SDC 20

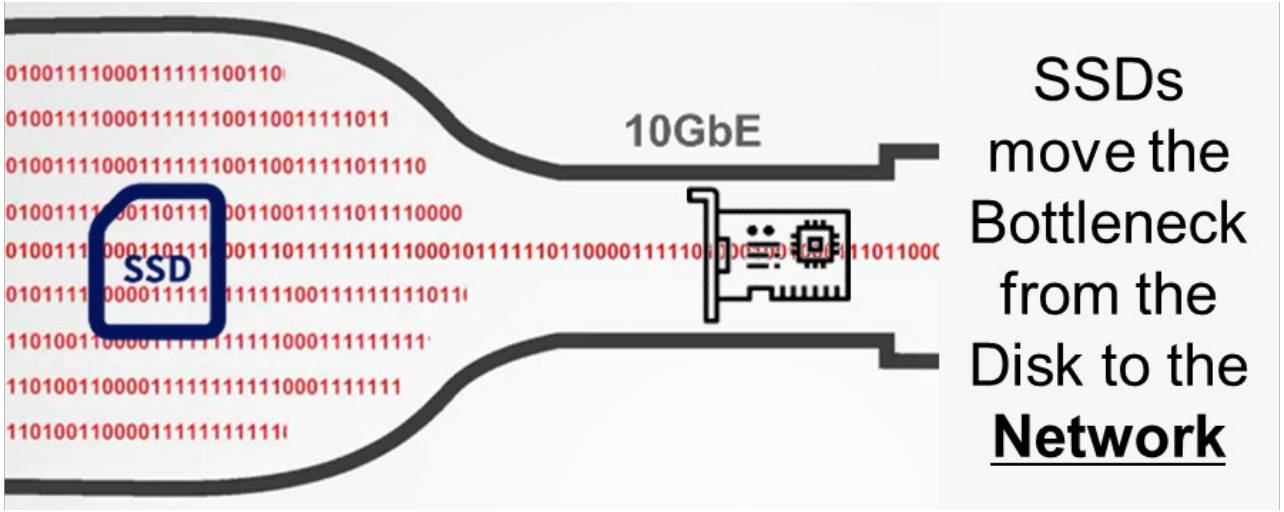
BY Developers FOR Developers

Storage Developer Conference
September 22-23, 2020

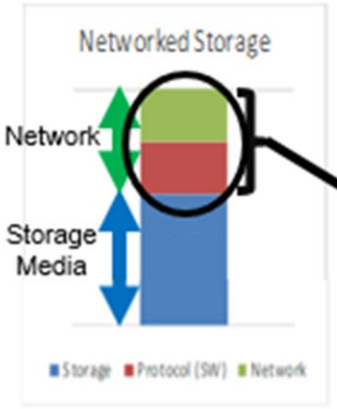
NVMe-oF on RDMA Performance Challenges and Solutions in Commodity Servers

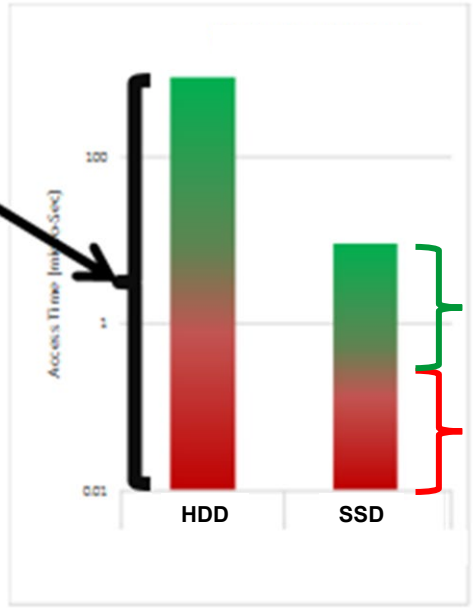
Yamin Friedman & Rob Davis
Nvidia networking business unit

SSDs Create a Storage Networking Performance Bottleneck



200Gb Network Speeds and NVMe-oF on RoCE to the Rescue

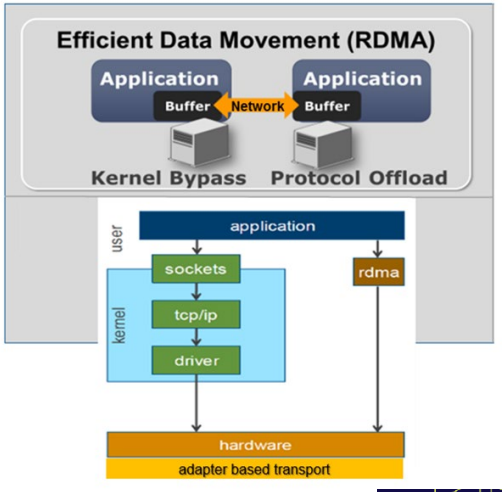




Faster Network

Faster Protocol

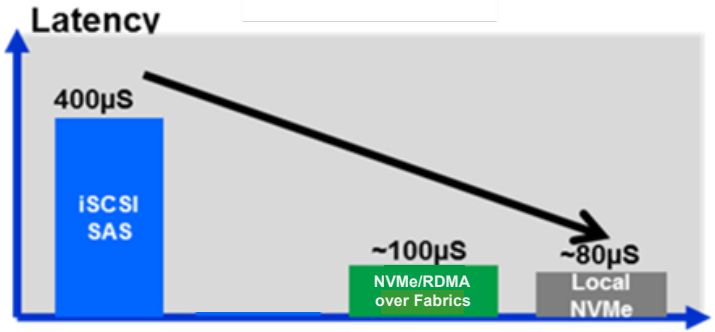
NVMe-oF



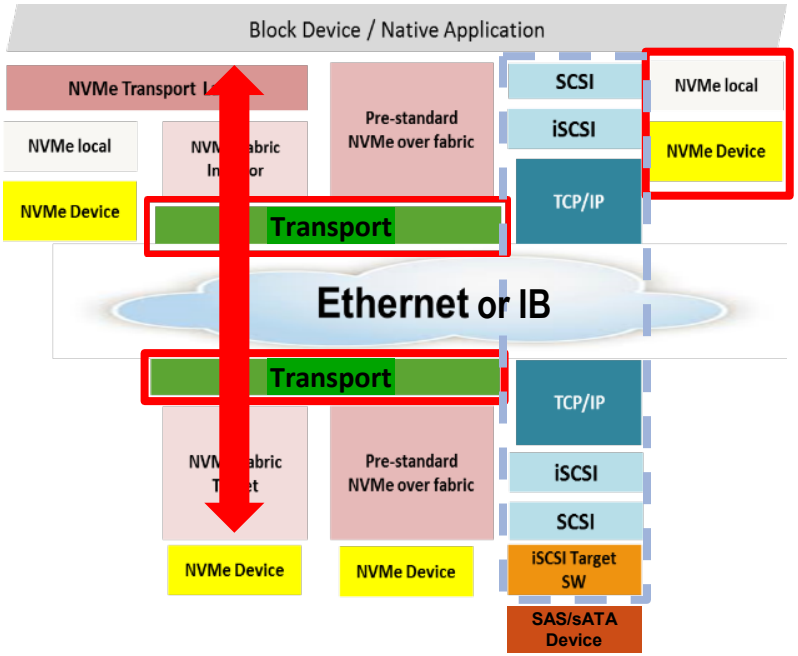
NVMe OVER FABRICS POWERED BY RoCE

What is NVMe over Fabrics (NVMe-oF)?

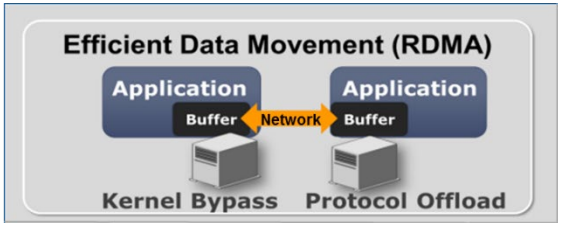
- Extending NVMe protocol over a fabric
  - NVMe commands and data structures are transferred end to end
  - Enables SAN features for NVMe based storage systems
- Bypassing legacy stacks for performance
- First products and early demos all used RDMA
  - Standard supports multiple transports
- Impressive performance gains

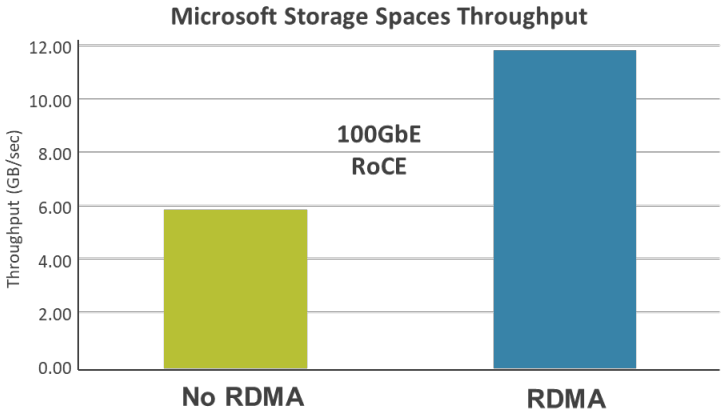


https://www.theregister.co.uk/2018/08/16/pavilion_fabrics_performance/



What is RDMA?



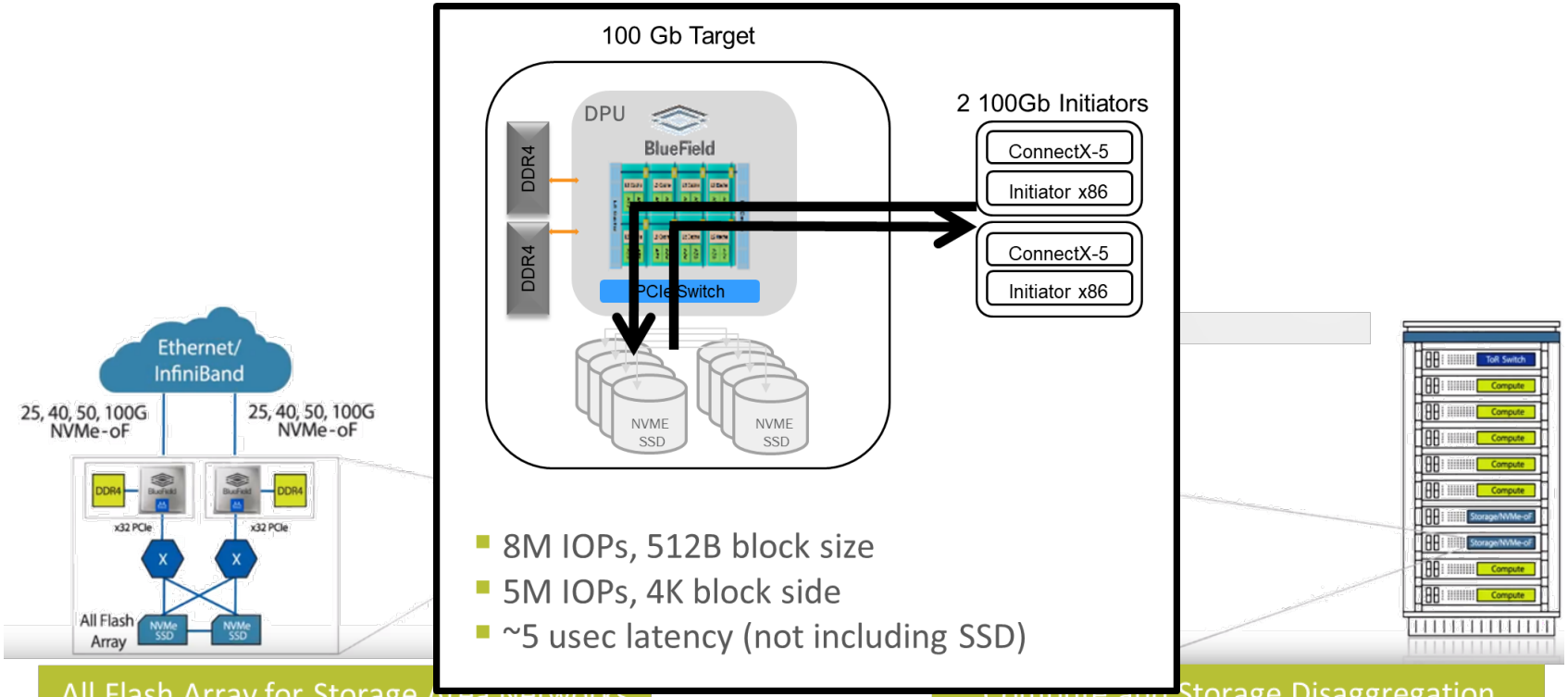


With RDMA

- 2x Better Bandwidth
- Half the Latency
- 33% Lower CPU

See MS demo: https://www.youtube.com/watch?v=u8ZYhUjSUoI

Applications That Need Performance

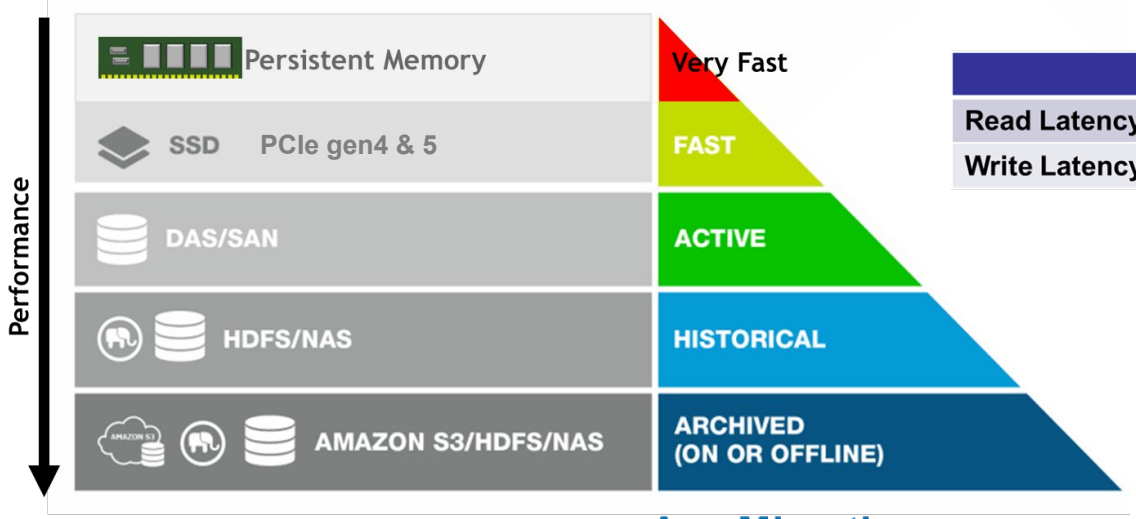


- 8M IOPs, 512B block size
- 5M IOPs, 4K block side
- ~5 usec latency (not including SSD)

All Flash Array for Storage Area Networks

Compute and Storage Disaggregation

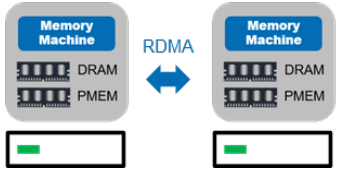
Applications That Need Even More Performance



	PM	NAND
Read Latency	~100ns	~20,000ns
Write Latency	~500ns	~50,000ns

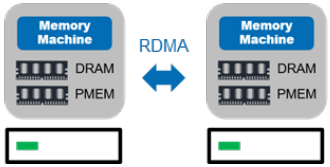
HA

Replicating PM to other servers for failover



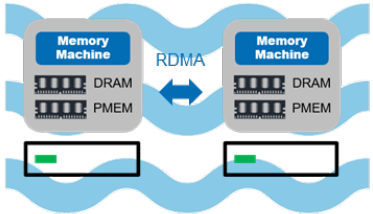
App Migration

Memory Machine migrates app + memory to other servers



Scale-Out

Create memory lakes by pooling server memory



System interrupt overload

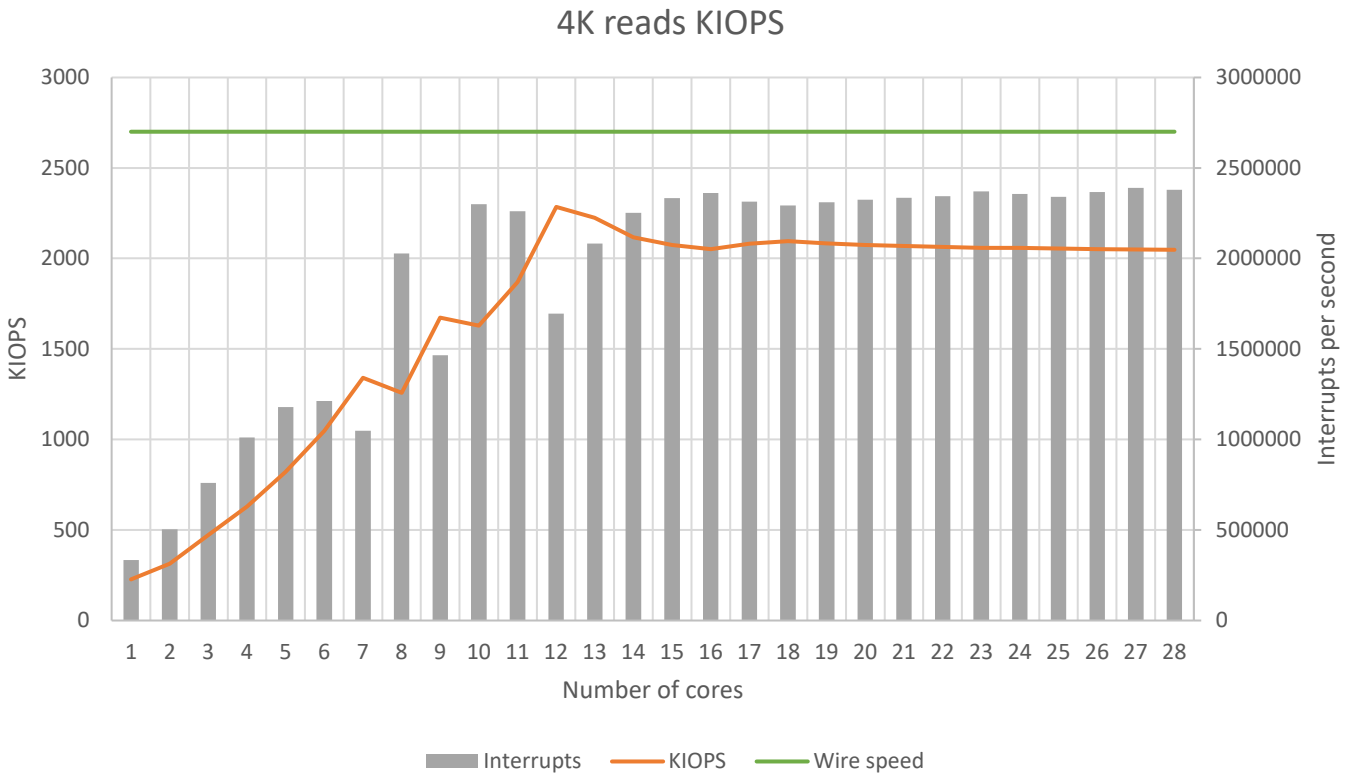
System interrupts and RDMA

- In order to handle networking traffic the kernel uses interrupts
- In NVMe-oF storage flows there are multiple interrupts per IO
- When the system is overloaded with interrupts it interferes with the ability to perform RDMA

Test setup

- CPU: Intel(R) Xeon(R) Platinum 8176M CPU @ 2.10GHz 28 cores
- Linux version: 5.7.0-rc3 with added shared CQs
- Network adapter: Mellanox ConnectX-5 Ex

Interrupt overload



Dynamic Interrupt Moderation (DIM)

Interrupt Moderation

- Interrupt moderation is a mechanism for aggregating more work per system interrupt
- RDMA completion queues can be configured to arm a system interrupt only once enough work has been gathered or a timeout reached
- There is no single configuration that is optimal for a specific system and time

Dynamic interrupt moderation

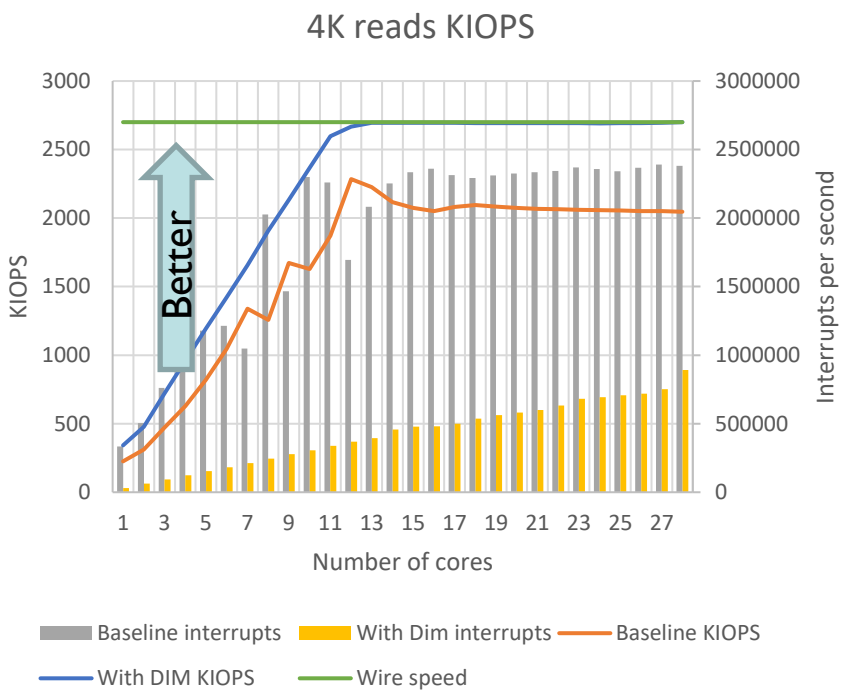
- In order to effectively use interrupt moderation we need a method that adapts to the current state

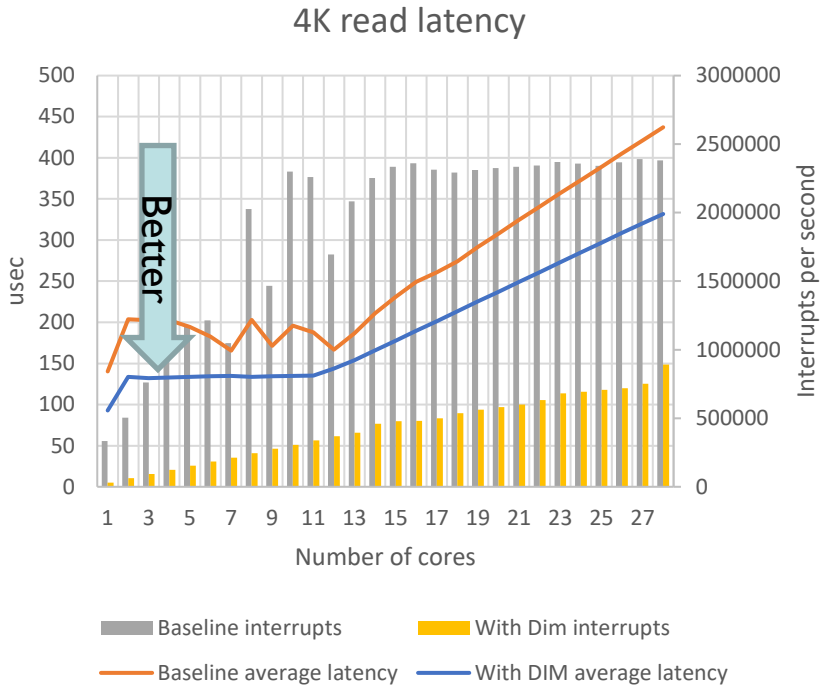
- DIM monitors the current state of the system and determines the optimal parameters moment to moment

DIM algorithm

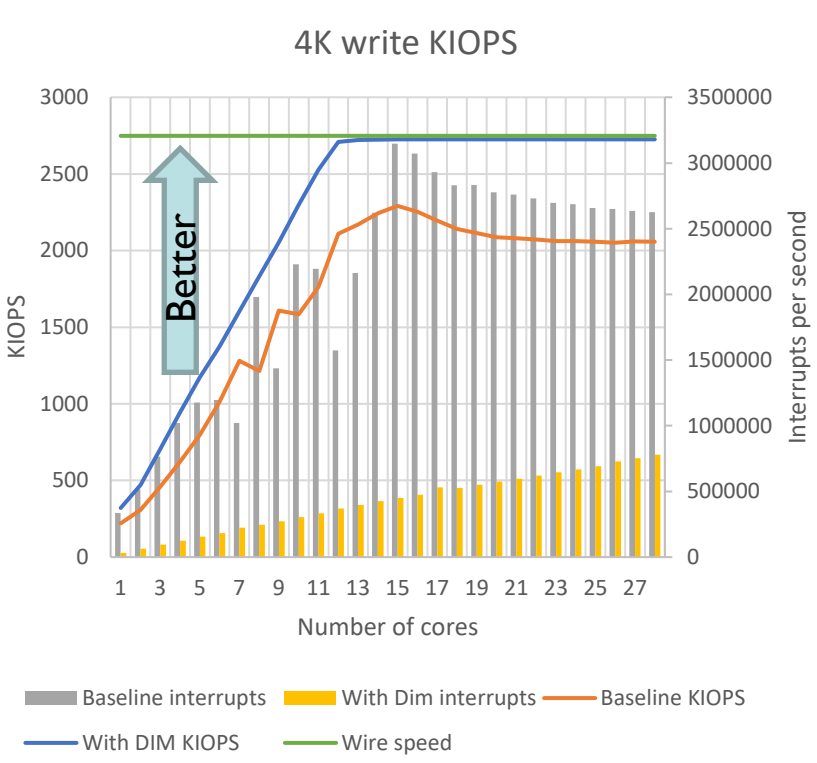
- Statistics collected:
  - Number of completions polled
  - Number of interrupts
- Three stage algorithm:
  - Optimize for number of completions
  - Optimize for ratio of completions to interrupts
  - Reduce moderation if ratio is low

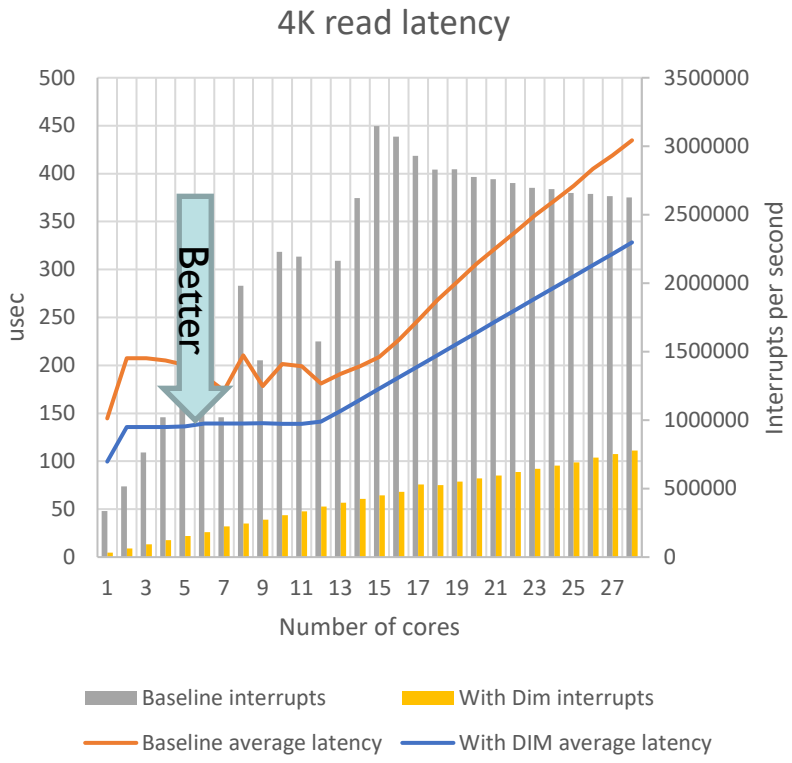
4K read comparison





4K write comparison





Shared Completion Queues

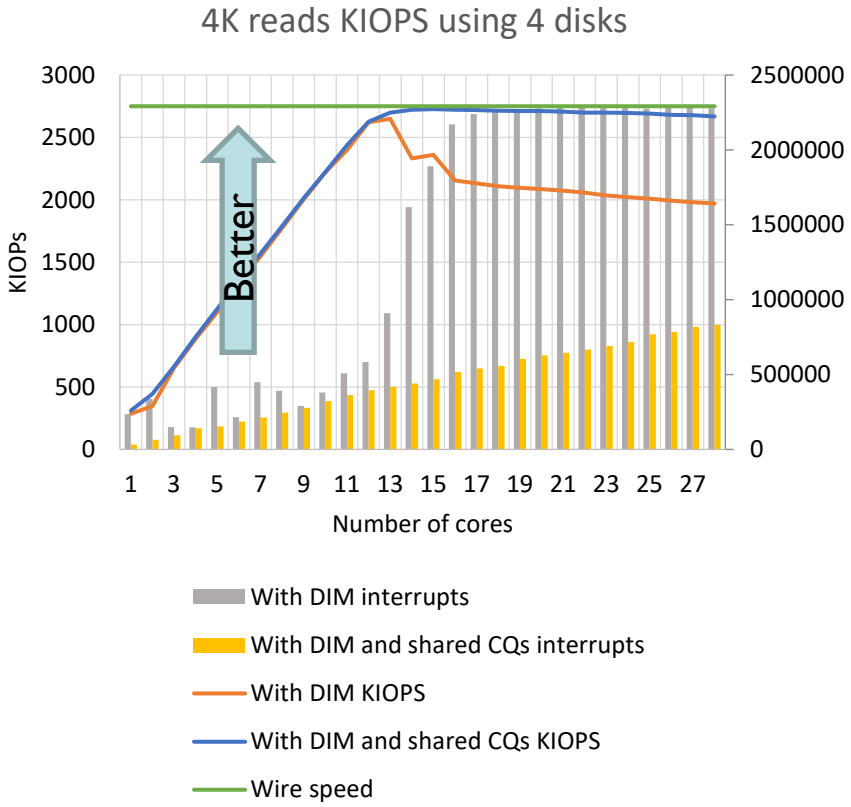
Interrupt moderation across applications

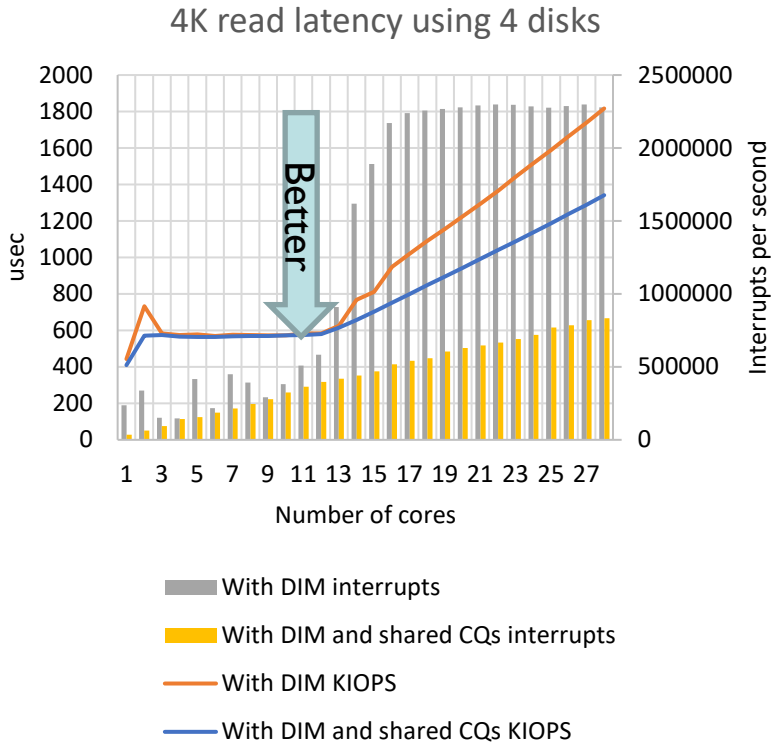
- As presented DIM is very effective in reducing interrupts per CQ
- What happens when there are many applications each with their own set of CQs?

Shared CQs

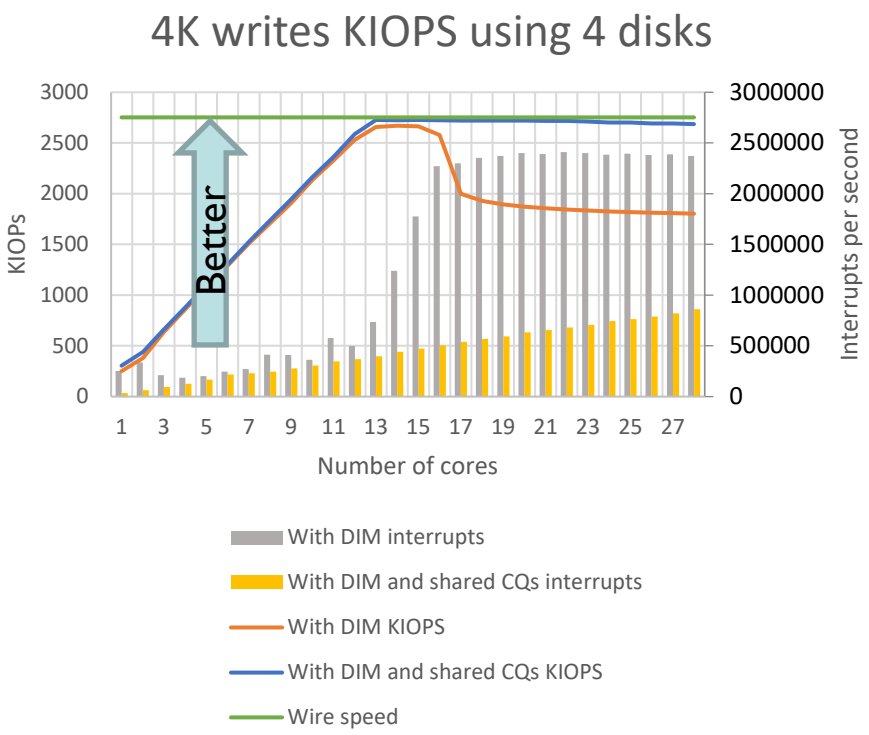
- In the Linux kernel driver CQ interrupt handling is provided as a service for all applications
- In this model each application opening its own set of CQs is an inefficient use of system resources
- With shared CQs we provide an API that provides full functionality while reducing system overhead

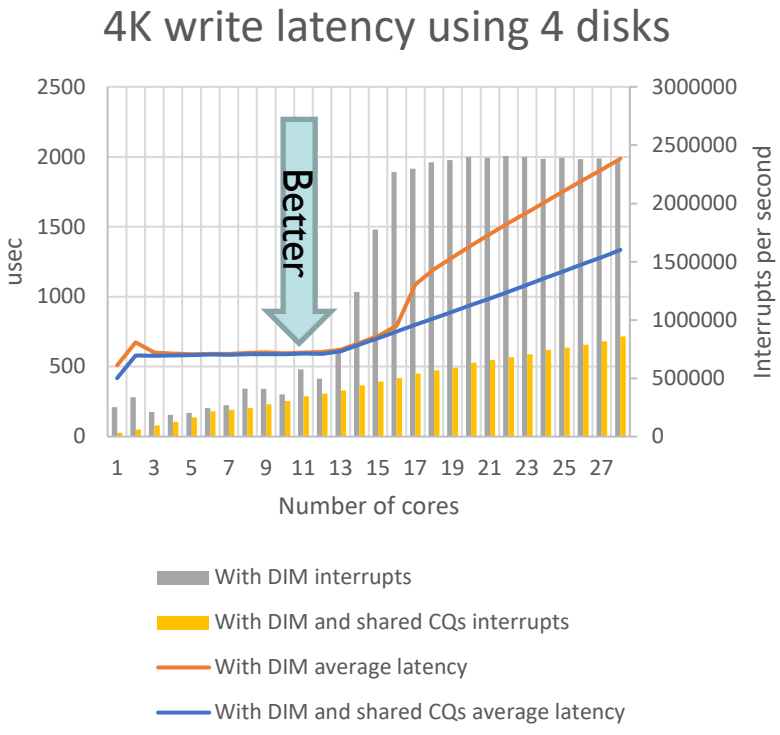
4K read 4 disk comparison





4K read 4 disk comparison





Concluding remarks

Conclusions

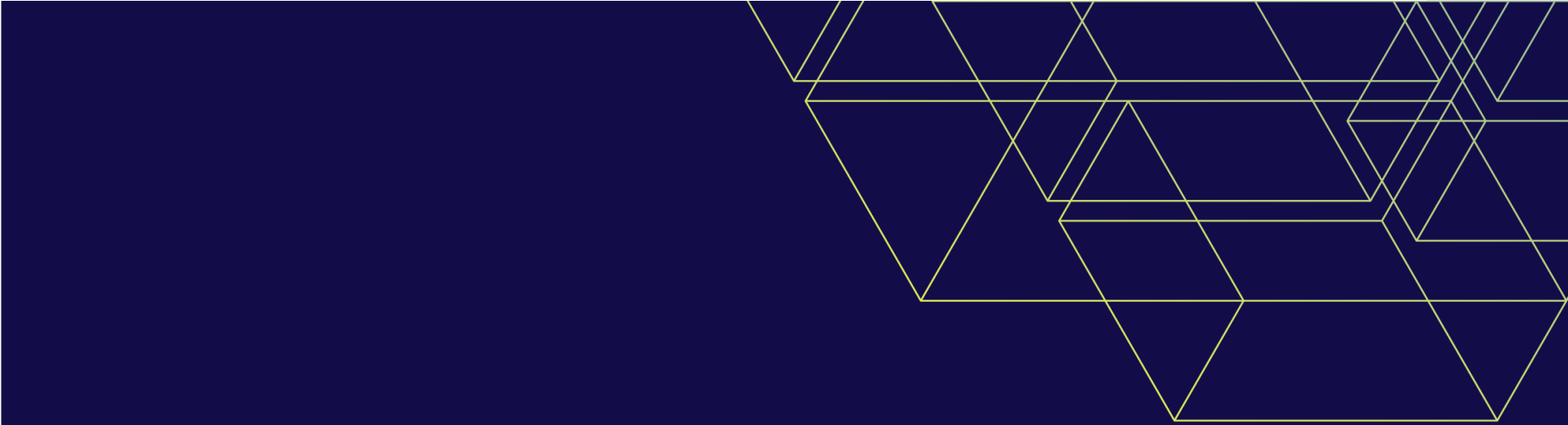
- Discovered hardware limitation on commodity servers for handling interrupts during RDMA
- Provided two features to upstream Linux to overcome this limitation

Key takeaways

- While optimizing our storage solutions, bottlenecks may be found throughout the hardware and software stack
- Solutions may need to be very flexible and perhaps cross abstraction layers

Contact information

- Yamin Friedman: yaminf@nvidia.com
- Rob Davis: rdavis@nvidia.com



Please take a moment
to rate this session.

Your feedback matters to us.