[Master Thesis]

Cache Optimization of Virtual Network I/O to Achieve 100 Gbps

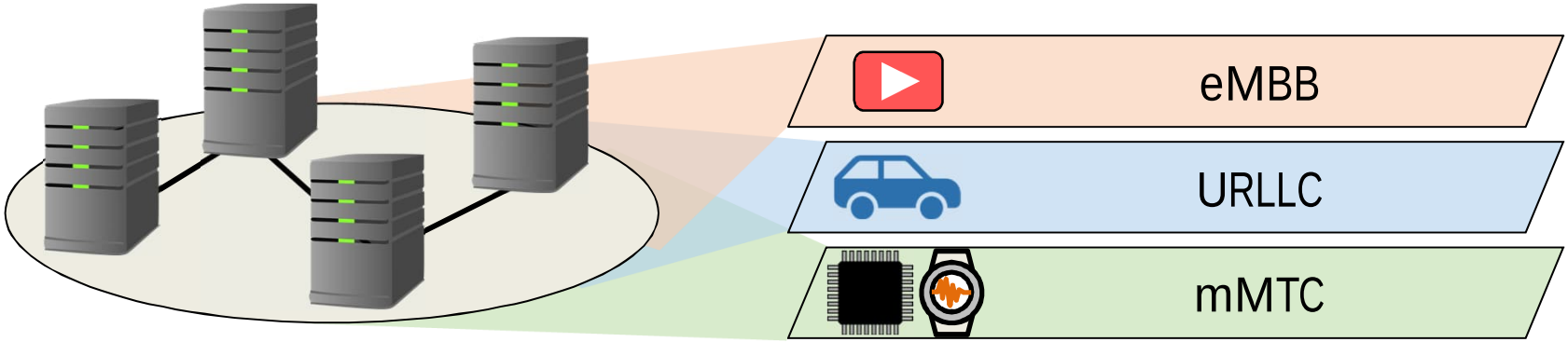
Graduate School of Engineering
Nagoya Institute of Technology
Daichi Takeya

2023/02/09

Cloud Native Network Functions

Replacing dedicated equipment with COTS servers!

Multiple (independent) logical networks

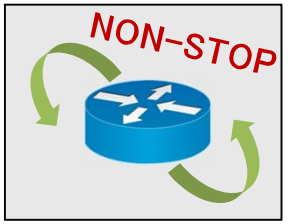


Core Network

Network Slicing

Cloud-Native Network Function

Secure Multi-Tenant



Easy-to-update

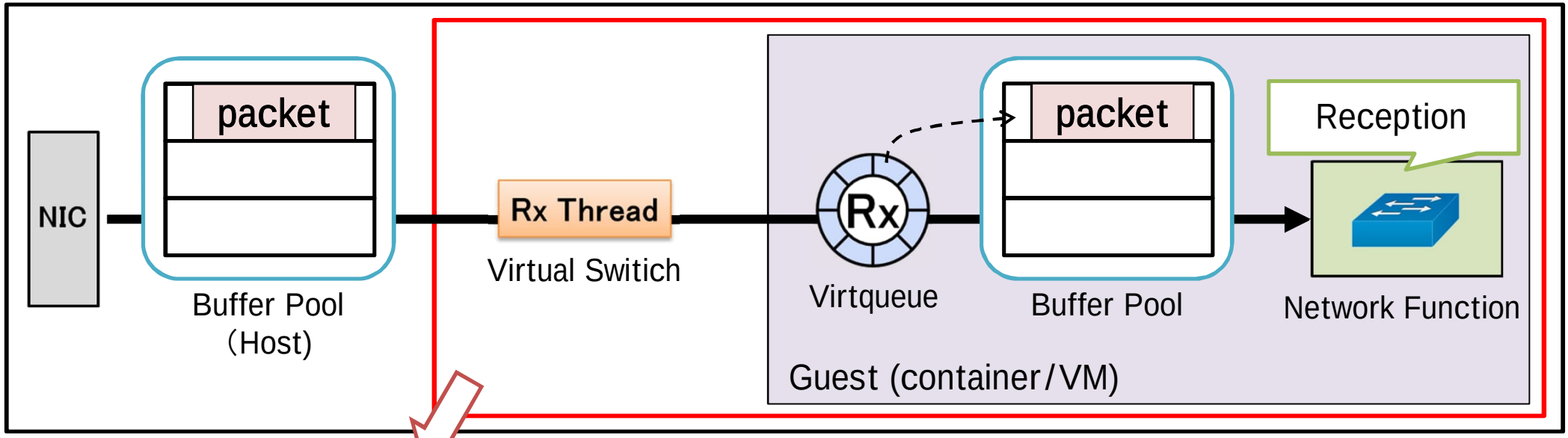
Gigantic network traffic!

CNF's poor performance

(1/100 of 6G's requirement)

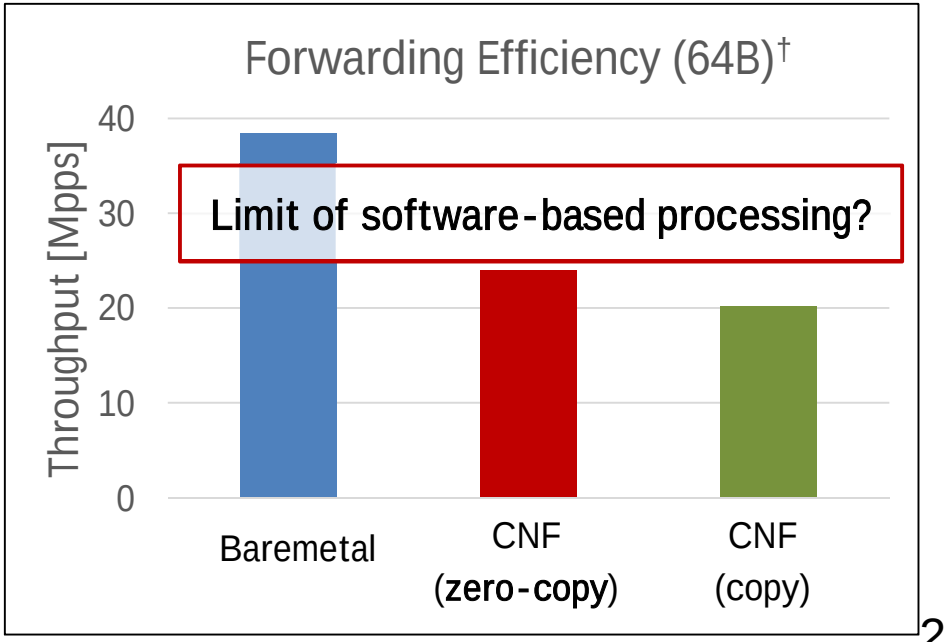
Virtual Network I/O in vhost-user/DPDK

Virtual Network I/O (Rx)



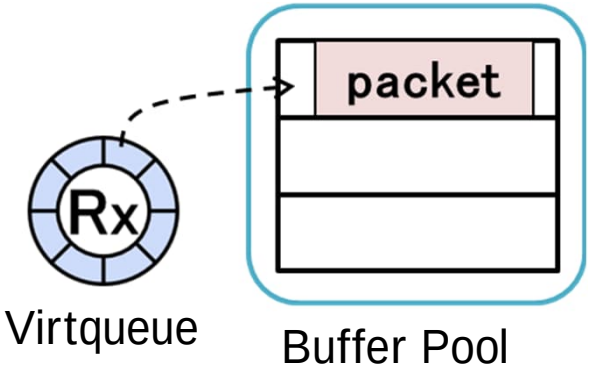
- Packet Copies } bottleneck? (No)
- Buffer Management } lightweight (can be bottleneck?)
- Virtqueue operations

Pursue the performance limit of software-based packet processing from the viewpoint of CPU caches



† R. Kawashima, "Software Physical/Virtual Rx Queue Mapping toward High-Performance Containerized Networking", IEEE Transactions on Network and Service Management, 2021

Effects of Cache Misses on Performance



- - Application/protocol independent
- - Support of various NIC offloading featues

100+ cache/memory accesses per packet

Two cache misses per packet

L1 Cache Hit Ratio	98.3%
L2 Cache Hit Ratio	54.0%
L3 Cache Hit Ratio	96.0%

Hit ratio of vhost-user/DPDK

100 Gbps (5.1ns / packet)

+8ns (two L1 cache misses)

39 Gbps (13.1ns / packet)

Significant decrease of the performance!

Example: Effects of cache misses (64B)

Thorough optimization of CPU cache usage!

The Purpose of This Study

- Understanding the pure bottlenecks of virtual network I/O
- Finding out a way of achieving ultimate performance

Current vhost-user/DPDK has a room for significant performance enhancement

Exhaustive investigations of possible designs and implementations

Our Target

Explicit Cache Control

CLFLUSH instructions

DPDK

Virtqueue Structure

Non-temporal instructions

Data Size

Packet Batching

DEMOTE instructions

Memory Alignment

Packet Buffer Structures

Software Prefetching

Zero-copy

BIOS-level

Entry Sizes

Cache Bandwidth

Caching Algorithms in Buffer Pool

Access Patterns

Data Sharing between Cores

Adjacent Cache Line Prefetch

SIMD instructions

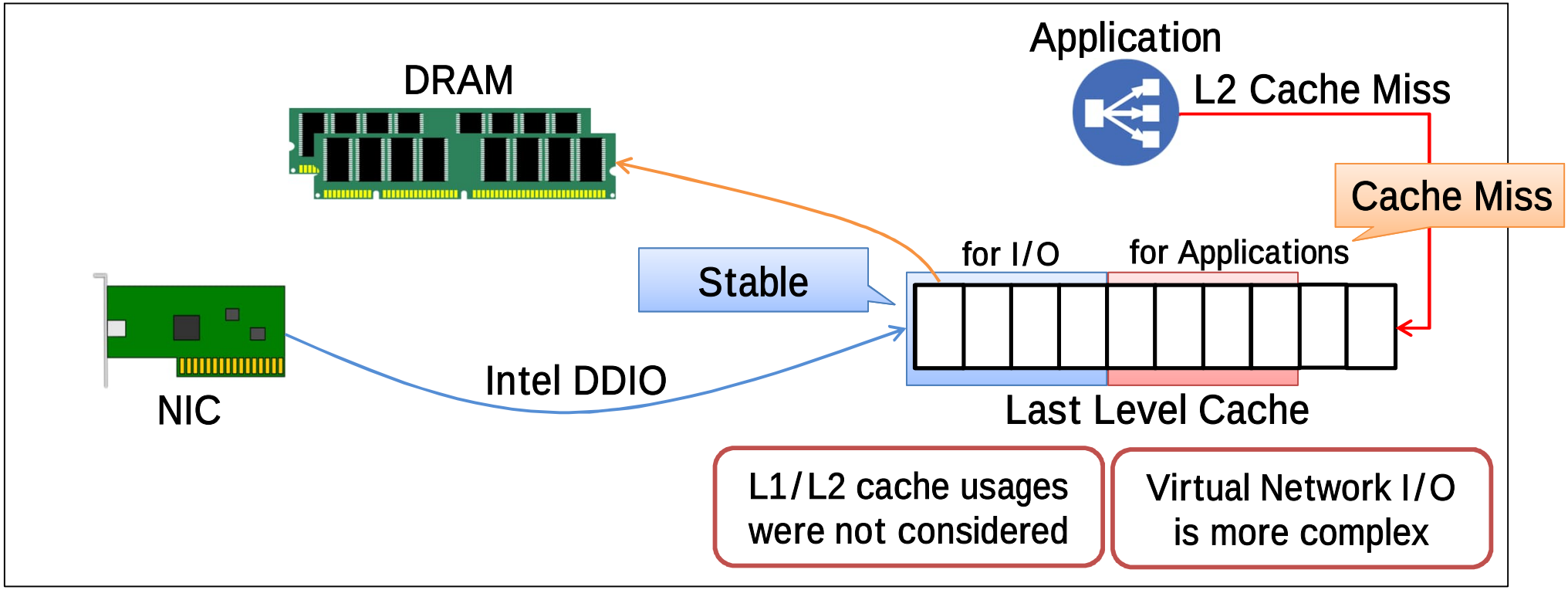
Hardware Prefetchers

etc...

141 evaluation items

Related Work

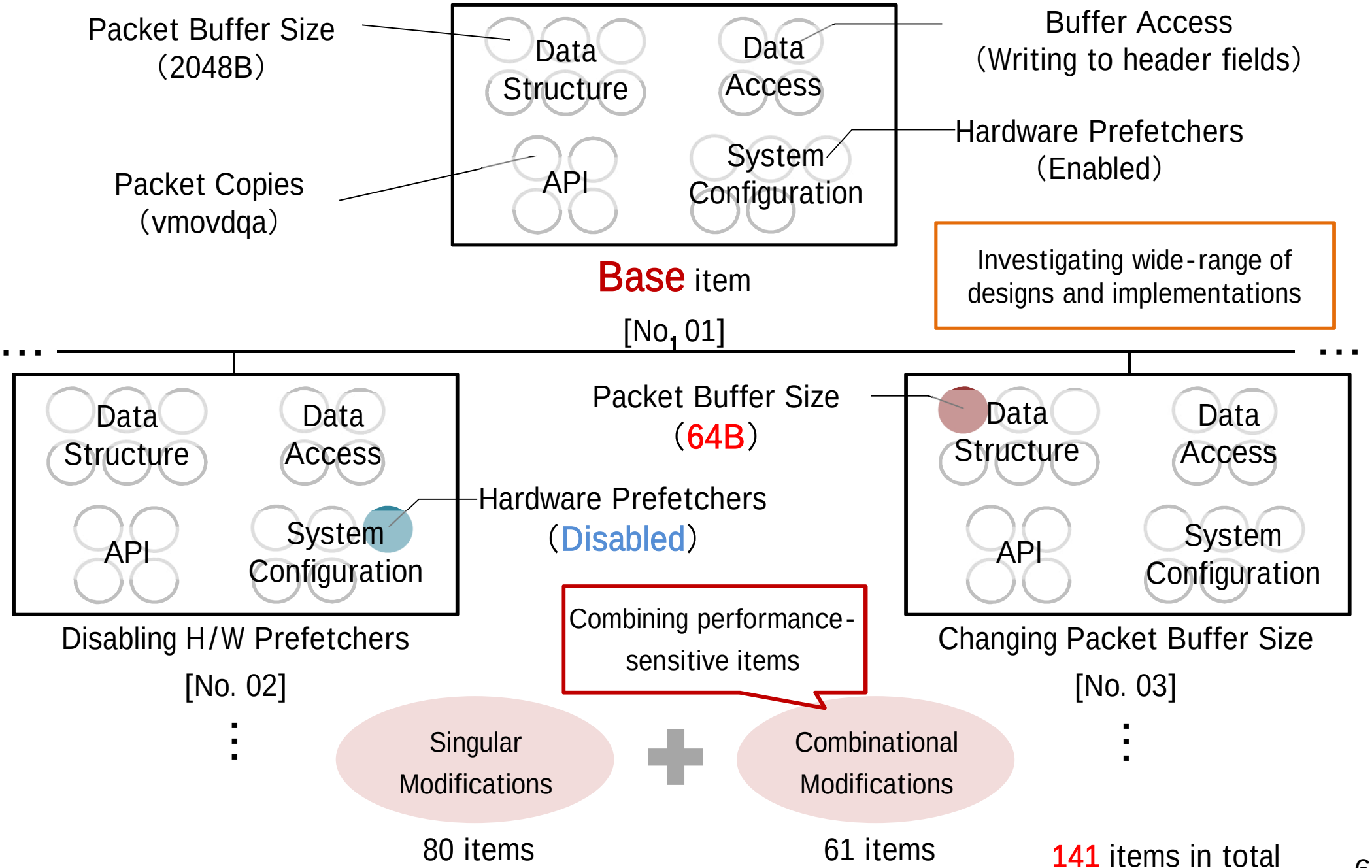
- Explicit LLC Separation between NIC and Applications
  - Stable performance can be expected



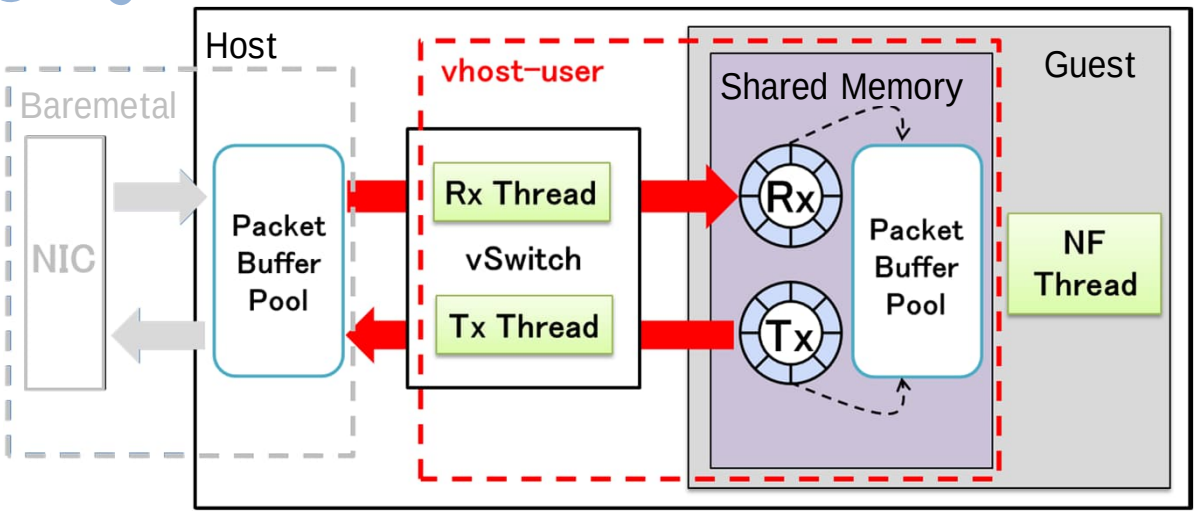
† A. Farshin, A. Roozbeh, G.Q.M. Jr., and D. Kostić, "Reexamining direct cache access to optimize i/o intensive applications for multi-hundred-gigabit networks", USENIX ATC 20

†† S. Thomas, R. McGuinness, G.M. Voelker, and G. Porter, "Dark packets and the end of network scaling" , ANCS '18

Design of Evaluation



EIVU Platform

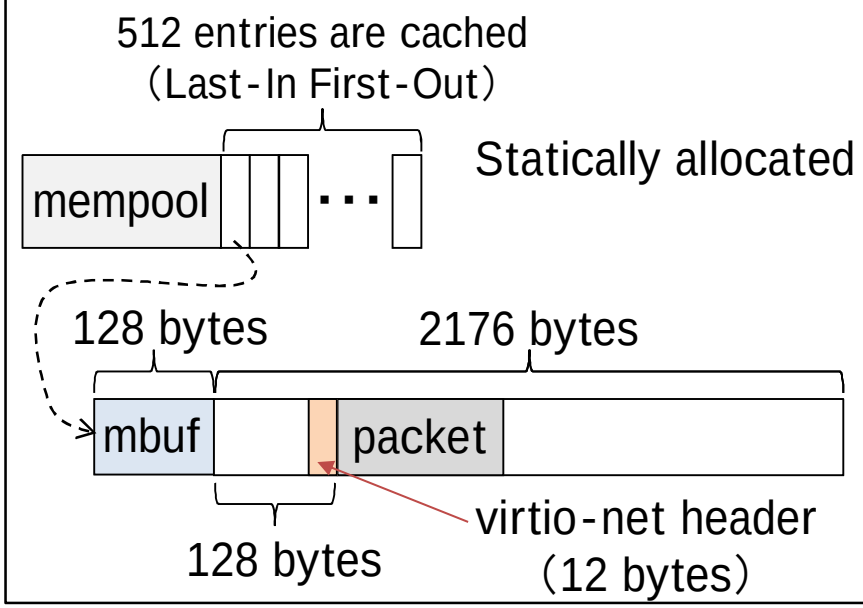


EIVU (Essential Implementation of Vhost-User)

Extracting the essential I/O processing

- Ease of modifications
- Hardware independent

Ex.: Structure of Packet Buffer



DPDK-like implementation

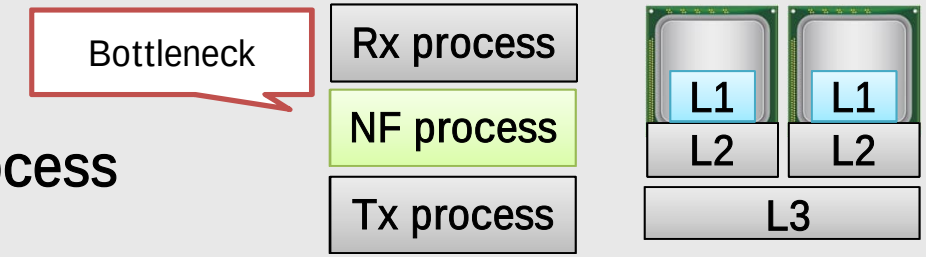
- Software Prefetching
- Lock-free virtqueues
- Packet batching
- Polling-based etc.

Throughput (64B): 28 Mpps
(vhost-user: 16 Mpps)

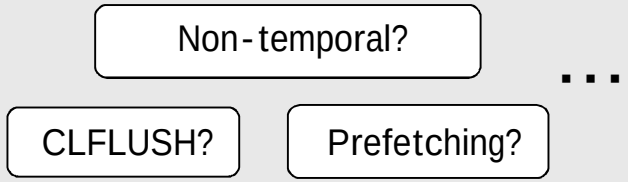
Evaluations are conducted on EIVU

Evaluation

1. Investigating the effect of CPU cache usage on the performance of the NF process

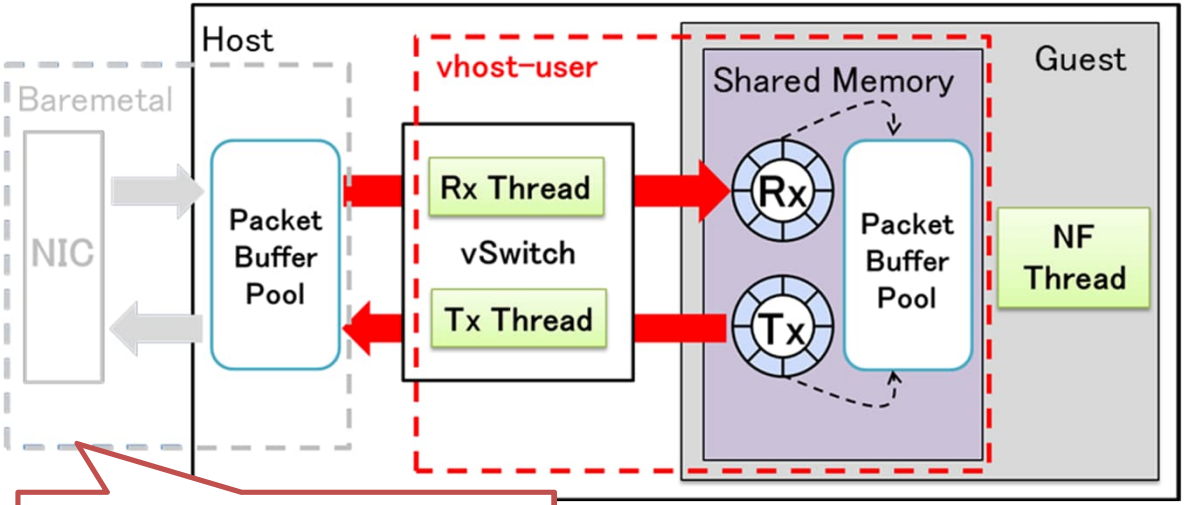


2. Identifying the essential performance factors



3. Considering a way to achieve ultimate performance

Evaluation Environment



NIC independent

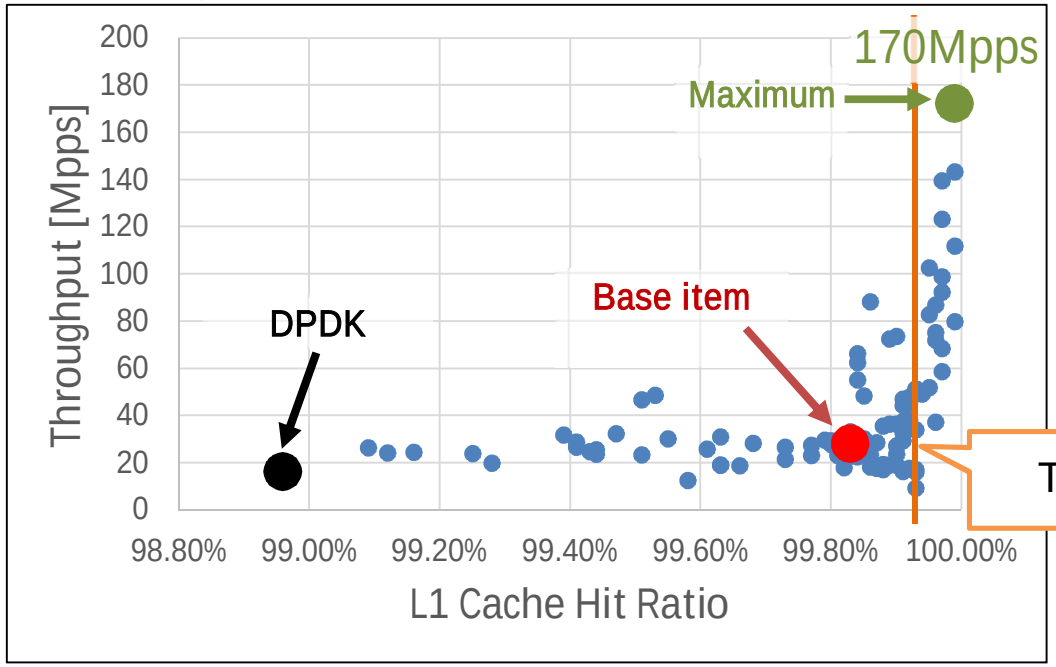
Measured items

- Throughput
- No. of cache accesses
- No. of caches misses
- Efficiency of prefetching
- Stalled cycles for LFB
- No. of RFO requests etc…

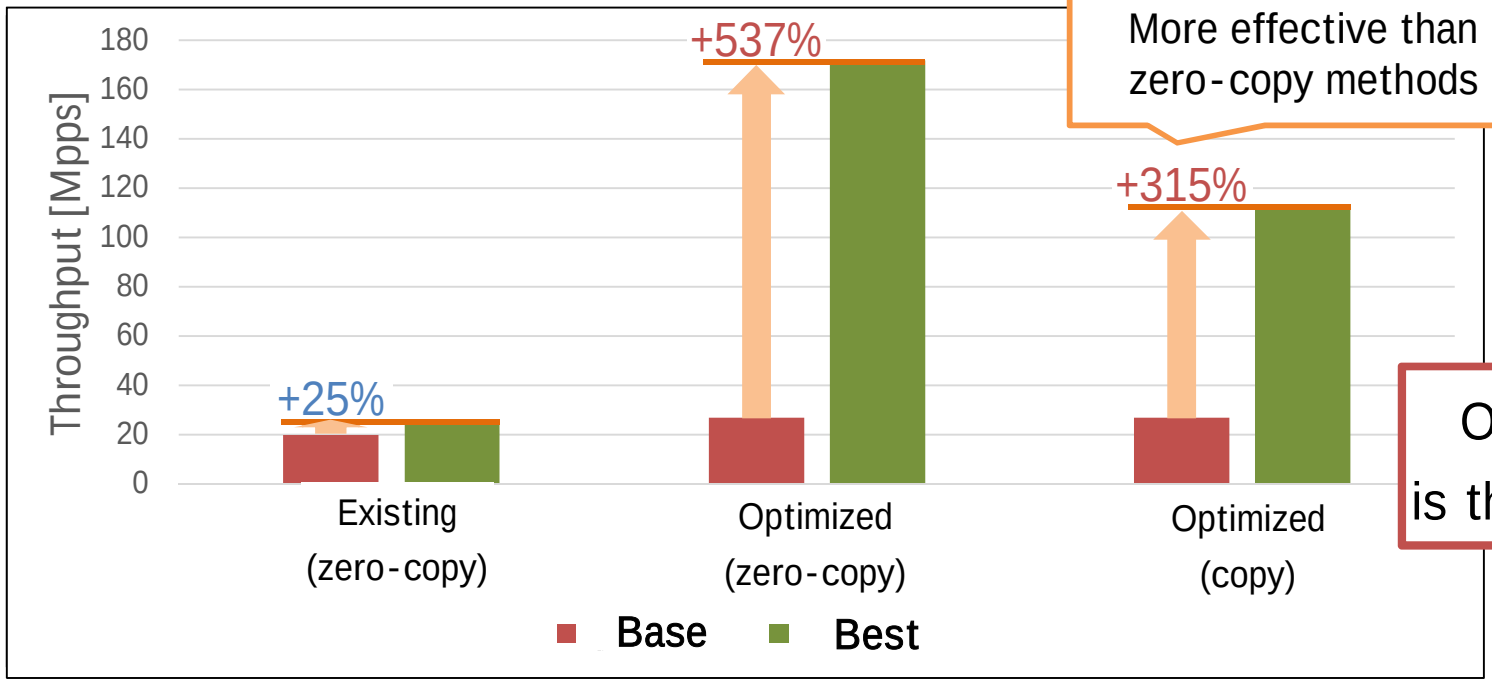
20 items in total

	Servers			
	Server 1	**Server 2**	**Server 3**	**Server 4**
		CPU		
Type	Core i9 11900K	Core i7 9800X	Core i9 13900K	Threadripper 5965WX
Clock	3.5 GHz	**3.8 GHz**	3.0 GHz	**3.8 GHz**
L1d	48 KB	**64** KB	48 KB	32 KB
L2	0.5 MB	1.0 MB	**2.0** MB	0.5 MB
L3 (shared)	16 MB	16 MB	**36** MB	32 MB
		Memory		
Clock	3200 MHz	2133 MHz	**4800** MHz	3200 MHz
		Performance		
EIVU (Base)	28 Mpps	19 Mpps	16 Mpps	45 Mpps
DPDK	16 Mpps	16 Mpps	16 Mpps	18 Mpps

Effect of L1 Cache Usage on Throughput

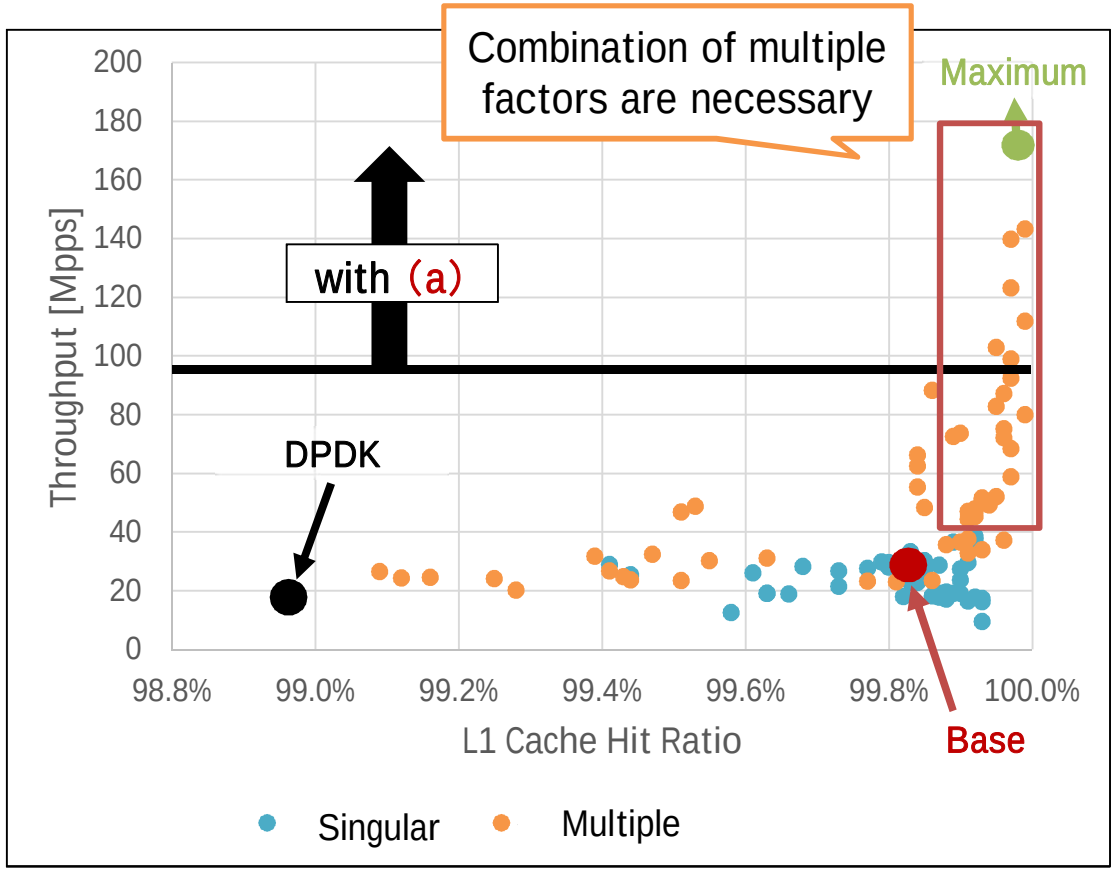


Significant increasement was seen for over 99.9% hit ratio
(L1 hit ratio of DPDK was 98.9%)



Optimizing L1 cache usage is the key to achieve 100 Mpps

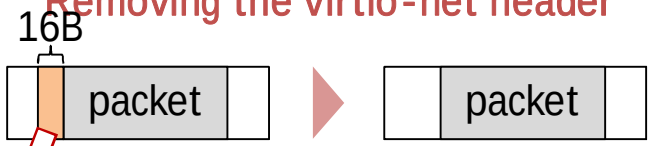
The Essential Performance Factors



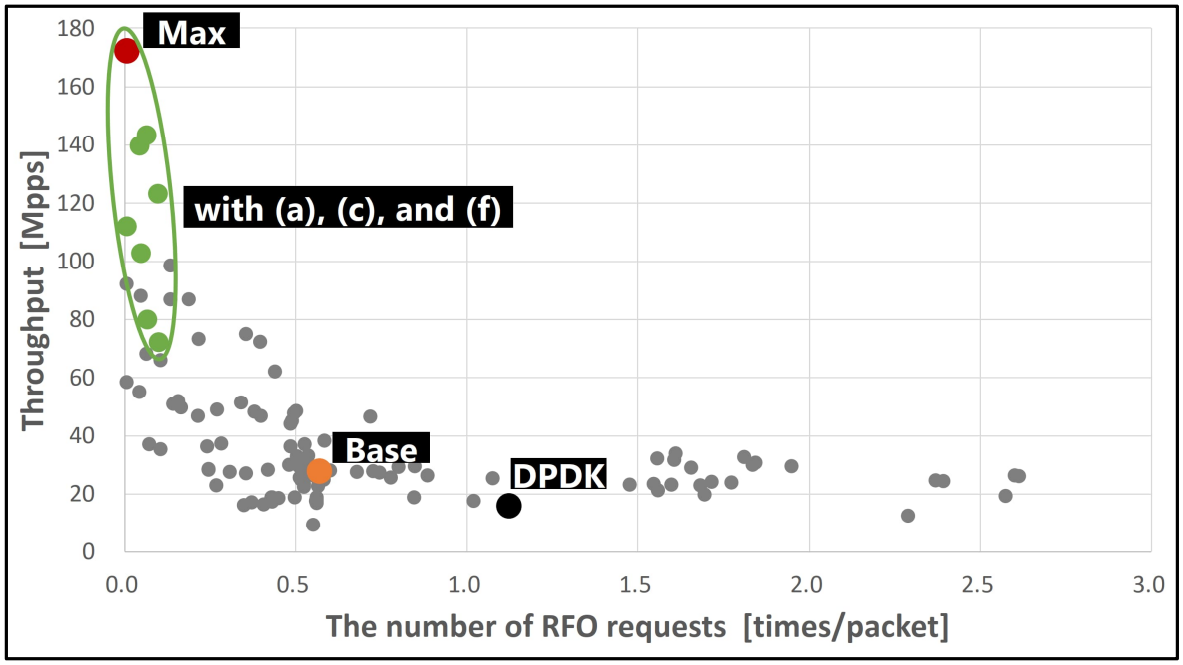
Factor	Description
(a)	Removing the virtio-net header 16B packet ▶ packet
(b)	Reducing the size of queue entry Rx 16B ▶ 4B
(c)	Removing the buffer header Data Field ▶ Data Field Header Field (128B)
(d)	Removing the tailroom packet (2KB) ▶ packet (Packet Size)
(e)	Zero-copy Host packet Guest Packet Copy
(f)	Prolonged packet batching 32 ▶ 1024 Host ··· Guest

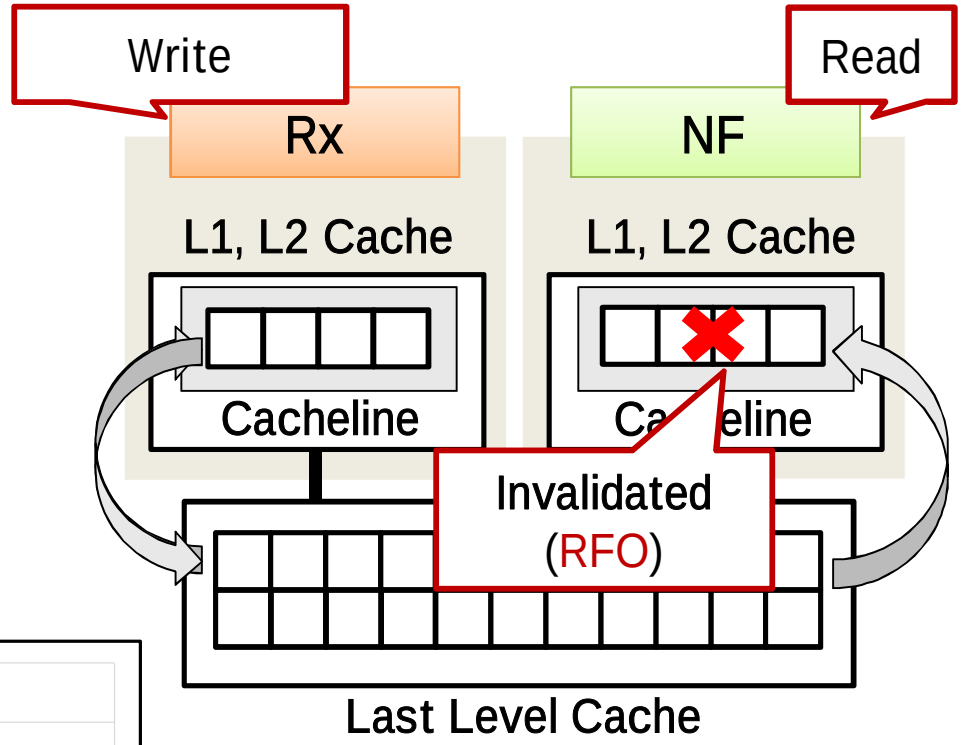
- **Re-design of data structures** is imperative!
- **Realistic design** is the most challenging theme!

Major Cause of Cache Misses

Factor	Description
(a)	Removing the virtio-net header 

1. Written by the Rx process
2. Read by the NF process



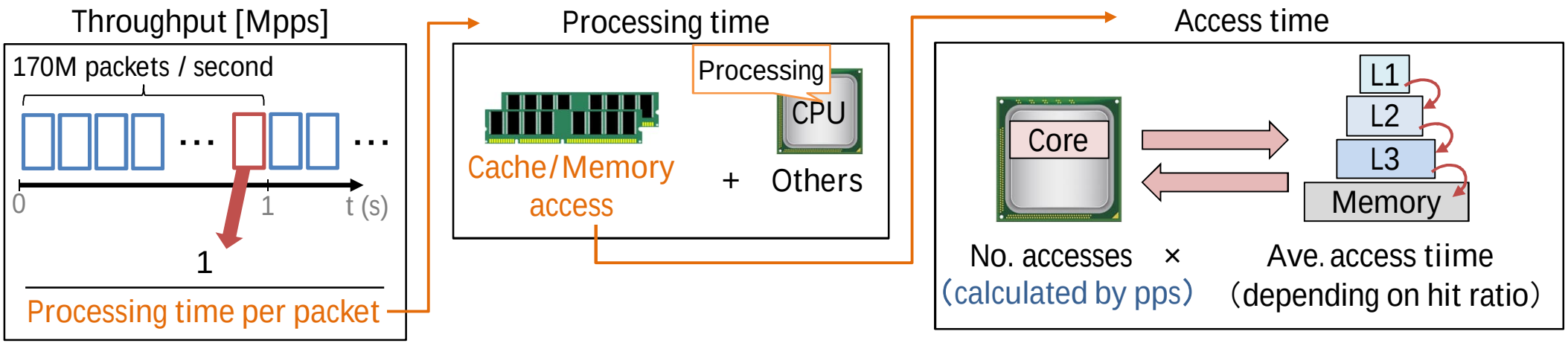


Keeping the cachelines on the cache can cause frequent invalidations!

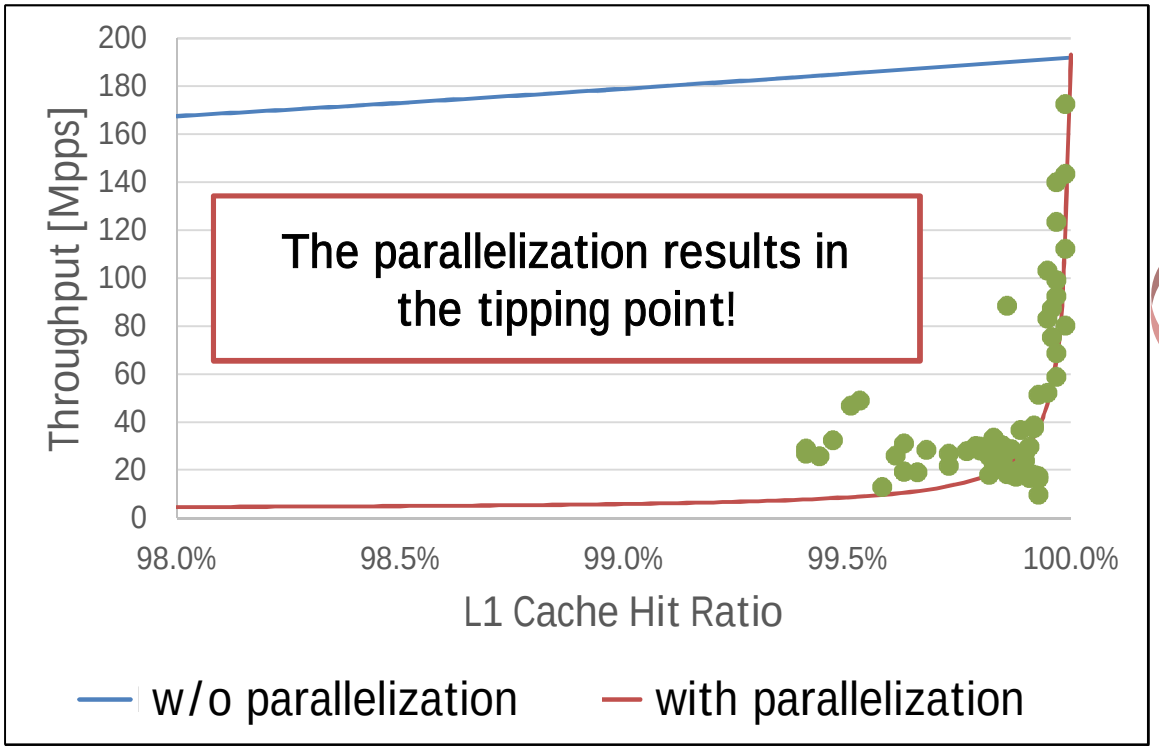
Minimizing cache invalidations is the key to exceed the tipping point!

Mathematical Analysis

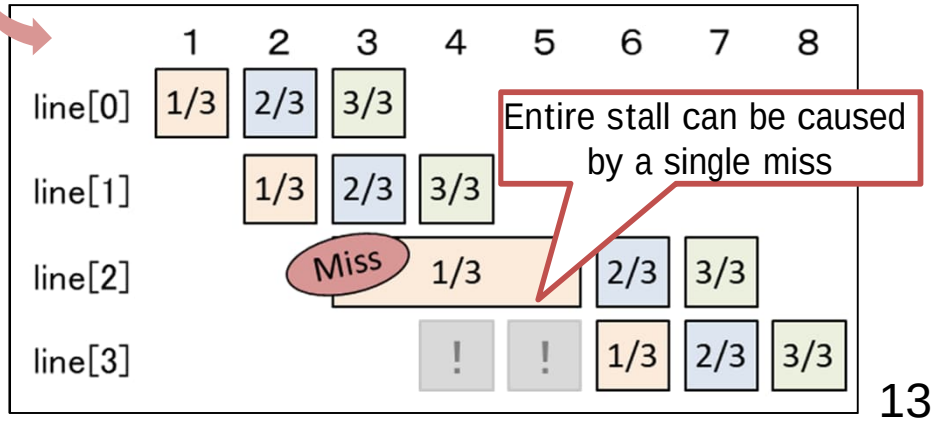
- Why does the tipping point appear?



No. of actual accesses was
far larger than the calculation!



Cache accesses must be parallelized!



Conclusion

- Identifying the bottlenecks of virtual network I/O
  - CPU (L1) cache usage is the key to understand the performance
  - 100+ Mpps throughput is theoretically possible by exceeding the tipping point
  - L1 cache misses negate the significant effect of the parallelization

- Future Work
  - Practical design of data structures needs to be re-devised