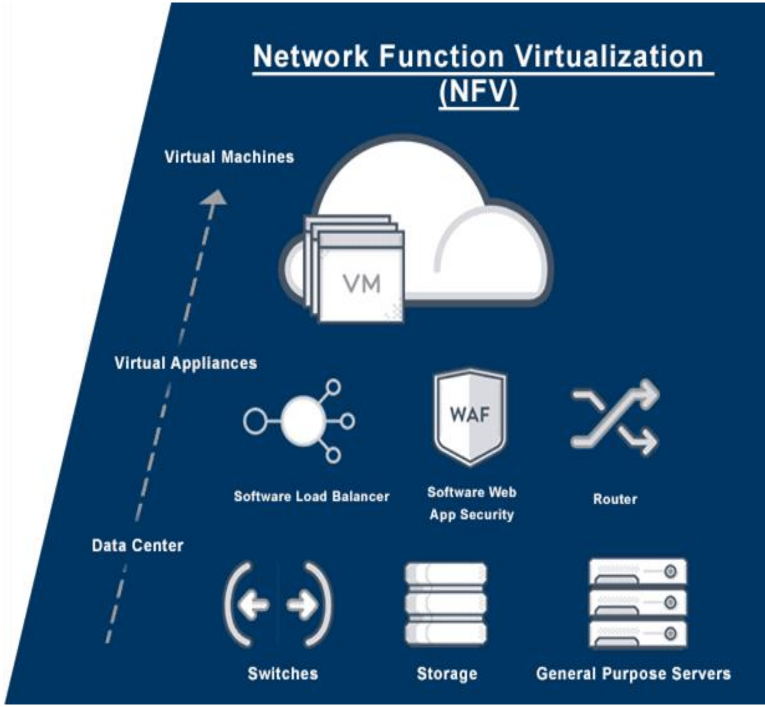
CS 773

Intel® Data Direct I/O Technology

Hari Sharan
harisharan@cse.iitb.ac.in

TRENDS IN NETWORK VIRTUALIZATION

- Virtualization of network capabilities
  - ➢ Software Load Balancers, Routers, Firewalls
  - ➢ Software based network functions on shared general purpose servers
- Demand for high bandwidth
  - ➢ Video streaming
  - ➢ IoT devices
  - ➢ 5G technologies
- Expectations from network virtualization
  - ➢ High throughput
  - ➢ Low latency
  - ➢ Service time guarantees

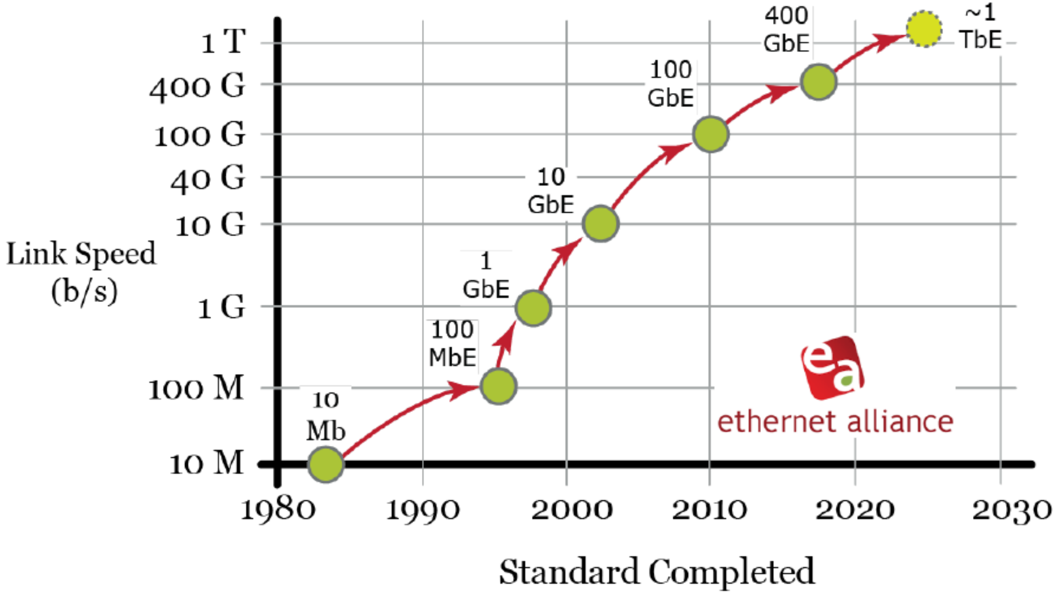


ETHERNET SPEEDS & PROCESSING POWER

- Network bandwidth increasing at faster pace then processor speeds

- Server receiving 64B packets @100 Gbps needs to process packet in 5.12ns

- Software network functions not able to match performance of traditional hardware middle boxes



Source: Ethernet Alliance Roadmap

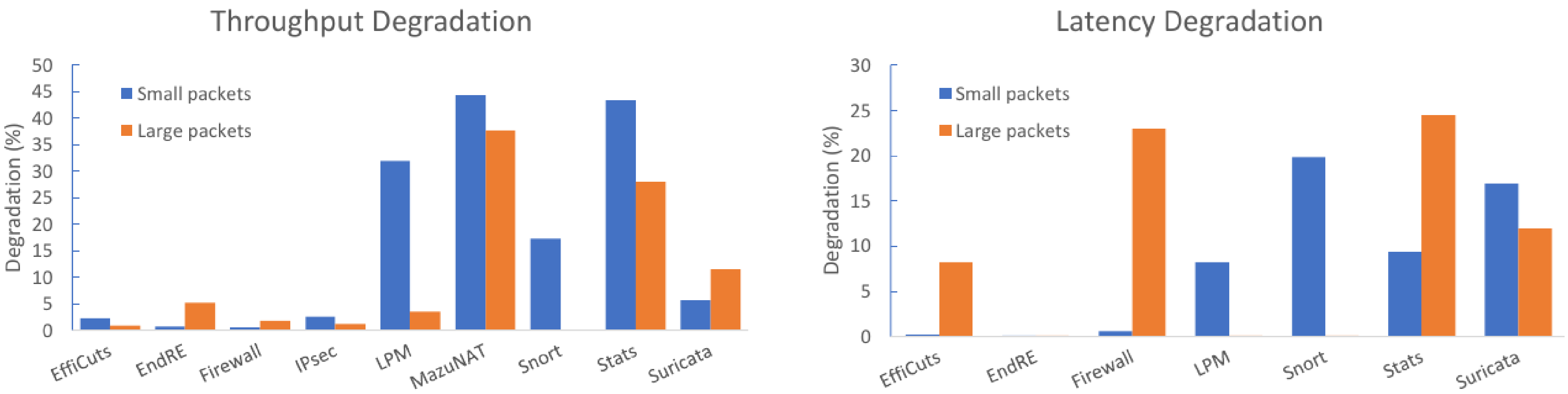
PERFORMANCE ISSUES IN NFV WORKLOADS

- Keeping up with demand for **high bandwidth applications** and **rise in link speeds**
  - ➢ Lesser time to process a packet
  - ➢ Focus on optimizing packet processing for higher throughput

- Providing **service guarantees using software Network Functions**
  - ➢ Predictable performance as provided by hardware

- **Performance isolation** in shared multi-tenant environment
  - ➢ Performance degradation due to contention for shared CPU resources

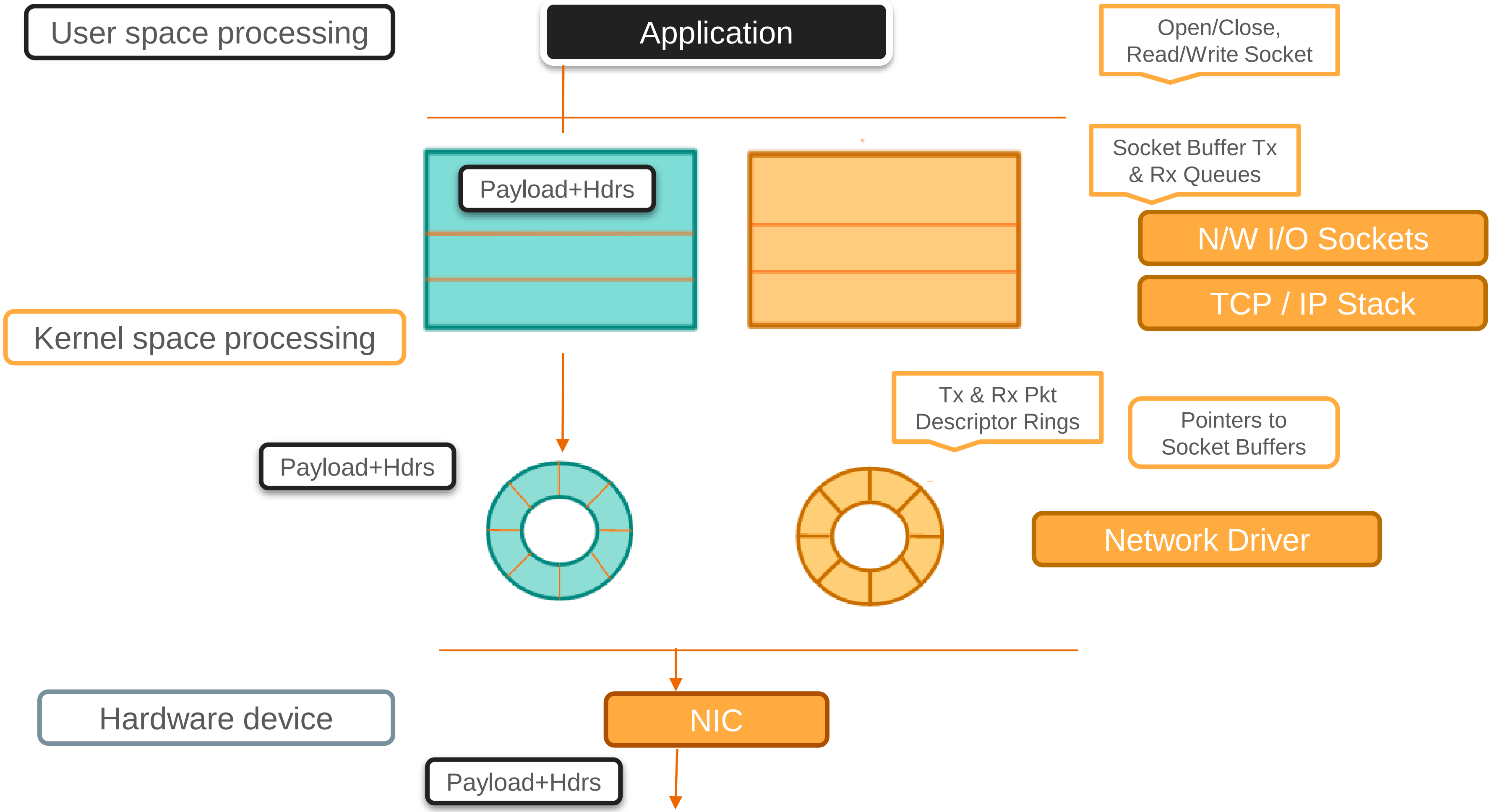
EFFECTS OF CO-RUNNING WORKLOADS ON PERFORMANCE



Maximum performance degradation for minimum and MTU-sized packets without isolation.

Source: ResQ: Enabling SLOs in Network Function Virtualization, NSDI 2018

HOW NETWORK I/O HAPPENS (Tx)



User space processing

Application

Open/Close, Read/Write Socket

Payload+Hdrs

Socket Buffer Tx & Rx Queues

N/W I/O Sockets

TCP / IP Stack

Kernel space processing

Tx & Rx Pkt Descriptor Rings

Pointers to Socket Buffers

Payload+Hdrs

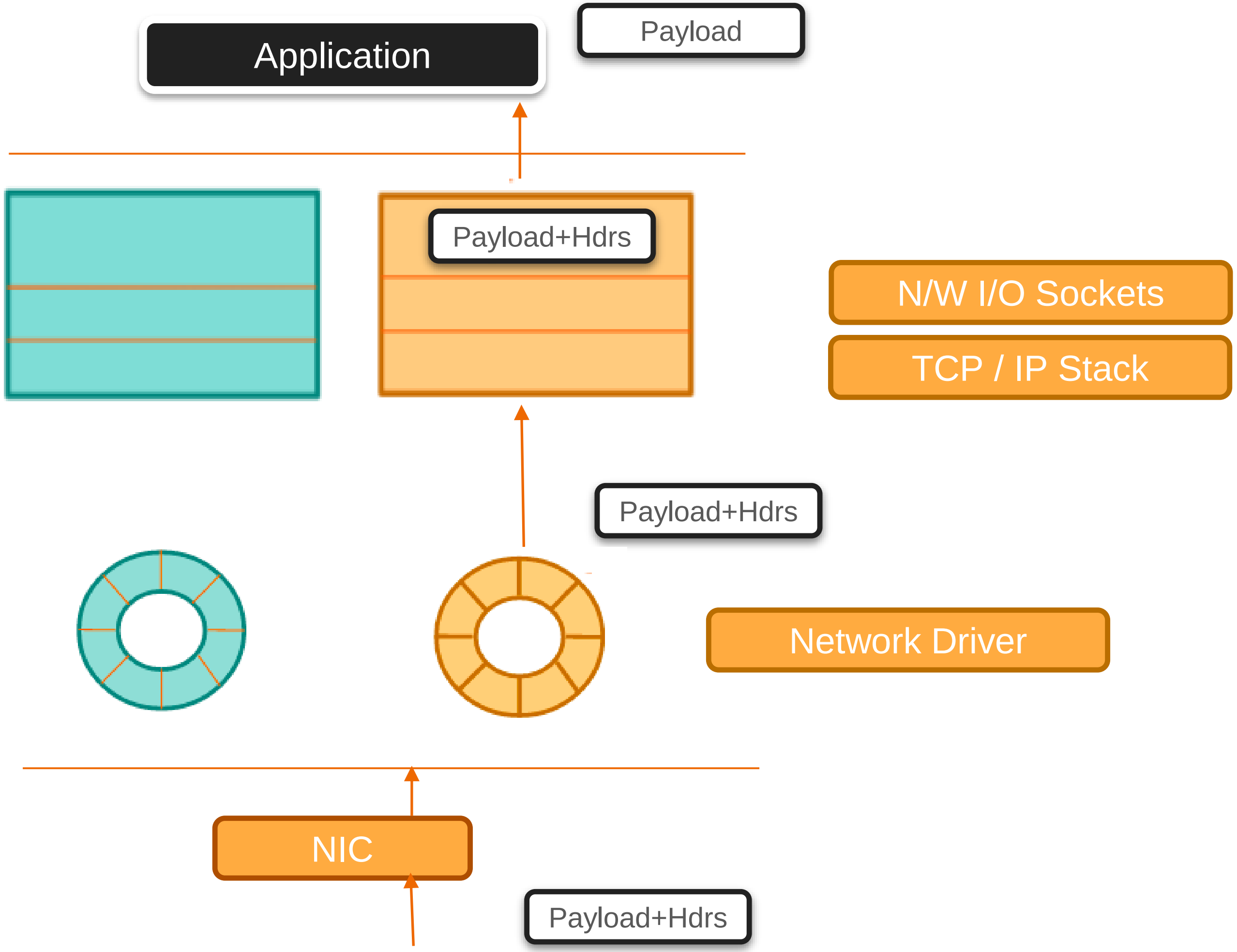
Network Driver

Hardware device

NIC

Payload+Hdrs

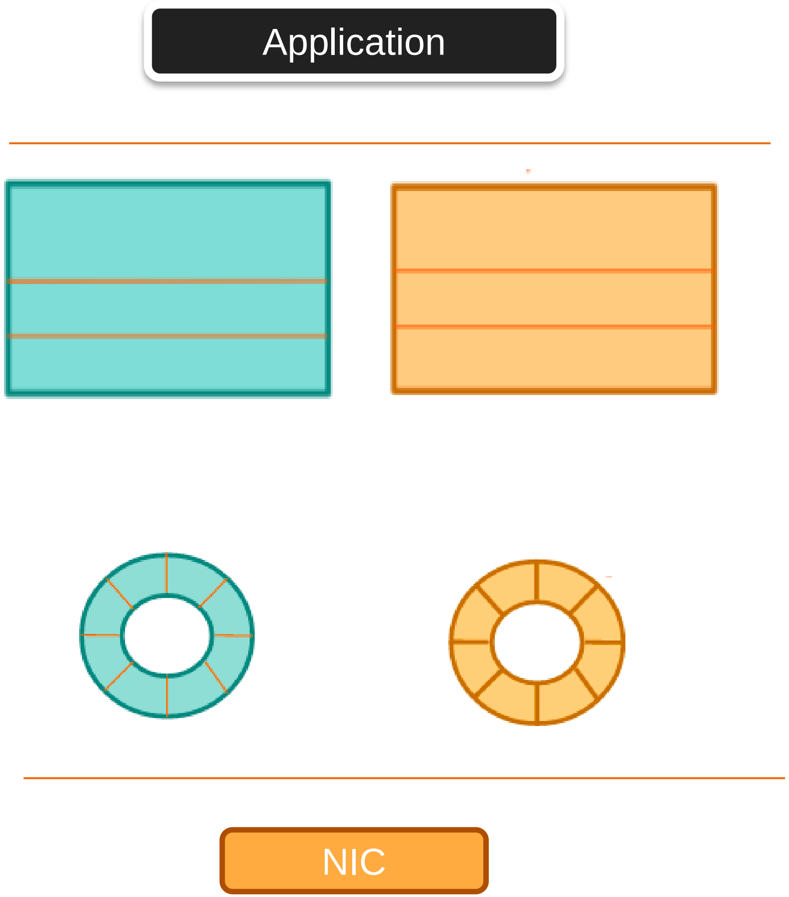
HOW NETWORK I/O HAPPENS (Rx)



NETWORK I/O OVERHEADS & S/W SOLUTIONS

SOME OVERHEADS

- Switching from user mode to kernel mode
- Multiple copying of packets
- Interrupt processing



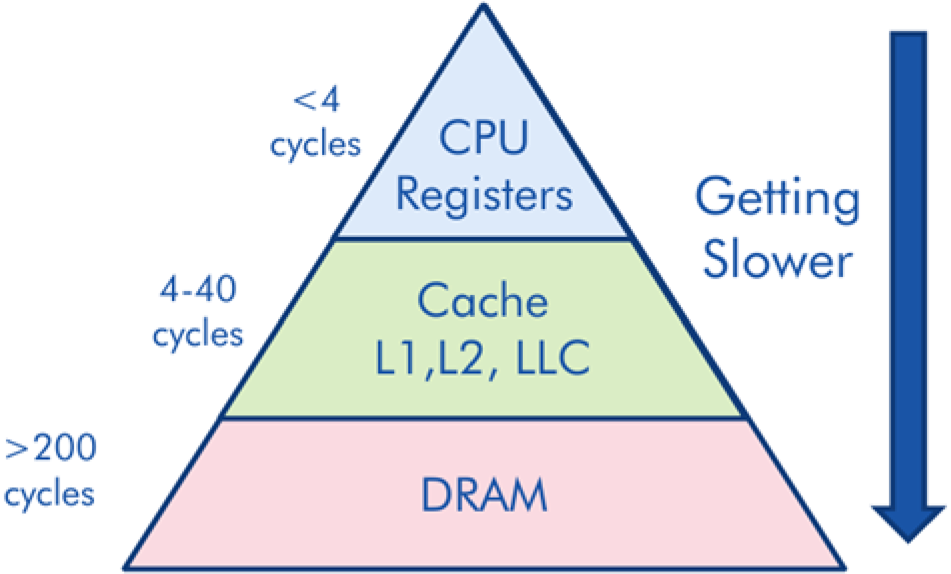
SOME SOLUTIONS

- Kernel bypass techniques
- NAPI (New API) to reduce interrupt load
- Receive side scaling at NIC

ROLE OF MEMORY ACCESS SPEEDS

- Socket buffers, network driver Tx / Rx rings frequently accessed during packet processing

- Faster packet processing if memory requests for these data structures serviced from cache



Memory Hierarchy

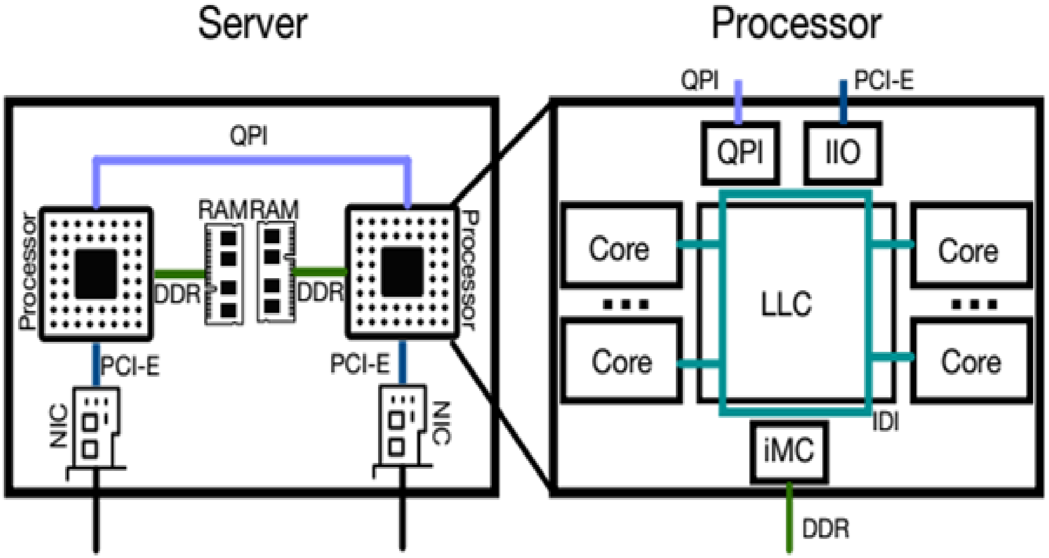
Source: Farshin DCA, Usenix ATC 2020

INTEL PROCESSOR UNCORE RESOURCES

- Processor uncore resources - Last Level Cache (LLC), integrated Memory Controller (iMC), Integrated I/O Controller (IIO) shared among multiple cores.

- Separate cores & L1/L2 caches for different workloads

- Sharing of uncore resources leads to
  - Resource contention
  - Performance degradation



Multi-core Processor Components

Source: ResQ, NSDI 2018

INTEL DATA DIRECT I/O TECHNOLOGY

- Microarchitecture level optimization technique

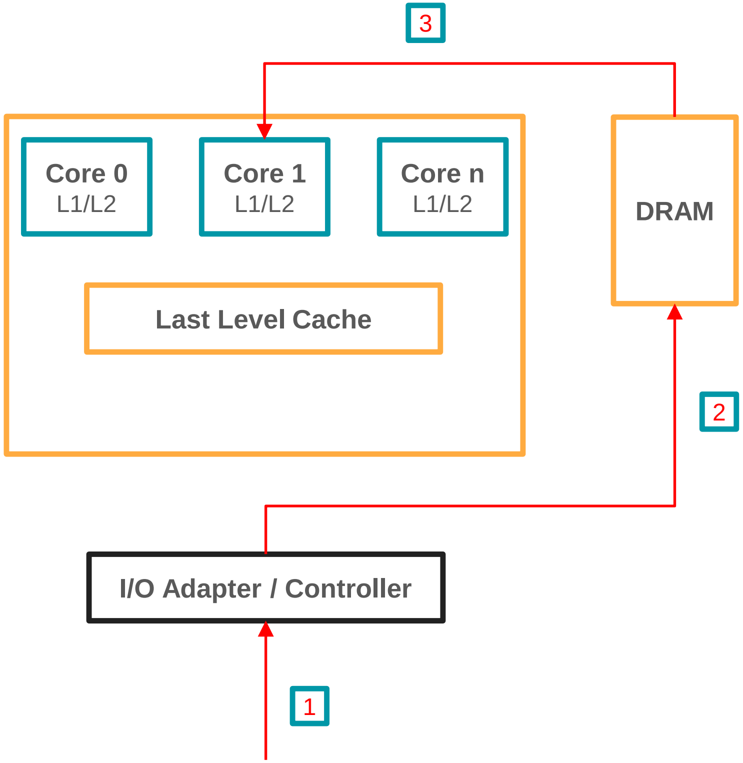
- Introduced with Intel® Xeon ® processor E5 product family

- A platform technology that enables I/O data transfers that require far fewer trips to memory

- Makes the processor cache the primary destination and source of I/O data rather than the main memory.

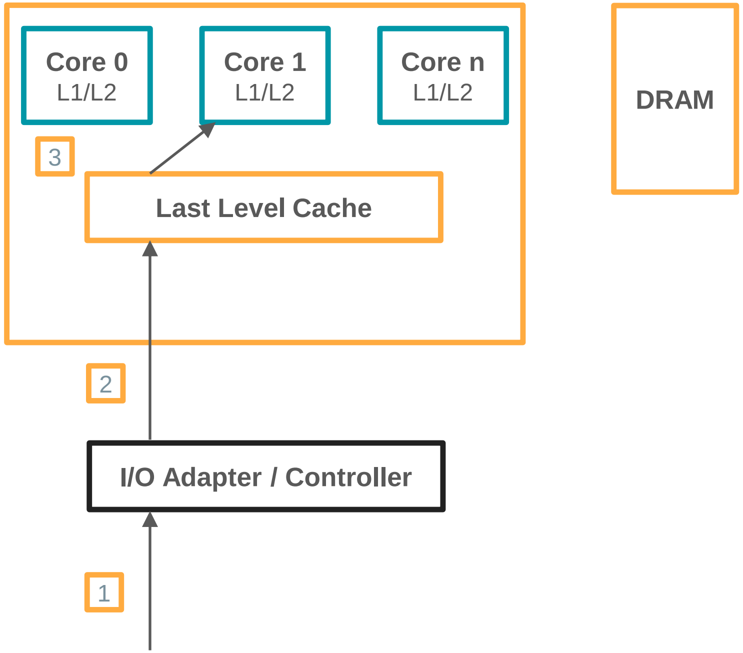
INTEL DATA DIRECT I/O TECHNOLOGY

- Traditionally, incoming data from NIC DMA’ed to DRAM.
- Data fetched to cache for processing
- Requires a memory write and a memory read before processing begins
- Inefficient due to
  - Number of accesses to Main Memory
  - Access latency to I/O data
  - Memory bandwidth usage



INTEL DATA DIRECT I/O TECHNOLOGY

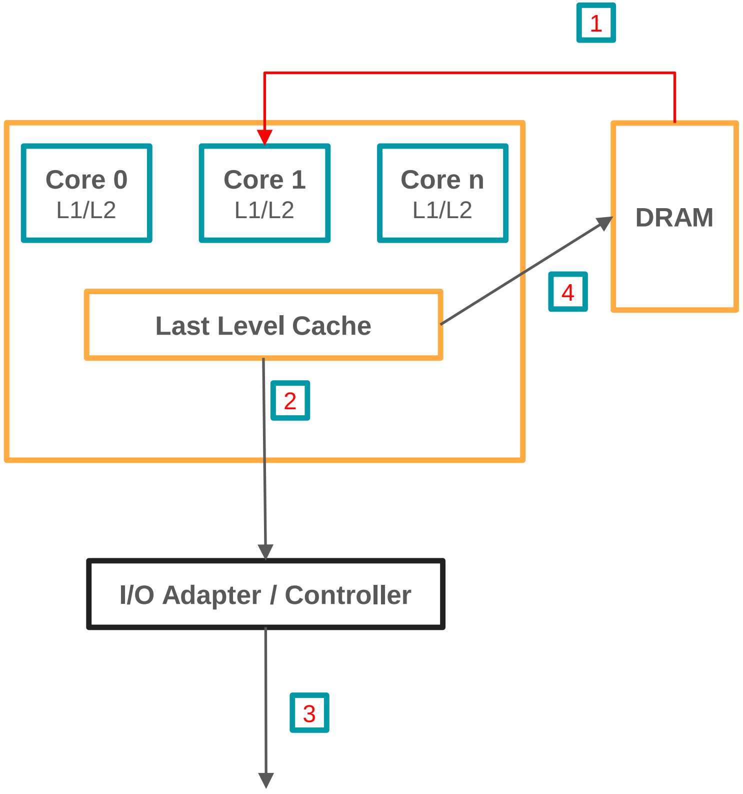
- With Intel DDIO
  - Avoid multiple reads from and writes to memory
  - Reduces latency, memory bandwidth requirements



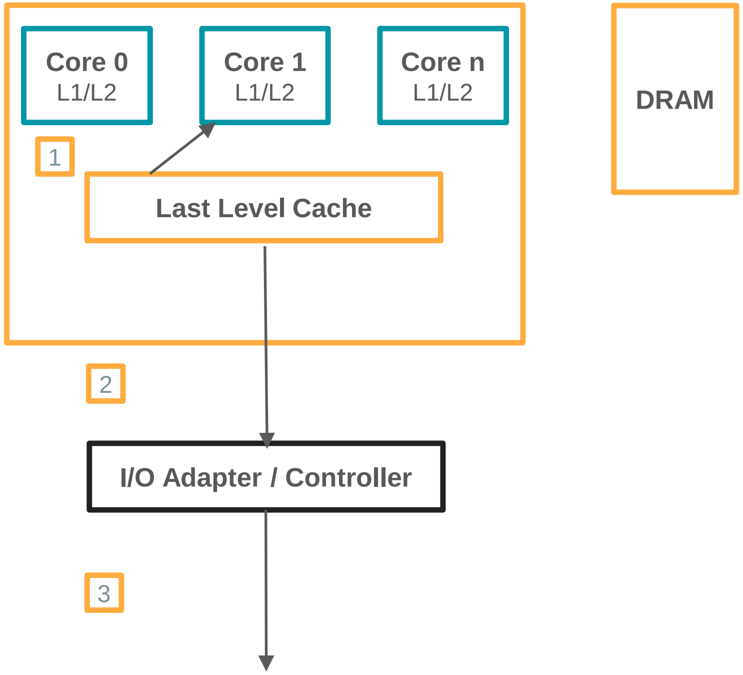
INTEL DATA DIRECT I/O TECHNOLOGY

- DDIO limited to 10% of LLC (2 LLC ways) and cannot be partitioned using CAT
- DDIO ways can be written by cores, whereas I/O can’t write into core ways
- If this limit is exceeded, new inbound I/O will continue to go directly to cache, but the least-recently used I/O will be written to memory to make room for the new data.
- No changes to drivers, applications, operating systems or firmware

I/O READ



Without Intel DDIO

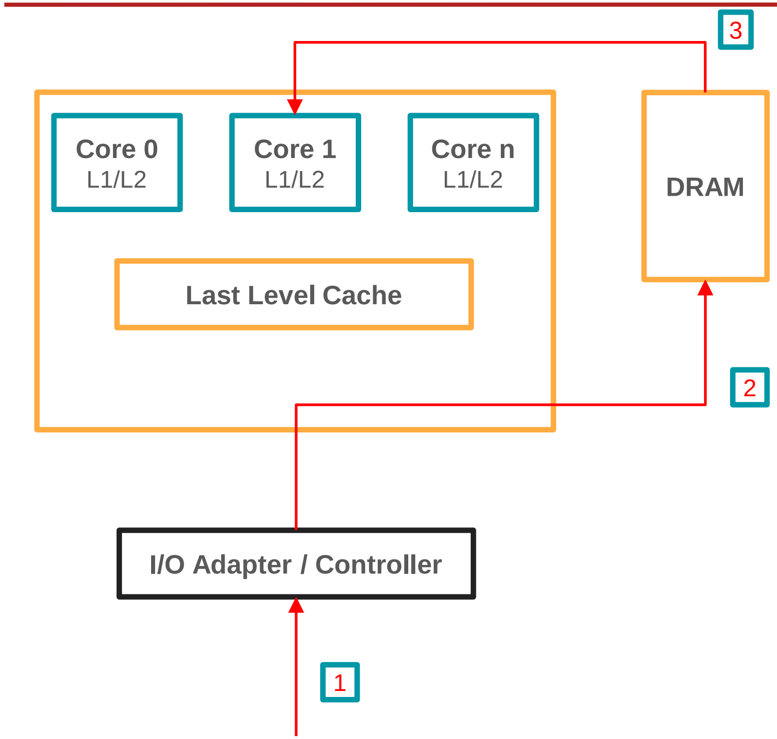


With Intel DDIO

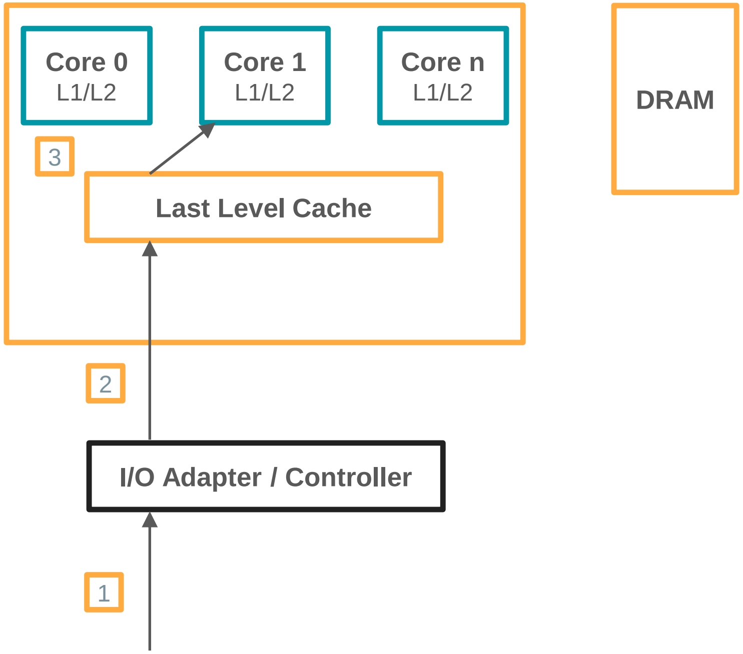
I/O READ

- I/O reads are achieved with fewer trips to memory
- The data in the cache is not disturbed by an I/O data consumption operation
- I/O reads to memory are non-allocating
  - If line is not present in LLC it is not allocated
  - If line is in LLC, it stays there
  - If it is in one of higher caches, it moves or gets copied to LLC
- LLC Hit: Data serviced from LLC Cache line
- LLC Miss: Data serviced from Memory

I/O WRITE



Without Intel DDIO



With Intel DDIO

I/O WRITE

Two modes of operation:

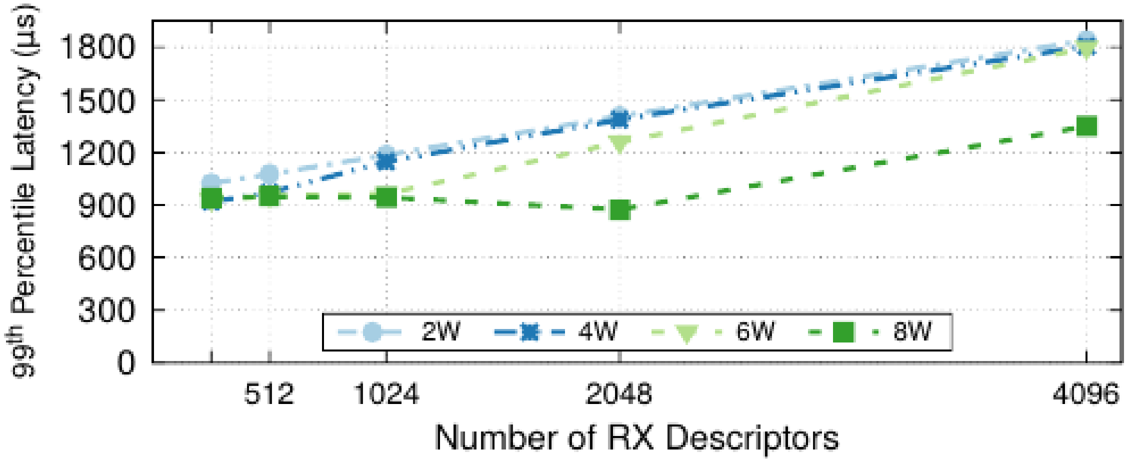
- Write Update (LLC Hit): Cache line is overwritten with new data
  - If the memory addresses for the data being delivered already exists in the LLC

- Write Allocate (LLC Miss): Requires allocation of Cache line in LLC.
  - If the memory addresses for the data being delivered does not exist in LLC, then no trips to memory are needed.

TUNING INTEL DDIO

- Model Specific Register (MSR) - “IIO LLC WAYS”
- For Skylake, default value is 0x600 (two bits set)
- Max value on testbed CPU 0x7FF (11 bits set - same as LLC ways)
- Disabling DDIO
  - Globally- setting the Disable_All_Allocating_Flows bit in “iiomiscctrl” register
  - Per-root PCIe port- setting bit NoSnoopOpWrEn and unsetting bit Use_Allocating_Flow_Wr in “perfctrlsts_0” register

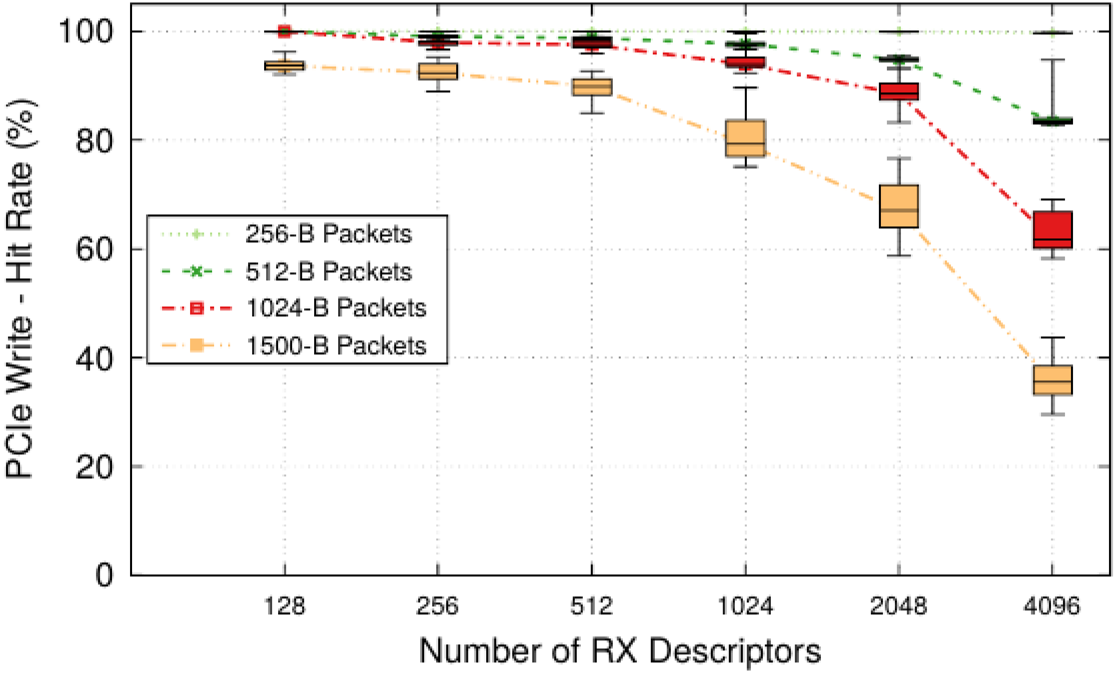
TUNING INTEL DDIO



Using more DDIO ways ("W") enables cores to forward large packets at 100 Gbps with large no. of descriptors achieving better or similar latency

Source: Reexamining Direct Cache Access to Optimize I/O Intensive Applications, Usenix ATC 2020

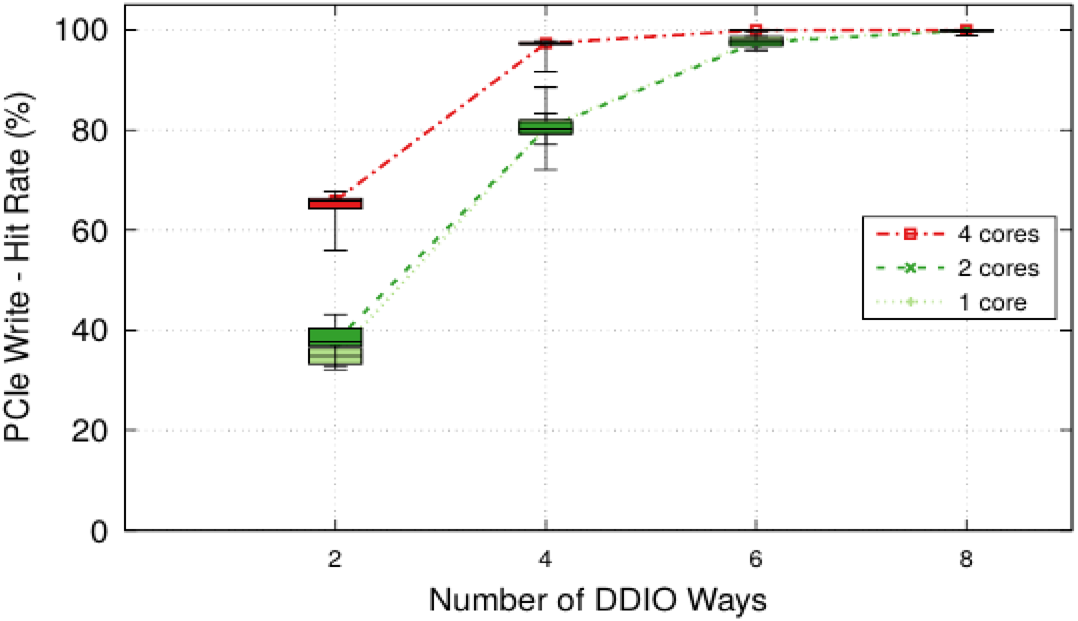
Packet Size & Rx Descriptors



Increasing the no. of packet descriptors and/or packet size adversely affects the performance of 2-way DDIO

Source: Reexamining Direct Cache Access to Optimize I/O Intensive Applications, Usenix ATC 2020

No. of Cores & DDIO Capacity



Increasing no. of DDIO ways have similar effect as Increasing the no. of Cores (Forward 1500-B packets with 4096 Rx Descriptors)

Source: Reexamining Direct Cache Access to Optimize I/O Intensive Applications, Usenix ATC 2020

PERFORMANCE ISSUES

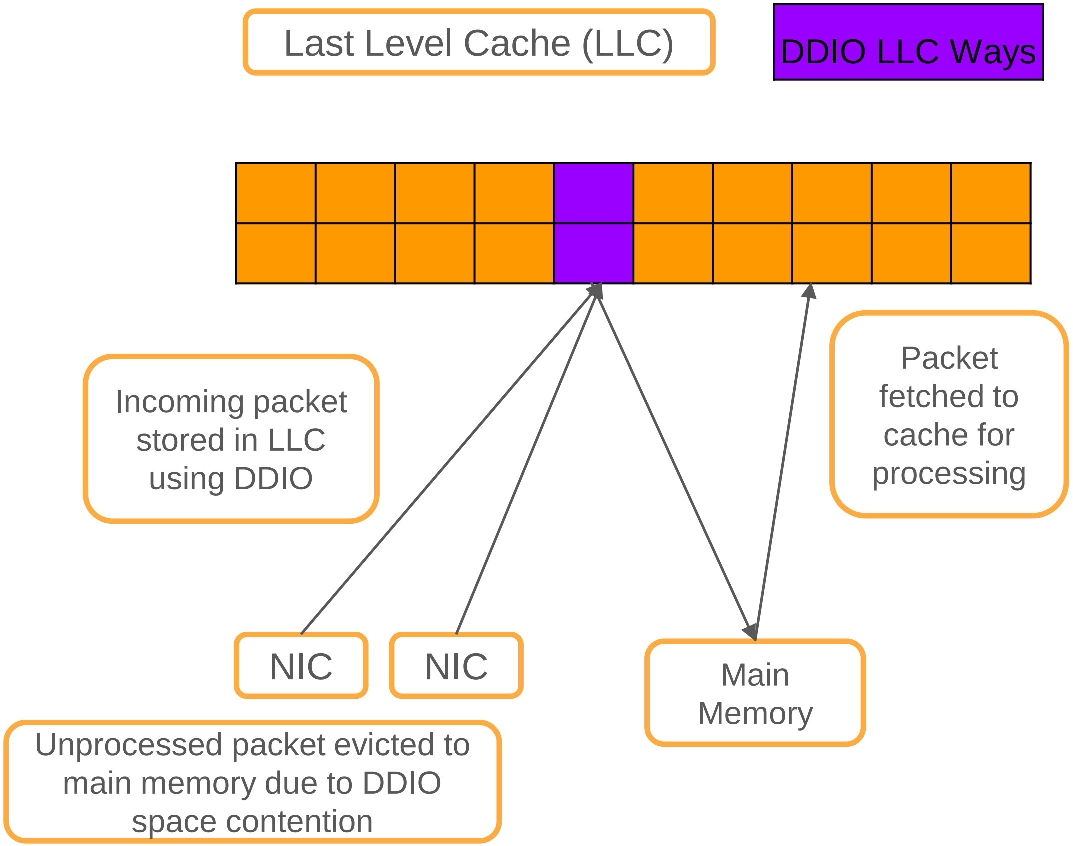
Leaky DMA Problem

DDIO limited to 10% of LLC capacity

DDIO LLC space contention

Proposed solution:
CAT + Buffer Sizing

Drawback:
Packet drops due to shallow buffers



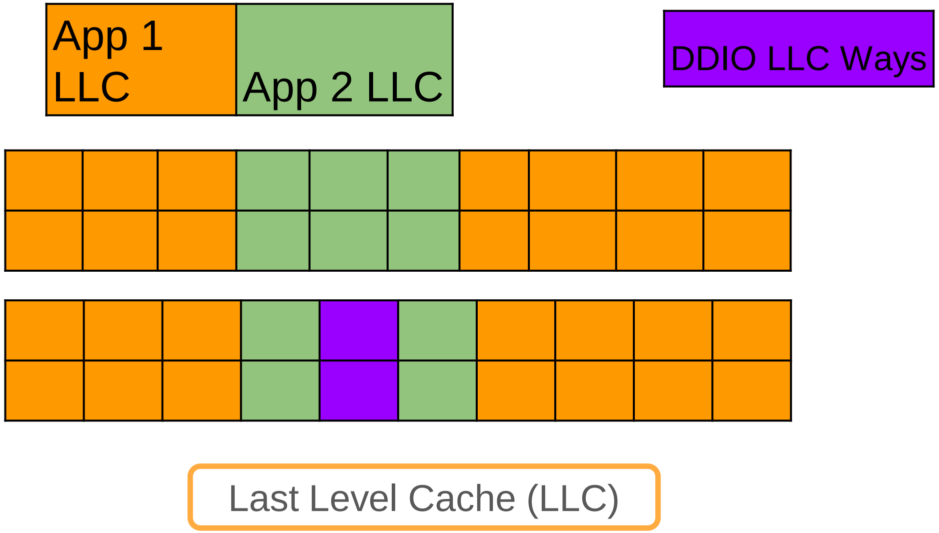
Last Level Cache (LLC)
DDIO LLC Ways
Incoming packet stored in LLC using DDIO
Packet fetched to cache for processing
NIC
NIC
Main Memory
Unprocessed packet evicted to main memory due to DDIO space contention

ResQ: Enabling SLOs in Network Function Virtualization, NSDI 2018

PERFORMANCE ISSUES

Latent Contender Problem

- DDIO’s LLC ways may be allocated to certain cores running LLC-sensitive workloads.
- Although LLC ways are isolated from the core’s point of view, DDIO still contends with the cores for the LLC



CONCLUSION

- Intel DDIO reduce Latency, increases system I/O bandwidth and throughput by writing the incoming I/O packets directly to LLC

- With DDIO tuning, possible to increase performance in comparison to default DDIO configuration

- Increasing no. of DDIO ways have similar effect as increasing the no. of cores

- Performance issues such as Leaky DMA, Latent Contender requires proper tuning of DDIO and workload characteristics

Source: Reexamining Direct Cache Access to Optimize I/O Intensive Applications, Usenix ATC 2020

THANK YOU