THE QUEST FOR LOW LATENCY STORAGE

Rick Coulson
Senior Fellow, Intel NVM Solutions Group
Outline

• The history of storage latency and where we stand today
• The promise of Storage Class Memory (SCM) and 3D Xpoint™ Memory
• Extensive compute platform changes driven by the quest for lower storage latency with SCM / 3D Xpoint™ Memory
  – As traditional storage
  – As persistent memory
• Innovation opportunities abound
1956: IBM RAMAC 350

5 MBytes

$57,000

$15200/Mbyte

~1.5 Random IOPs*

600ms latency

* IOPs depends on the workload and is a range

* Other names and brands may be claimed as the property of others.
1980: IBM 3350

300 Mbytes
$60,000
$200/MByte
30 random IOPs

33ms latency

*Other names and brands may be claimed as the property of others.
1983: IBM3380

2.52 GBytes

$82,000

$36/MByte

~160 IOPS total

25ms latency

Two hard disk assemblies each with two independent actuators each accessing 630 MB gigabyte within one chassis

*Other names and brands may be claimed as the property of others.
2007: 15K RPM HDD

15K RPM HDD

About 200 random IOPs*

~5ms latency

IOPS scaling problem was addressed through HDDs in parallel in Enterprise

* IOPs depends on the workload and is a range
2016: 10K RPM HDD

1.8TB

~150 IOPs

6.6ms latency
2016 NVMe NAND SSD

2TB
500,000+ IOPs
~60 usec latency
The Continuing Need For Lower Latency

RAMAC 305
100 Hz best case “clock”

RAMAC 350
600ms

~100x
Access time reduction

10K RPM
~6ms access

~10,000x
Access time reduction

NAND SSD
~60us access

~40,000,000x
Clock speed increase

Core™ i7
~4 Ghz clock

59 Years

*Other names and brands may be claimed as the property of others.
Lower Storage Latency Requires Media and Platform Improvements

- First HDD
- Modern HDD
- NAND SSD
- NVMe NAND SSD

Media Bottlenecks
Platform HW / SW bottlenecks

Drive for Lower Latency

- Persistent Memory
- 3D XPoint™ memory (SCM)
- Ultra fast SSD
Addressing Media Latency: Next Gen NVM / SCM

Scalable Resistive Memory Element

Resistive RAM NVM Options

<table>
<thead>
<tr>
<th>Family</th>
<th>Defining Switching Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase Change</td>
<td>Energy (heat) converts material between crystalline (conductive) and amorphous (resistive) phases</td>
</tr>
<tr>
<td>Magnetic Tunnel Junction (MTJ)</td>
<td>Switching of magnetic resistive layer by spin-polarized electrons</td>
</tr>
<tr>
<td>Electrochemical Cells (ECM)</td>
<td>Formation / dissolution of “nano-bridge” by electrochemistry</td>
</tr>
<tr>
<td>Binary Oxide Filament Cells</td>
<td>Reversible filament formation by Oxidation-Reduction</td>
</tr>
<tr>
<td>Interfacial Switching</td>
<td>Oxygen vacancy drift diffusion induced barrier modulation</td>
</tr>
</tbody>
</table>

Scalable, with potential for near DRAM access times
3D XPoint™ Technology

Crosspoint Structure
Selectors allow dense packing and individual access to bits

Scalable
Memory layers can be stacked in a 3D manner

Breakthrough Material Advances
Compatible switch and memory cell materials

High Performance
Cell and array architecture that can switch states 1000x faster than NAND
A NEW CLASS OF NON-VOLATILE MEMORY

1000X FASTER THAN NAND

1000X ENDURANCE OF NAND

10X DENSER THAN DRAM

*Results have been estimated or simulated using internal analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance
3D XPoint™ Technology Instantiation

INTEL® OPTANE™ SSDS

DIMMS BASED ON 3D XPoint™
3D Xpoint™ Technology Video

Please excuse the marketing
3D Xpoint™ Technology Video
Demonstration of 3D Xpoint™ SSD Prototype

NAND TECHNOLOGY
Intel® SSD DC P3700 Series

7.32X IOPS PERFORMANCE

3D XPOINT™ TECHNOLOGY
Early SSD Prototype

IOPS

IOPS

10,700

78,300

READ QUEUE DEPTH

1
Need to Address System Architecture To Go Lower

Latency (usecs)

NAND MLC NVMe SSD (4kB read)
3D Xpoint NVMe SSD (4kB read)
DIMM Memory (64B read)
Block Storage Platform Changes
Addressing Interface Efficiency With NVMe / PCI

SSD NAND technology offers ~500X reduction in media latency over HDD

NVMe™ eliminates 20 µs of controller latency

3D XPoint™ SSD delivers < 10 µs latency

~7X

3D XPoint™ Persistent Memory

Drive Latency
Controller Latency (i.e. SAS HBA)
Software Latency

Source: Storage Technologies Group, Intel
NVMe Delivers Superior Latency
Platform HW/SW Average Latency Excluding Media 4KB

PCIe NVMe approaches theoretical max of 800K IOPS at 18us

Source: Storage Technologies Group, Intel
NVMe/PCIe Provides More Bandwidth

PCle/NVMe provides more than 10X the Bandwidth of SATA. Even More with Gen 4

Source: Storage Technologies Group, Intel
Storage SW Stack Optimizations

Much of the storage stack designed with HDDs latencies in mind

• No point in optimizing until now
• Example: Paging algorithms with seek optimization and grouping
Synchronous Completion for Queue Depth 1?

Async (interrupt-driven)

OS cost = \( Ta + Tb = 4.9 + 1.4 = 6.3 \, \mu s \)

Sync (polling)

OS cost = 4.4 \, \mu s

From Yang: FAST ‘12 -10th USENIX Conference on File and Storage Technologies

From Yang: FAST ‘12 -10th USENIX Conference on File and Storage Technologies

From Yang: FAST ‘12 -10th USENIX Conference on File and Storage Technologies
Standards for Low Latency Replication

In most Datacenter usage models, a storage write does not “count” until replicated.

High replication overhead diminishes the performance differentiation of 3D XPoint™ technology.

NVMe over Fabrics is a developing SNIA specification for low overhead replication.
Summary: Block Storage Platform Changes

Move to PCIe based storage
Streamlined command set NVMexpress
OS / SW stack optimizations
Fast replication standards
Persistent Memory Oriented Platform Changes

DIMMS BASED ON 3D XPoint™
INTEL DIMMs
Based on 3D XPoint™ Technology

- DDR4 electrical & physical compatible
- Required support delivered by next generation Intel® Xeon® platform
- Up to 4X system memory capacity, at significantly lower cost than DRAM
- Can deliver big memory benefits without modifications to OS or applications

INTEL DIMM
(based on 3D XPOINT™ Technology)

Future Xeon® Processor

DDR4 DIMM
(acts as write-back cache)
Why Persistent Memory?

- NAND MLC NVMe SSD (4kB read)
- 3D XPoint NVMe SSD (4kB read)
- 3D XPoint DIMM Memory (64B read)

Latency (usecs)

- NAND MLC NVMe SSD: 30
- 3D XPoint NVMe SSD: 20
- 3D XPoint DIMM Memory: 5

Inset: Comparison of read speeds for different storage technologies.
Open NVM Programming Model

SNIA Technical Working Group
Initially defined 4 programming modes required by developers

Spec 1.0 developed, approved by SNIA voting members and published

- Interfaces for PM-aware file system accessing kernel PM support
- Interfaces for application accessing a PM-aware file system
- Kernel support for block NVM extensions
- Interfaces for legacy applications to access block NVM extensions
NVM Library: pmem.io

64-bit Linux Initially

- Open Source
  - [http://pmem.io](http://pmem.io)
- libpmem
- libpmemobj
- libpmemblk
- libpmemlog
- libvmem
- libvmmalloc
Write I/O Replaced with **Persist Points**

- Application
- Standard File API
- Load/Store
- User Space
- NVM Library
- No Page Cache
- pmm-Aware File System
- MMU Mappings
- pmem-aware File System
- Kernel Space
- Traditional APIs
  - msync()
  - FlushViewOfFile()
- NVML API
  - pmem_persist()
Storage Class Memory Support in the Windows Operating System

Neal Christiansen
Principal Development Lead
Microsoft
nealch@microsoft.com
The Data Path

MOV

L1 L1
L2

Core

L1 L1
L2

Core

L1 L1
L2

Core

L1 L1
L2

Core

L1 L1
L2

L3

Memory Controller

NVDIMM

Memory Controller

NVDIMM
New Instructions For Flushing Writes

MOV

Core
L1 L1
L2

Core
L1 L1
L2

Core
L1 L1
L2

Core
L1 L1
L2

L3

Memory Controller

NVDIMM

NVDIMM

CLFLUSH, CLFLUSHOPT, CLWB

PCOMMIT
### Flushing Writes from Caches

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLFLUSH addr</td>
<td>Cache Line Flush: Available for a long time</td>
</tr>
<tr>
<td>CLFLUSHOPT addr</td>
<td>Optimized Cache Line Flush: New to allow concurrency</td>
</tr>
<tr>
<td>CLWB addr</td>
<td>Cache Line Write Back: Leave value in cache for performance of next access</td>
</tr>
</tbody>
</table>
## Flushing Writes from Memory Controller

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCOMMIT</td>
<td>Persistent Commit: Flush stores accepted by memory subsystem</td>
</tr>
<tr>
<td>Asynchronous DRAM Refresh</td>
<td>Flush outstanding writes on power failure</td>
</tr>
<tr>
<td></td>
<td><strong>Platform-Specific Feature</strong></td>
</tr>
</tbody>
</table>
Example Code

```
MOV X1, 10
MOV X2, 20
MOV R1, X1
SFENCE
CLFLUSHOPT X1
CLFLUSHOPT X2
SFENCE
PCOMMIT
SFENCE
```

**Comments**

- X2, X1 are in pmem
- Stores to X1 and X2 are globally visible, but may not be persistent
- X1 and X2 moved from caches to memory
- Ensures PCOMMIT has completed
Join the Discussion about Persistent Memory

Learn about the Persistent Memory programming model
- [http://www.snia.org/forums/ssi/nvmp](http://www.snia.org/forums/ssi/nvmp)

Join the pmem NVM Libraries Open Source project
- [http://pmem.io](http://pmem.io)

Read the documents and code supporting ACPI 6.0 and Linux NFIT drivers
- [https://git.kernel.org/cgit/linux/kernel/git/djbw/nvdimm.git/log/?h=nd](https://git.kernel.org/cgit/linux/kernel/git/djbw/nvdimm.git/log/?h=nd)
- [https://github.com/pmem/ndctl](https://github.com/pmem/ndctl)
- [https://github.com/01org/prd](https://github.com/01org/prd)

**Intel Architecture Instruction Set Extensions Programming Reference**

**Intel 3D XPoint™ Memory**
Persisted Memory Summary

New storage model for low latency
New instructions to support persistence
OS support
Lots of innovation opportunity
Low Latency Ahead

Persistent Memory

3D XPoint™ memory

<1 usec

NVMe SSD

Ultra fast SSD

<10 usec