SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Software Challenges for the Changing IO Landscape

Daniel Waddington

IBM Research, Almaden

Disclaimer

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

IN ADDITION, THIS INFORMATION IS BASED ON IBM’S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE.

IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION.

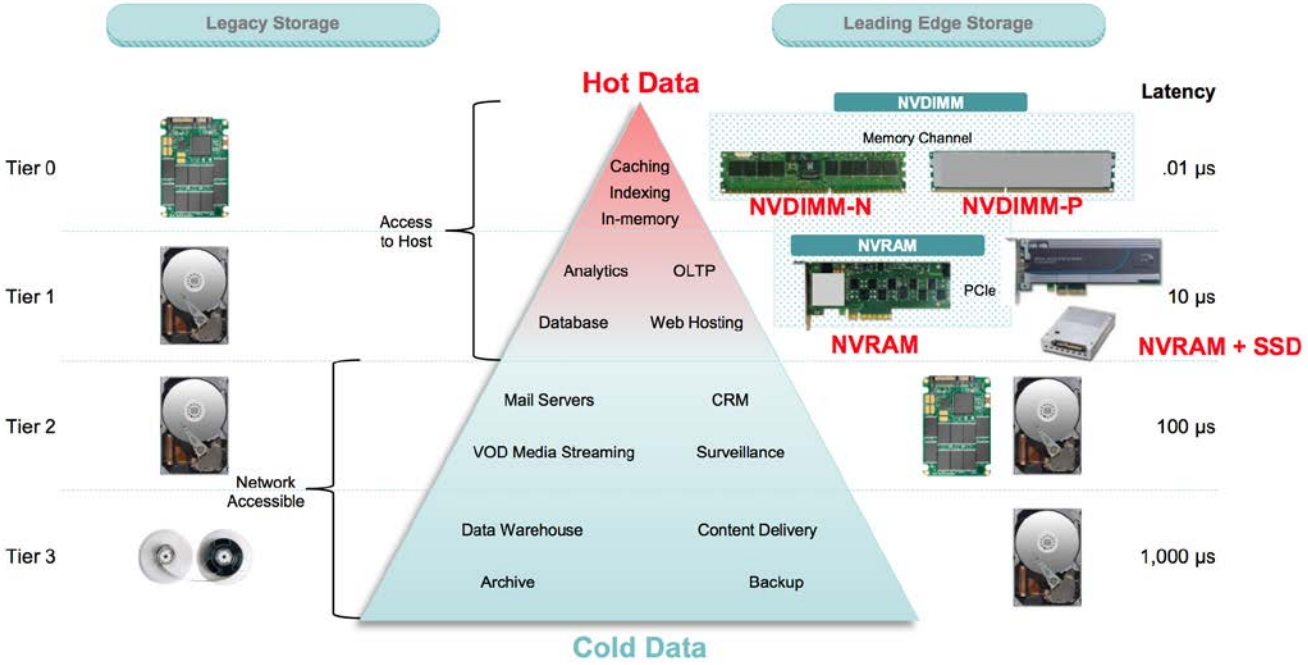
NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, OR SHALL HAVE THE EFFECT OF:

• CREATING ANY WARRANTY OR REPRESENTATION FROM IBM (OR ITS AFFILIATES OR ITS OR THEIR SUPPLIERS AND/OR LICENSORS); OR

• ALTERING THE TERMS AND CONDITIONS OF THE APPLICABLE LICENSE AGREEMENT GOVERNING THE USE OF IBM SOFTWARE.

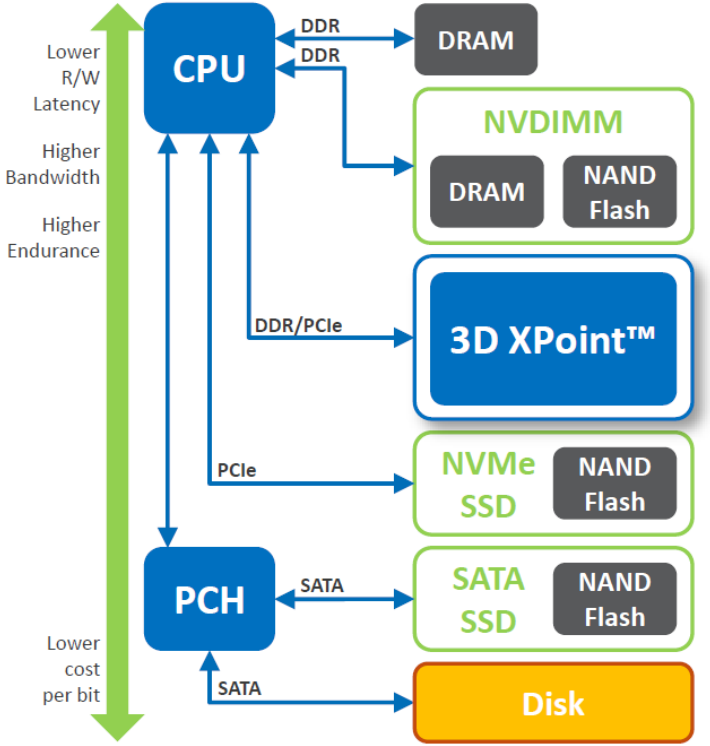
IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion. Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision. The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

A New Memory Hierarchy

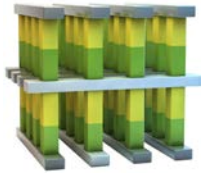


http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2015/20150812_S202A_Martinez.pdf

Intel/Micron 3DXpoint



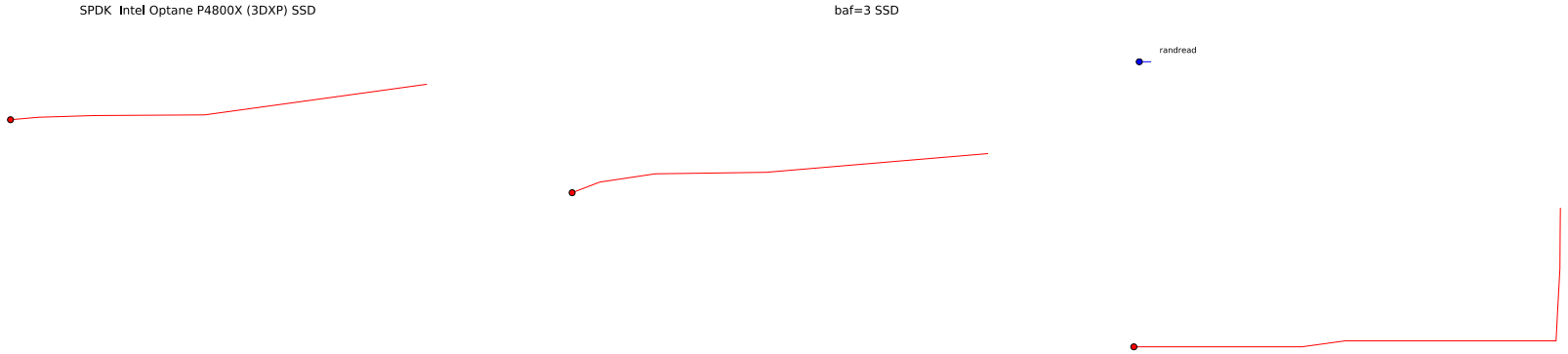
- 3DXpoint – new memory media technology (PCM like)
- High write-performance (symmetric with read)
- Order of magnitude more stable latency
  - less complex firmware compared to NAND
- **Optane** P4800X is the initial SSD form factor (adapter card) offering based on NVMe



- **Apache Pass** is an NV-DIMM version planned for 2018/2019 that will operate with Intel's new Purley architecture
  - hybrid memory controller

Intel Optane Performance

- Optane is an NVMe solution using 3DXpoint
- Add-in adapter card (PCIe)
- High write-performance (symmetric with read)
- Order of magnitude more stable latency (no delayed garbage collection)

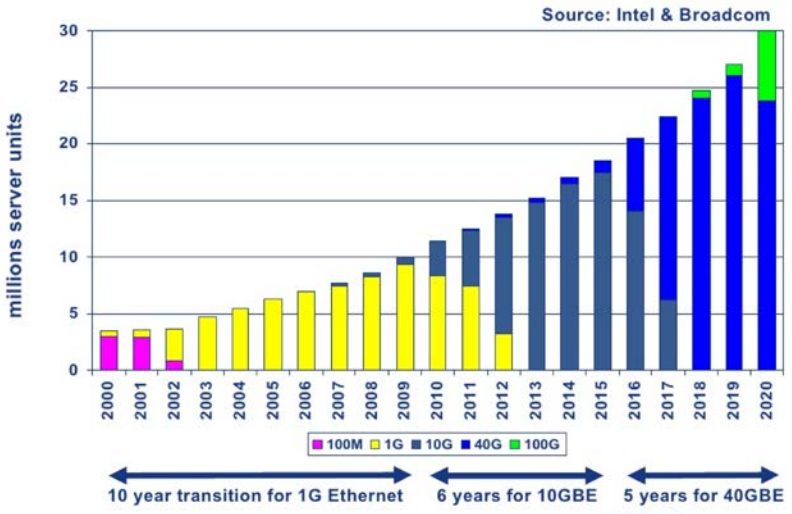


Other State-of-the-Art NVMe Storage

- Micron will market 3DXpoint under Quantx brand name
  - expected write latency is <20µs and read latency is <10µs
  - U.2 200GB to 1.6TB XPoint SSD achieves 900
  - 900GB to 1.6TB XPoint SSD in a half height, half length PCIe 3.0 x8 card achieves 1.8 million IOPS with 25 DWPD over 5 years
- Samsung Z-SSD (based on Z-NAND)
  - shares the fundamental structure of V-NAND and has a unique circuit design and controller that can maximize performance, with four times faster latency and 1.6 times better sequential reading than the Samsung PM963 NVMe** SSD
  - speculation that it is based on variation on SLC NAND
  - expect sequential read/write up to 3.2GB/s
  - random read 750 KIOPS, random write 160 KIOPS
  - ~12-20 usec rand read, 16usec rand write, 30 DWPD
- Many other NVMe companies/solutions

Network Trends

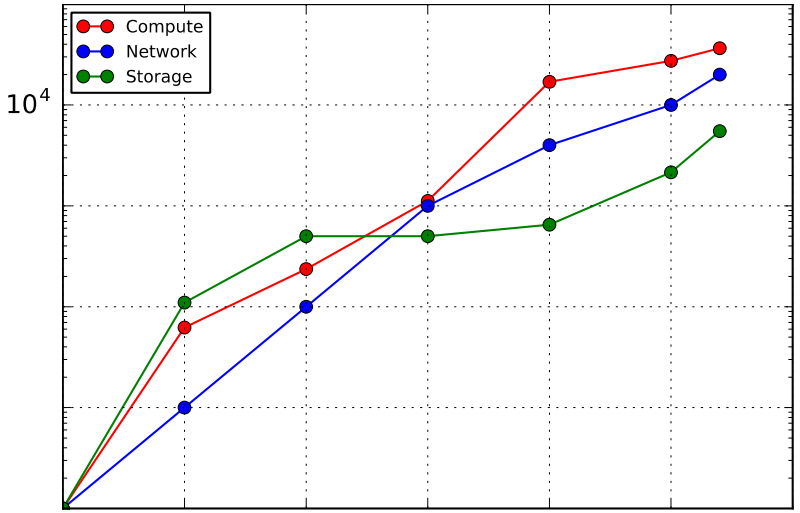
- 40GbE and100GbE data center connectivity is the new norm
- 200GbE available today
- > 100M PPS small packet (10 ns/packet)
- < 400 ns switch cut-through latency
- 10ms / 500 miles
- Long-haul / Internet latency remains dominant



Relative Growth Trends

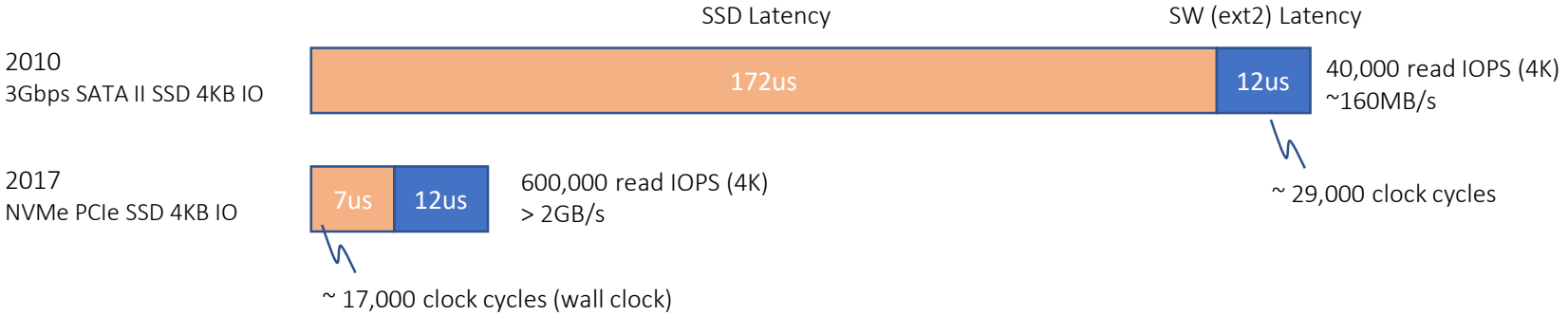
- CPU performance is flattening out
- Storage is on an “up-tick”

- CPU is the new scarce resource
  - multi-core/parallelism
    is the primary growth dimension



Key Observation

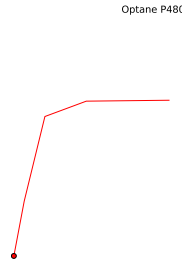
- Storage is no longer the primary system bottleneck

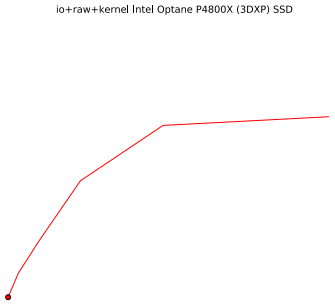


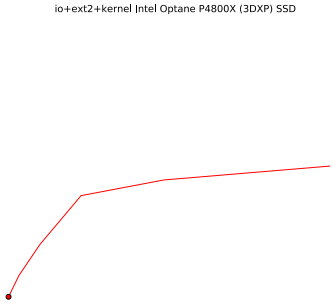
- Persistent memory (NVDIMM-N) is imminent
  - 64B data granularity
  - near DRAM-latency 0.1µsec latency (~240 clock cycles)
  - direct CPU micro-instruction addressable (cache line granularity)

Software Overhead

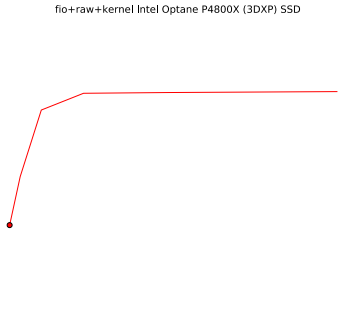
- Raw performance
- Kernel driver (raw block)
- Kernel driver (ext2)

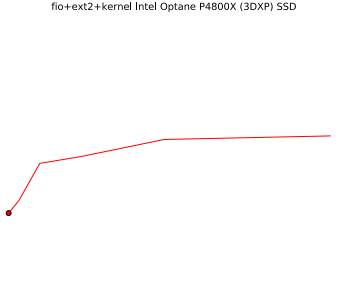






QD=1



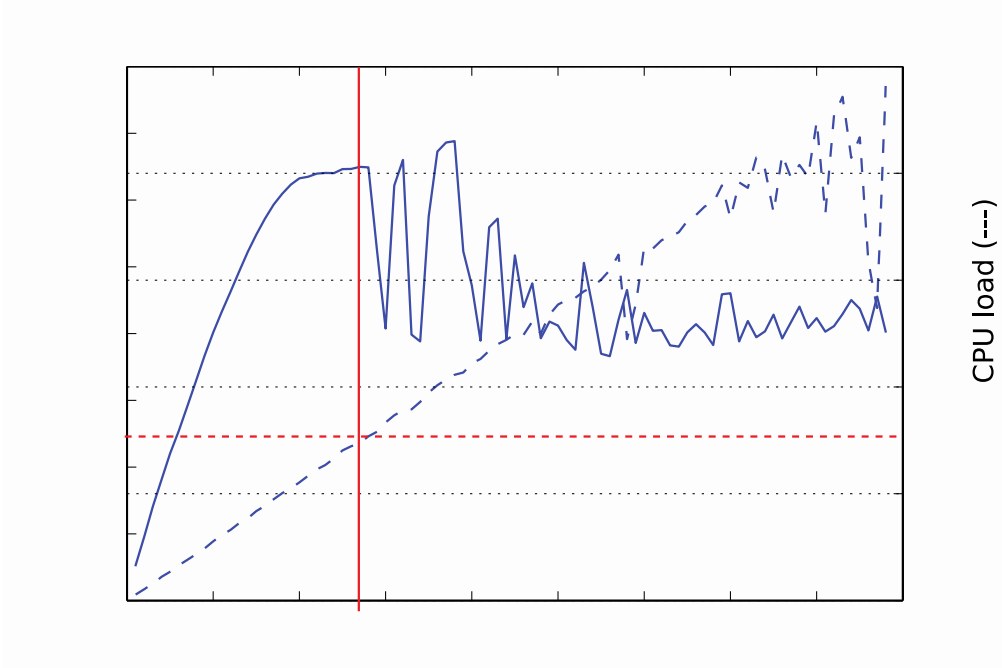


QD=32

Software Overhead

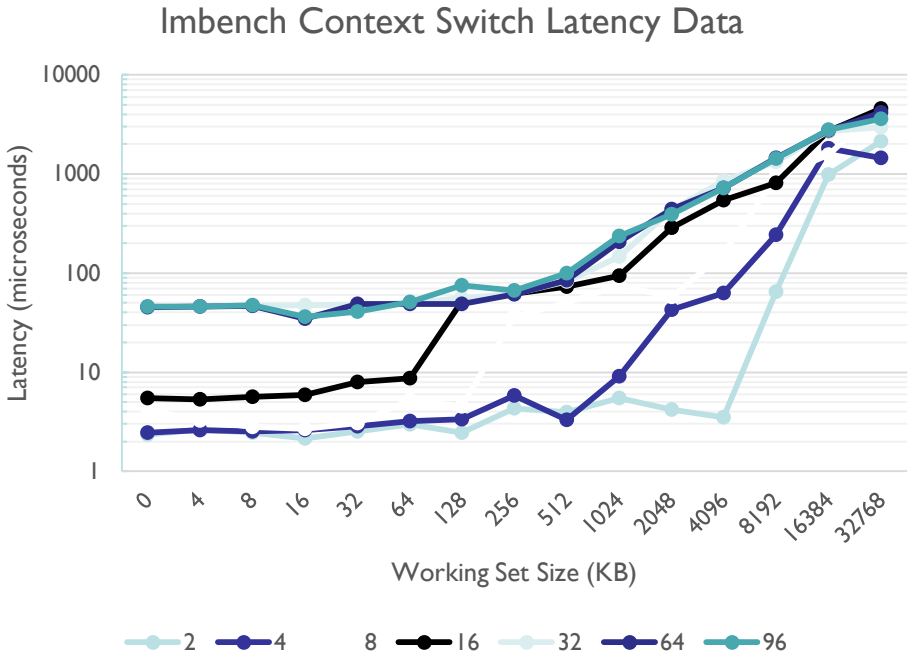
- Linux ext4 file system scaling
  - Intel E5-2699 x2 server
  - x24 NVMe Samsung 172Xa SSD
  - ‘fio’ micro-benchmark
  - no file sharing

- Maximum throughput achieved is 3.2M IOPS (12.21 GB/s) which is realized at a load of ~26 threads (one per device) and 30% total CPU capacity



Context Switching Overhead

- Context switches typically take between 7,000 and 80,000 clock cycles depending on working set size and cache state



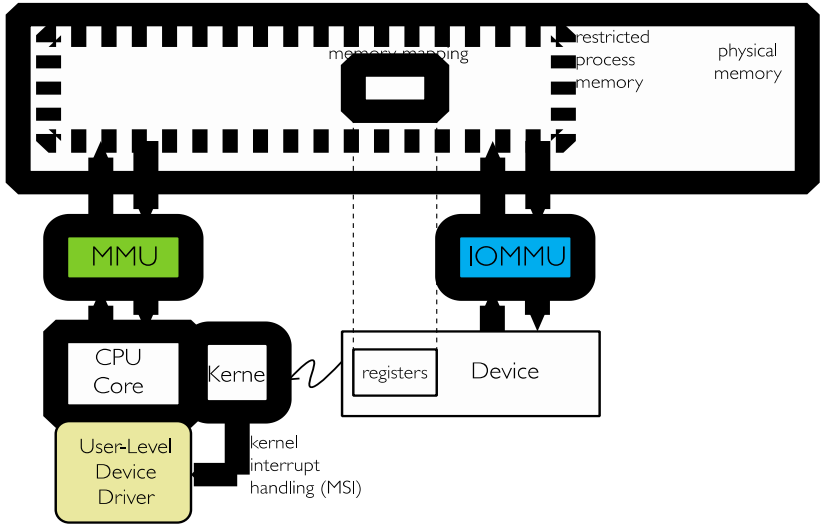
Intel E5-2650 @ 2.2GHz
2x 24-core processors

32K L1 cache size
256K L2 cache
30MB L3 cache

1000 iterations on lat_ctx (lmbench)

Rethinking IO

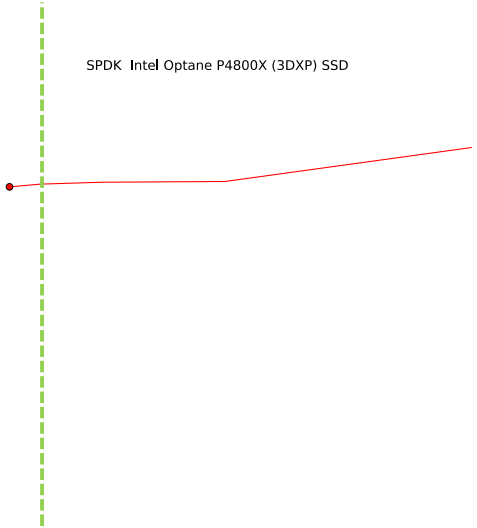
1. User-level subsystems
   - "Lift" conventional IO stack into userspace
     - block device, caching, file system, kv-store
   - Protect with IOMMU – key enabler
   - Interrupt coalescing / atomic masking devices
   - Run at Ring 3 privilege and non-root (ideally)
   - LGPL software licensing (for Linux)



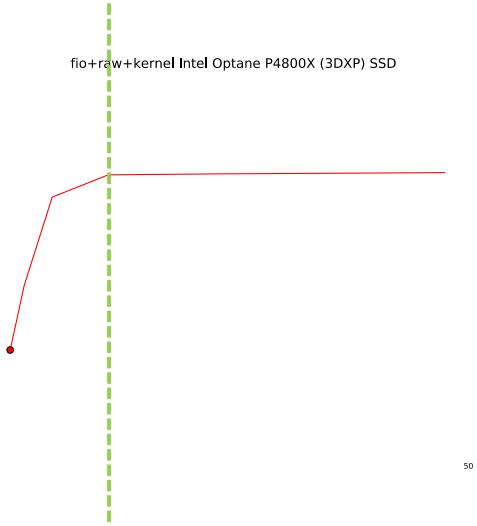
2. Tailor IO to application domains
   - Composition
   - Coordination through shared memory and lock-free data structures

Performance Potential

- Less latency (worst case latency >40% reduction, mean latency >50% reduction)
- Less CPU resources are spent on IO
  - beyond 3M IOPS per core



SPDK 4K QD=32 LBAF=0

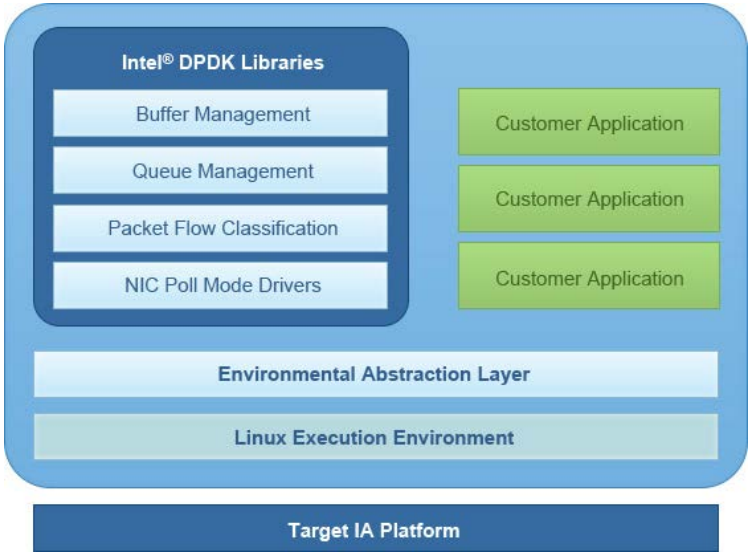


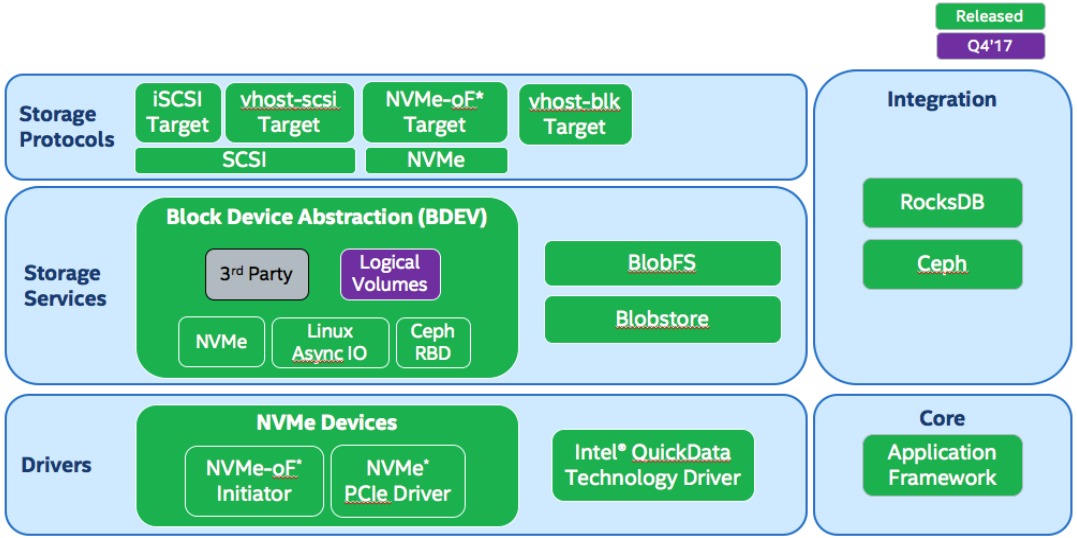
Kernel Raw Block Device 4K QD=32 LBAF=0

Emerging User-level Ecosystem

- Data Plane Development Kit (DPDK) - http://dpdk.org
- Storage Performance Development Kit (SPDK) - http://spdk.io
- Fast Data Project (FD.io) - http://fd.io
- Seastar - http://seastar-project.org
- UNVMe - https://github.com/MicronSSD/unvme
- IBM Crail - https://github.com/zrlio/crail
- NVMeDirect - https://github.com/nvmedirect
- IX-Dataplane Operating System (Stanford) - https://github.com/ix-project/ix
- IBM Research Comanche – TBA
- …

DPDK/SPDK

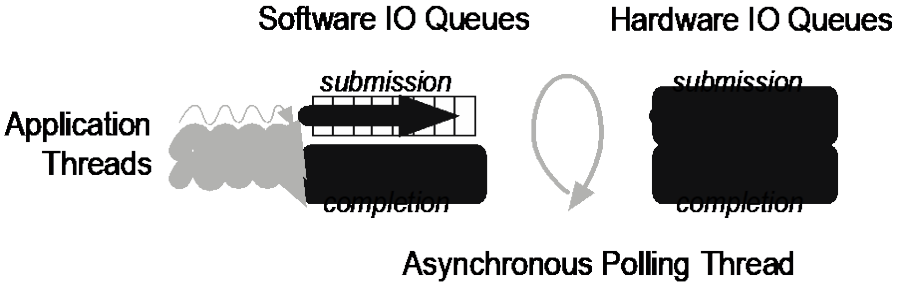




Cost of User-level IO

- Interrupts are not delivered to userspace
  - signaled from kernel

- Polling driven
  - lock-free queues
  - asynchronous task management
  - sleeping queues can be used to reduce busy-waiting but incur a wake-up cost



Key Challenges

- Integrating with existing applications (e.g., POSIX based)
  - I/O interceptor method
- Protected sharing of storage devices
  - software multiplexing
  - NVMe SR-IOV
- Memory flush optimization
  - explicit
  - checksums
  - page table attributes
- Memory paging/swapping
  - user-level PF handling (e.g., via POSIX mprotect/mmap, Dune)

Consumption-as-Memory

- Block, object, file, database are all storage paradigms
- NV-DIMM and fast NVMe opens up a consumption-as-memory paradigm
- Consumption-as-memory means taking our existing DRAM-base (synchronous) programming model and extending it to persistent memory
  - avoids the need to translate or serialize between memory and storage
- Programming language support is either explicit and implicit (orthogonal)

  Types and Persistent in Database Programming Languages – ACM Surveys '87 (Atkinson [Glasgow], Buneman [U.Penn])
  Orthogonally Persistent Object Systems (Napier88) – VLDB'95 (Atkinson [Glasgow], Morrison [St. Andrews])
  An Orthogonally Persistent Java – SIGMOD'96 (Atkinson et al. Glasgow)
- Typically requires heterogeneous heap management

  NV-Heaps – ASPLOS'11 (Coburn et al., UCSD)
  Mnemosyne: Lightweight Persistent Memory – ASPLOS'11 (Volos, Tack, Swift, U. Wisconsin-Madison)
  memkind: User Extensible Heap Manager - http://memkind.github.io/memkind/ (Cantalupo et al.)
  memif: Programming Heterogeneous Memory Asynchronously – ASPLOS'16 (Lin & Lu)
  pVM – Eurosys'16 (Kannan, Gavrilovska, Schwan, Georgia Tech)

Key Challenges

- Early technology
  - cost and system hardware integration (NV-DIMM)
  - endurance
- New memory semantics
  - heterogeneous heaps (pmem.io)
  - different cost, performance and properties for each heap
  - avoiding heap pollution
- Integrating with existing languages
  - extending type systems (many types)
  - new language design – compilers/adoption
- Adding “richer” services without interception point (e.g., durability, encryption)

Conclusions

- Storage is on the up – CPU is the new scarce resource
  - thousands of cycles per IOP
- Kernel bypass helps: 1.) eliminate system calls and 2.) ease development of lightweight tailored IO stacks
  - user-level device drivers and IO services
  - programming language flexibility
  - Microkernel-esque architecture (we can learn from this community)
- Challenges
  - legacy
  - tools and languages for persistent memory

Recommended Reading: ACM Communications

- Attack of the Killer Microseconds, Luiz Barroso, Mike Marty, David Patterson, Parthasarathy Ranganathan
  - http://cacm.acm.org/magazines/2017/4/215032-attack-of-the-killer-microseconds/fulltext

The computer systems we use today make it easy for programmers to mitigate event latencies in the nanosecond and millisecond time scales (such as DRAM accesses at tens or hundreds of nanoseconds and disk I/Os at a few milliseconds) but significantly lack support for microsecond (µs)-scale events. This oversight is quickly becoming a serious problem for programming warehouse-scale computers, where efficient handling of microsecond-scale events is becoming paramount for a new breed of low-latency I/O devices ranging from datacenter networking to emerging memories