STORAGE INDUSTRY SUMMIT

Convergence of Storage and Memory Developing the Needed Ecosystem

JANUARY 20, 2016, SAN JOSE, CA

Dejan Vucinic

HGST San Jose Research Center

Latency in Context: Finding Room for NVMs in the Existing Software Ecosystem

SNIA SSSI | SOLID STATE STORAGE

Credits and acknowledgements

- HGST San Jose Research Center and HGST Research Japan
  - Martin Lueker-Boden
  - Chao Sun
  - Damien Le Moal
  - Dejan Vucinic
  - Zvonimir Bandic
  - Qingbo Wang
  - Md Kamruzzaman
  - Filip Blagojevic
  - Cyril Guyot
  - Robert Mateescu
  - Luiz Franca-Neto
  - Minghai Qin
  - Kiran Gunnam
  - Arup De

Outline

- Where should we attach emerging NVMs?
  - Low latency parallel, or high speed serial?
- PCI Express limitations and DC Express
  - Reducing overhead of a high bandwidth high latency serial bus for faster local access
- Kernel limitations: device driver or userspace access to storage
  - Need many CPU cores to surpass 1e6 IOPS through the kernel
- Reducing latency spikes
  - Emerging NVMs are a perfect match for real-time workloads, no GC
- Scale-out with RDMA
  - Bypassing the remote CPU for low latency access to networked storage

Where should we attach emerging NVMs?

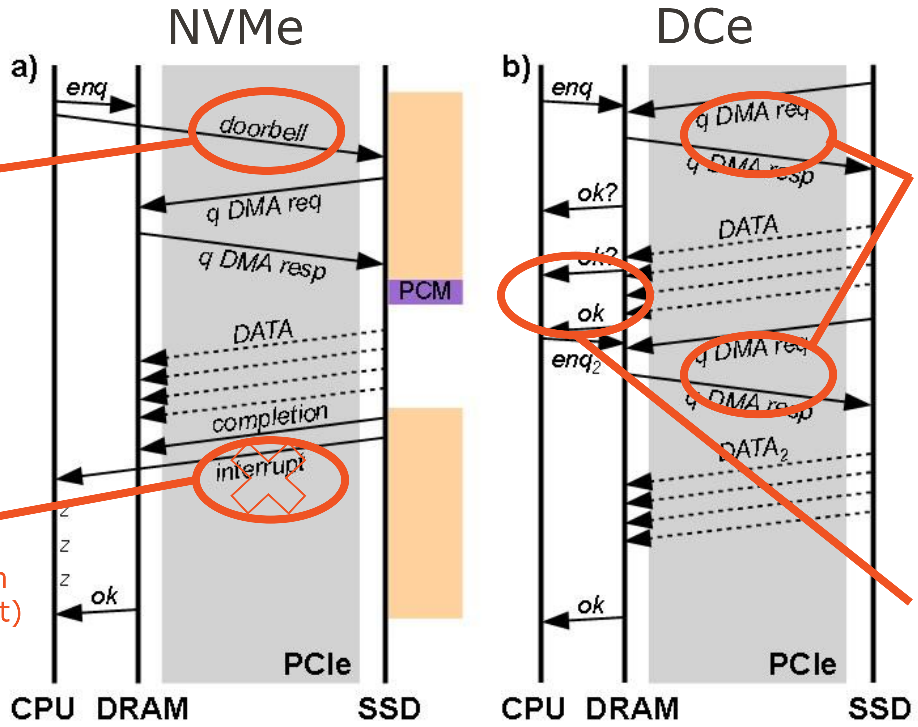
- Today’s NVMe SSDs using NAND flash media have low latencies, typically less than 100 µs
- Emerging NVM (eNVM) media will have even lower latency than NAND flash, some of them by 1000x!
- Low media latency puts pressure on device interfaces: we need to be able to extract this value to justify the higher cost

- The trouble with NV-DIMM
  - High bandwidth, low latency, power proportional, coherent interface seems like the perfect place to attach eNVM, but:
  - Existing “main” memory bus not well suited for asymmetric and stochastic media latency
  - eNVM media are 10x slower than DRAM, require:
    - wear leveling
    - error correction
    - data protection at rest (i.e. encryption)
  - deep changes to memory controller and cache hierarchy!
  - Power budget and room for chips very limited

HGST Research NVDIMM circa 2012

Minimize impact of high PCIe latency instead

- A leaner PCIe storage protocol: DC Express



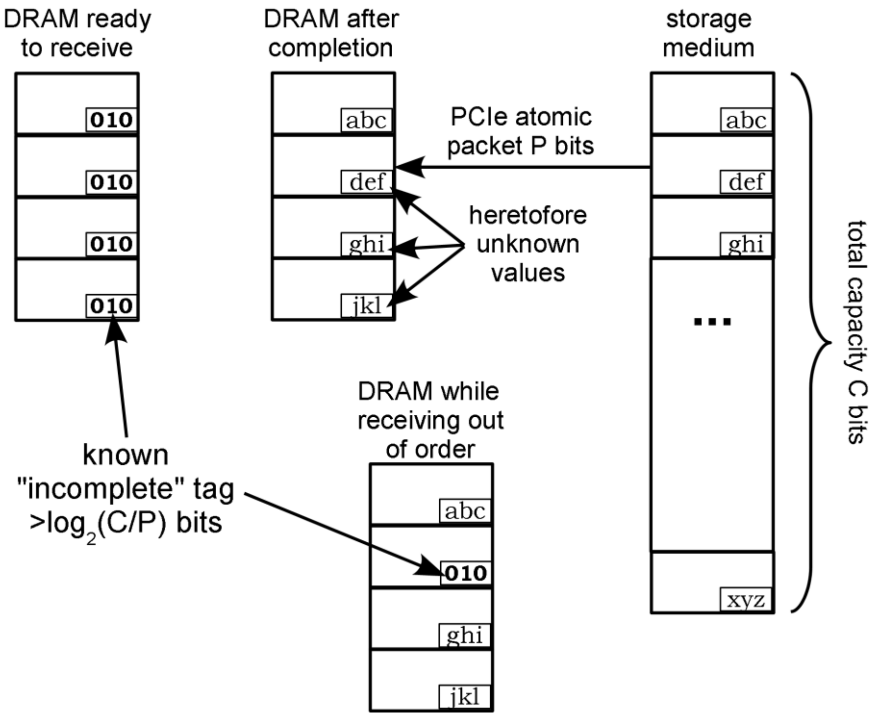
PIO register write for initiation

interrupt for completion (in polling NVMe driver replaced with polling from CPU on completion packet)

polling queue from device for initiation

polling data buffer from CPU for completion

How to know it’s done without an explicit signal



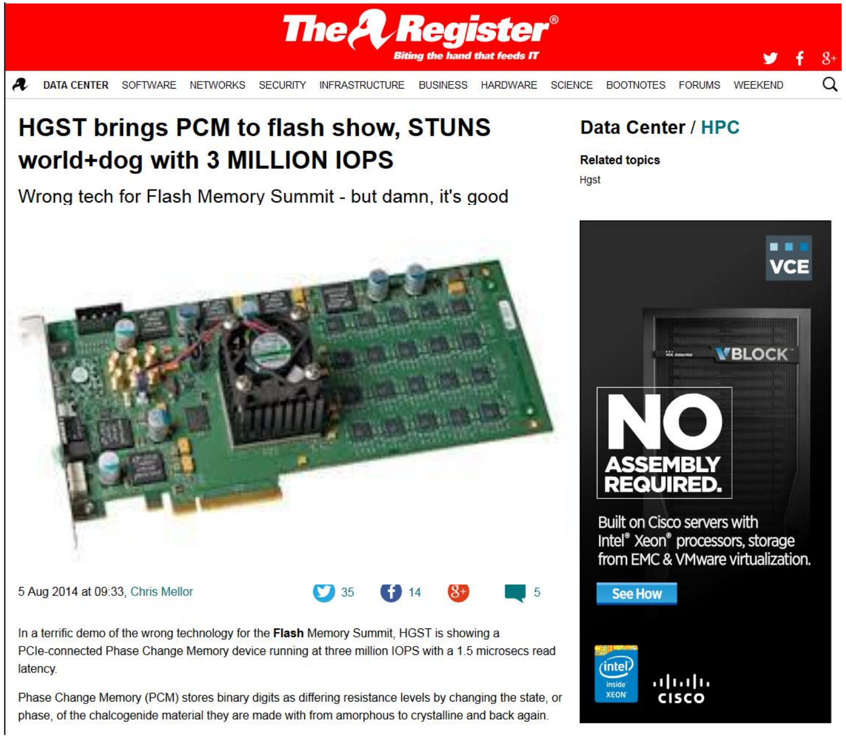
e.g. for a 128 GB SSD and 128 B TLP, 31 bits ensure existence of a unique tag

Various strategies for selecting tag:

- host at random w/ timeout
- device using algorithm

Blast from the past: DC express at gen2x4

- Last year at FMS:



The Register®
Biting the hand that feeds IT
DATA CENTER SOFTWARE NETWORKS SECURITY INFRASTRUCTURE BUSINESS HARDWARE SCIENCE BOOTNOTES FORUMS WEEKEND

HGST brings PCM to flash show, STUNS world+dog with 3 MILLION IOPS

Wrong tech for Flash Memory Summit - but damn, it's good

5 Aug 2014 at 09:33, Chris Mellor 35 14 5

In a terrific demo of the wrong technology for the **Flash** Memory Summit, HGST is showing a PCIe-connected Phase Change Memory device running at three million IOPS with a 1.5 microsecs read latency.

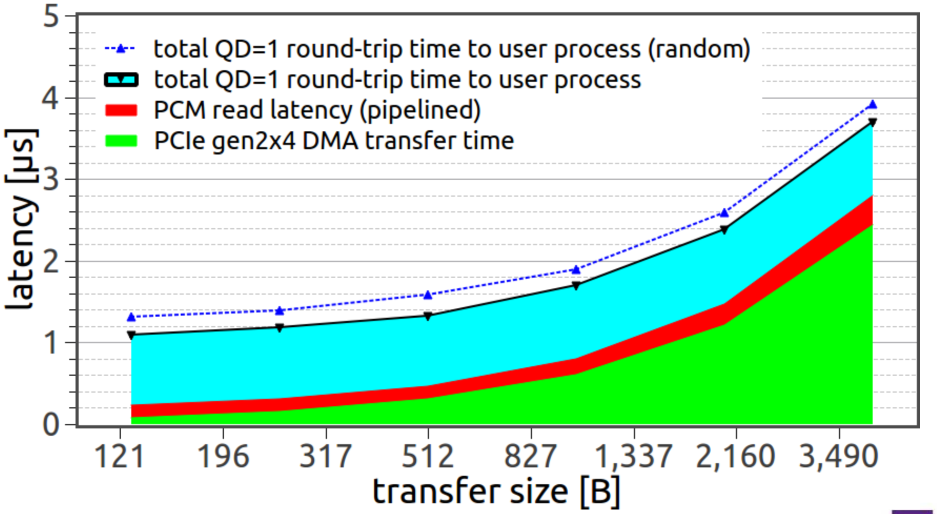
Phase Change Memory (PCM) stores binary digits as differing resistance levels by changing the state, or phase, of the chalcogenide material they are made with from amorphous to crystalline and back again.

Data Center / HPC

Related topics

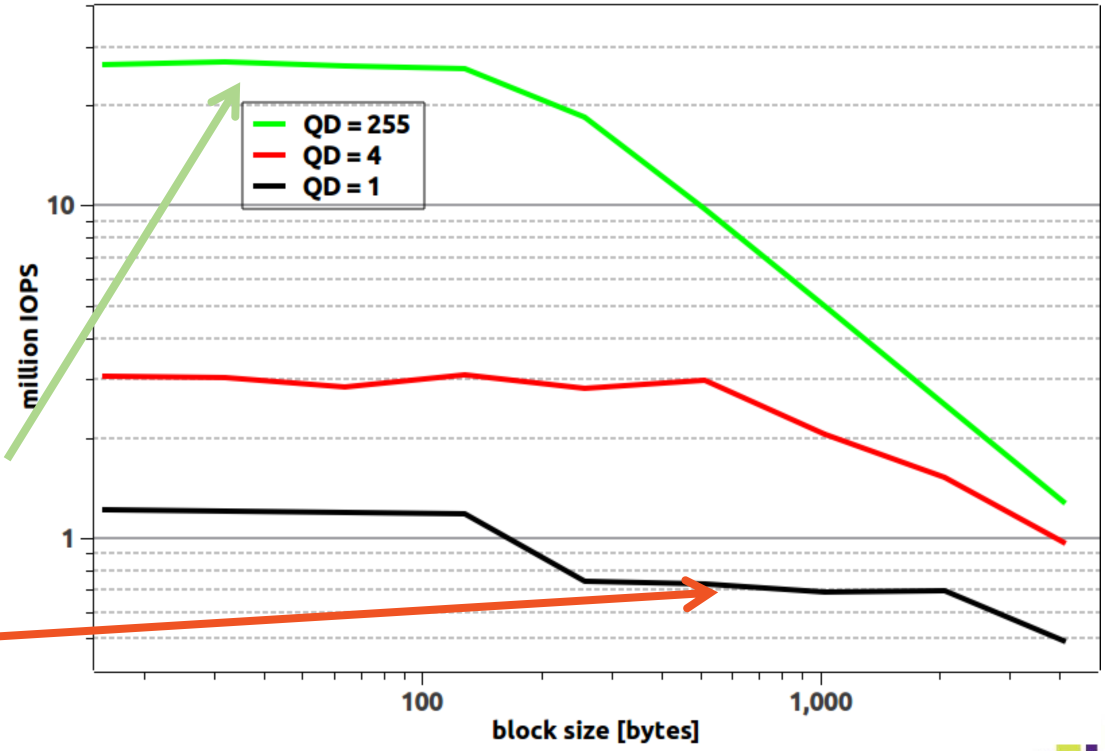
Hgst

- HGST innovation: DC Express
  - new, leaner PCIe storage protocol
  - minimizes number of packets per command



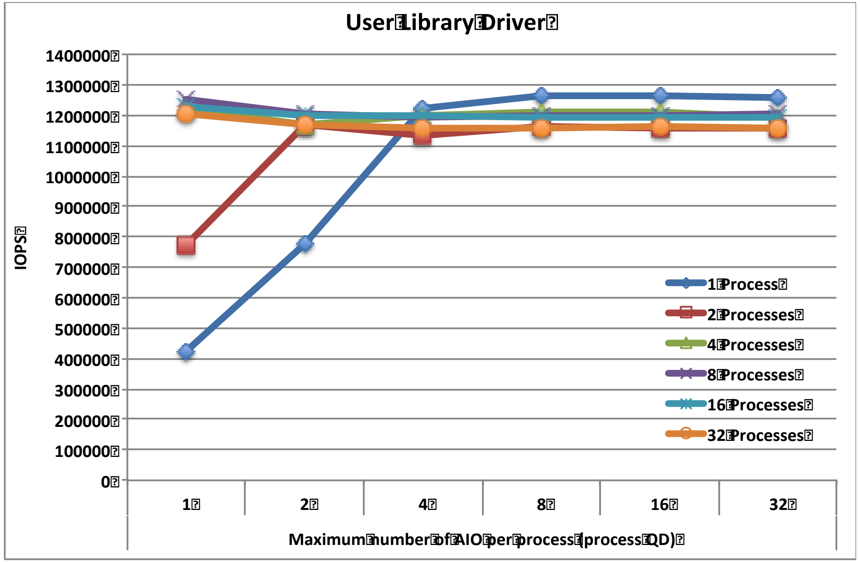
DC express bare protocol performance at gen3x8

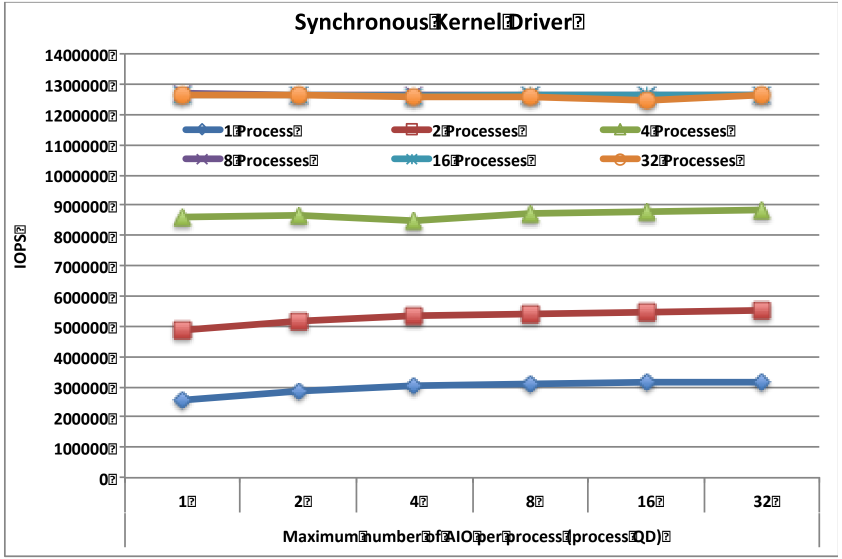
- Random read from SRAM, testing overhead of protocol alone
- Max bandwidth much higher, over 20 MIOPS on some systems for <128 B transfers
- At QD=1 no significant latency improvement over gen2x4, PCIe is limiting



But wait! Kernels aren’t ready!

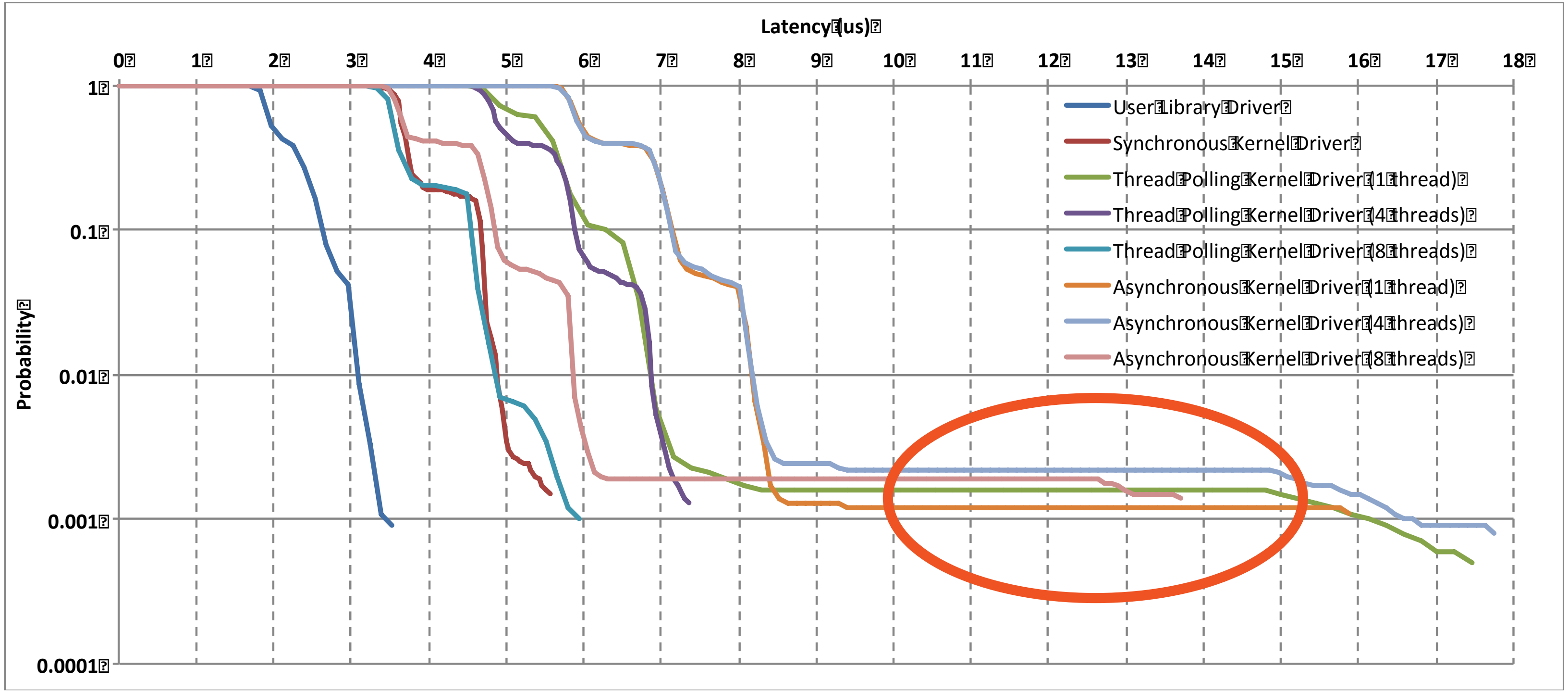
- 8 CPU cores required to exceed 1 MIOPS through Linux kernel: context switch overhead
- Userspace access achieves same performance at QD=4 using only 1 core





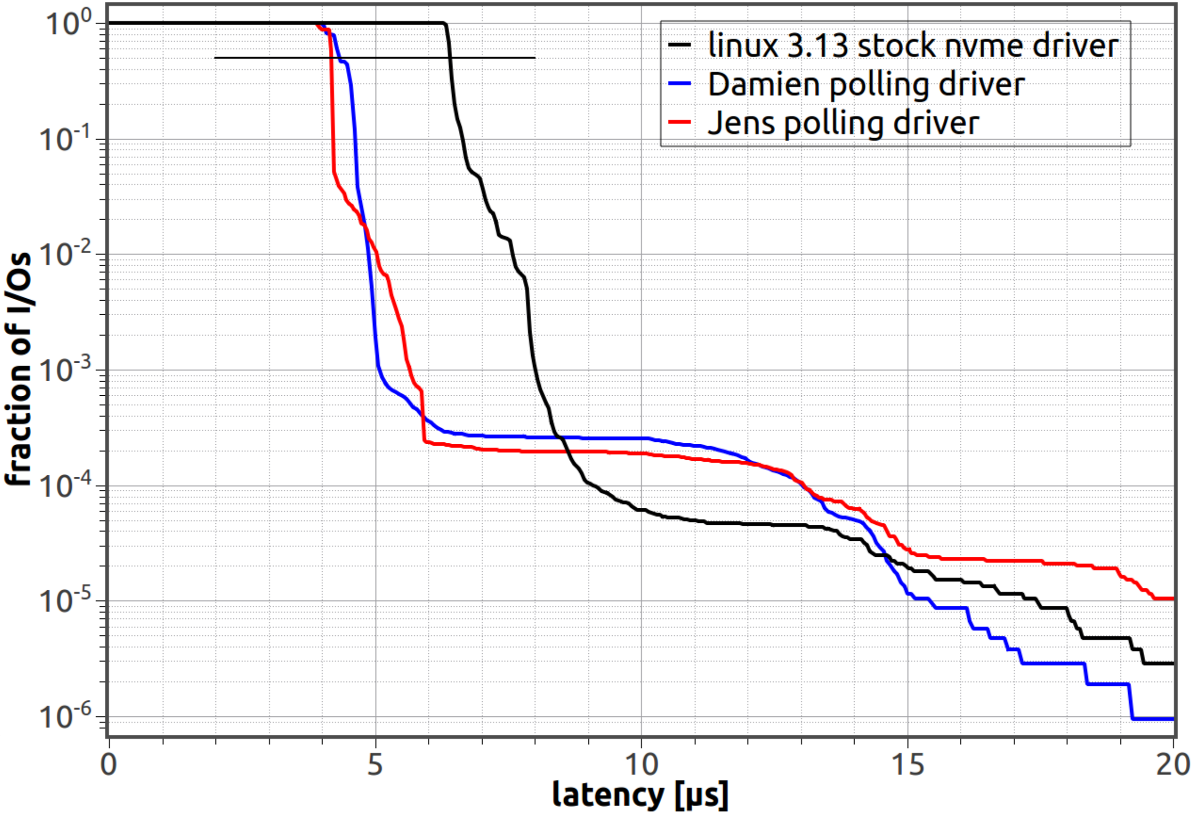
- Presented at NVMW’15 in San Diego by Damien Le Moal

Worst-case latencies under a full Linux OS have a long tail



- Presented at NVMW'15 in San Diego by Damien Le Moal

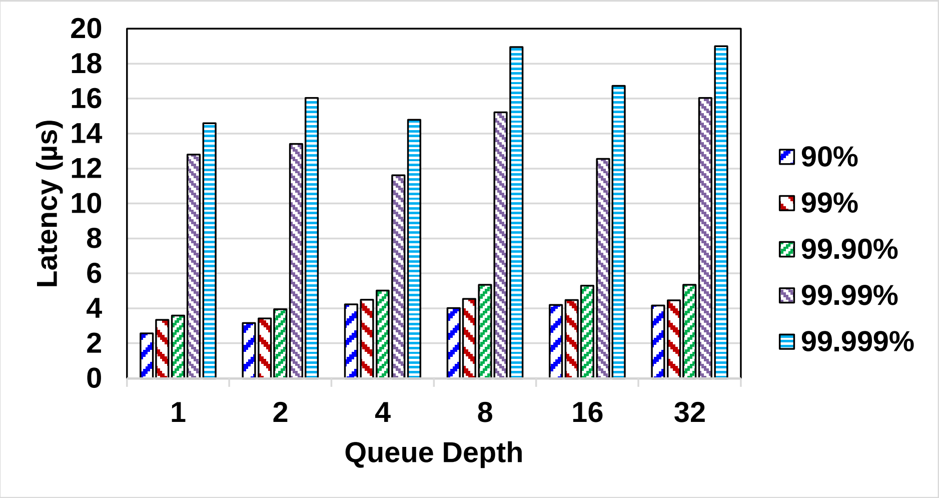
For comparison: polling and stock NVMe driver



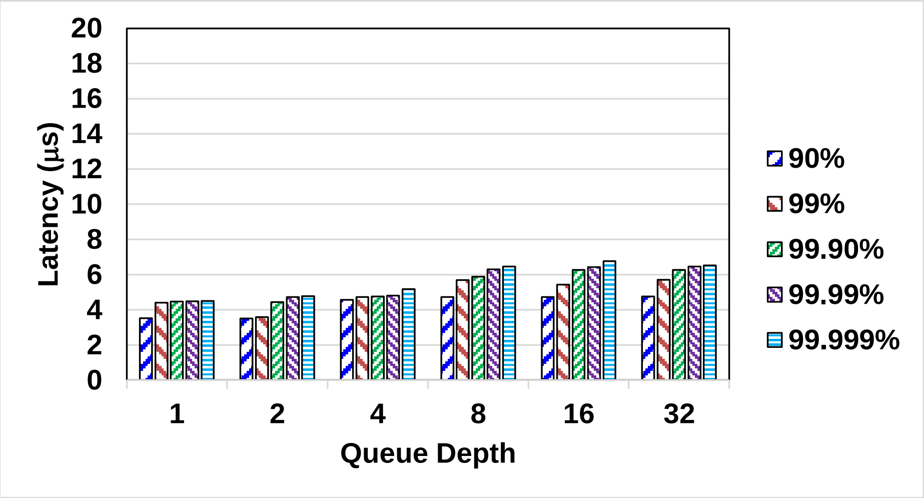
- Avoids interrupt context switch, > 2 µs latency savings

Latency tail can be tamed

Stock Linux 3.13



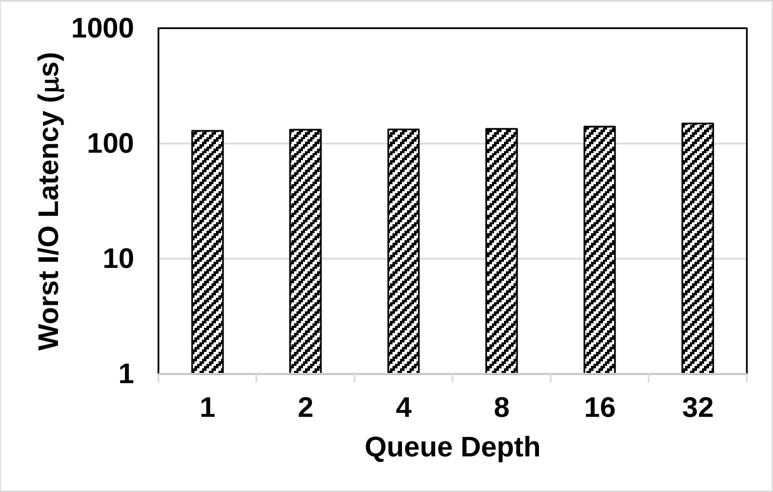
Treated Linux 4.0.5



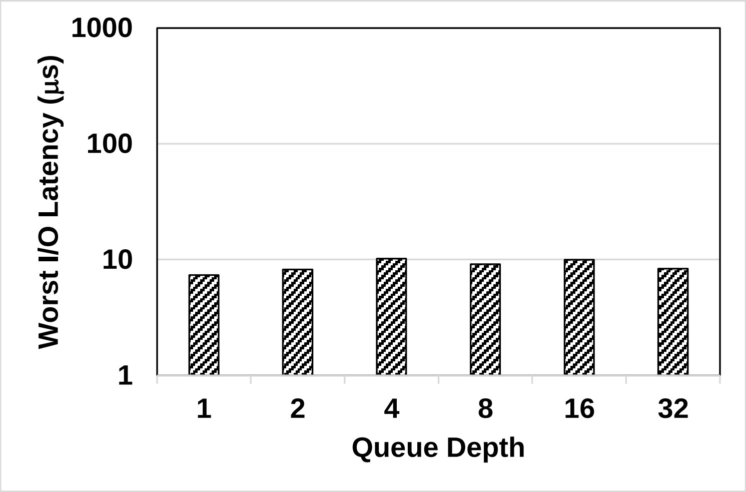
- 9 treatments including disabling C-States and TurboBoost
- Five-nines under 7 µs
- Presented at NVMW'16 in San Diego by Chao Sun

The worst observed latency of a PCM SSD on Linux

Stock Linux 3.13

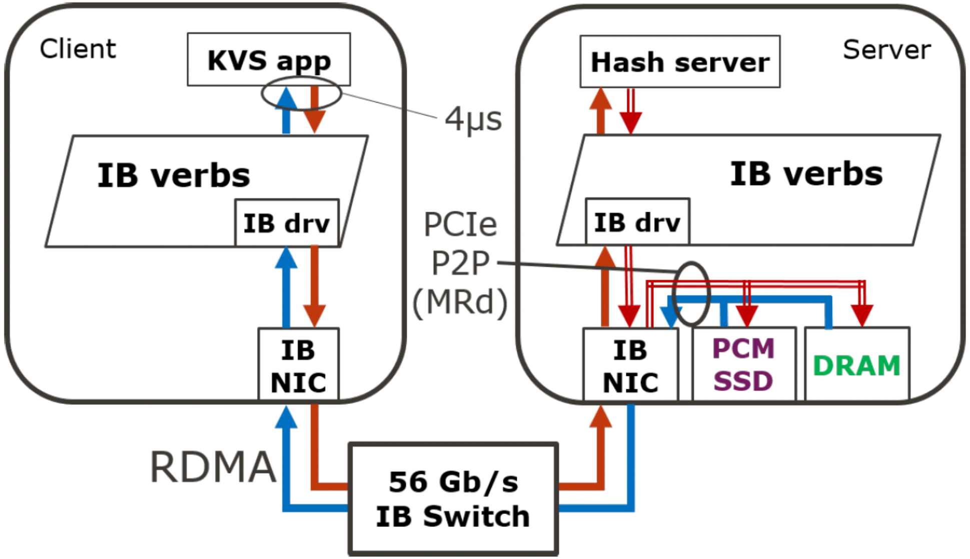


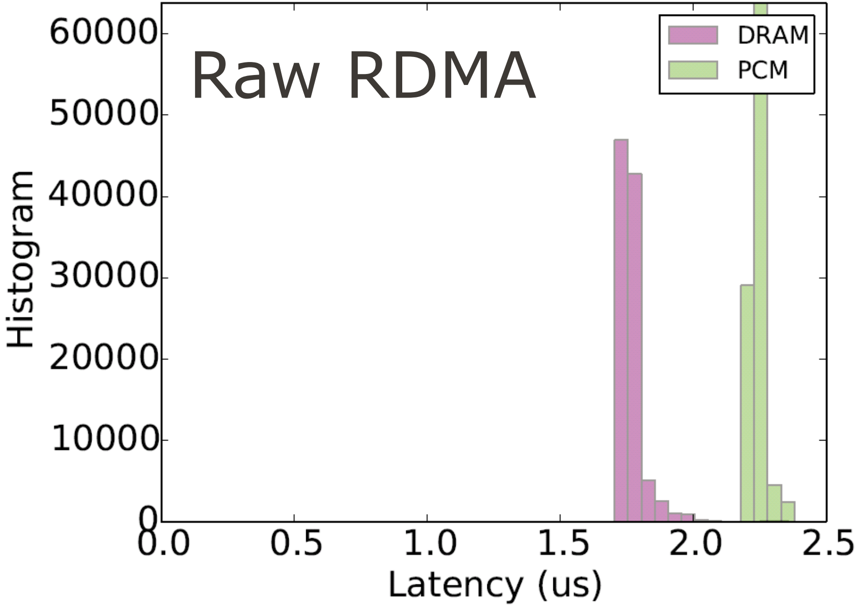
Treated Linux 4.0.5



- PCM is excellent storage technology for real-time environments
- Presented at NVMW’16 in San Diego by Chao Sun

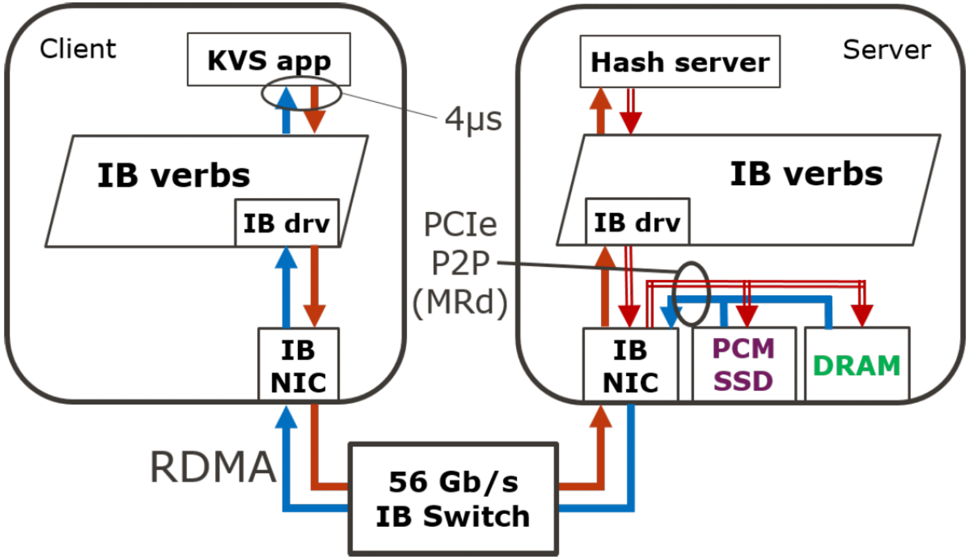
Remote PCM has performance similar to remote DRAM

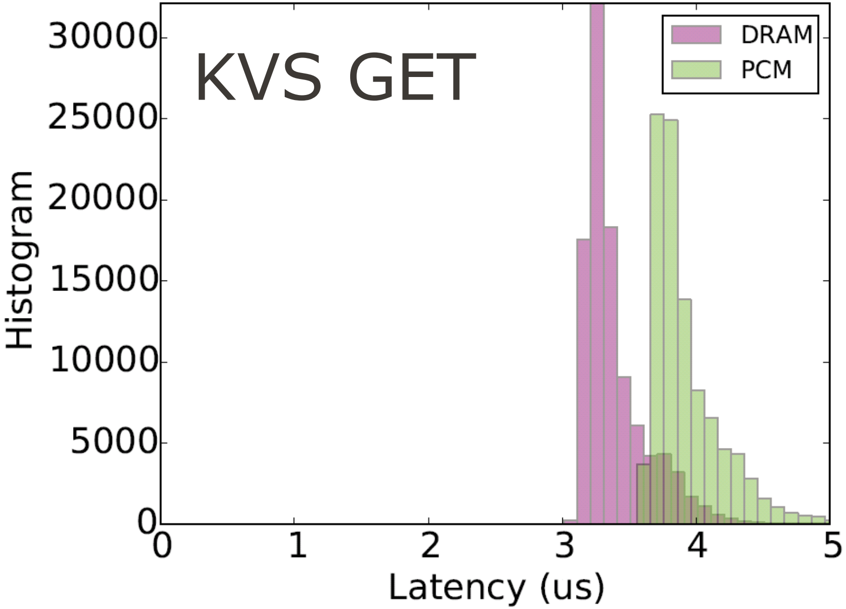




- Raw RDMA access to remote PCM via PCIe peer2peer is 26% slower than to DRAM

Remote PCM has performance similar to remote DRAM





- This gets effectively hidden in other system latency: KVS latency impact is only 14%
- Presented at NVMW’16 in San Diego by Martin Lueker-Boden

Conclusions

- Emerging NVM-based SSDs are going to have 10x improved latency and much less latency jitter compared to NAND based SSDs
- HGST Research working hard to enable seamless adoption of eNVM SSDs into the existing datacenter ecosystem
- Using peripheral interfaces is slightly sub-optimal from the latency standpoint but feasible today
  - Software stack not ready for faster anyway
  - In many applications end-to-end performance comparable to DRAM, at less $$$
- Future will see deep changes to cache hierarchy and memory controller architectures to extract even higher value from eNVMs

Helping the World Harness
the Power of Data with Smarter Storage Solutions