Understanding Roadblocks in Virtual Network I/O:

A Comprehensive Analysis of CPU Cache Usage

D. Takeya[†]

R. Kawashima[†]

Y. Nakayama[‡]

T. Hayashi[‡]

H. Matsuo[†]

Contact: *R. Kawashima* <kawa1983@ieee.org>

[†]Nagoya Institute of Technology [‡]BOSCO Technologies, Inc.

POINTS

Goal

Identify the performance bottleneck

Idea

Focus on CPU L1 cache usage

Past focus: packet copy

Result

Show a potential to achieve 100+ Mpps

6x higher than DPDK/vhost-user

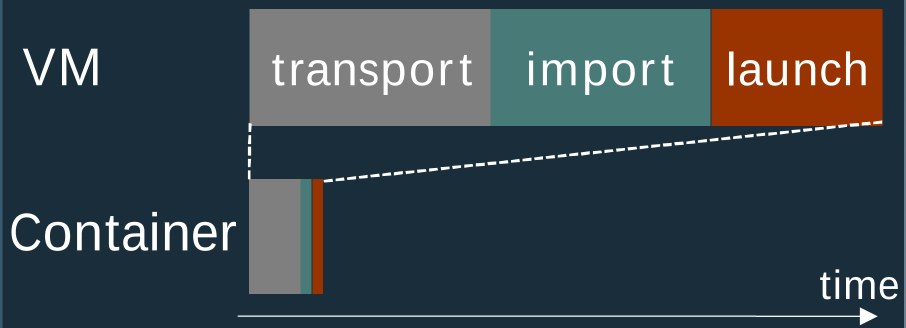
AGENDA

1. INTRODUCTION
   CNF, Virtual Network I/O, Zero-copy
2. CACHE & VIRTUAL NETWORK I/O
   Why CPU cache?, CPU cache usage
3. OUR STUDY
   Goals, Approach, Evaluation design
4. RESULTS
   Environment, Throughput, Analysis
5. CONCLUSION
   Conclusion and future work

CNF (Cloud-native Network Function)

Container-formed network functions

Lightweight



Agile adaptation

Portable

- Standardized
  - Image format & Runtime
  - OCI
- Immutable

Reproducibility

Multiplexing

- Separation
  - CPU, memory, ...
  - Namespace (network, ...)

Resource efficiency

Eco-system

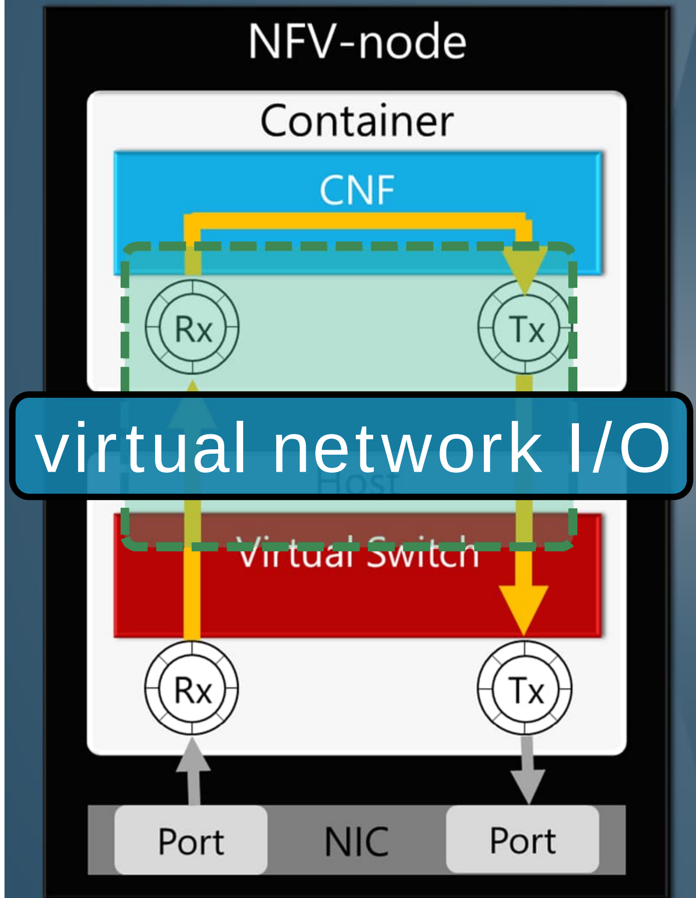
- Container-as-a-Service (CaaS)
- CNCF
  - e.g.) Kubernetes

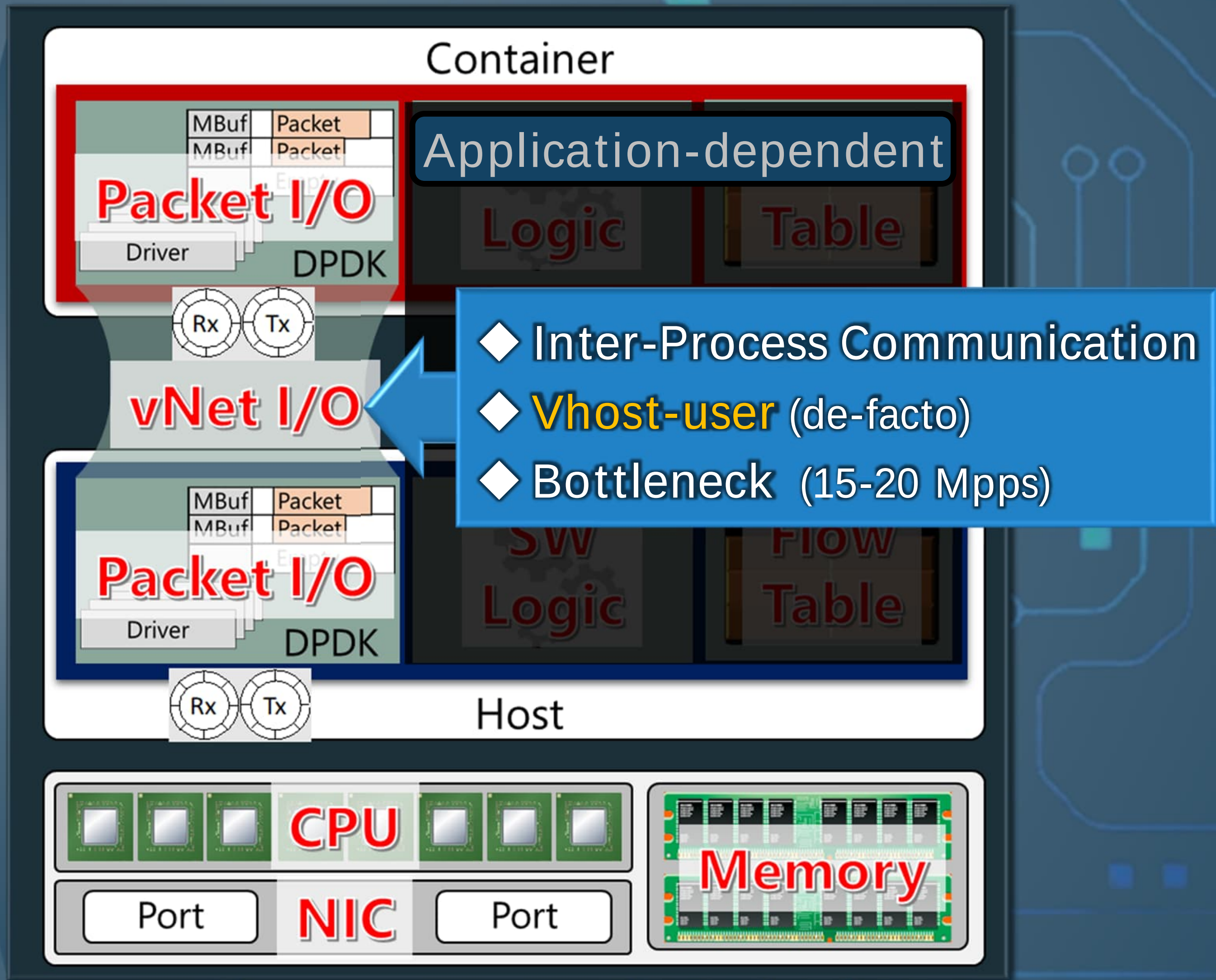
Technology asset

Can CNFs replace other forms of NFs?

Virtual Network I/O

Performance critical part of CNFs

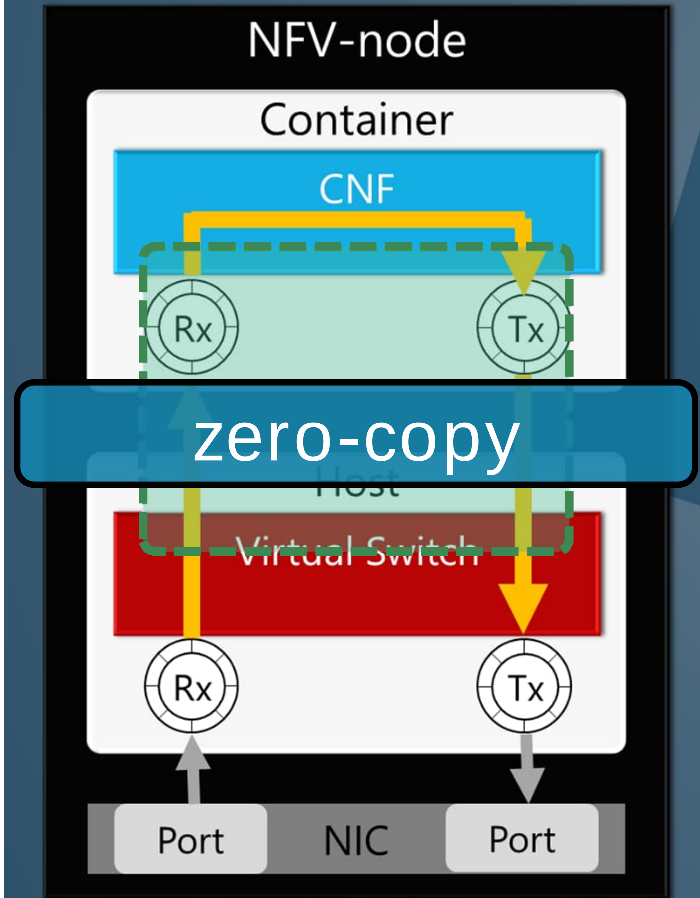




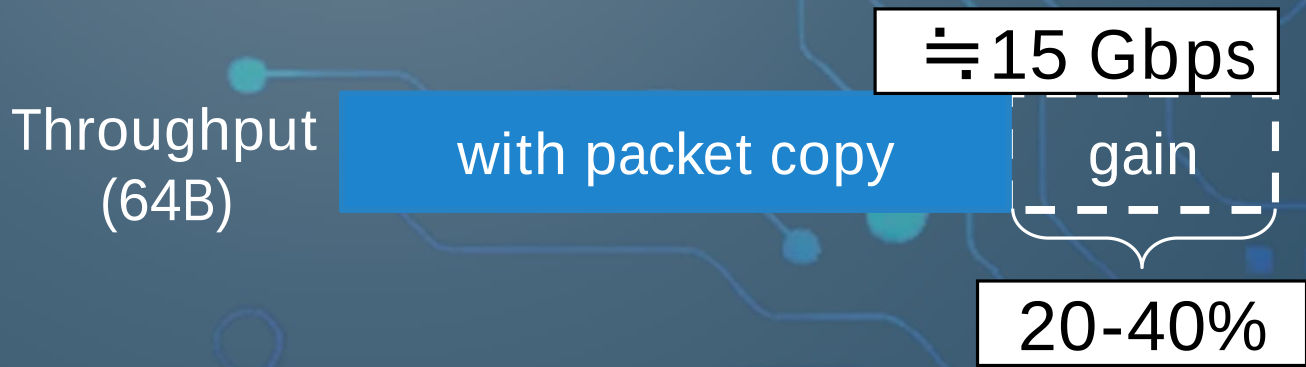
Why does virtual network I/O halve throughput?

Zero-copy

Past studies focused on packet copy



- Packet (memory) copy is removed
- Various implementations
  - NetVM (2014), OpenNetVM (2016)
  - ZCopy-Vhost (2017)
  - IOVTee (2018)
- Marginal effect on performance



Isn’t packet copy the true bottleneck?

AGENDA

1. INTRODUCTION
CNF, Virtual Network I/O, Zero-copy

2. CACHE & VIRTUAL NETWORK I/O
Why CPU cache?, CPU cache usage

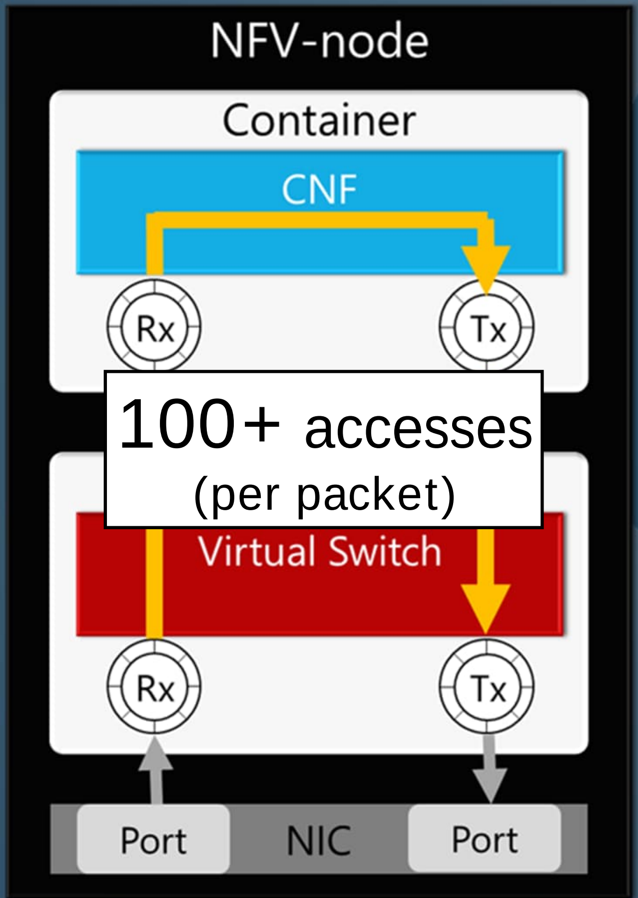
3. OUR STUDY
Goals, Approach, Evaluation design

4. RESULTS
Environment, Throughput, Analysis

5. CONCLUSION
Conclusion and future work

Why CPU Cache?

Every little bit adds up



- Cache is always accessed
- Virtual Network I/O
  - Due to packet copy? or queue handling?
- Penalty of cache misses

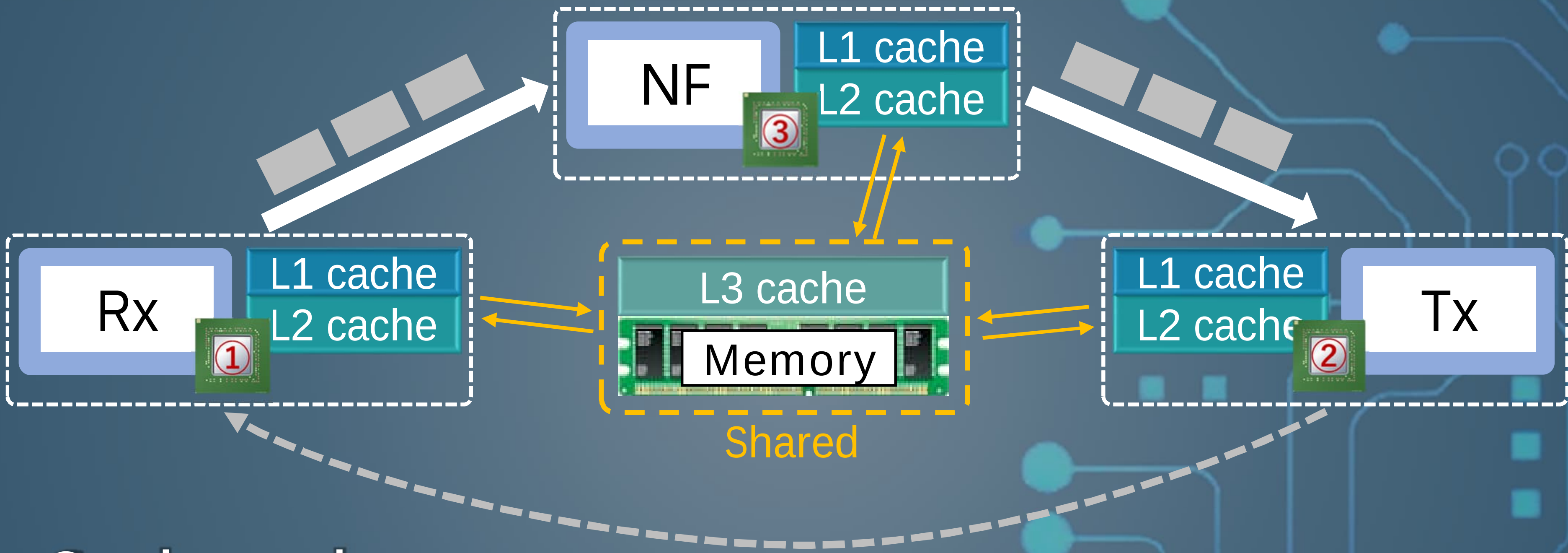
Performance cost

Cache miss ≒ Packet copy (64B)

Why does virtual network I/O need frequent cache accesses?

CPU Cache Usage (in Virtual Network I/O)

Three-body problem in cache/memory



- Cache coherency
  - Invalidation, RFO (Read For Ownership)
- False sharing
  - H/W prefetching implicitly causes false sharing

Are design and implementation of packet framework ideal?

AGENDA

1. INTRODUCTION
   CNF, Virtual Network I/O, Zero-copy
2. CACHE & VIRTUAL NETWORK I/O
   Why CPU cache?, CPU cache usage
3. OUR STUDY
   Goals, Approach, Evaluation design
4. RESULTS
   Environment, Throughput, Analysis
5. CONCLUSION
   Conclusion and future work

GOALS

Understand the true bottleneck in virtual network I/O

Unveil the effect of cache usage on performance

Assess a possibility of fair speed-up

APPROACH

Exhaustive experiments and analyses

Base
(DPDK-equivalent)

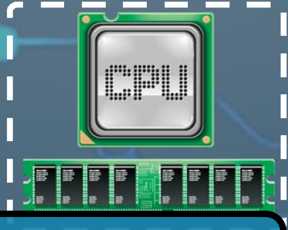
+

- Various structural designs
- Various implementations
- Various configurations

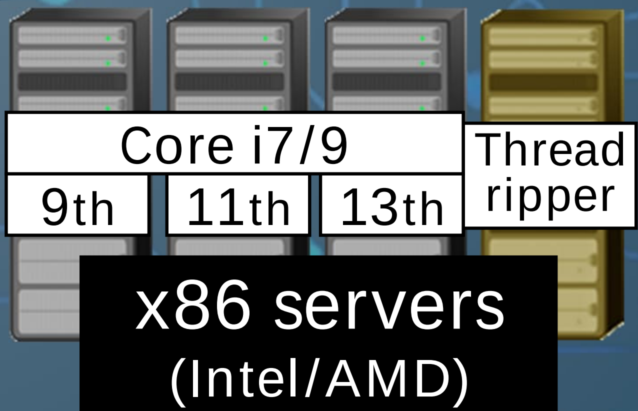
141 evaluation items

EIVU platform

Real H/W devices



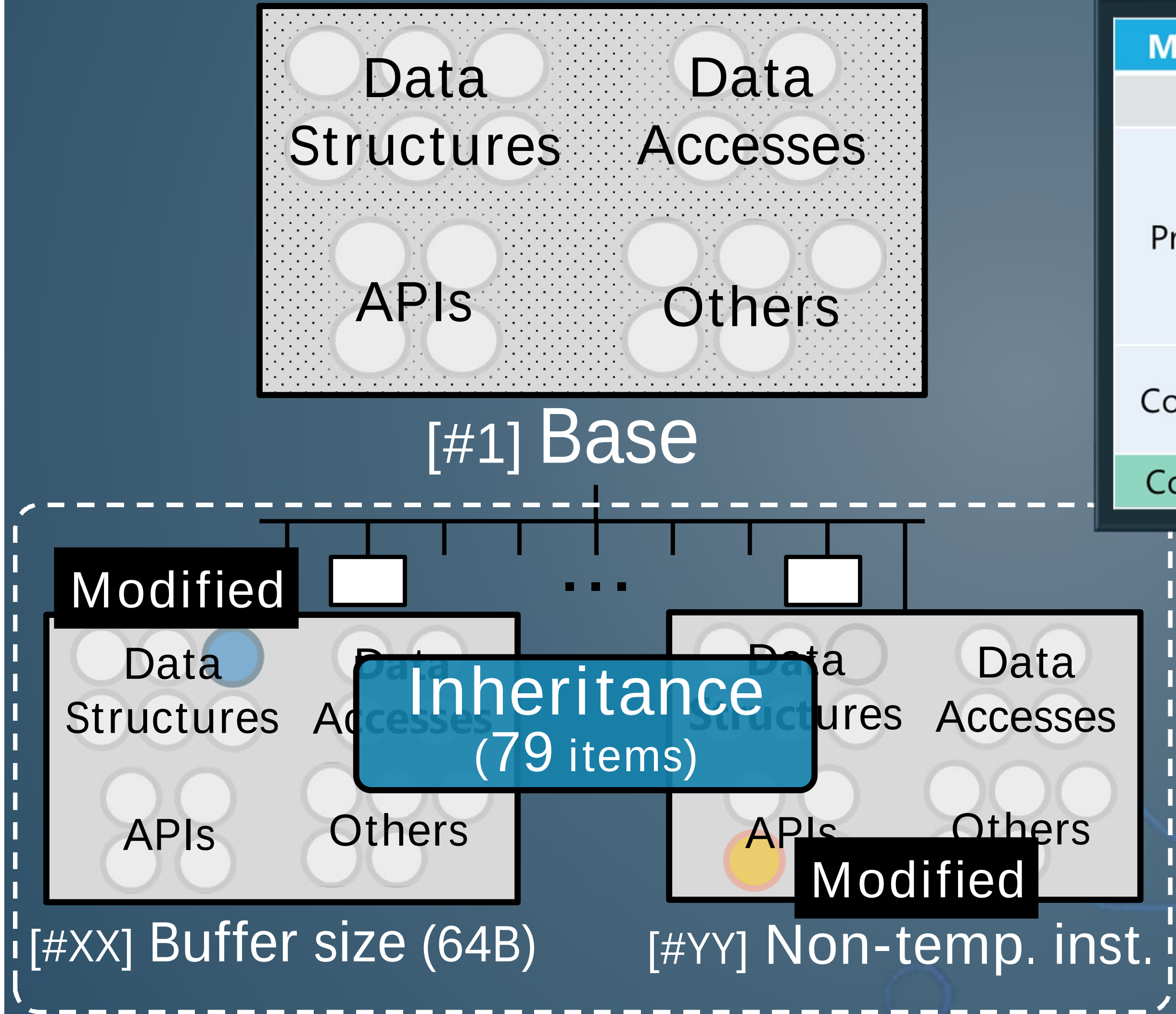
Actual cache behavior



Unveil the effect of cache usage on performance

Evaluation Design

Inheritance and Multiplexing



Modification	Category	Details
Base	Baseline item (DPDK-equivalent)	
Programmatic	Data Structure	I/O queue, packet buffer, …
	Data Access	read/write
	API	memory copy, cache control
	Others	zero-copy
Configurational	Runtime	various sizes
	BIOS	H/W prefetching
Combinational	Combination of above items	

https://sdnnitech.github.io/EIVU/eval/evaluation.html

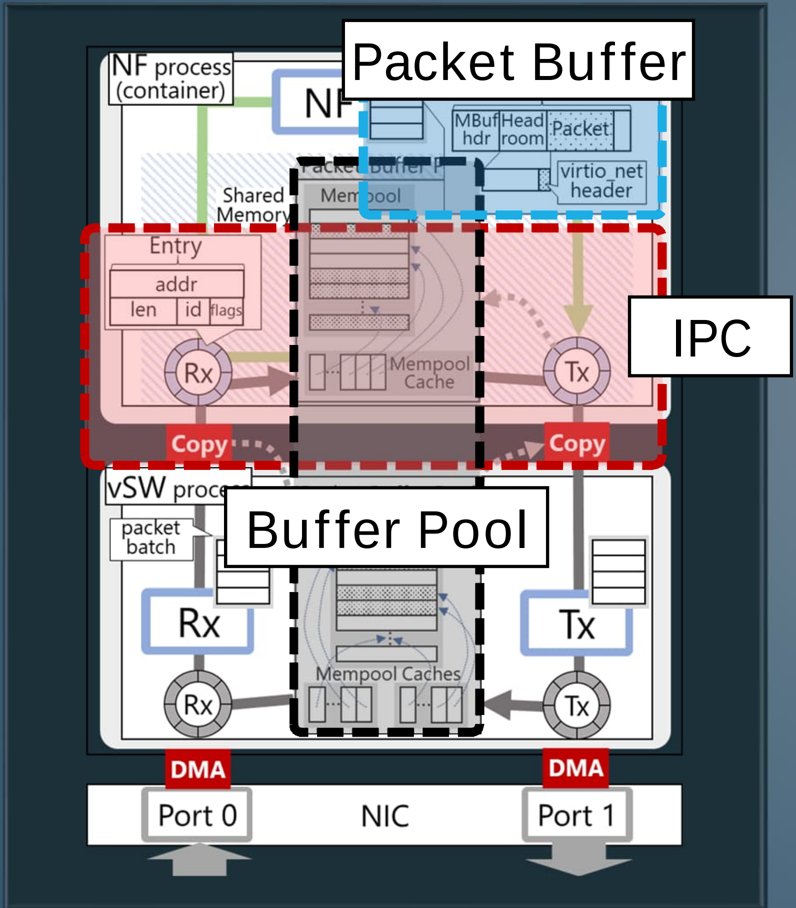
Multiplexing
(61 items)

141 items in total!

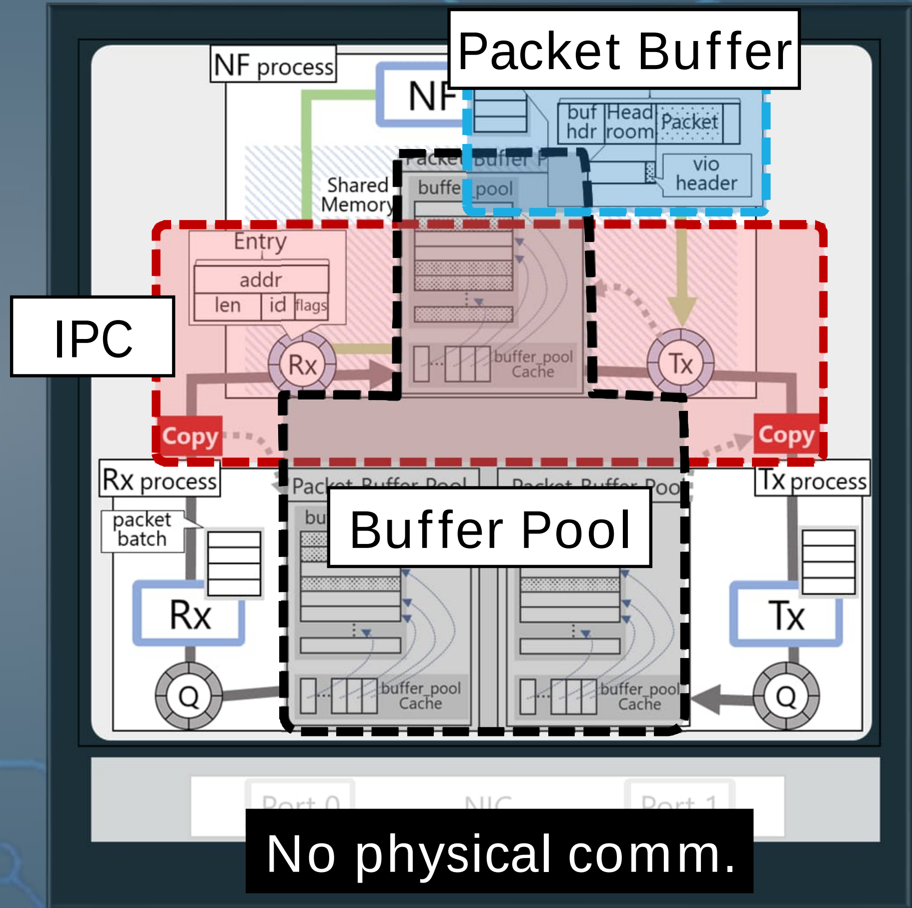
Extremely customizable framework is essential

EIVU (Essential Implementation of Vhost-User)

Easy-to-customize evaluation framework



DPDK/vhost-user



EIVU

https://github.com/sdnnitech/EIVU

Equivalent design/implementation and performance

AGENDA

1. INTRODUCTION
   CNF, Virtual Network I/O, Zero-copy
2. CACHE & VIRTUAL NETWORK I/O
   Why CPU cache?, CPU cache usage
3. OUR STUDY
   Goals, Approach, Evaluation design
4. RESULTS
   Environment, Throughput, Analysis
5. CONCLUSION
   Conclusion and future work

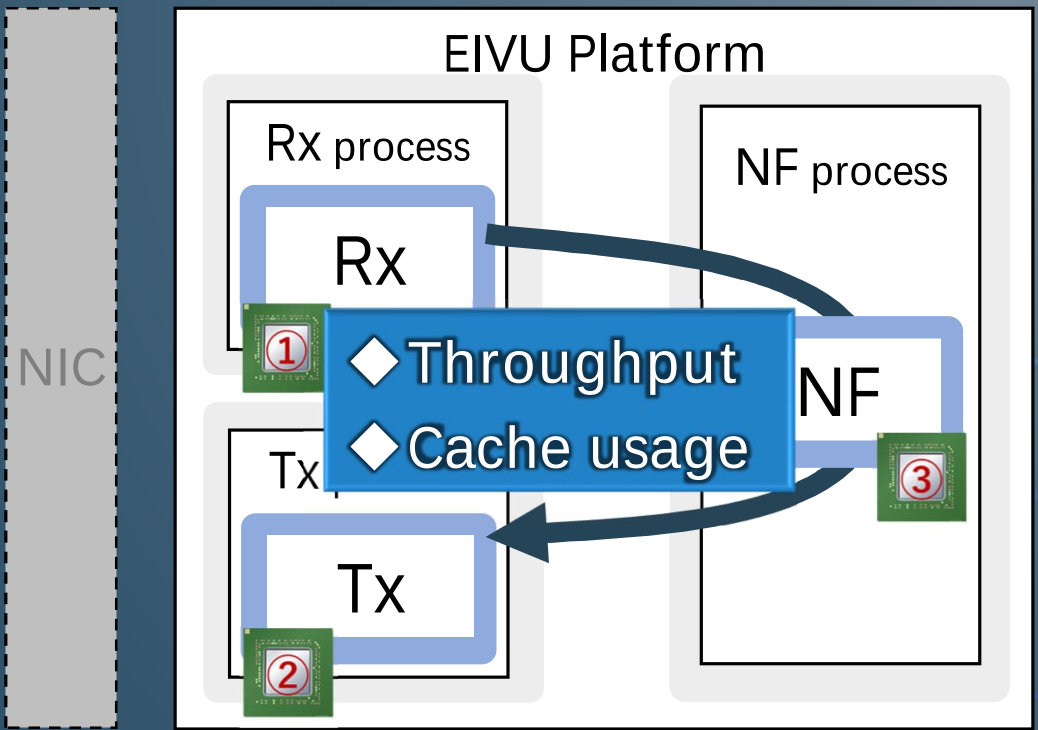
Contents & Environment

To what extent does cache usage affect?

What to show

Cache usage > Packet copy

More important

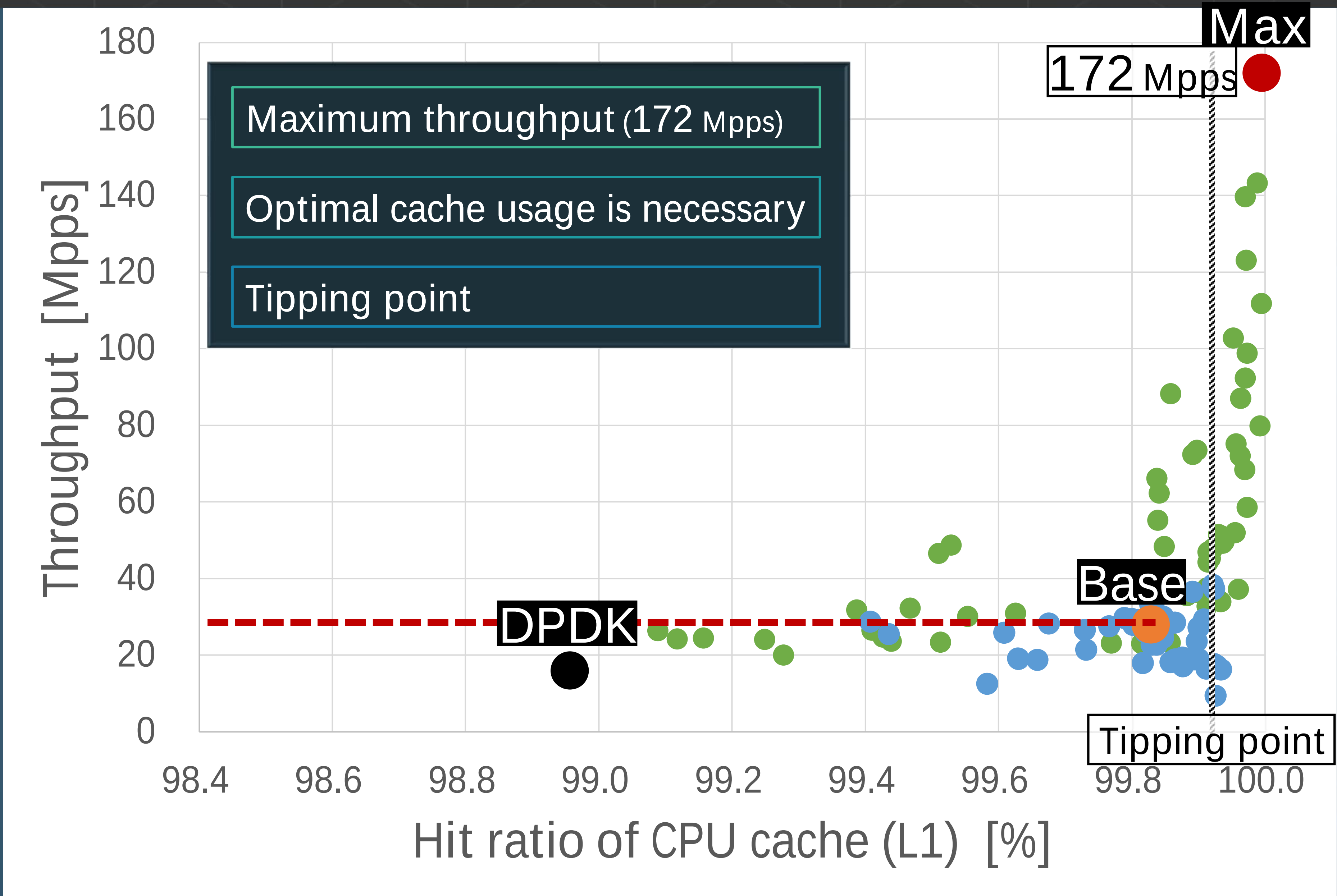


Machine spec.

	Server 1	Server 2	Server 3	Server 4
	Servers			
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

How did the results go?

Throughput vs. L1 Cache Usage

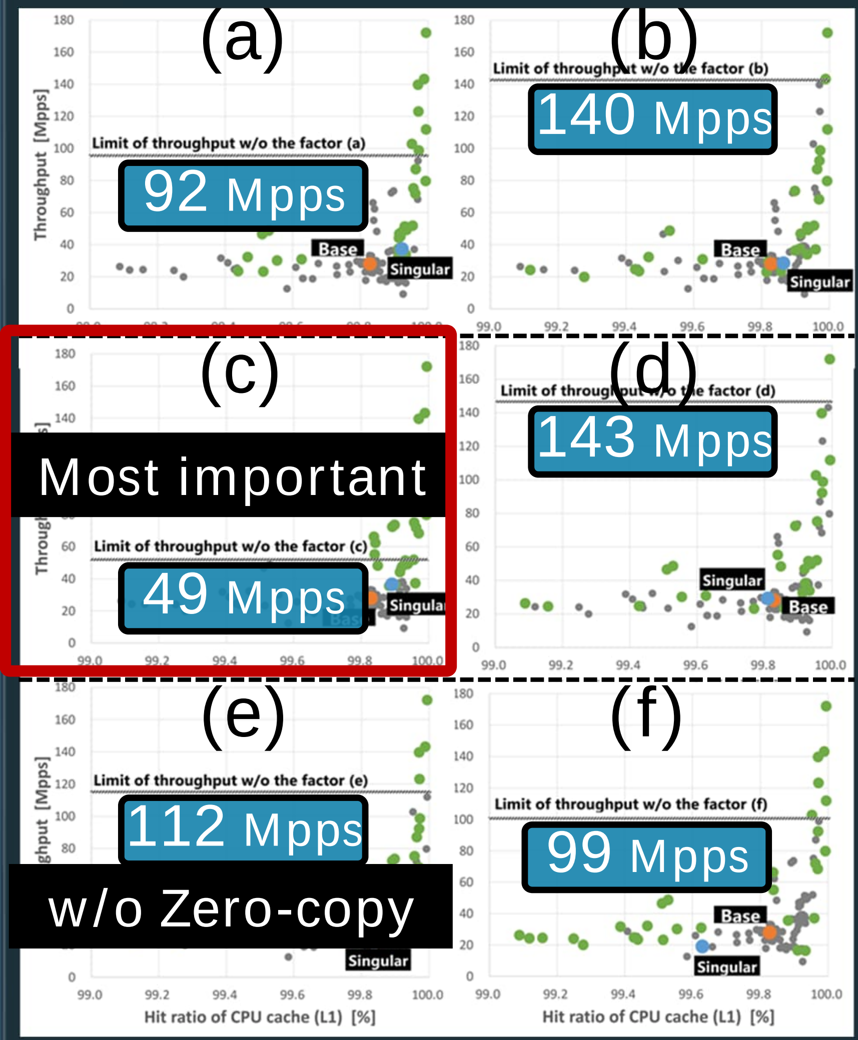


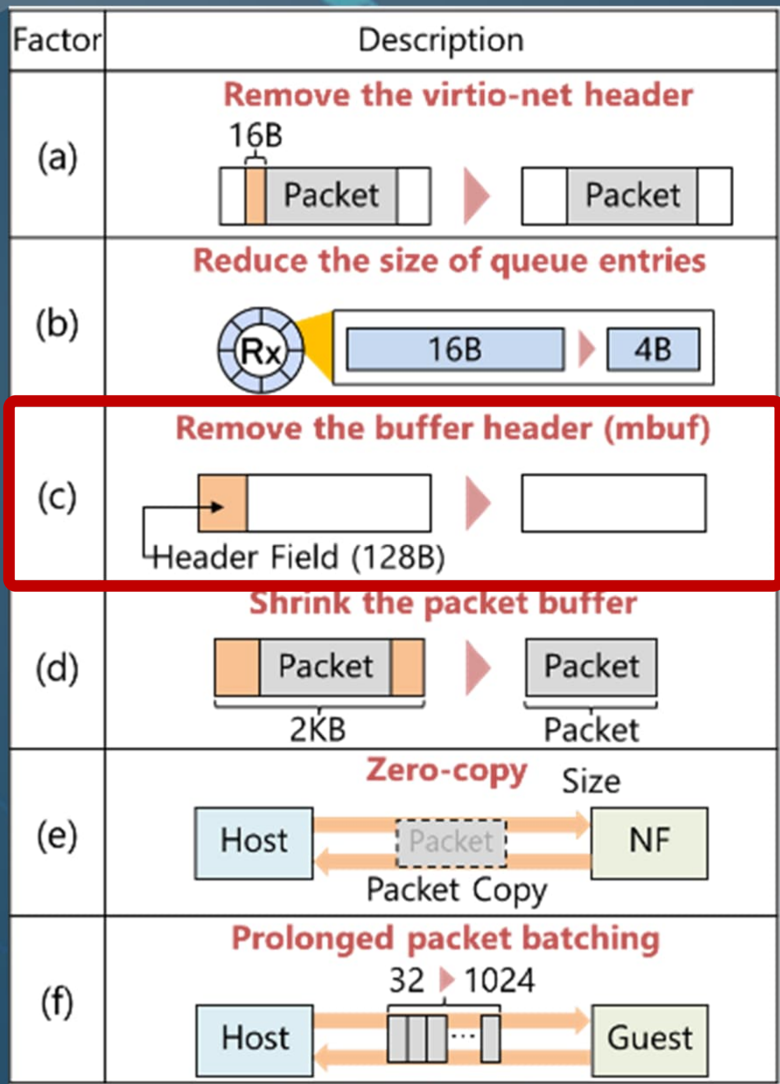
What was the performance bottleneck?

Analysis

Look deep inside the best-case item!

Six factors

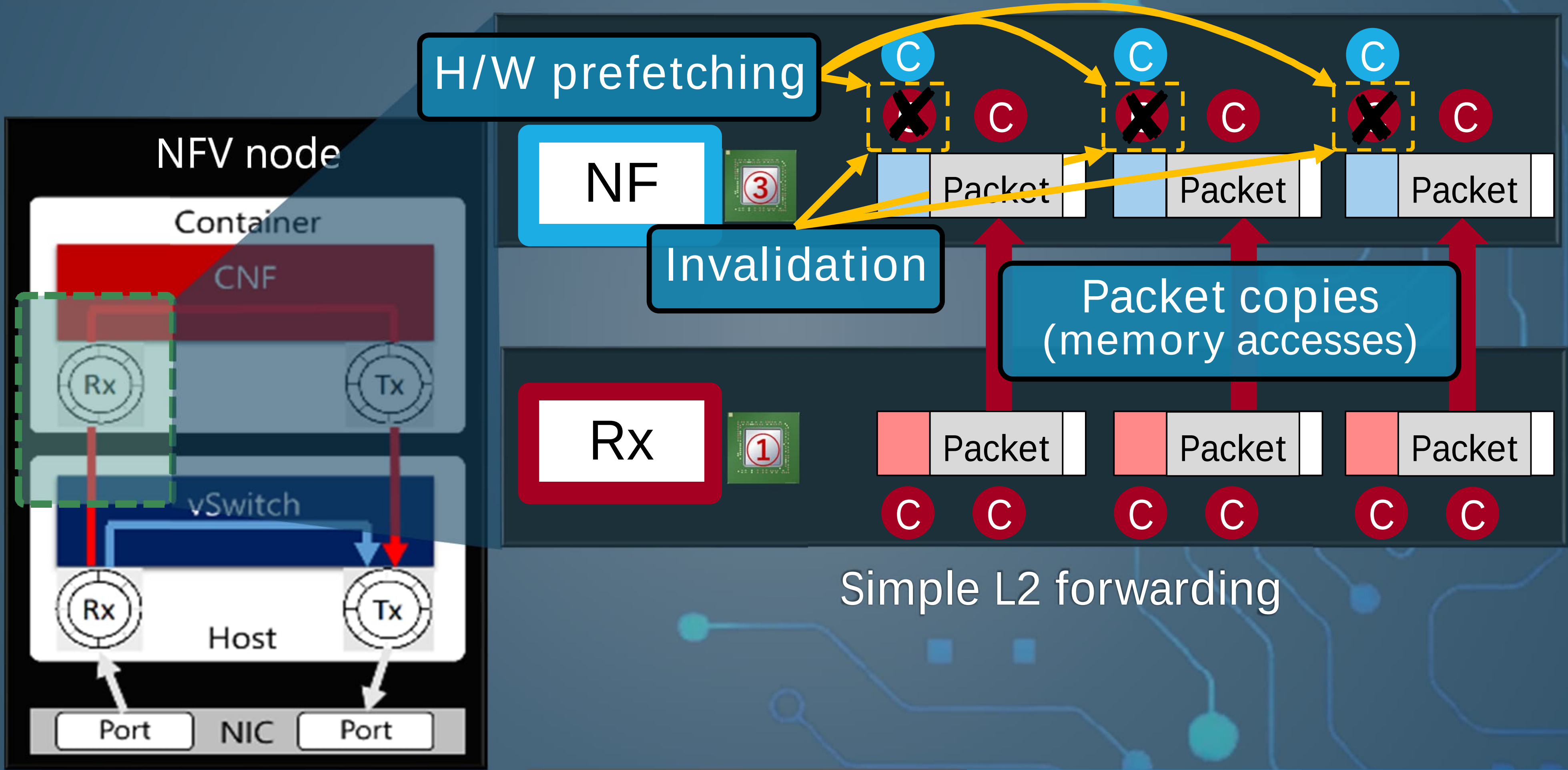




Why is factor (c) so influential?

Performance Bottleneck

The buffer header causes implicit conflicts



Future challenge

Re-design of packet buffer structure

AGENDA

1. INTRODUCTION
   CNF, Virtual Network I/O, Zero-copy
2. CACHE & VIRTUAL NETWORK I/O
   Why CPU cache?, CPU cache usage
3. OUR STUDY
   Goals, Approach, Evaluation design
4. RESULTS
   Environment, Throughput, Analysis
5. CONCLUSION
   Conclusion and future work

Conclusion and Future Work

Theme: Performance issue of virtual network I/O

	Then	Now
Throughput:	15-20 Mpps	100+ Mpps (potential)
Focus:	Packet copy	CPU cache usage
Approach:	Zero-copy	Re-design (structure)

Not the true bottleneck

- ◆Over 99.99% of L1 hit ratio is necessary
- ◆Implicit cache conflicts need to be avoided

Challenge: Re-design of packet buffer structure

RESOURCES

EIVU platform

https://github.com/sdnnitech/EIVU

Evaluation design

https://sdnnitech.github.io/EIVU/eval/evaluation.html

Result

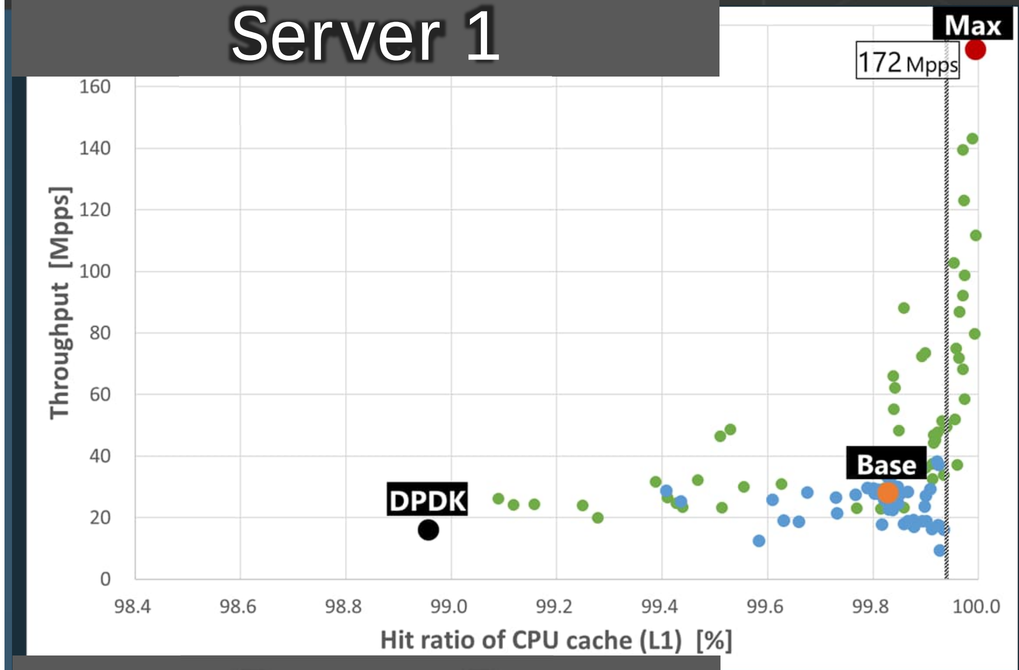
https://sdnnitech.github.io/EIVU/eval/results.html

Mathematical analysis

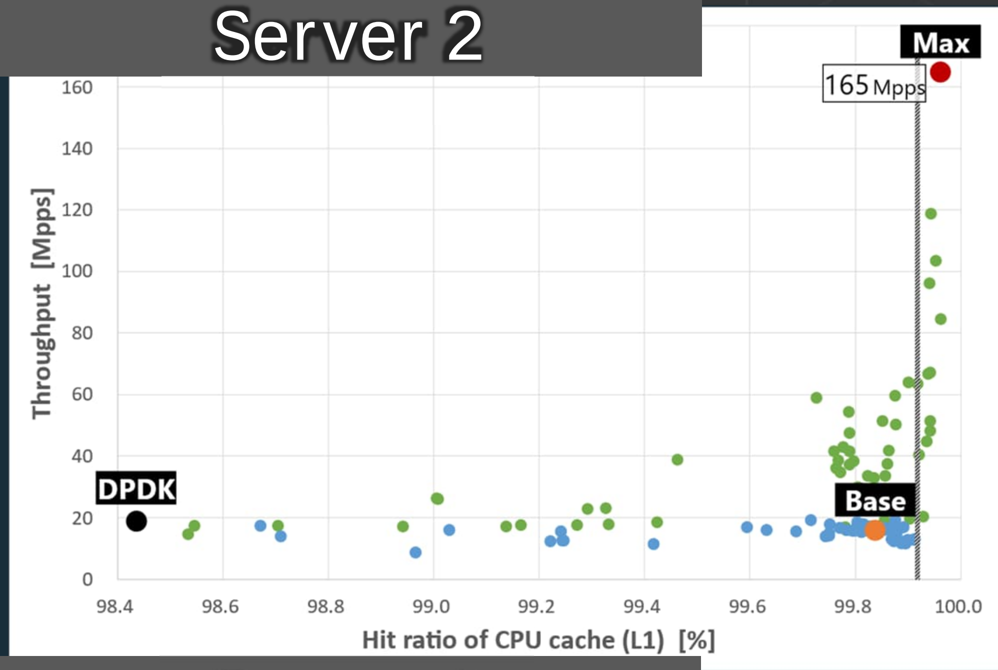
https://github.com/sdnnitech/CESim

[Appendix] Results on the Other Servers

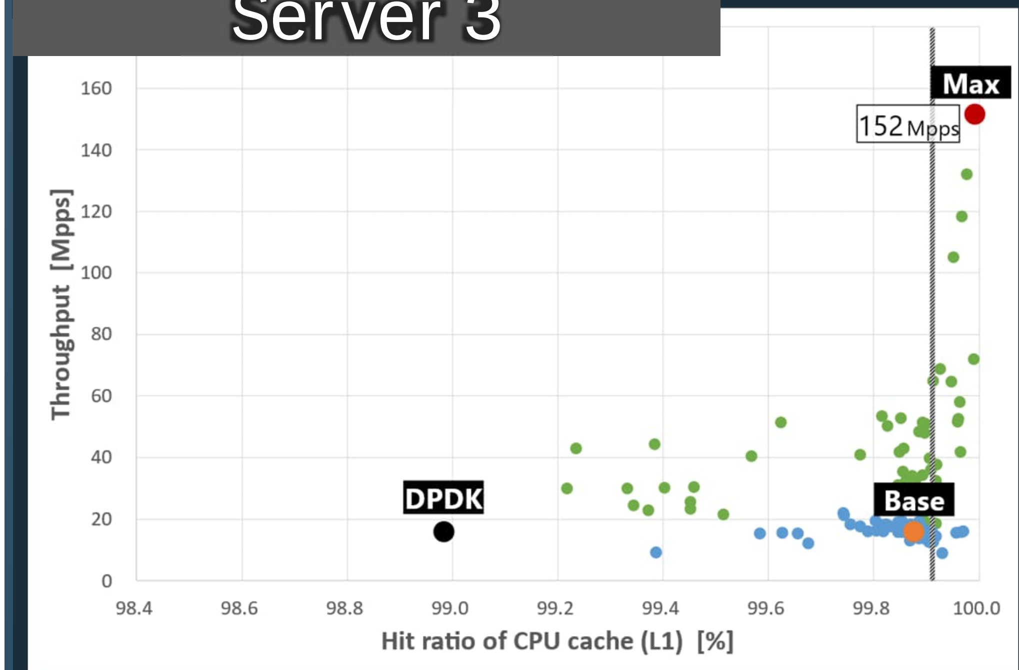
Server 1



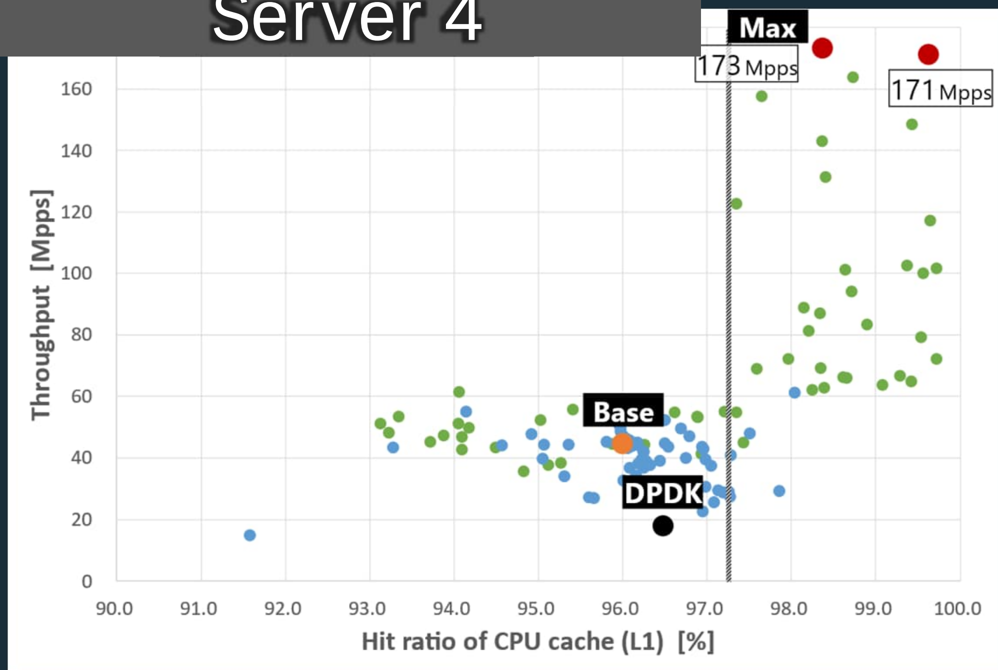
Server 2



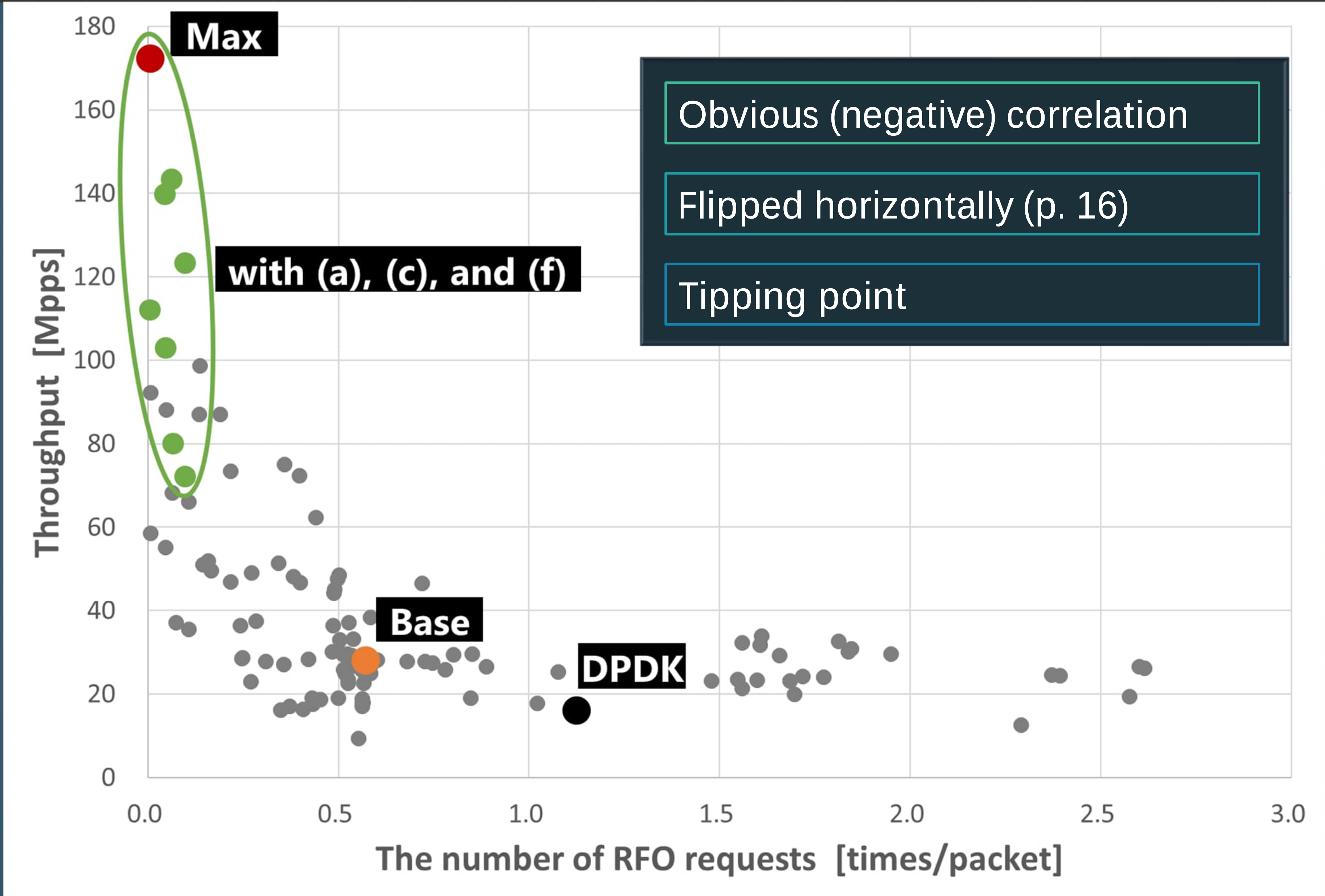
Server 3



Server 4



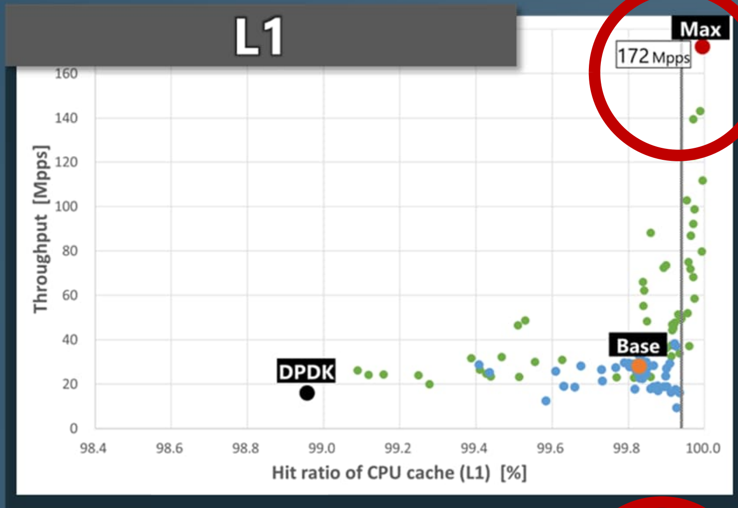
[Appendix] Impact of Cache Invalidations



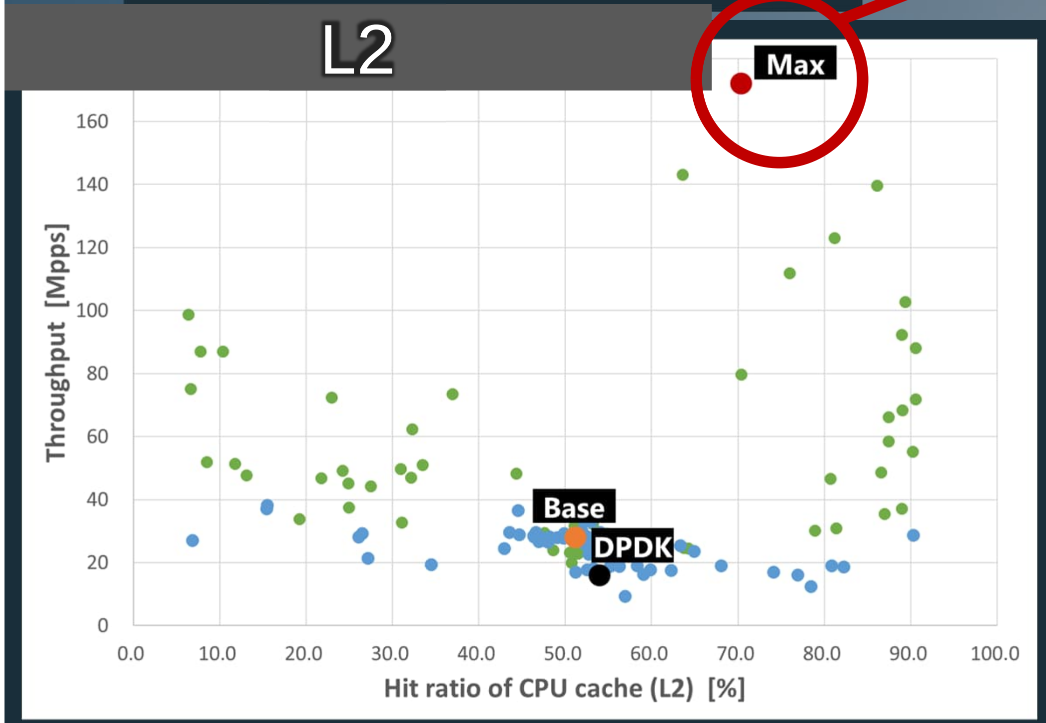
- Obvious (negative) correlation
- Flipped horizontally (p. 16)
- Tipping point

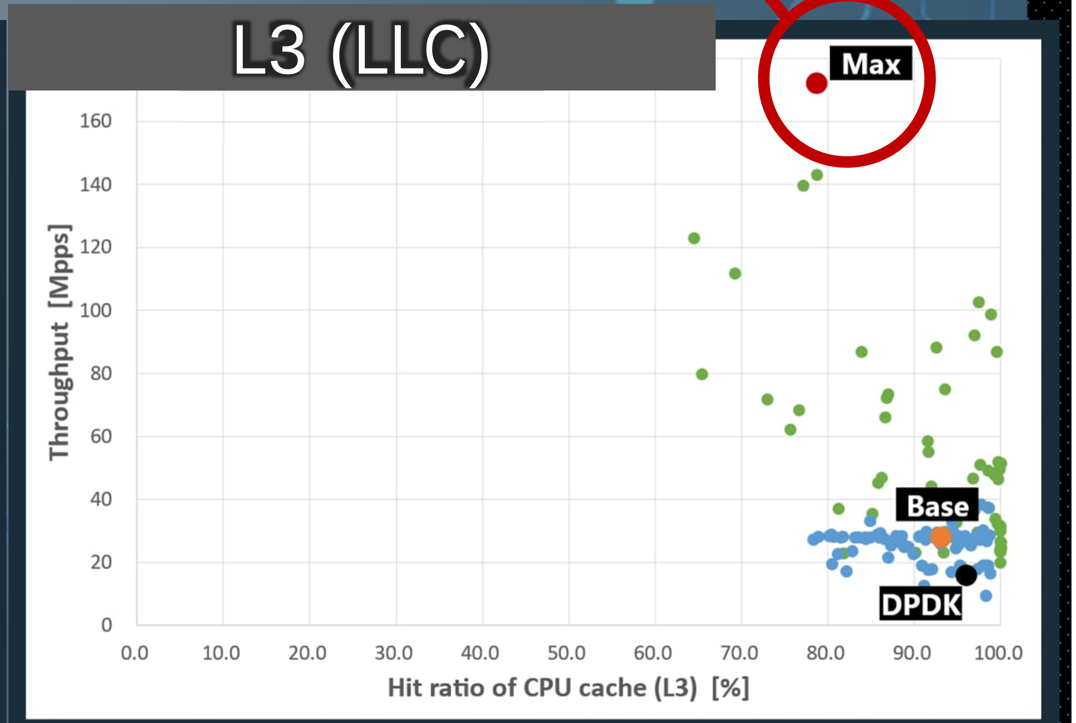
Major cause of L1 cache misses is invalidation!

[Appendix] Effects of L2 and L3 Caches



- Accesses to L2/L3 caches seldom occur
- Hit ratio of L2/L3 caches does not affect throughput





L2/L3 cache usages have little impact on throughput!

[Appendix] Tipping Points

Why does the tipping points appear?

Experiment

- pp. 15-18
- Acquired values are useful
- Real environment is complex to dig in

Modeling

- Essential nature of packet forwarding
- Experimental results are feedbacked

Simulation

- Throughput vs. Cache usage
- Can reproduce the experimental results?

[Appendix] Modeling (Parameters)

Experiment

Best-case item
- Throughput
- Cache usage

- No. cache accesses (per packet)

Machine spec.
- CPU clock
- Access latency

Modeling

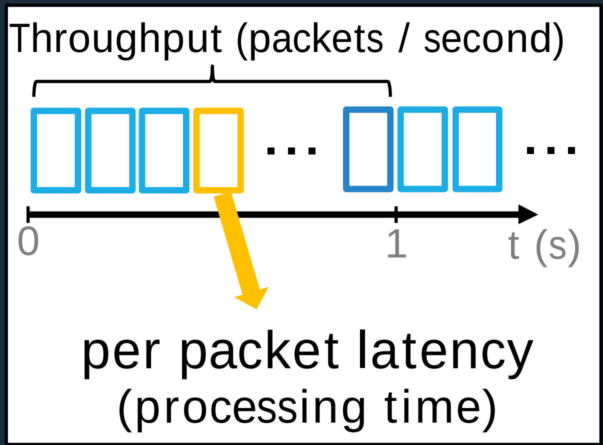
Constants
- Input parameters
- Pure proc. ratio (α)
- Acceleration factor (β)

Variable
- Cache hit ratio (L1)

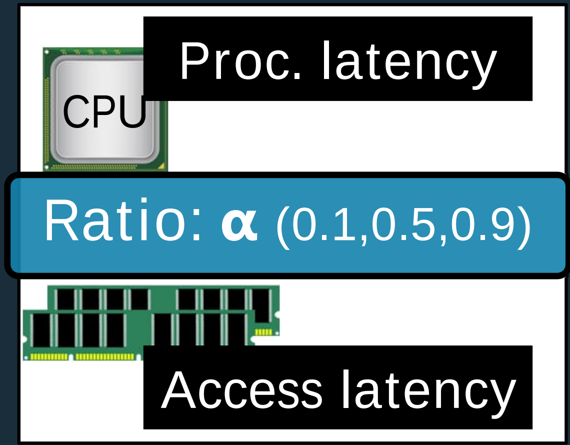
[Appendix] Modeling (Construction)

Simple model (Non-parallelized)

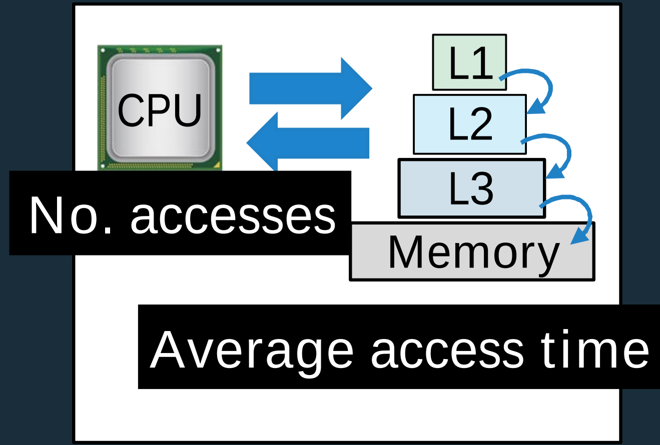
Calculate per packet latency



Calculate access latency



Vary L1 hit ratio

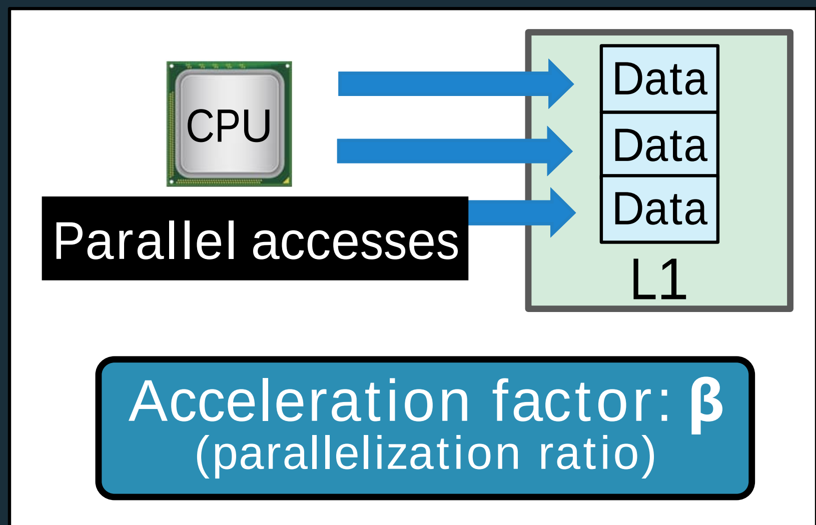


Calculate throughput!

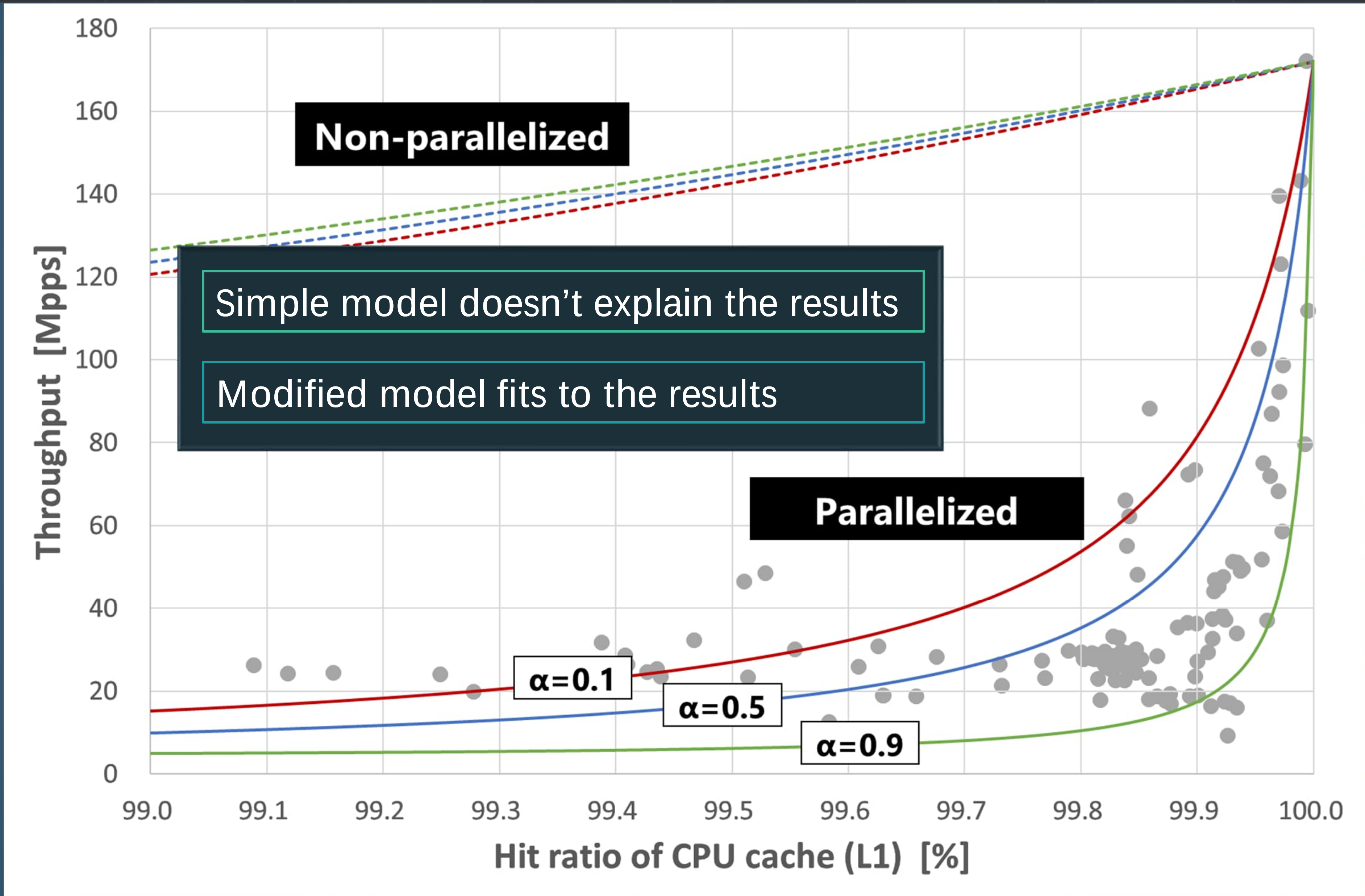
Modified model (Parallelized)

Tipping point doesn't appear on Simple model

(see next page)



[Appendix] Simulation



L1 cache miss would cancel parallelization effect!