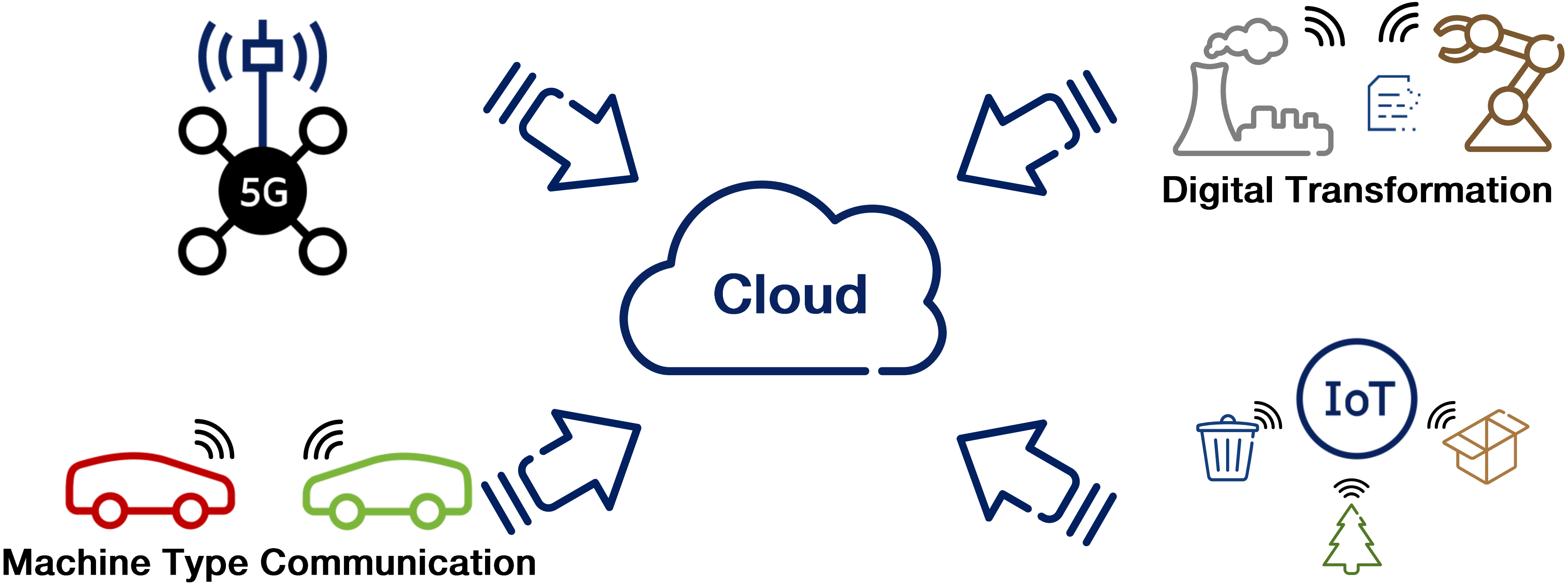
Make the Most out of Last Level Cache in Intel Processors

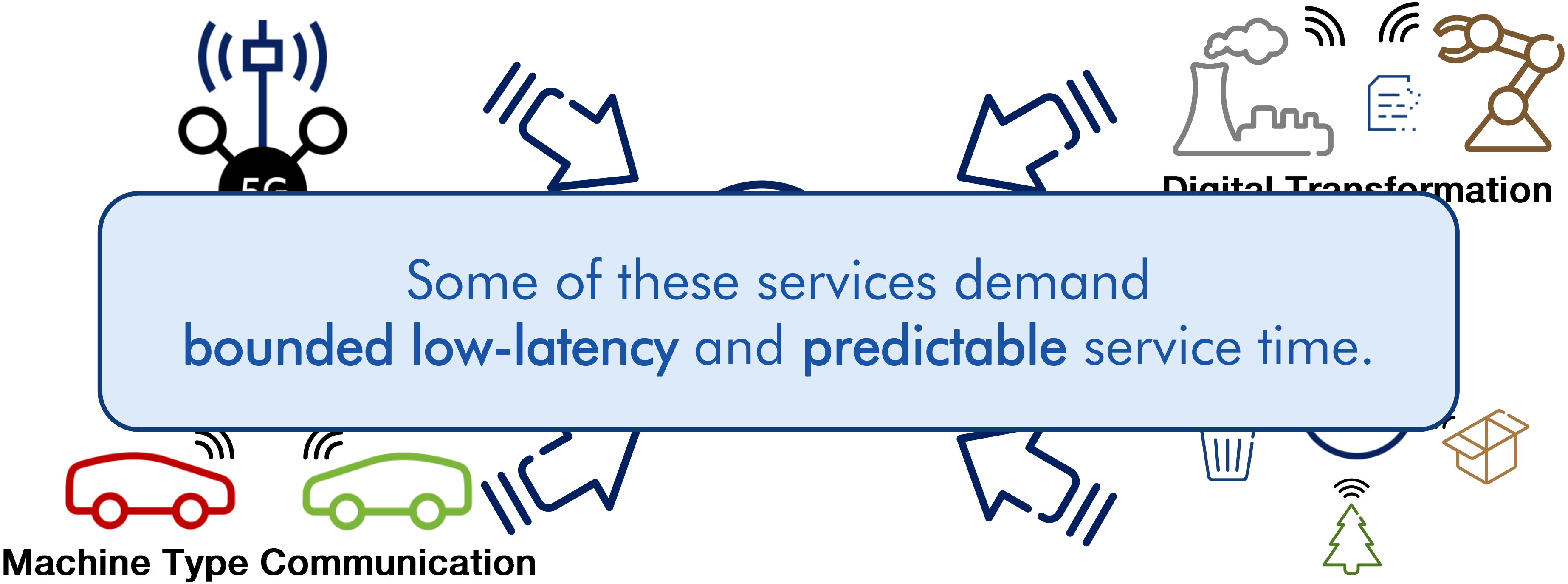
Alireza Farshin[*], Amir Roozbeh[*+], Gerald Q. Maguire Jr.[*], Dejan Kostić[*]

[*] KTH Royal Institute of Technology (EECS/COM) [+] Ericsson Research

Motivation

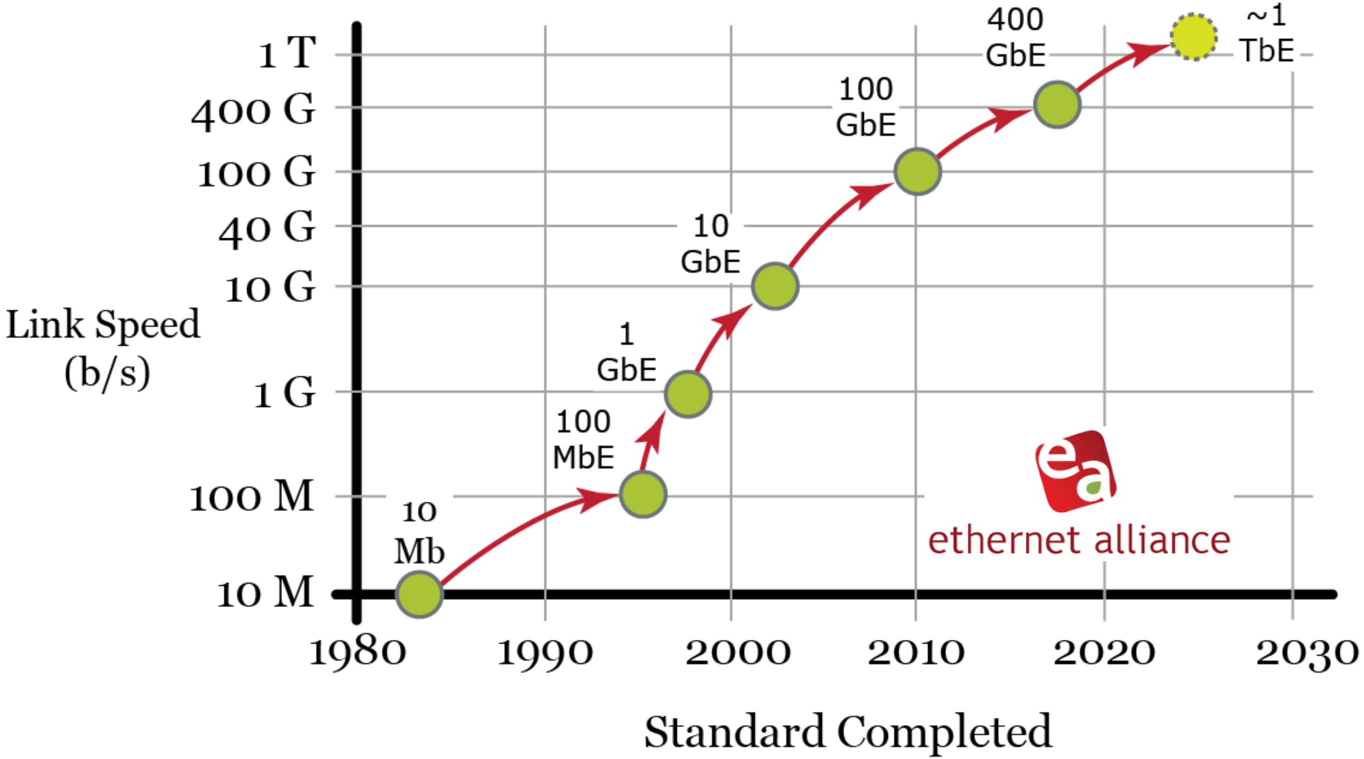

5G
Cloud
Digital Transformation
IoT
Machine Type Communication

Motivation

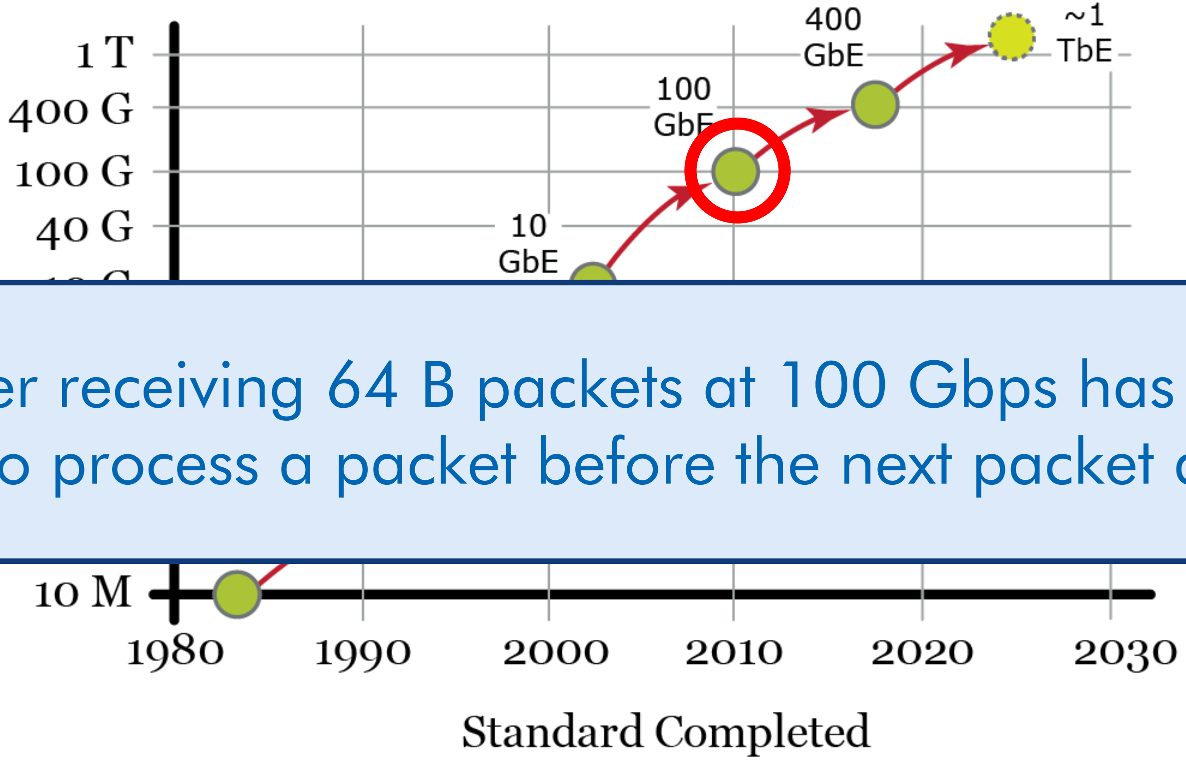


Some of these services demand **bounded low-latency** and **predictable** service time.

Motivation

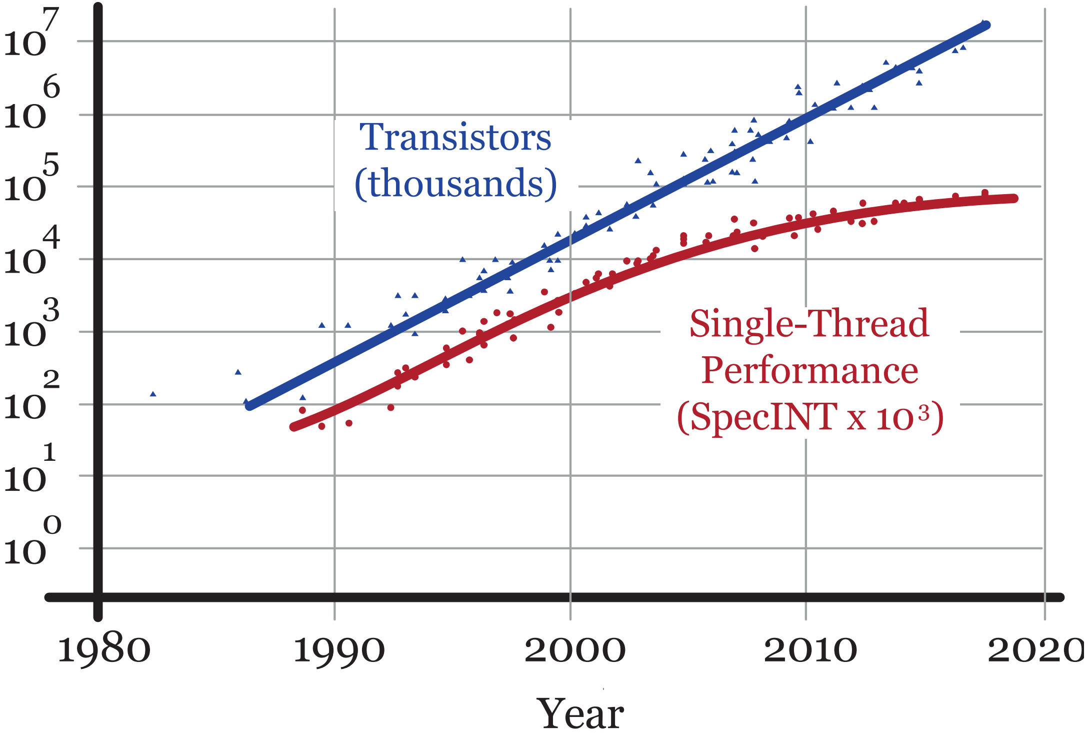

Link Speed (b/s)
1 T
400 G
100 G
40 G
10 G
1 G
100 M
10 M
1980
1990
2000
2010
2020
2030
Standard Completed
10 Mb
100 MbE
1 GbE
10 GbE
100 GbE
400 GbE
~1 TbE
ethernet alliance

Motivation



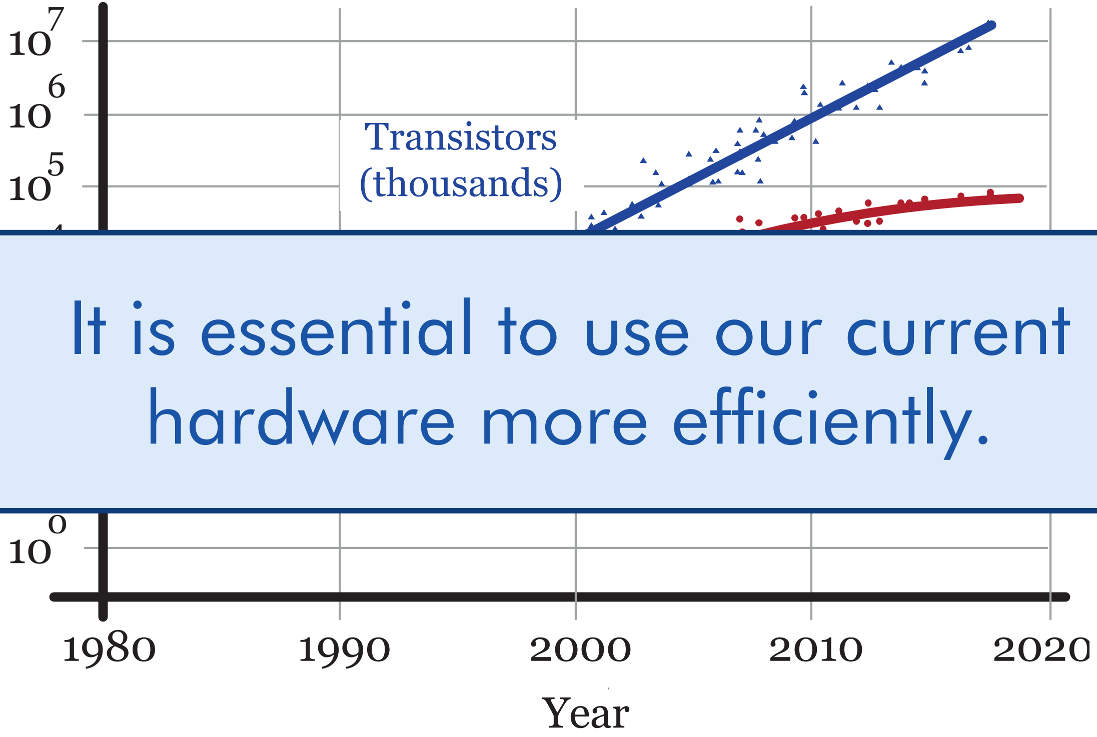
A server receiving 64 B packets at 100 Gbps has only **5.12 ns** to process a packet before the next packet arrives.

Motivation



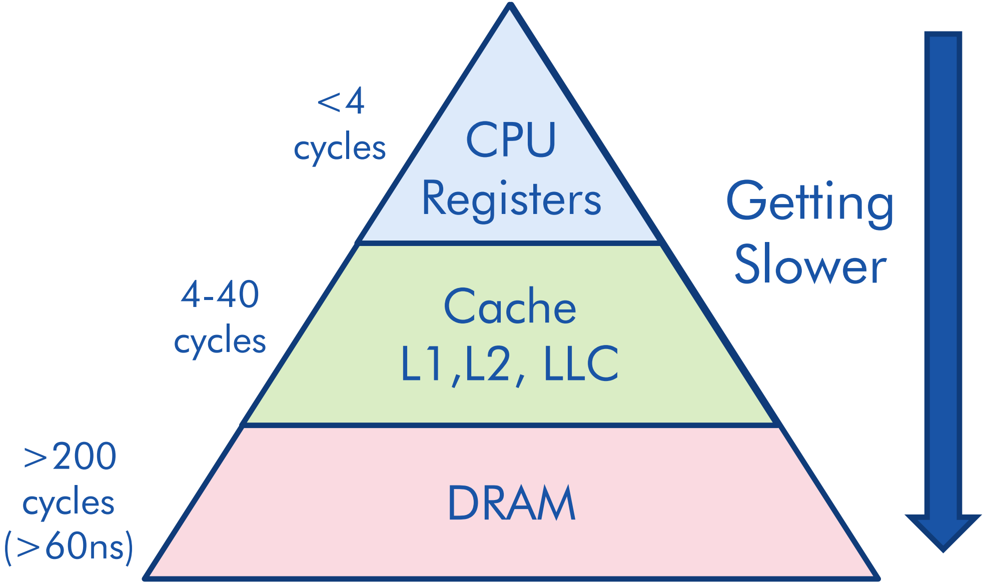
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
Transistors
(thousands)
Single-Thread
Performance
(SpecINT x 10^3)
1980
1990
2000
2010
2020
Year

Motivation



It is essential to use our current hardware more efficiently.

Memory Hierarchy



Memory Hierarchy

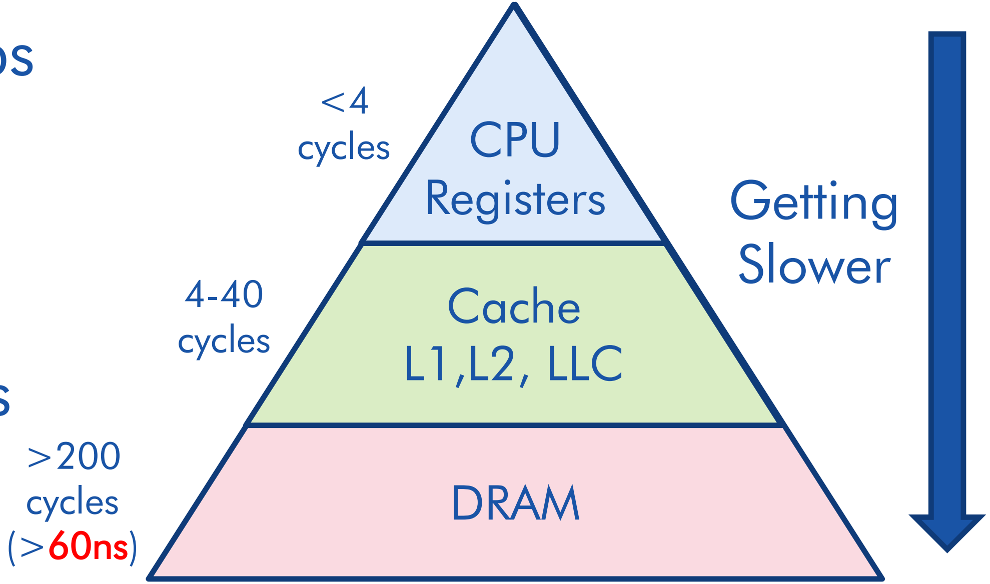
For a CPU that is running at 3.2 GHz, every 4 cycle is around 1.25 ns.

Memory Hierarchy

To keep up with 100 Gbps time budget (5.12 ns)

↓

Cache becomes valuable, as every access to DRAM is **expensive**



Memory Hierarchy

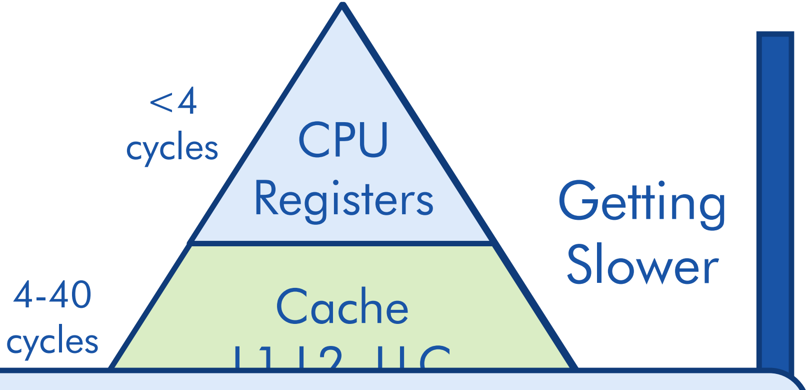
For a CPU that is running at 3.2 GHz, every 4 cycle is around 1.25 ns.

Memory Hierarchy

To keep up with 100 Gbps time budget (5.12 ns)

↓

Cache becomes



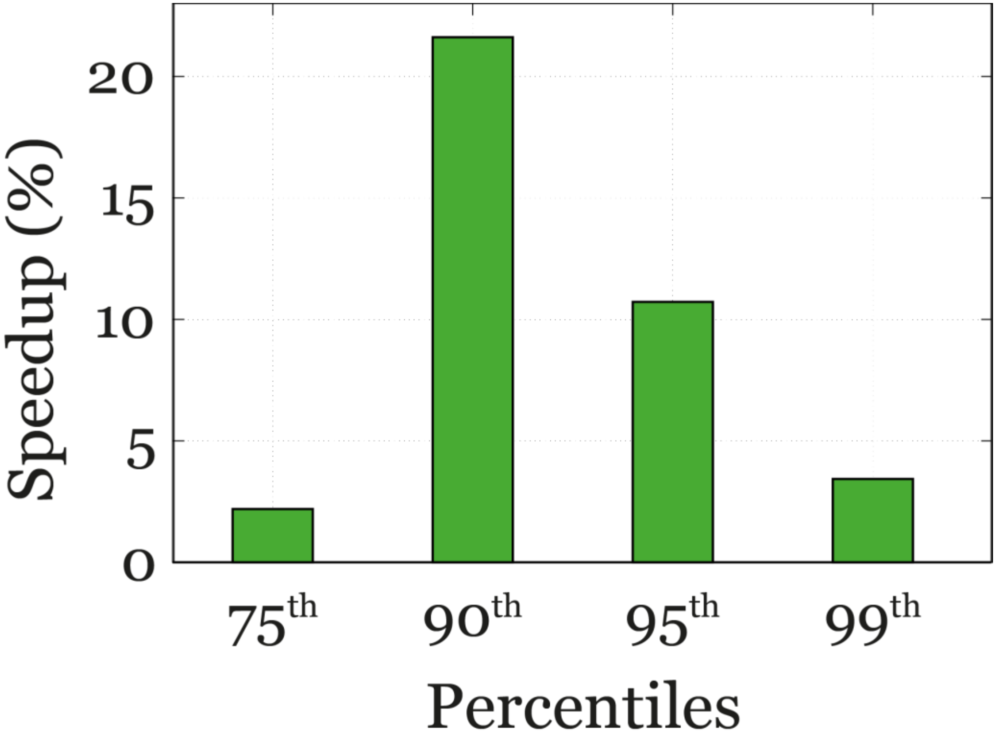
We focus on **better management** of cache.

Memory Hierarchy

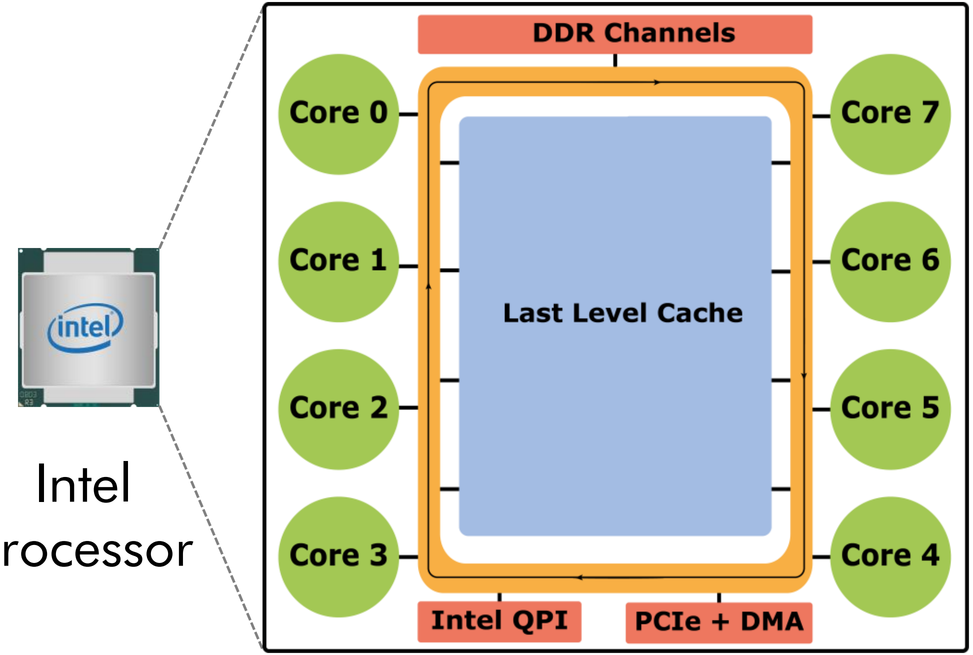
For a CPU that is running at 3.2 GHz, every 4 cycle is around 1.25 ns.

Better Cache Management

Reduce tail latencies of NFV service chains running at 100 Gbps by up to **21.5%**



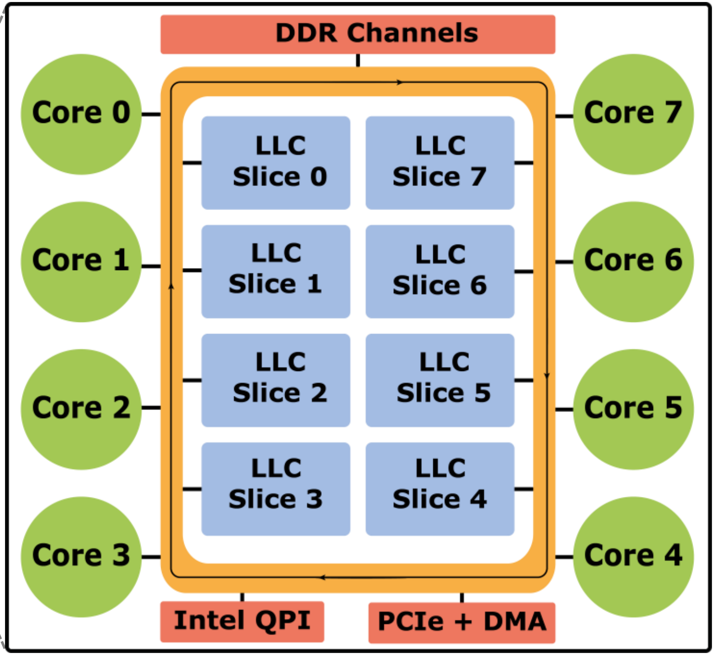
Last Level Cache (LLC)



Intel Processor

Non−uniform Cache Architecture (NUCA)

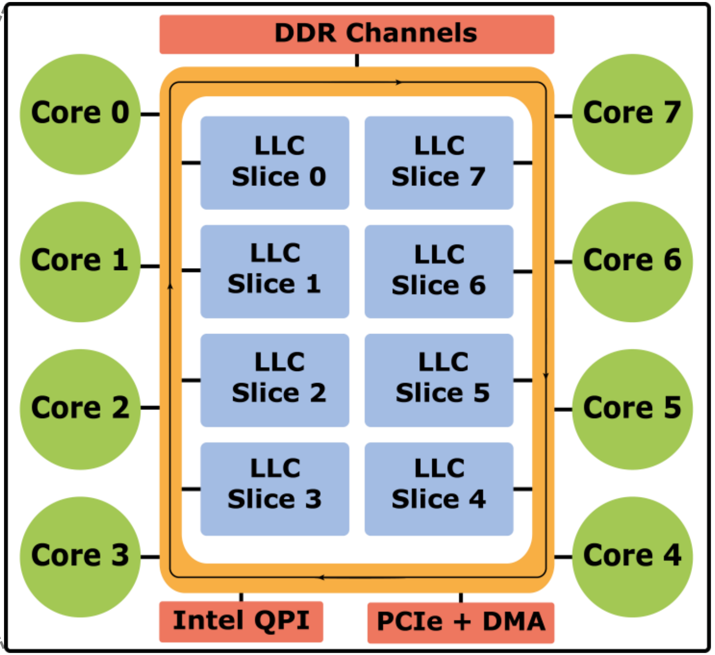
Intel Processor



Since Sandy Bridge (~2011), LLC is not unified any more!

Non–uniform Cache Architecture (NUCA)

Intel Processor



Intel's Complex Addressing

Determines the mapping between memory address space and LLC Slices.

Almost every cache line (64 B) maps to a different LLC slice.

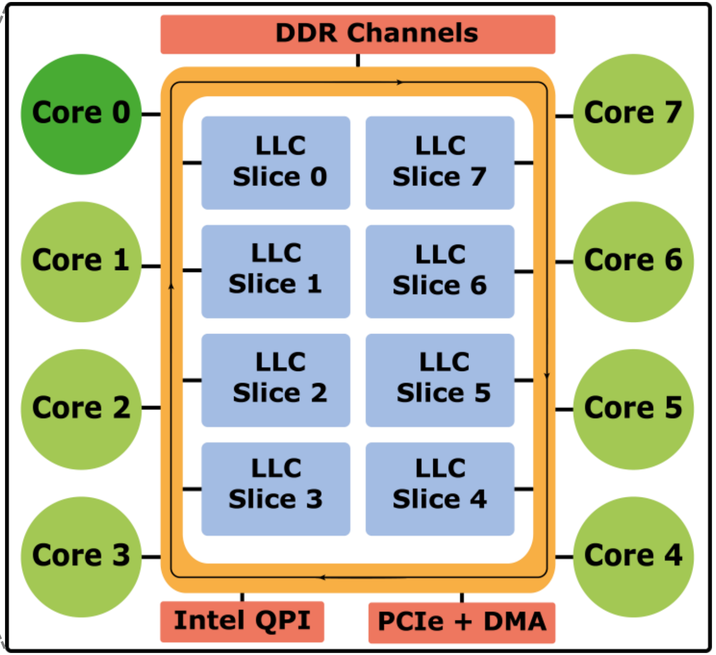
Known Methods:
Clémentine Maurice et al. [RAID '15]*

- Performance Counters

* Clémentine Maurice, Nicolas Scouarnec, Christoph Neumann, Olivier Heen, and Aurélien Francillon. 2015. Reverse Engineering Intel Last-Level Cache Complex Addressing Using Performance Counters.

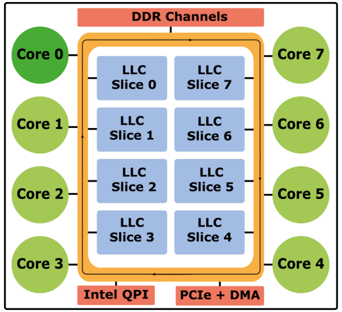
Measuring Access Time to LLC Slices

Intel Xeon
E5-2667 v3
(Haswell)

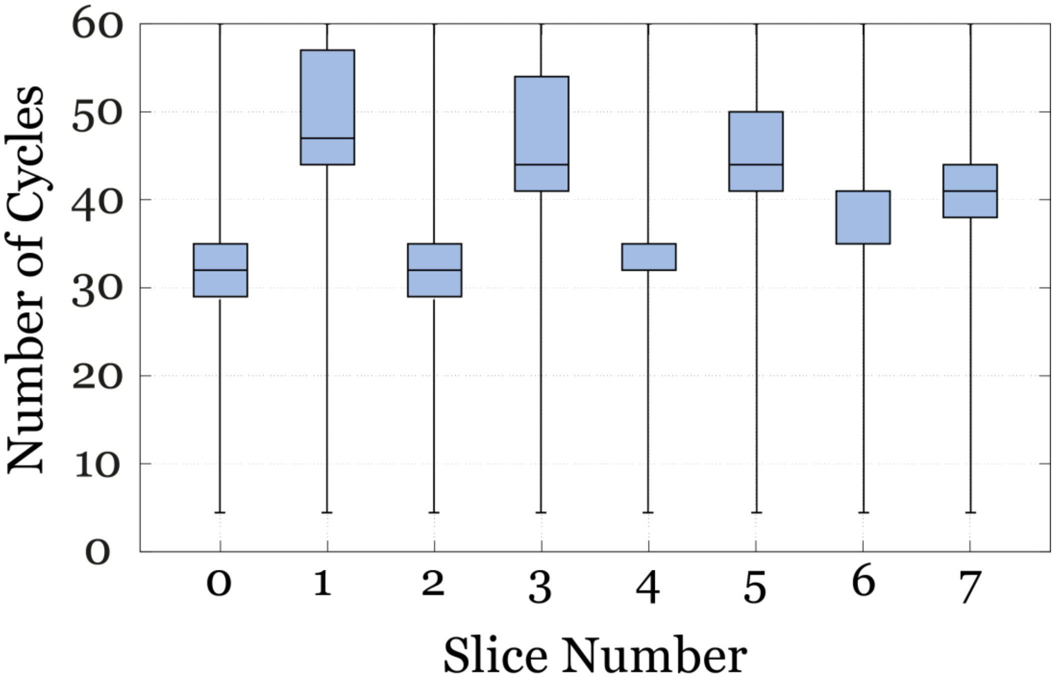


Different access time to different LLC slices

Measuring Access Time to LLC Slices

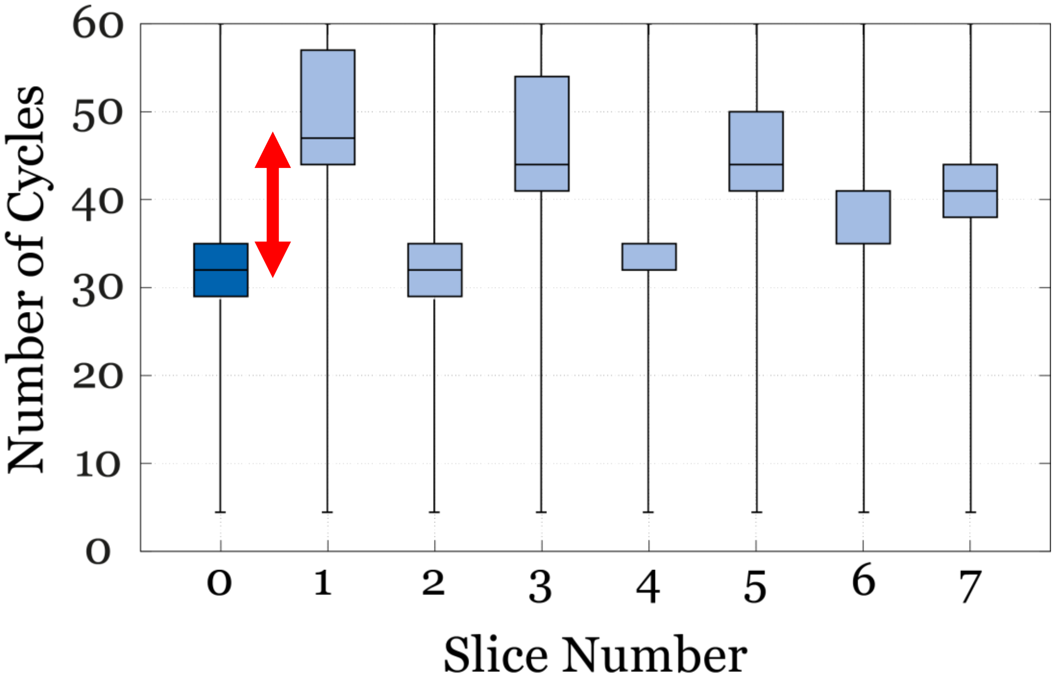


Measuring Read Access Time from Core 0 to all LLC slices



Opportunity

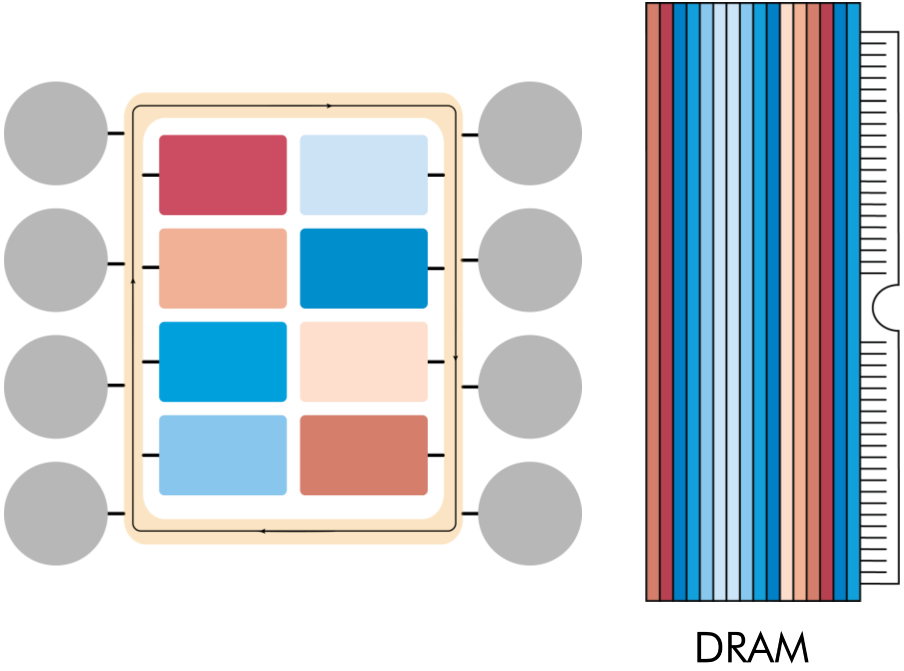
Accessing the **closer** LLC slice can save up to **~20 cycles**, i.e., 6.25 ns.



For a CPU that is running at 3.2 GHz.

Slice–aware Memory Management

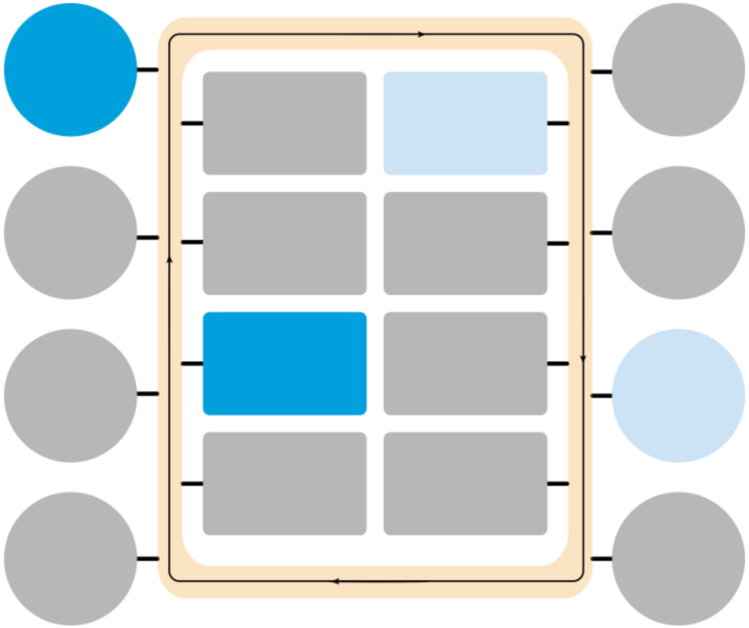
Allocate memory from physical memory in a way that it maps to the appropriate LLC slice(s).



Slice-aware Memory Management

Use Cases:

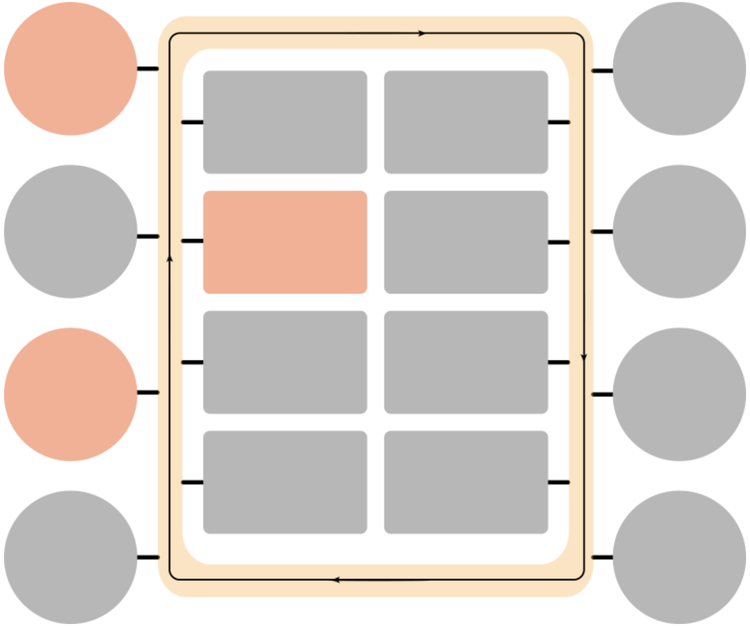
- Isolation



Slice–aware Memory Management

Use Cases:

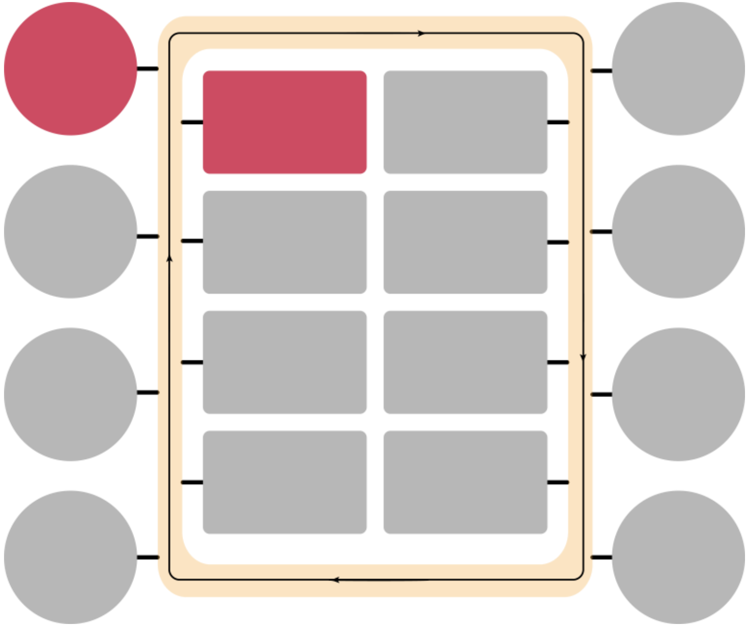
- Isolation
- Shared Data



Slice–aware Memory Management

Use Cases:

- Isolation
- Shared Data
- Performance

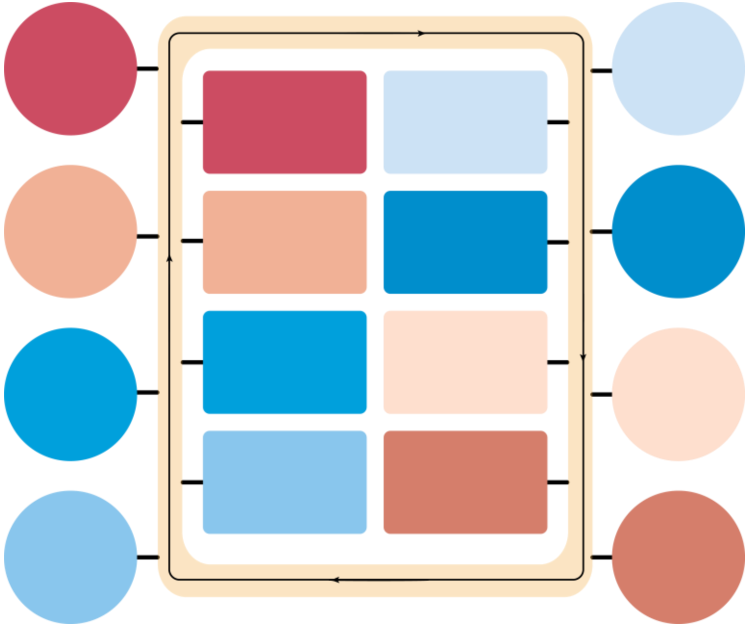


Slice-aware Memory Management

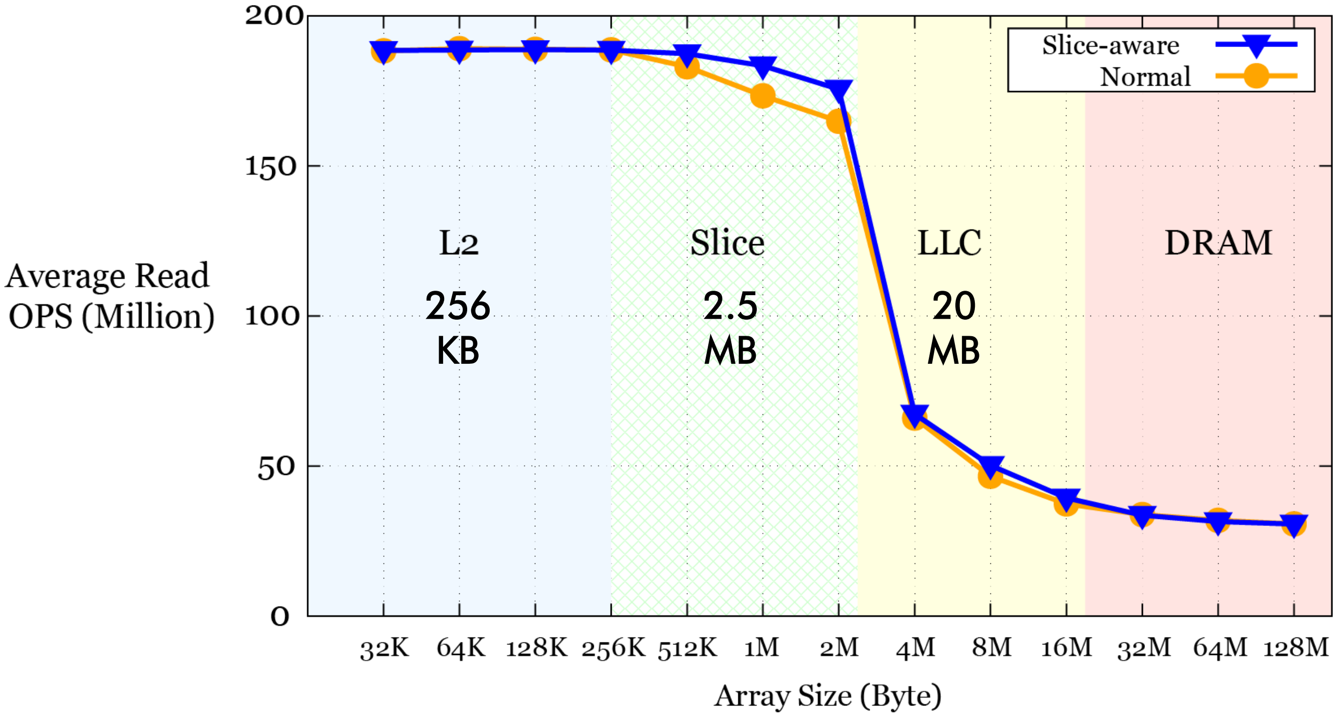
Use Cases:

- Isolation
- Shared Data
- Performance

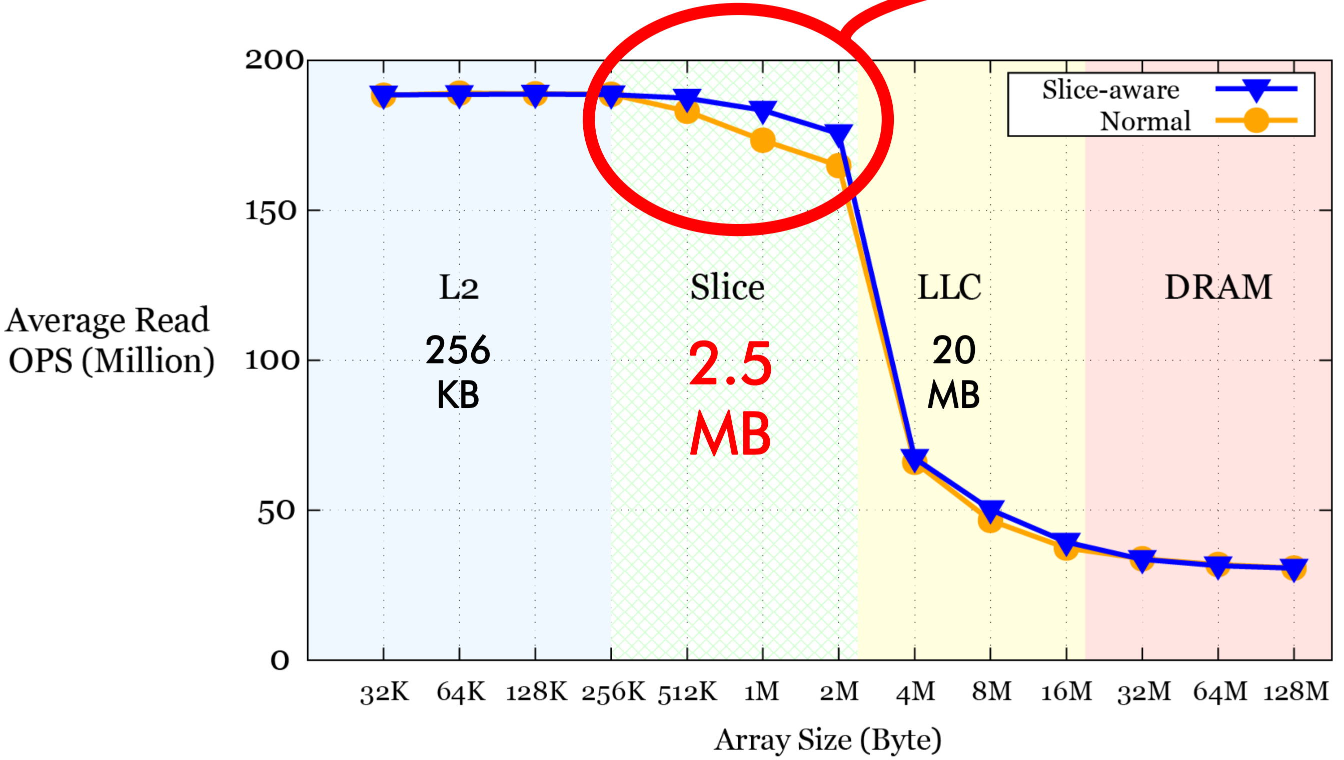
Every core is associated to its closest LLC slice.



Slice−aware Memory Management



Slice−aware Memory Management



Beneficial when **working set** can fit into a slice.

Slice–aware Memory Management

There are many applications that have this characteristic.

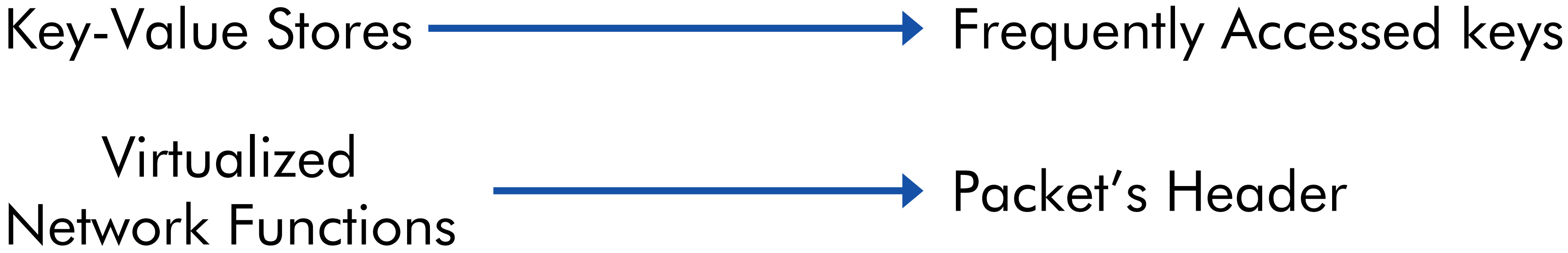
Slice–aware Memory Management

There are many applications that have this characteristic.

Key-Value Stores → Frequently Accessed keys

Slice–aware Memory Management

There are many applications that have this characteristic.

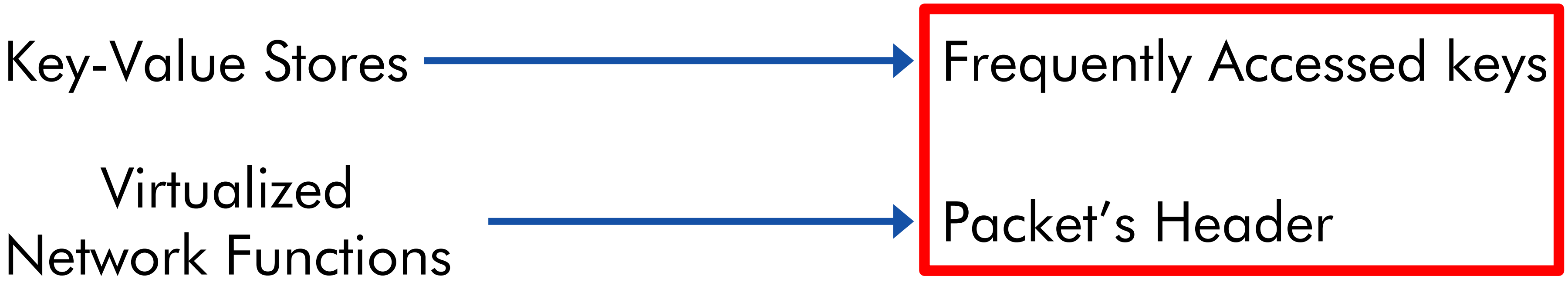


Key-Value Stores → Frequently Accessed keys

Virtualized Network Functions → Packet's Header

Slice–aware Memory Management

There are many applications that have this characteristic.



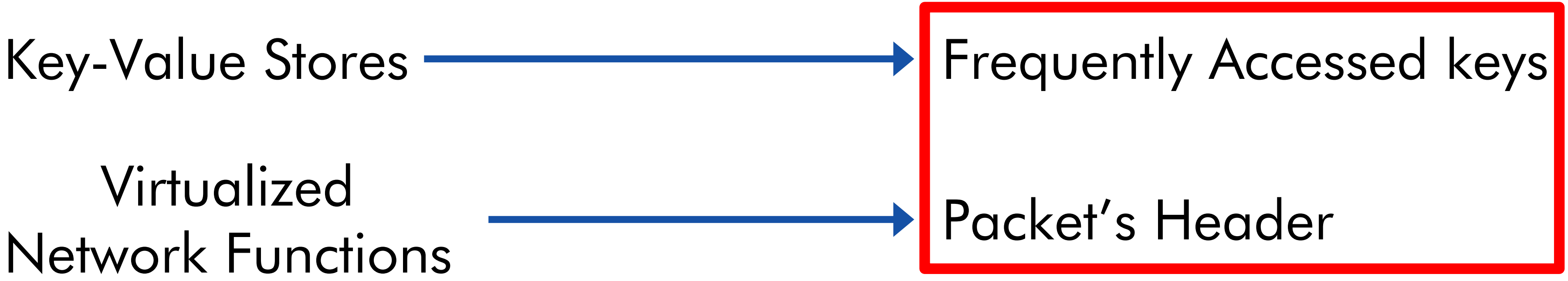
Key-Value Stores → Frequently Accessed keys

Virtualized Network Functions → Packet's Header

Can fit into a slice

Slice–aware Memory Management

There are many applications that have this characteristic.



Key-Value Stores → Frequently Accessed keys

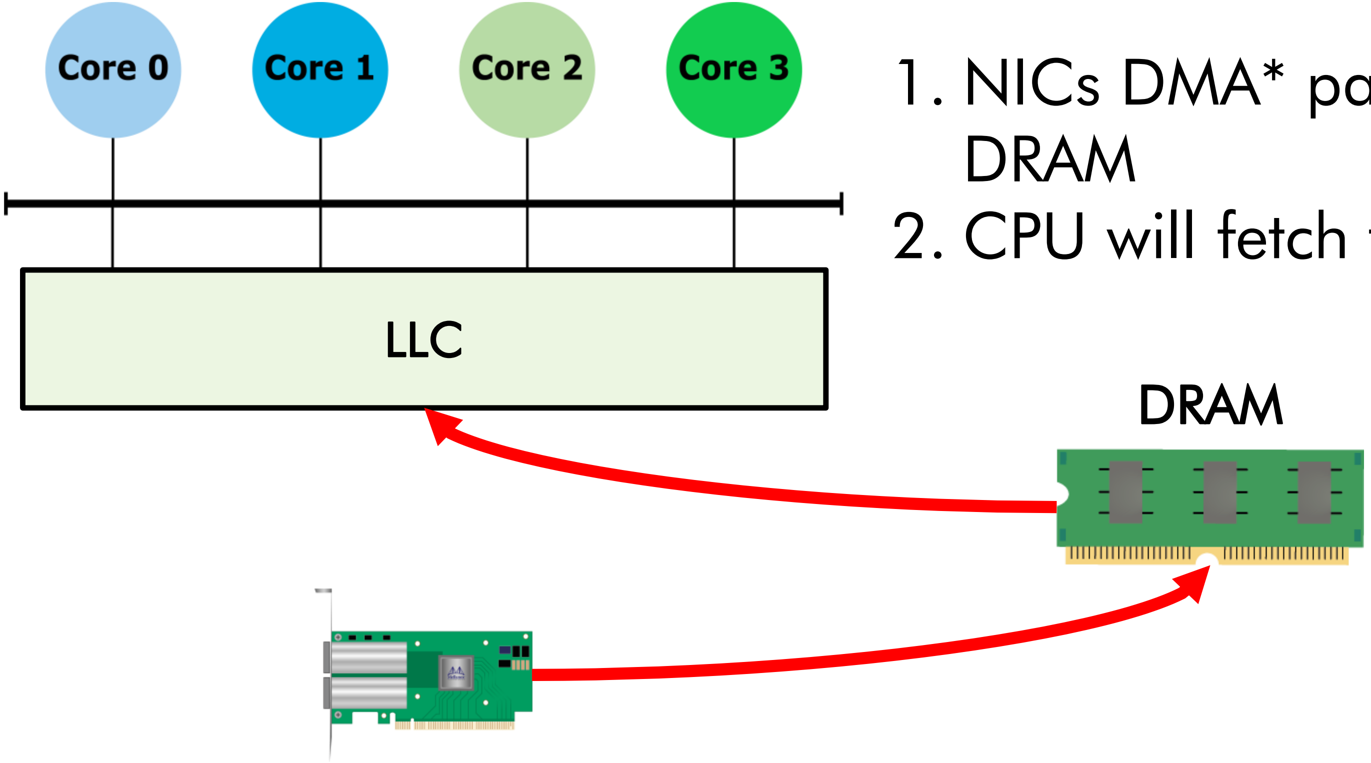
Virtualized Network Functions → Packet's Header

We focus on **virtualized network functions** in this talk!

CacheDirector

A network I/O solution which extends Data Direct I/O (DDIO) by employing Slice-aware Memory Management

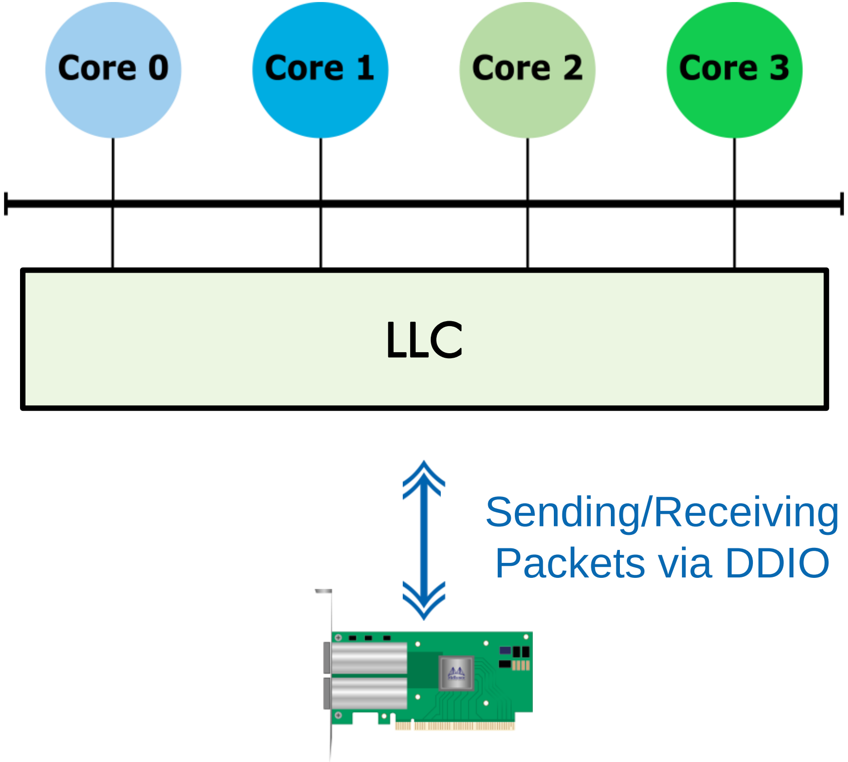
Traditional I/O



1. NICs DMA* packets to DRAM
2. CPU will fetch them to LLC

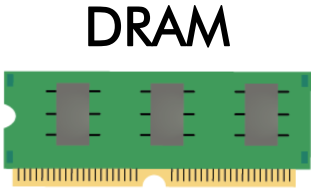
* Direct Memory Access (DMA)

Data Direct I/O (DDIO)



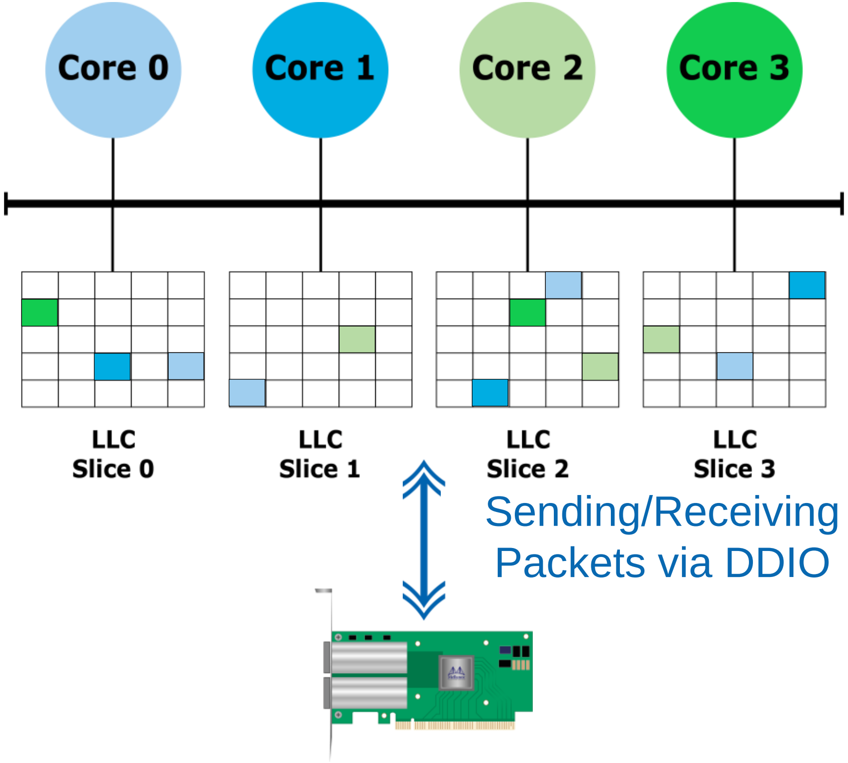
Sending/Receiving Packets via DDIO

DMA*-ing packets directly to LLC rather than DRAM.



* Direct Memory Access (DMA)

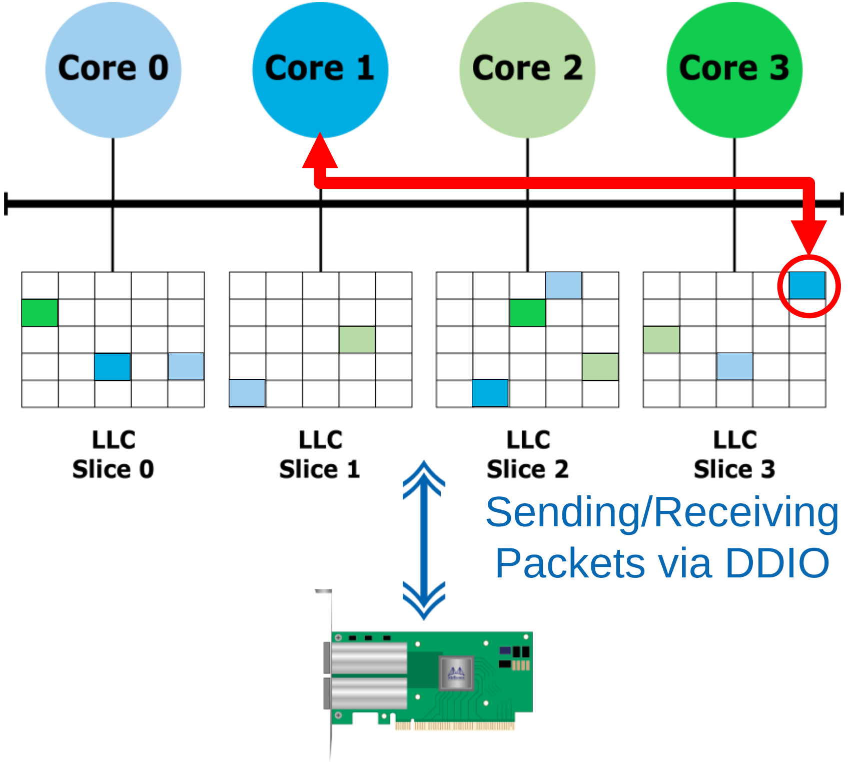
Data Direct I/O (DDIO)



Sending/Receiving Packets via DDIO

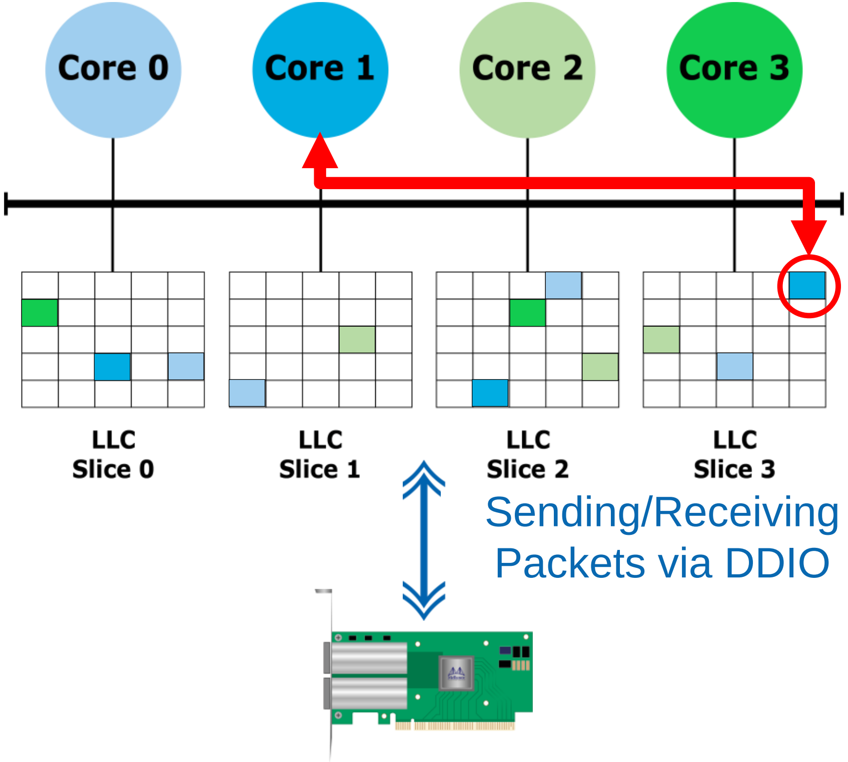
Packets go to random slices!

Data Direct I/O (DDIO)

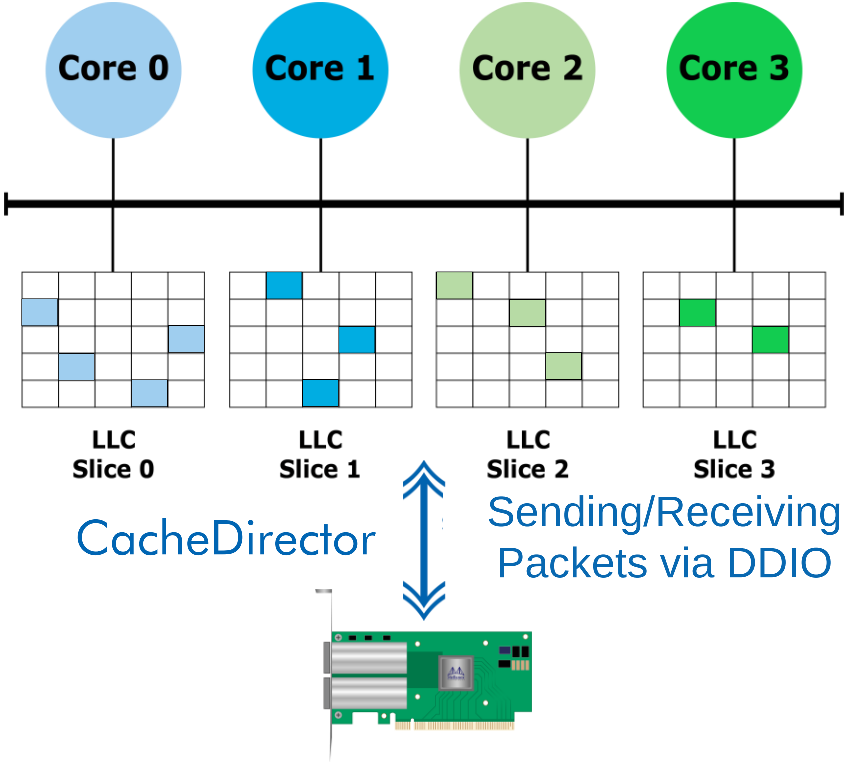


Packets go to random slices!

CacheDirector



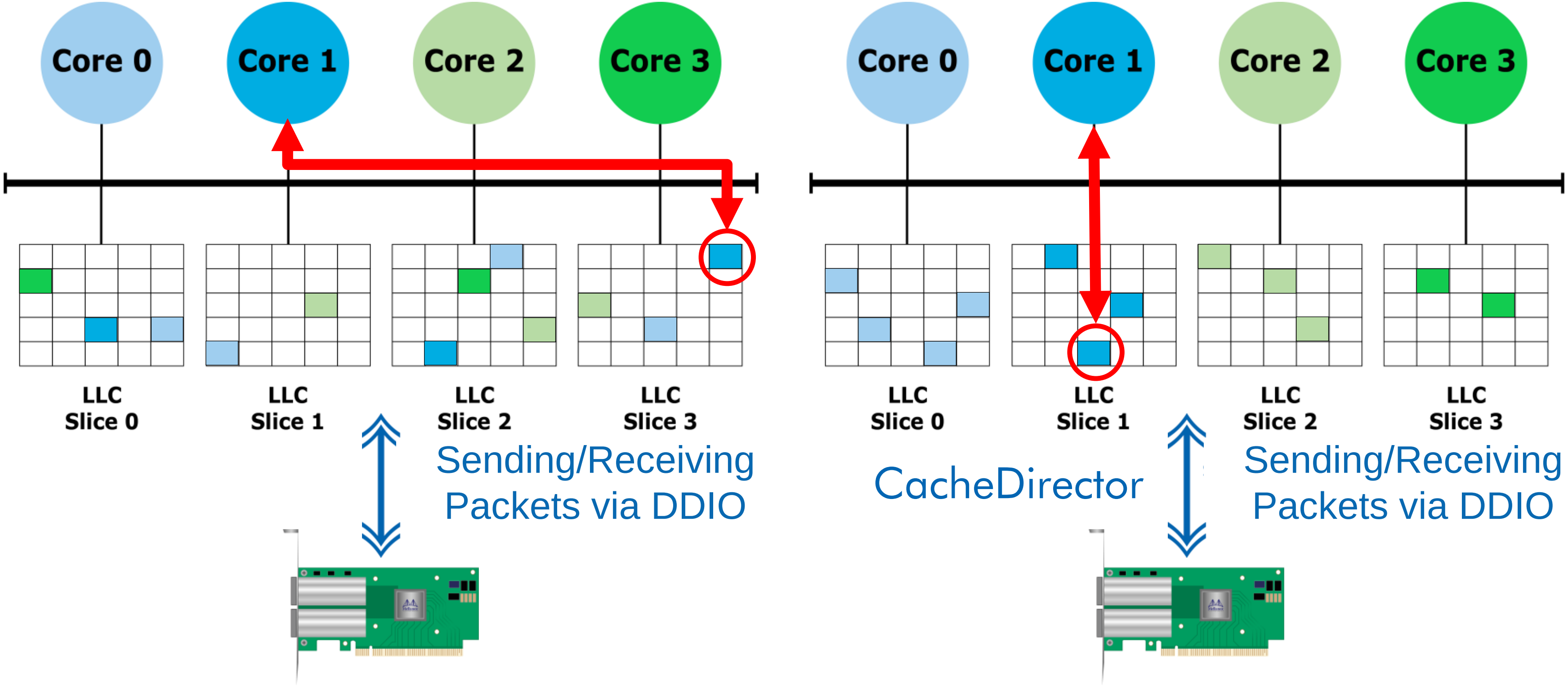
Sending/Receiving Packets via DDIO



CacheDirector

Sending/Receiving Packets via DDIO

CacheDirector



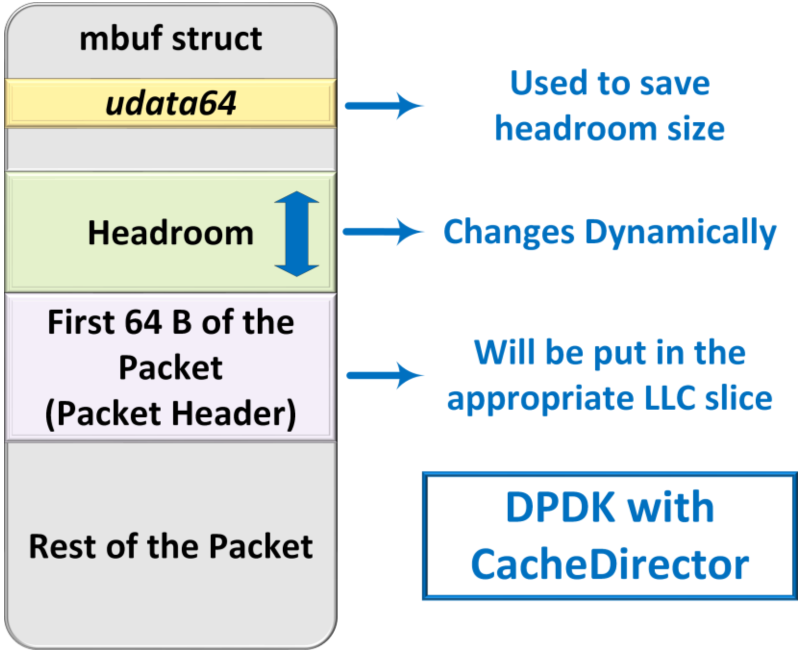
Sending/Receiving Packets via DDIO

CacheDirector

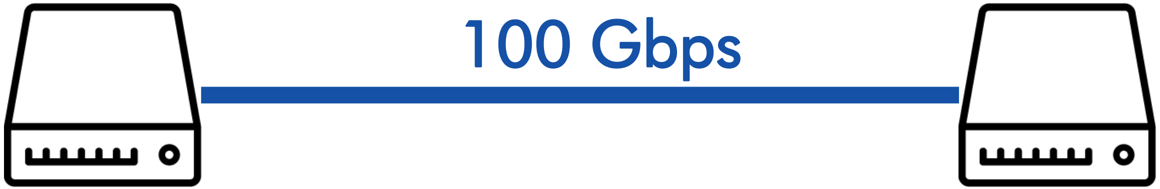
Sending/Receiving Packets via DDIO

CacheDirector

- Sends packet's header to the appropriate LLC slice.
- Implemented as a part of user-space NIC drivers in the Data Plane Development Kit (DPDK).
- Introduces dynamic headroom in DPDK data structures.


mbuf struct
udata64
Used to save headroom size
Headroom
Changes Dynamically
First 64 B of the Packet (Packet Header)
Will be put in the appropriate LLC slice
Rest of the Packet
DPDK with CacheDirector

Evaluation – Testbed

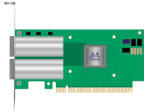


100 Gbps

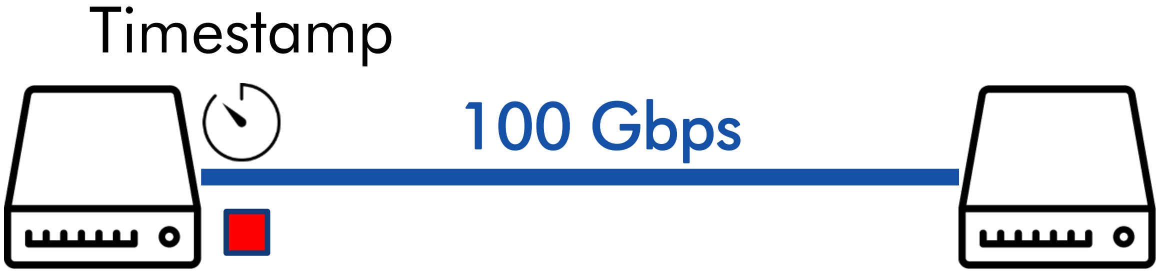
Packet Generator

Device under Test
Running VNFs

Intel Xeon E5 2667 v3


Mellanox ConnectX-4

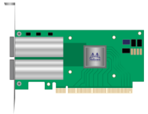
Evaluation – Testbed



Packet Generator
Actual Campus Trace

Device under Test
Running VNFs

Intel Xeon E5 2667 v3

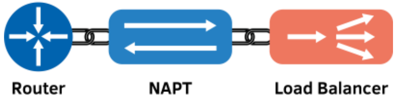


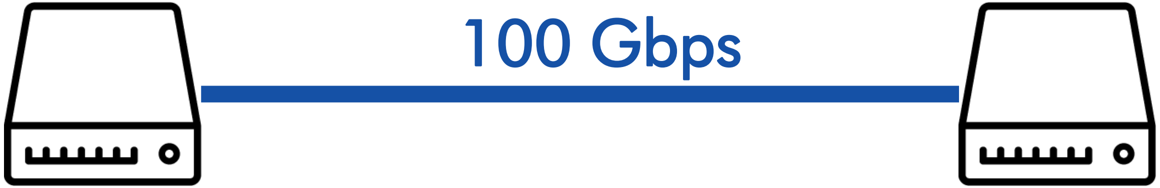
Mellanox ConnectX-4

Evaluation – Testbed

Metron [NSDI '18]*

Stateful NFV Service Chain

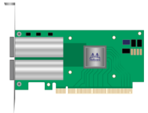




Packet Generator
Actual Campus Trace

Device under Test
Running VNFs

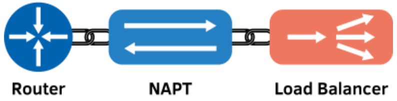
Intel Xeon E5 2667 v3


Mellanox ConnectX-4

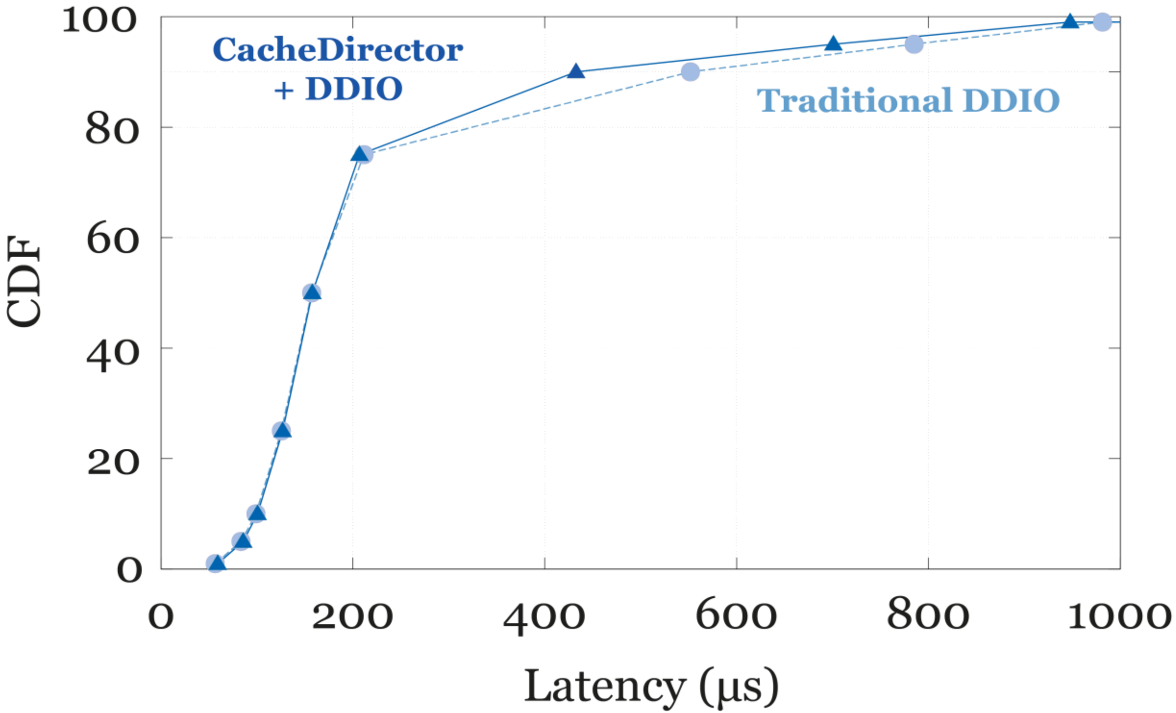
* Georgios P.Katsikas, Tom Barbette, Dejan Kostic, Rebecca Steinert, and Gerald Q. Maguire Jr. 2018. Metron: NFV Service Chains at the True Speed of the Underlying Hardware.

Evaluation — 100 Gbps

Stateful NFV Service Chain

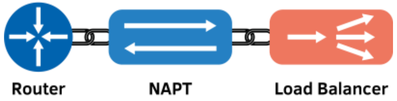


Achieved Throughput
~76 Gbps

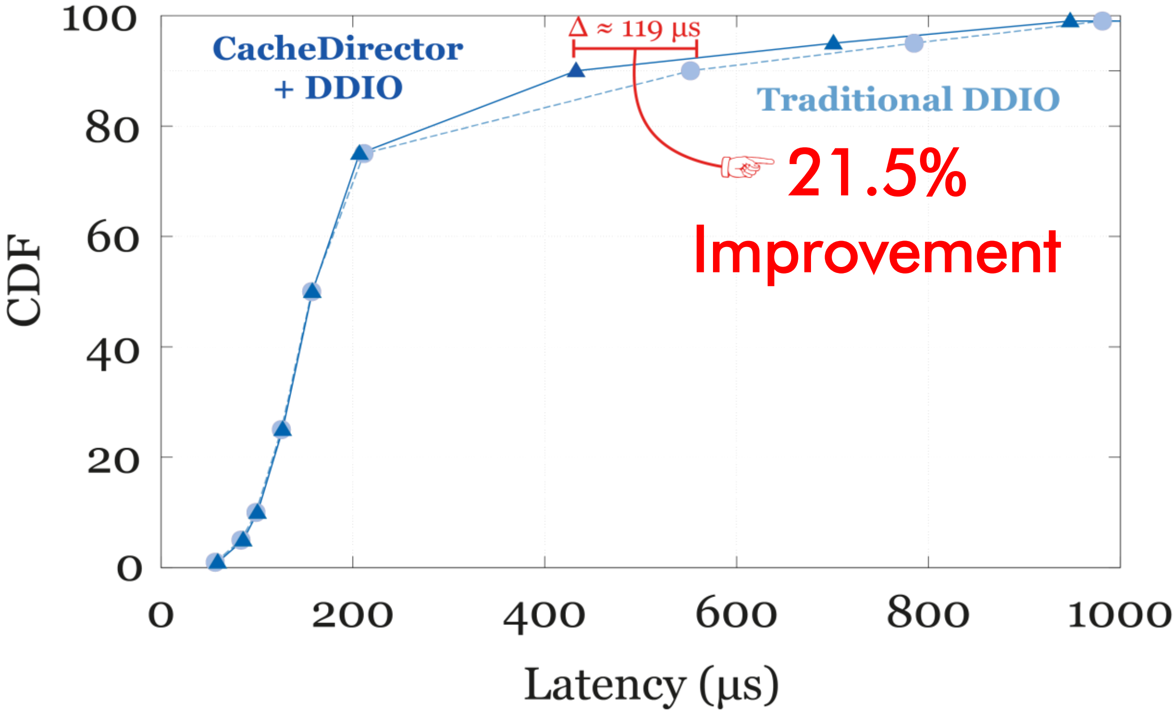


Evaluation — 100 Gbps

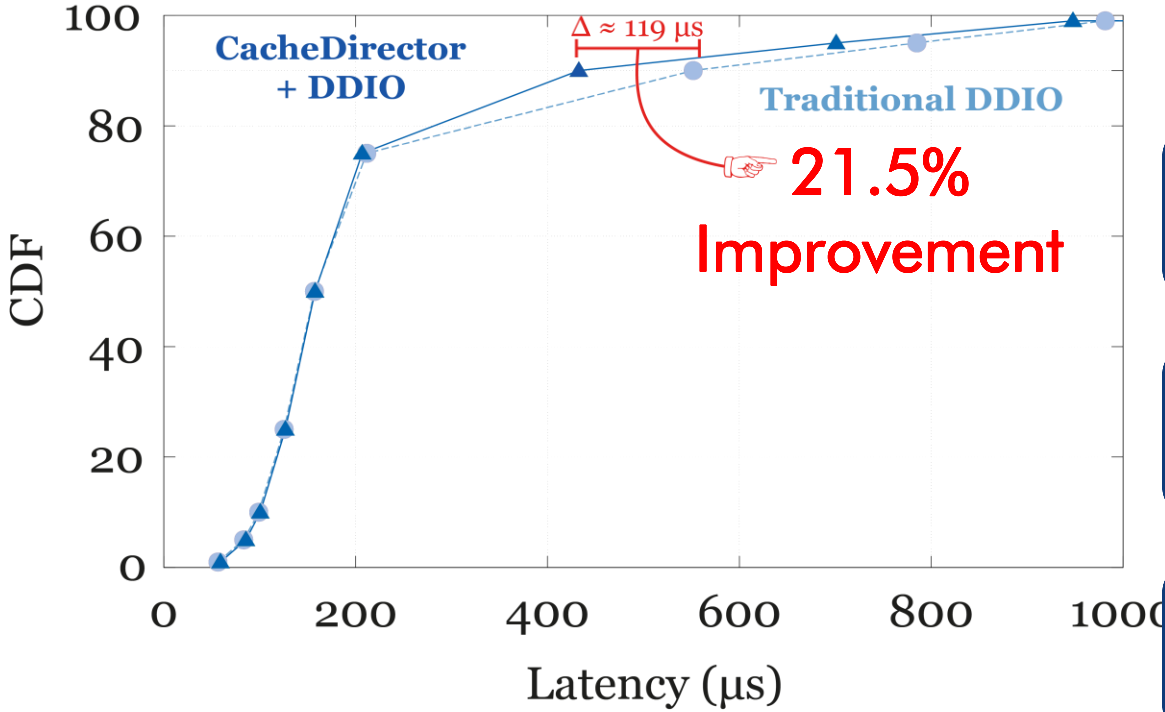
Stateful NFV Service Chain



Achieved Throughput
~76 Gbps

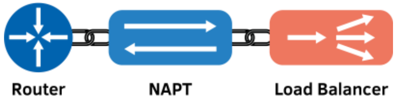


Evaluation — 100 Gbps



21.5% Improvement

Stateful NFV Service Chain



Achieved Throughput
~76 Gbps

Faster access to packet header

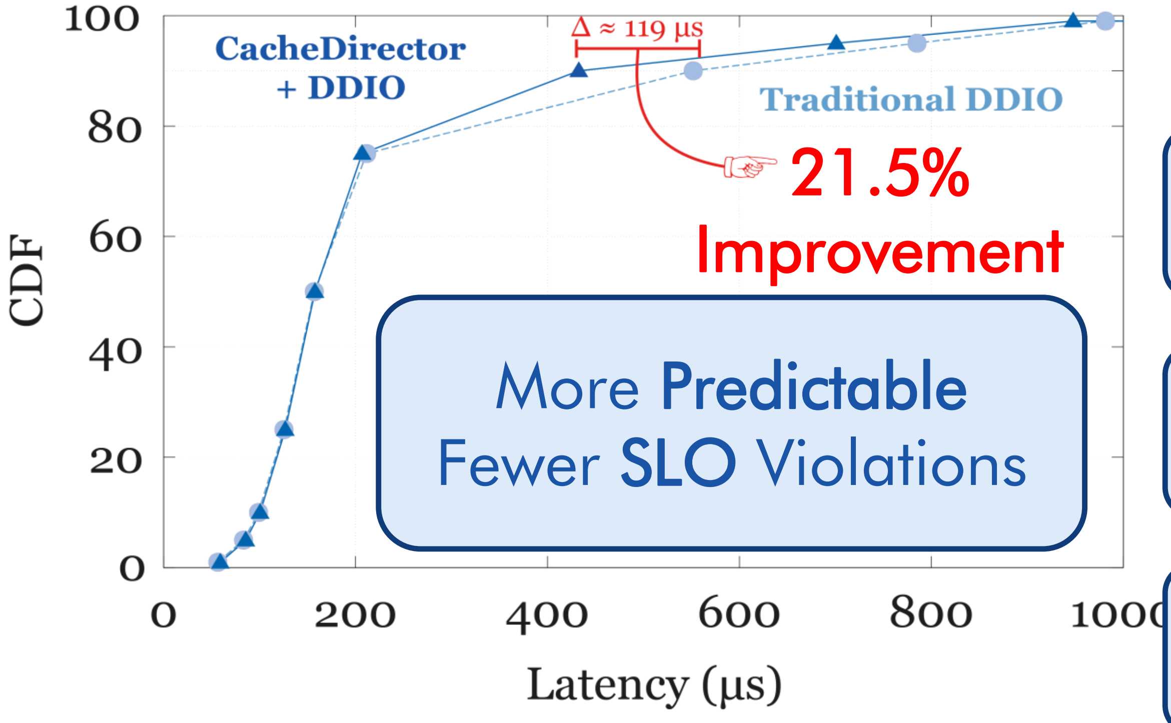
↓

Faster processing time per packet

↓

Reduce queueing time

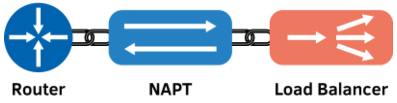
Evaluation — 100 Gbps



21.5% Improvement

More **Predictable**
Fewer **SLO** Violations

Stateful NFV Service Chain



Achieved Throughput
~76 Gbps

Faster access to packet header

Faster processing time per packet

Reduce queueing time

* Service Level Objective (SLO)

Read More …

- More NFV results
- Slice-aware key-value store
- Portability of our solution on Skylake architecture
- Slice Isolation vs. Cache Allocation Technology (CAT)
- More …

Our Paper

SCAN ME

Conclusion

- Hidden opportunity that can decrease average access time to LLC by ~20%

- Useful for other applications

  https://github.com/aliireza/slice-aware

- Meet us at the poster session

WASP WALLENBERG AUTONOMOUS SYSTEMS PROGRAM

SWEDISH FOUNDATION for STRATEGIC RESEARCH

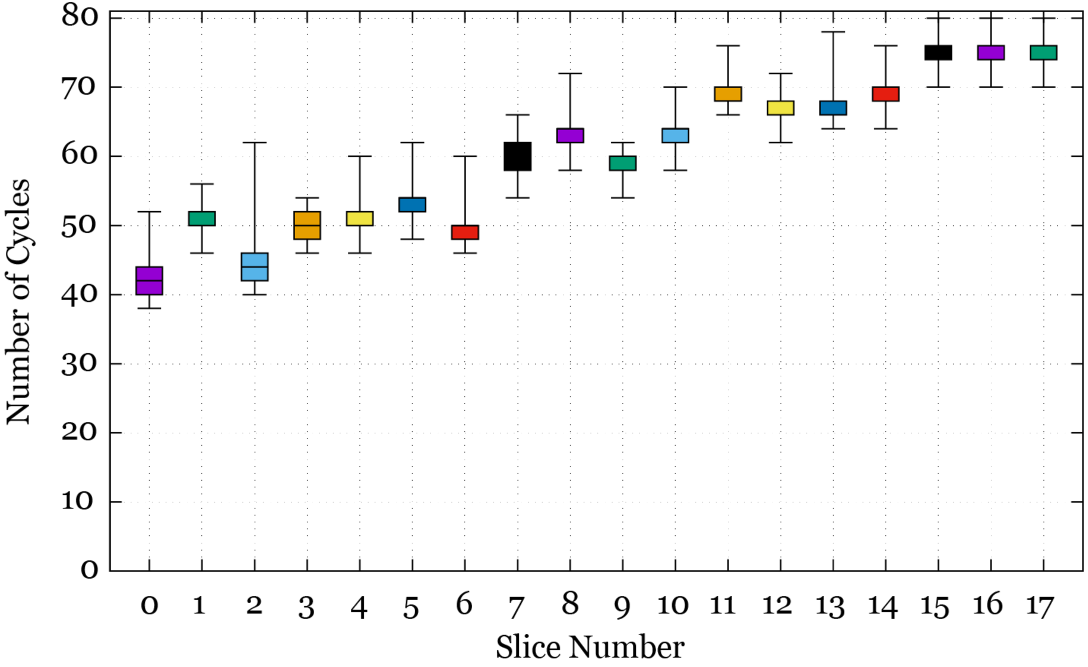
erc European Research Council Established by the European Commission

This work is supported by WASP, SSF, and ERC.

Backup

Portability

- Intel Xeon Gold 6134 (Skylake)
- Mesh architecture
- 8 cores and 18 slices
- Non-inclusive LLC
- Does not affect DDIO



Packet Header Sizes

- IPv4:

  14 B (Ethernet)+ 20 B (IPv4) + 20 B (TCP) < 64 B

- IPv6:

  14B (Ethernet) + 36 B (IPv6) + 20 B (TCP) > 64 B

Any 64 B of the packet can be placed in the appropriate slice

Limitations and Considerations

- Data larger than 64 B
  - Using linked-list and scatter data
  - Future H/W features:
    - Bigger chunks (e.g., 4k pages)
    - Programmable
- Slice Imbalance

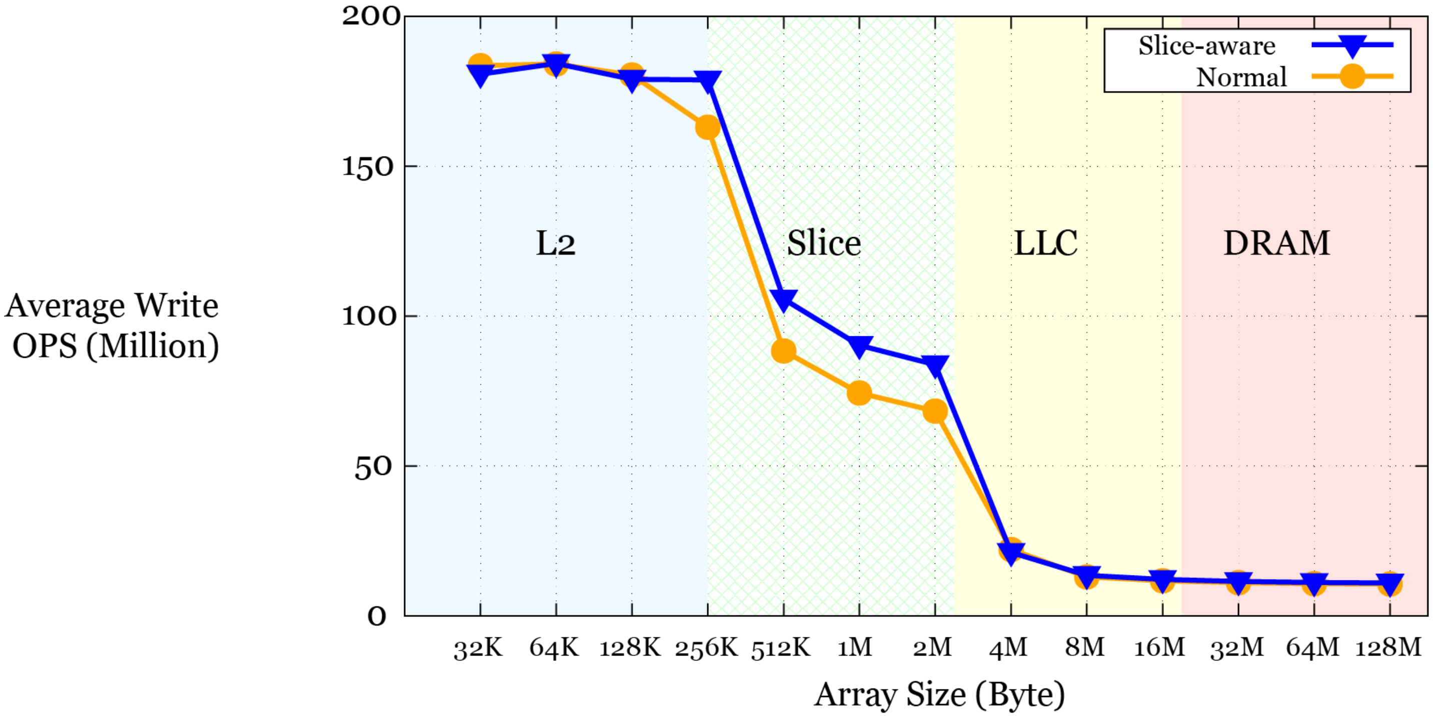
Limiting our application to smaller portion of LLC, but with faster access.

Relevant and Future Works

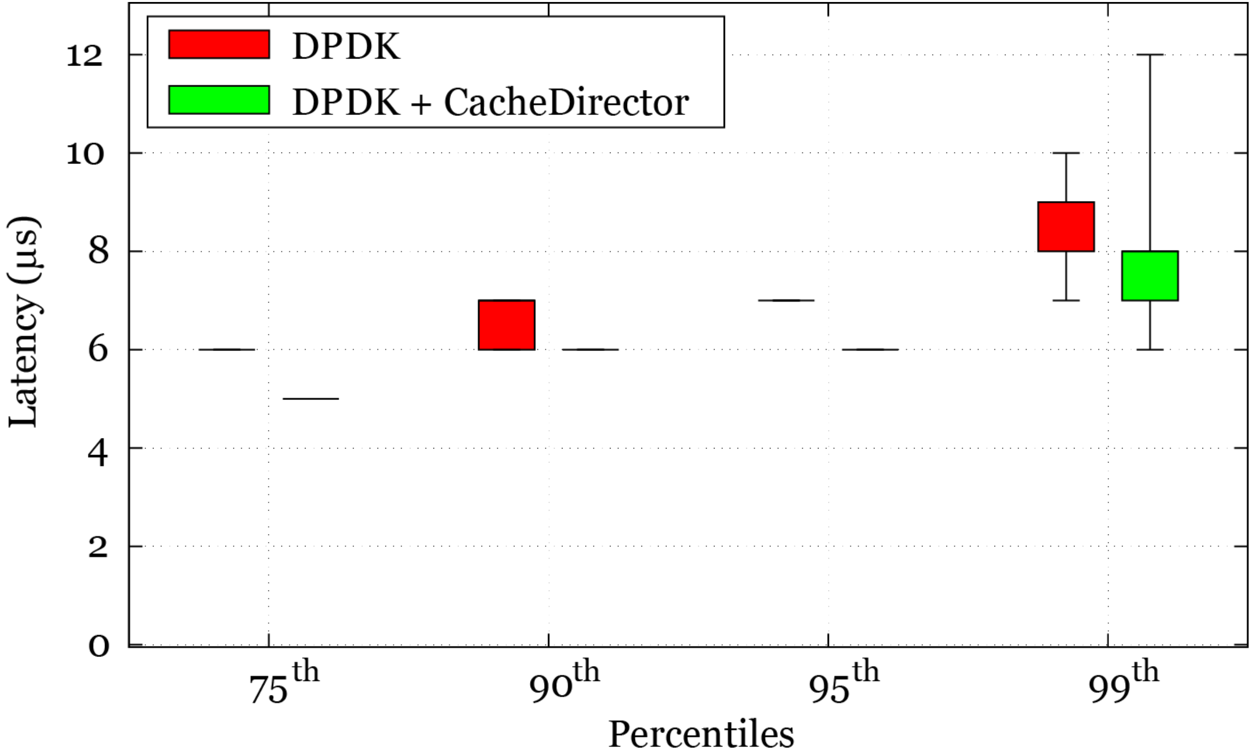
- NUCA
- Cache-aware Memory Management

  (e.g., Partitioning and Page Coloring)
- Extending CacheDirector for the whole packet
- Slice-aware Hypervisor

Slice-aware Memory Management



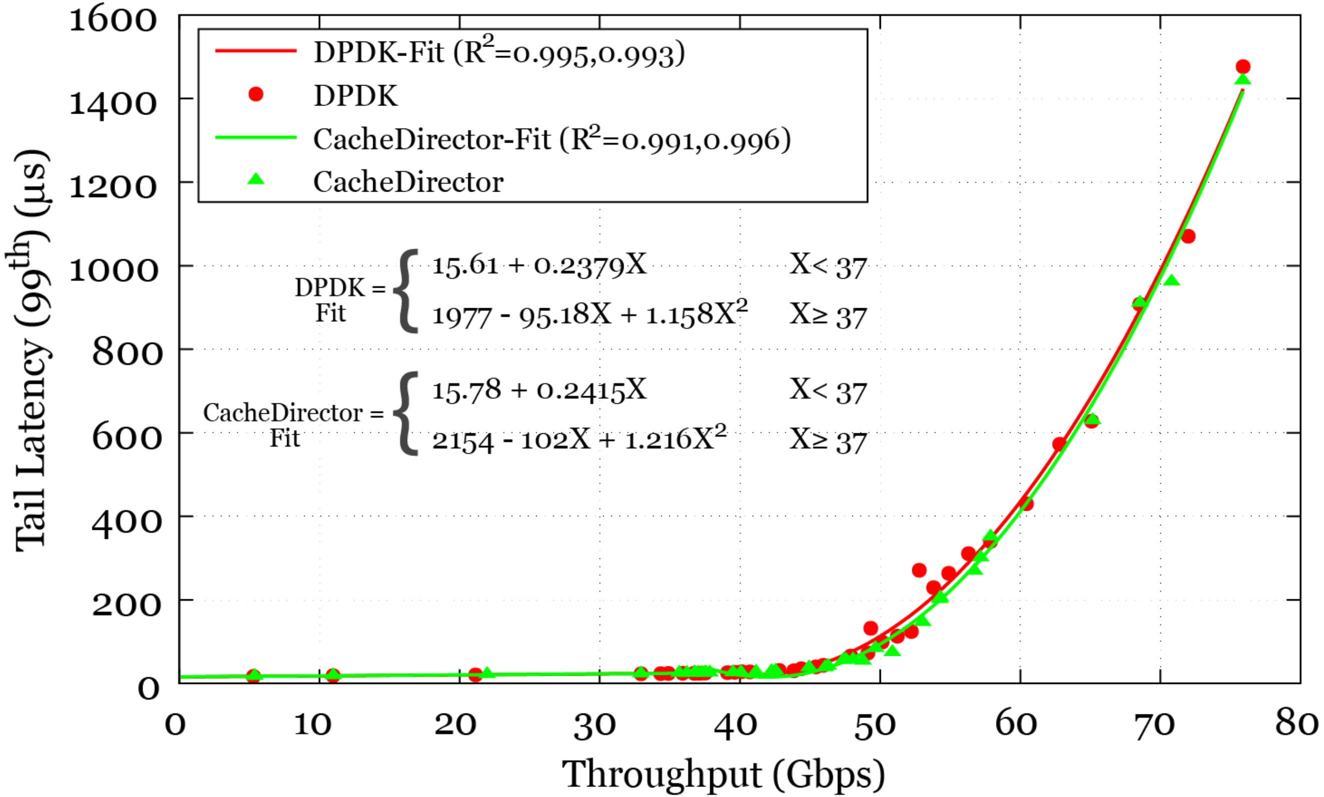
Evaluation — Low Rate



Simple Forwarding Application

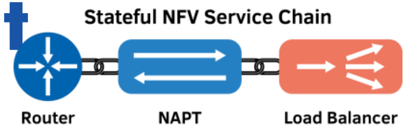
1000 Packets/s

Evaluation — Tail vs. Throughput



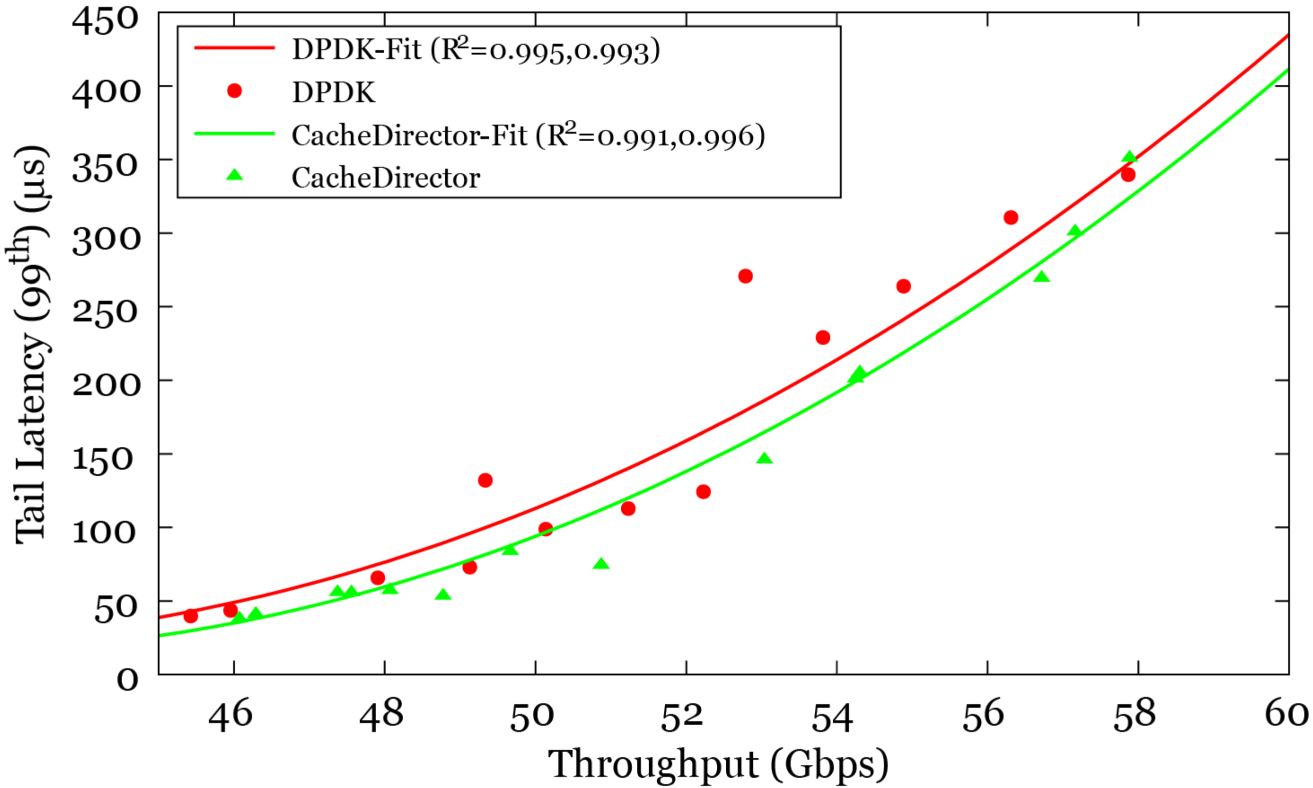
$$\text{DPDK Fit} = \begin{cases} 15.61 + 0.2379X & X < 37 \\ 1977 - 95.18X + 1.158X^2 & X \geq 37 \end{cases}$$

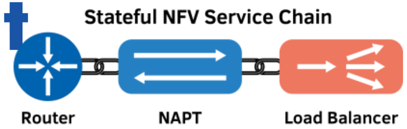
$$\text{CacheDirector Fit} = \begin{cases} 15.78 + 0.2415X & X < 37 \\ 2154 - 102X + 1.216X^2 & X \geq 37 \end{cases}$$



Slightly shifts the knee, which means CacheDirector is still beneficial when system is experiencing a moderate load.

Evaluation — Tail vs. Throughput





Slightly shifts the knee, which means CacheDirector is still beneficial when system is experiencing a moderate load.