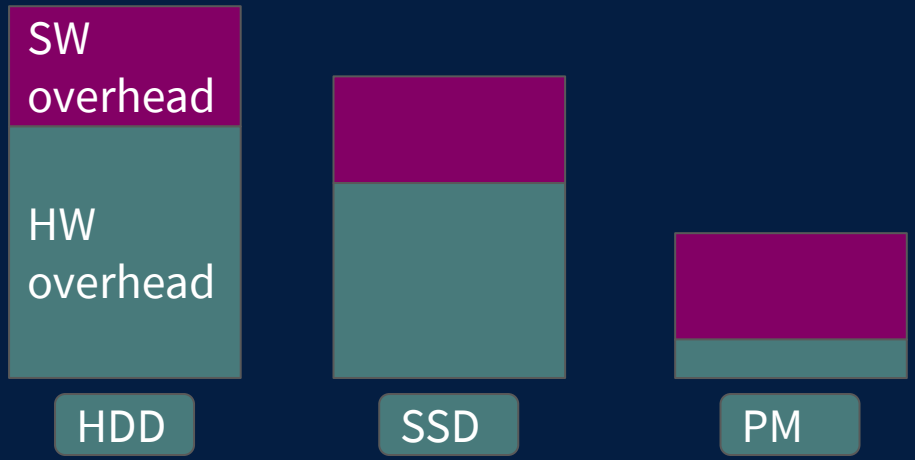
THE UNIVERSITY *of* EDINBURGH

Packets as Persistent In-Memory Data Structures

Michio Honda
University of Edinburgh
ACM HotNets 2021
10 November, online

Storage systems

Protect data over reboot or power loss



- Nova (FAST'16)
- NV-tree (FAST'15)
- NoveLSM (ATC'18)

Block devices **Byte addressable**

Where do software overheads come from?

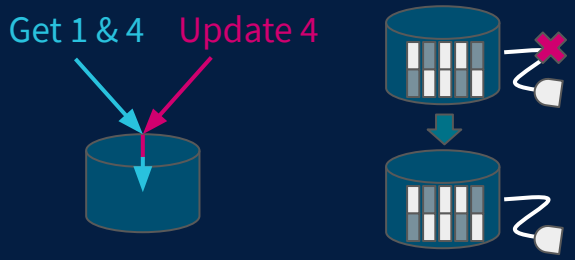
Storage properties

Integrity

- Ensure data is intact

Consistency

- Concurrent readers/writers
- Crash recovery



Searchability

- Organize data for quick read & write

Networking for PM storage systems

- RDMA
  - More CPU cycles for storage software
    - FileMR (NSDI'20)
    - Octopus (ATC'17)
    - Mojim (ASPLOS'15)

Can we enable efficient networked PM systems without RDMA?

Design options

- How to reduce storage software overheads?
  - In-storage computing 

  - External accelerator devices 

  - **Repurposing networking features to implement storage properties**
    - Checksums, timestamps, sequence numbers
    - NIC offloadings

Design rational 1

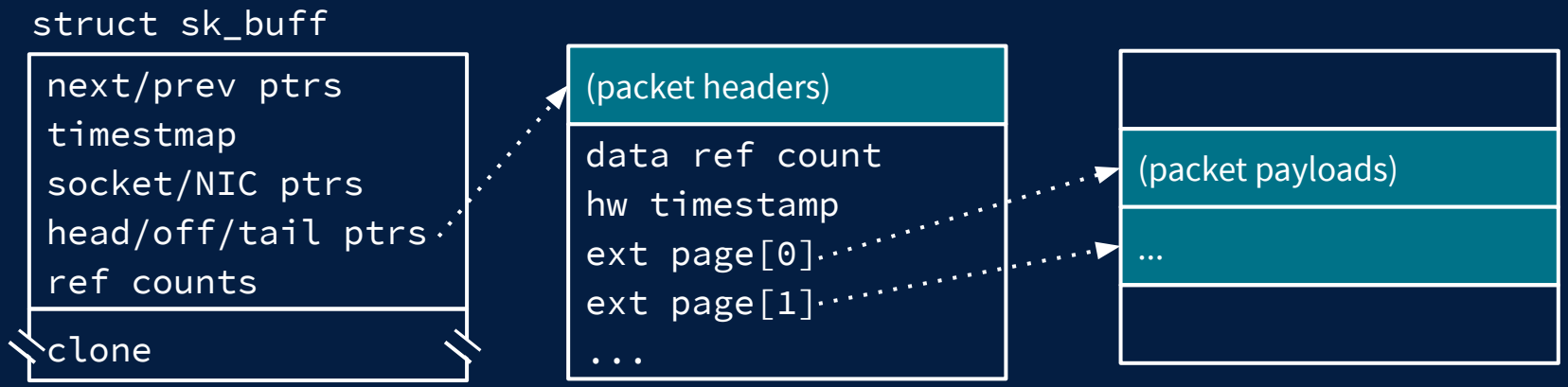
*Both network protocols and storage software are designed to survive **dumb**, **faulty** hardware*

- Networks: reorder, drop and corrupt packets
- Storage: reorder writes, corrupt data and falsely respond

Design isolation: in-memory and on-disk data structures

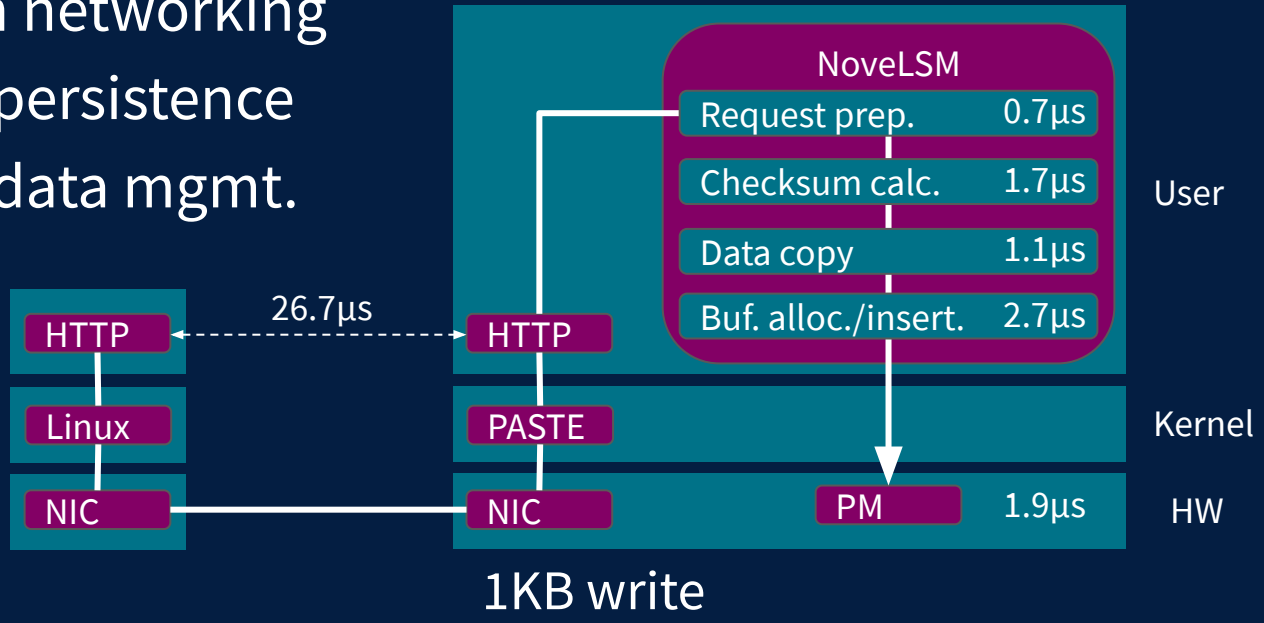
Design rational 2

Packet representation in the network stack is efficient in-memory data structures



Are data management overheads really high?

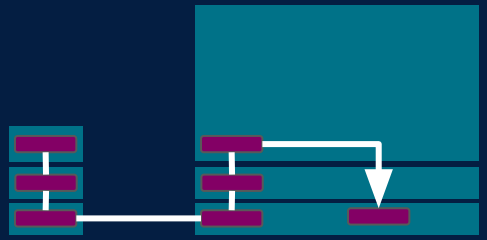
- 34.7µs in total
  - 26.7µs in networking
  - 1.9µs in persistence
  - 6.2µs in data mgmt.



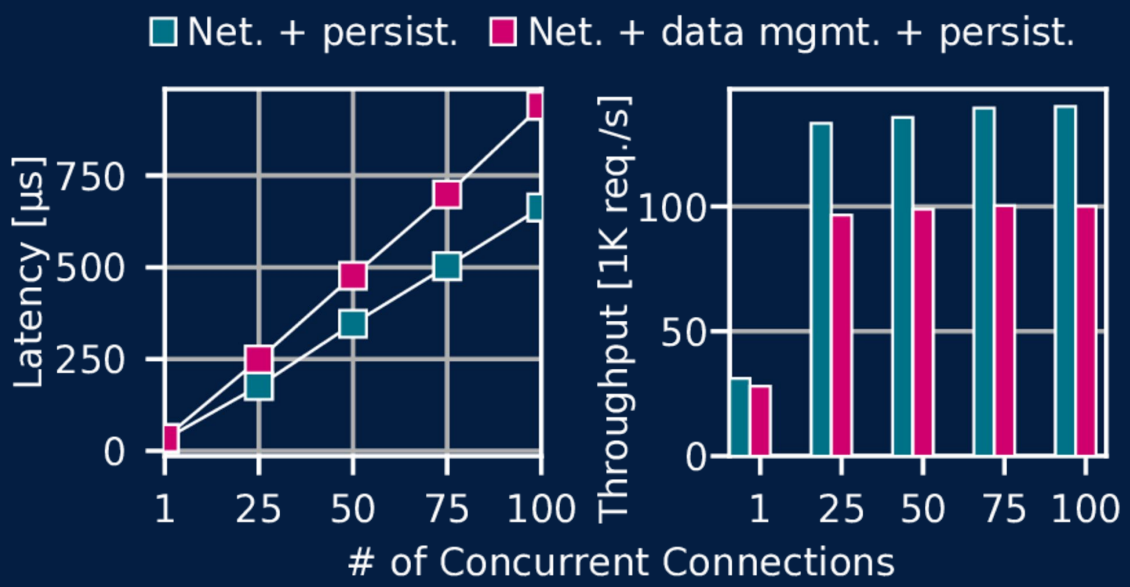
1KB write

Processing delays increase request backlog

Net. + persist.



Net. + data mgmt. + persist.



Data management overheads are significant

Summary

- We should turn the networking *overheads* into *assets* to reduce data management overheads
- Research agenda
  - Persistent packet metadata structures
  - NIC offloading
  - Transport protocols

https://micchie.net