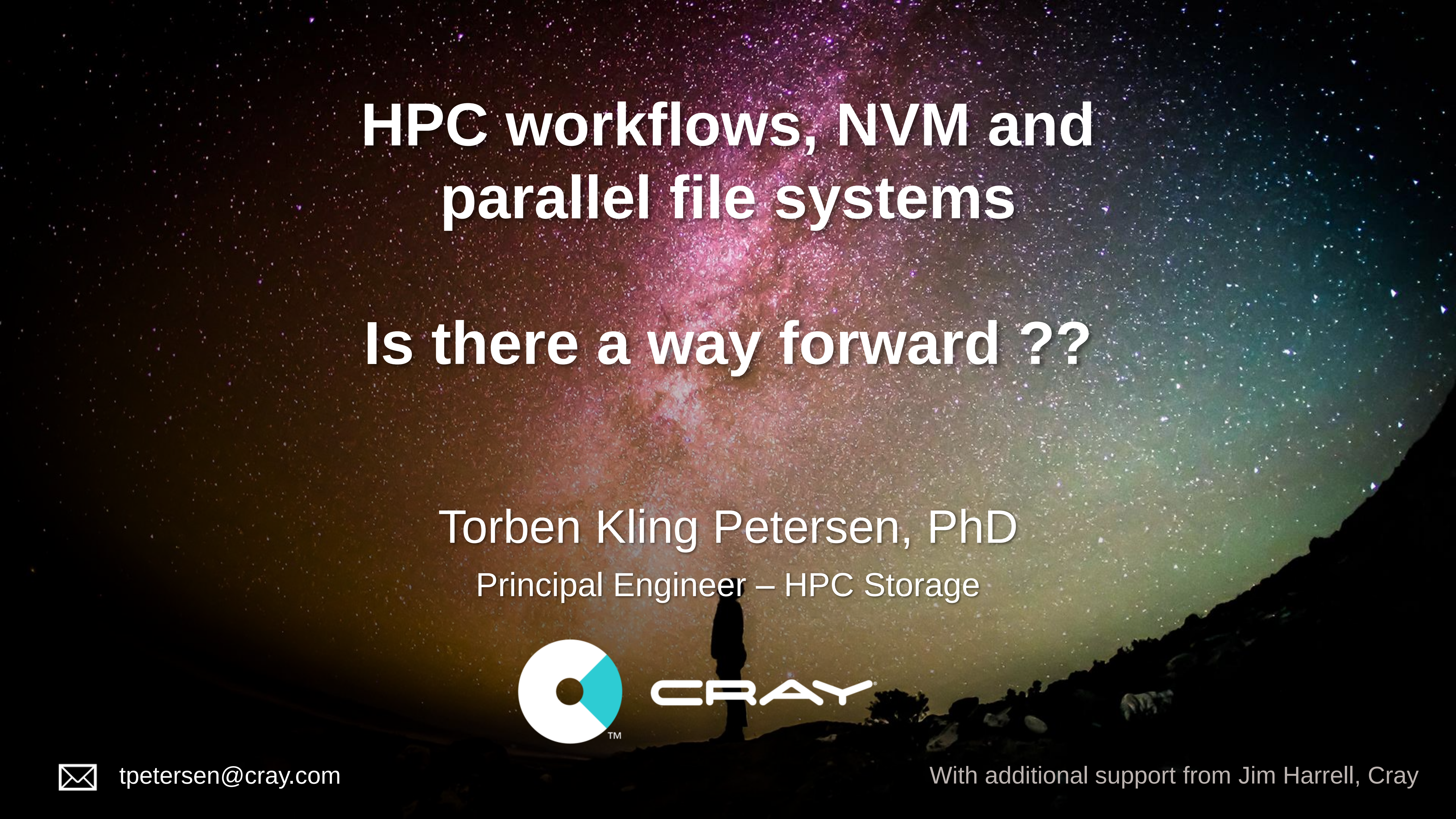


# HPC workflows, NVM and parallel file systems

# Is there a way forward ??

Torben Kling Petersen, PhD

Principal Engineer – HPC Storage

tpetersen@cray.com

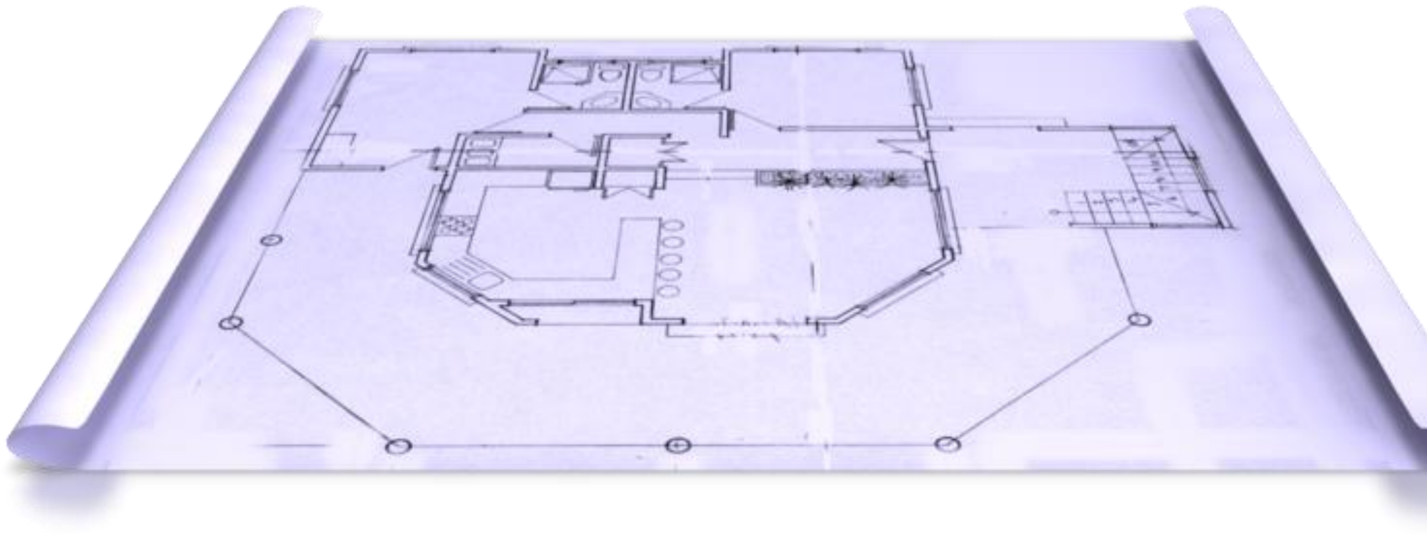
With additional support from Jim Harrell, Cray

# Initial caveats …

- I’m NOT pre-announcing any new products
- Opinionated view are mine and not Cray’s ….
- Nowadays, I’m a “storage guy” and compute is less interesting

- The fine print

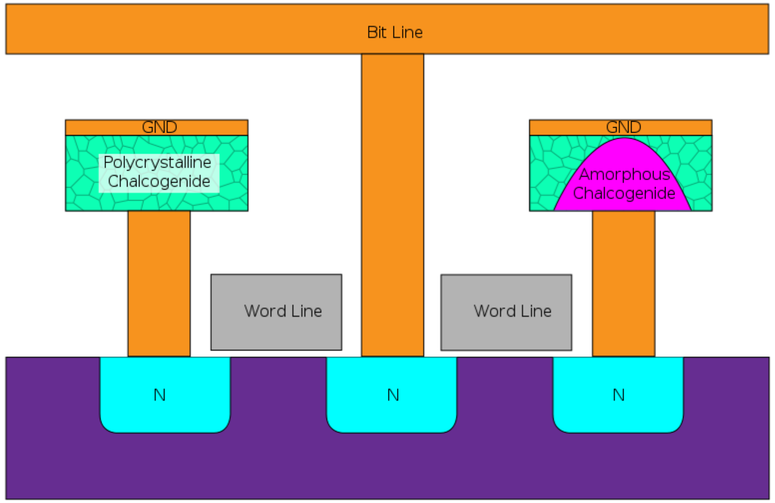
This presentation may contain forward-looking statements that are based on our current expectations. Forward looking statements may include statements about our financial guidance and expected operating results, our opportunities and future potential, our product development and new product introduction plans, our ability to expand and penetrate our addressable markets and other statements that are not historical facts.
These statements are only predictions and actual results may materially vary from those projected. Please refer to Cray's documents filed with the SEC from time to time concerning factors that could affect the Company and these forward-looking statements.



# Definition of NVM (WikiPedia …)

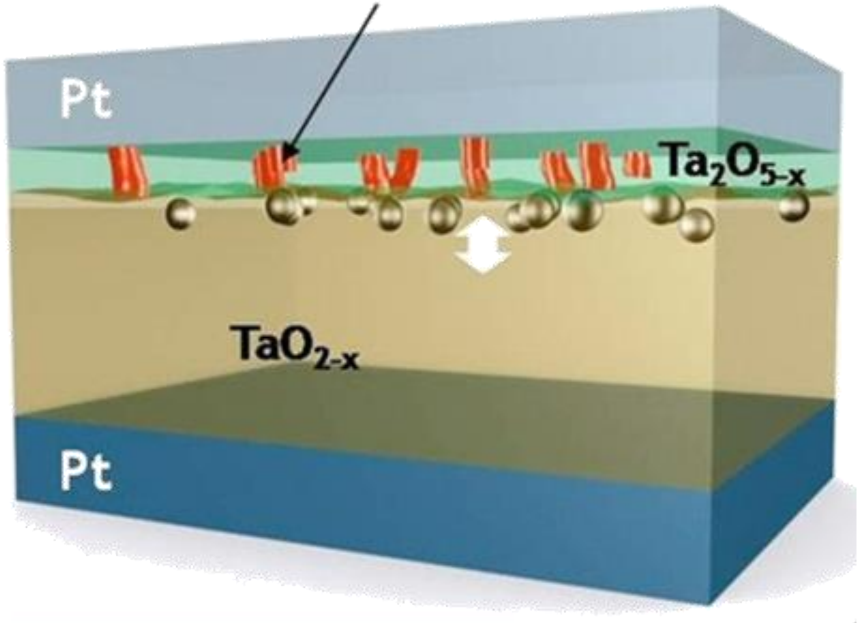
**Non-volatile memory** (**NVM**) or **non-volatile storage** is a type of computer memory that can retrieve stored information even after having been power cycled.
In contrast, volatile memory needs constant power in order to retain data.
Examples of non-volatile memory include flash memory, read-only memory (ROM), ferroelectric RAM, most types of magnetic computer storage devices (e.g. hard disk drives, floppy disks, and magnetic tape), optical discs, and early computer storage methods such as paper tape and punched cards.

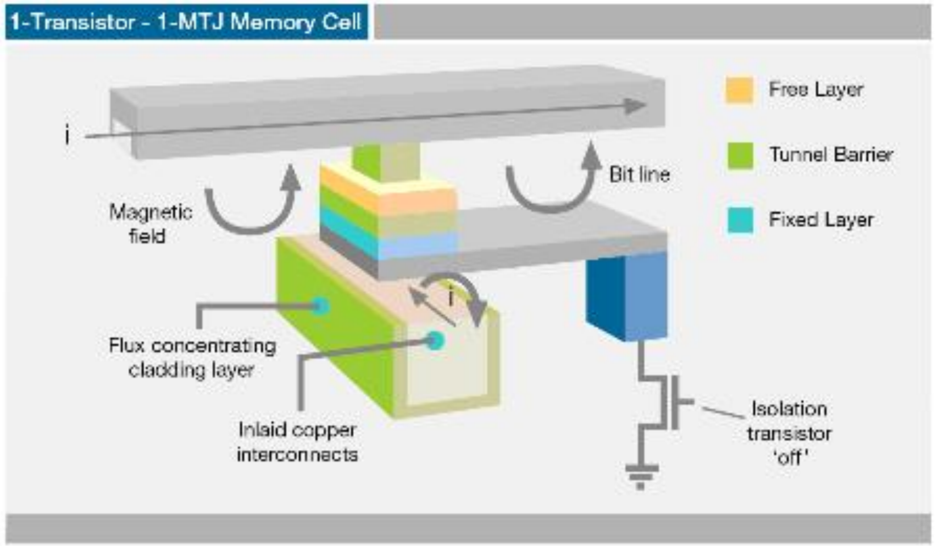
# New Devices – Challenging DRAM and Flash



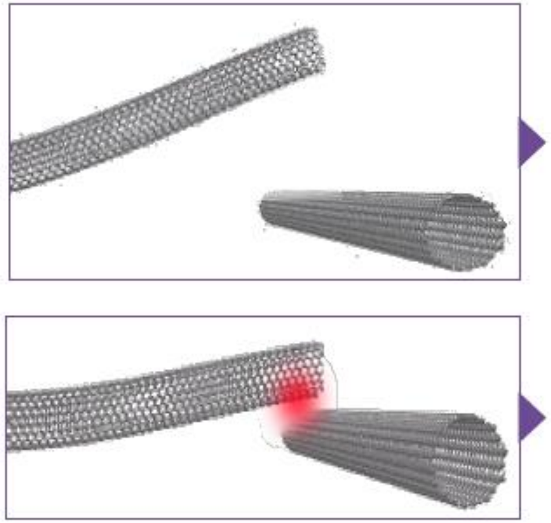
Phase Change Memory (PCM or PCRAM)

Resistive Memory (ReRAM)




1-Transistor - 1-MTJ Memory Cell
i
Magnetic field
Bit line
Free Layer
Tunnel Barrier
Fixed Layer
i
Flux concentrating cladding layer
Inlaid copper interconnects
Isolation transistor 'off'

Spin-Transfer Torque Magneto-resistive Memory (STT-MRAM)



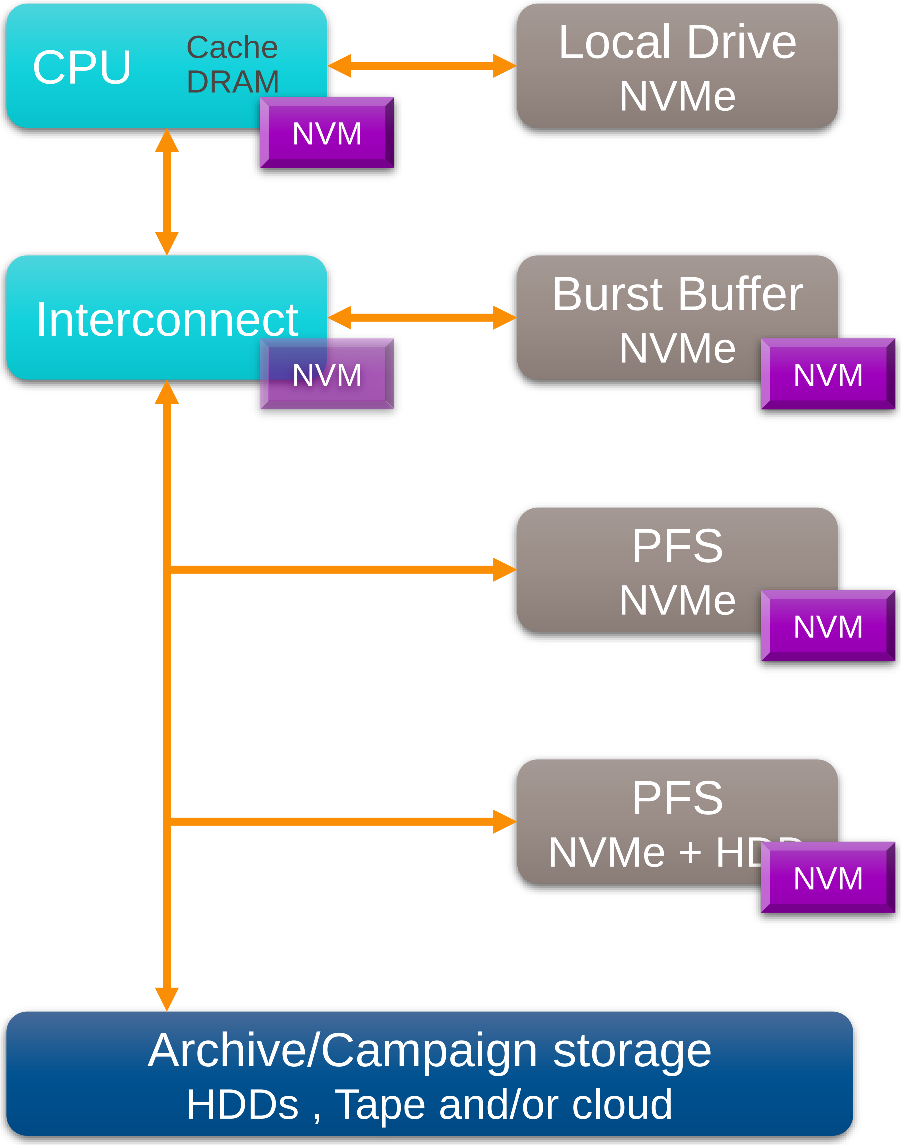
Carbon Nanotube (NRAM)

# Emerging Technologies

| | DRAM | NAND Flash | PCRAM / 3DXP | STT-MRAM | ReRAM | NRAM |
|---|---|---|---|---|---|---|
| **Availability** | Now | Now | Now | 2020 | Varies | 2020 |
| **Device Density** | >128 GB | Up to 1 TB | 256 GB | 4 GB | Varies | 16 GB |
| **R/W Latency** | 14ns / 14ns | 30µs / 120µs | 350ns / 1µs | 15ns / 30ns | 1µs / 1µs | 15ns / 15ns |
| **Endurance (writes)** | > 1E16 | > 1E4 | > 1E8 | > 1E12 | > 1E8 | >> 1E10 |
| **Retention** | < 300ms | 1 yr | 6hr to 3mo (depending on temperature) | 10 yrs | 10 yrs | "Forever" |

***Hic Sunt Dracones***

# The data IO path and options …



- Byte addressable or
- File and/or block based
- POSIX or ….
- File system or key value store
- Advanced vs simple IO protocol
- One logical name space or …
- Data integrity or replication …
- Object store or hierarchical …

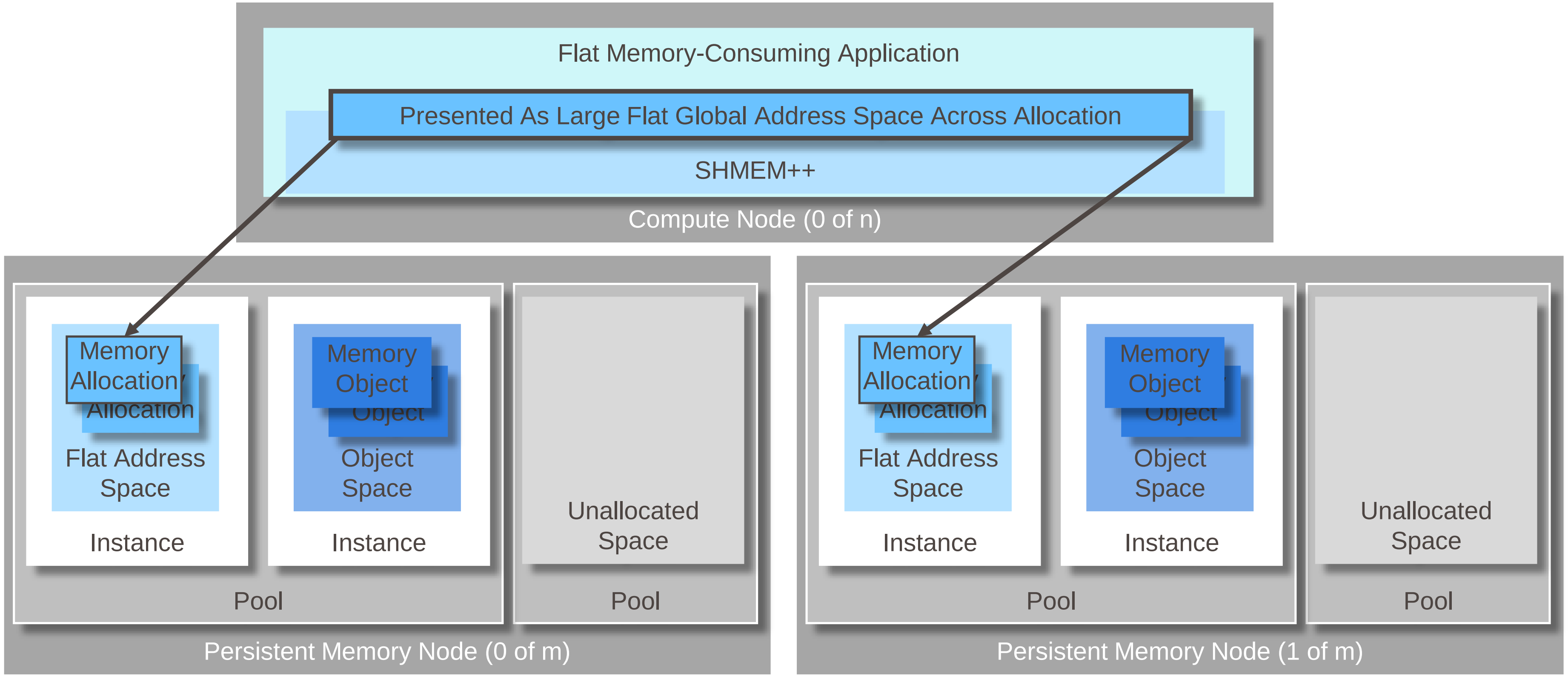
# Locality – Within Compute

- DIMM-like, M.2/U.2 formfactor or AIC ??
- Usage pattern
  - Memory Mode
  - App Direct Mode ..
    - Raw device Access
    - File System
    - NVMe Aware file system
    - Memory Access
- Application readiness / Compiler support …
- Remote data access

Too complicated ??

# NVM Memory - Cray Architecture

- Issues that require(d) some thought and discussion...
  - How does a user share remote memory or find it for reuse?



# So if node local NVM is not optimal, what then ???

nvm
EXPRESS

# NVMe Gen 4

- NVMe spec 1.4
  - Official multi-port support
  - Asymmetric name space support
  - IO Determinism
  - Shared Stream Write
- NVMe Gen 4 U.2 devices
  - Quad port (2x2 or 4x1)
  - Enhanced monitoring and persistent logs
- Includes both PCM and NAND

Pictures taken from https://news.samsung.com/medialibrary/

# Performance comparisons

| **Feature** | **NVMe 1.3 (Samsung PM1725)** | **NVMe 1.4 (Samsung PM1733)** |
|---|---|---|
| Streaming write (GB/s) | 3.1 | 4.8 |
| Streaming read (GB/s) | 3.5 | 8 |
| Random 8 k IOPS write | 190,000 | 800,000 |
| Random 8 k IOPS read | 350,000 – 400,000 | 1,500,000 |
| Ports | 2 | 4 |
| PCIe performance (theoretical) GB/s | 4.2 | 8.8 |
| Capacity DWPD=1 | Up to 12.8 TB | Up to 30.7 TB |
| Write perf (2U Array) | ~ 20 – 30 GB/s | ~ 60 - 80 GB/s |
| Read perf (2U Array) | ~ 30 - 40 GB/s | ~ 100 - 120 GB/s |

# File systems ?

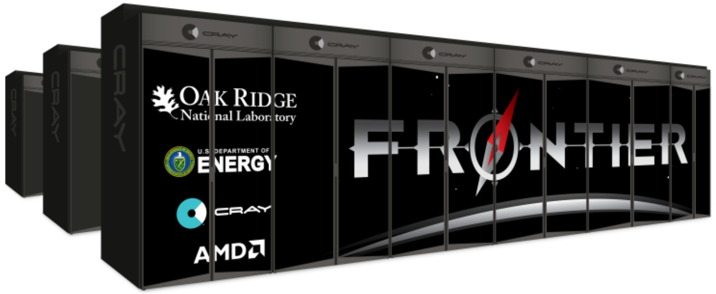
- Traditional parallel file systems
  - Lustre, Spectrum Scale, BeeGFS, PVFS etc
- Object Stores
  - Ceph, OpenIO, WekaIO, DAOS
- Key Value Stores
  - HDHIM, Mero, Memcached, Redis
- New hybrid architectures
  - ???

Lustre®

# Exascale examples

- NERSC – Perlmutter
  - Lustre file system >4 TB/sec and 30 PB all flash file system

- ORNL – Frontier
  - Lustre FS >10TB/s with 1 EB+



- LLNL – El Capitan
  - Next Gen ClusterStor
  - Probably Lustre ??

Lawrence Livermore National Laboratory
U.S. Department of Energy
NNSA
CRAY
EL CAPITAN

- ANL – Aurora
  - DAOS frontend with a Lustre backend

# Lustre - Existing and future enhancements

### Current features (2.12)

- Data on MDT (DoM)
- Progressive File Layouts (PFL)
- Multiple modify RPCs
- File Lever Replication
- Additional flash enhancements †

  + ***NXD****

### Forthcoming features

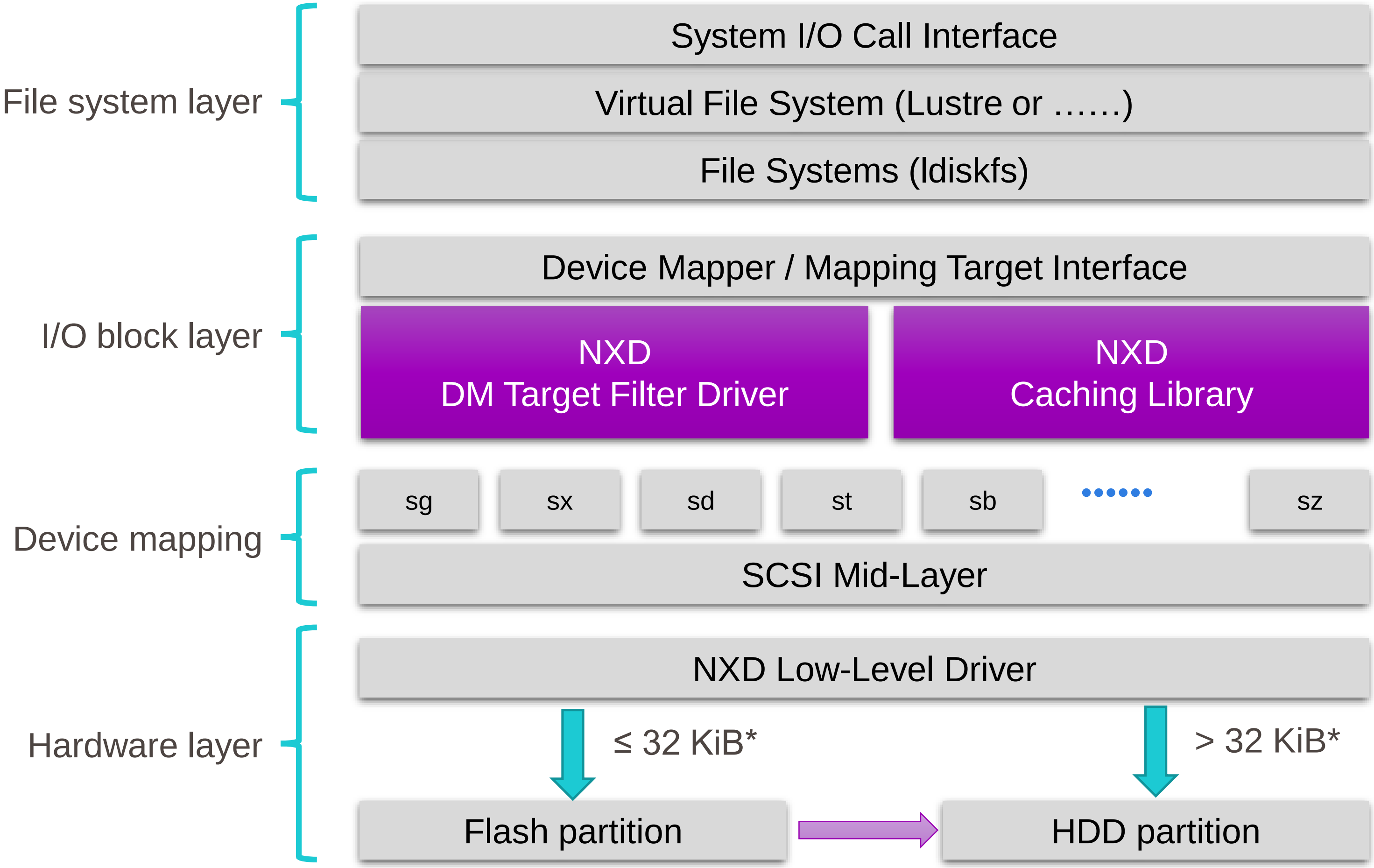
- OST Over Striping
- DNE3
- Lazy Size on MDT
- Persistent Client Cache (NVM support/DAX)
- Client side processing and compression
- Client Metadata Writeback Cache

| † CLIENT | NETWORK | METADATA |
|---|---|---|
| Small IO improvements<br>• Tiny writes [2.11] *<br>• Fast reads [2.11] *<br>• Lockless Direct IO * | • Lock Ahead [2.11] *<br>• Immediate short IO [2.11] *<br>• Small overwrites [2.11] *<br>• Data on MDT [2.11] | • DNE.2 [2.8] (reduce MDS bottlenecks)<br>• Progressive file layouts [2.10]<br>• Pool Quotas* |

* Contributions by Cray

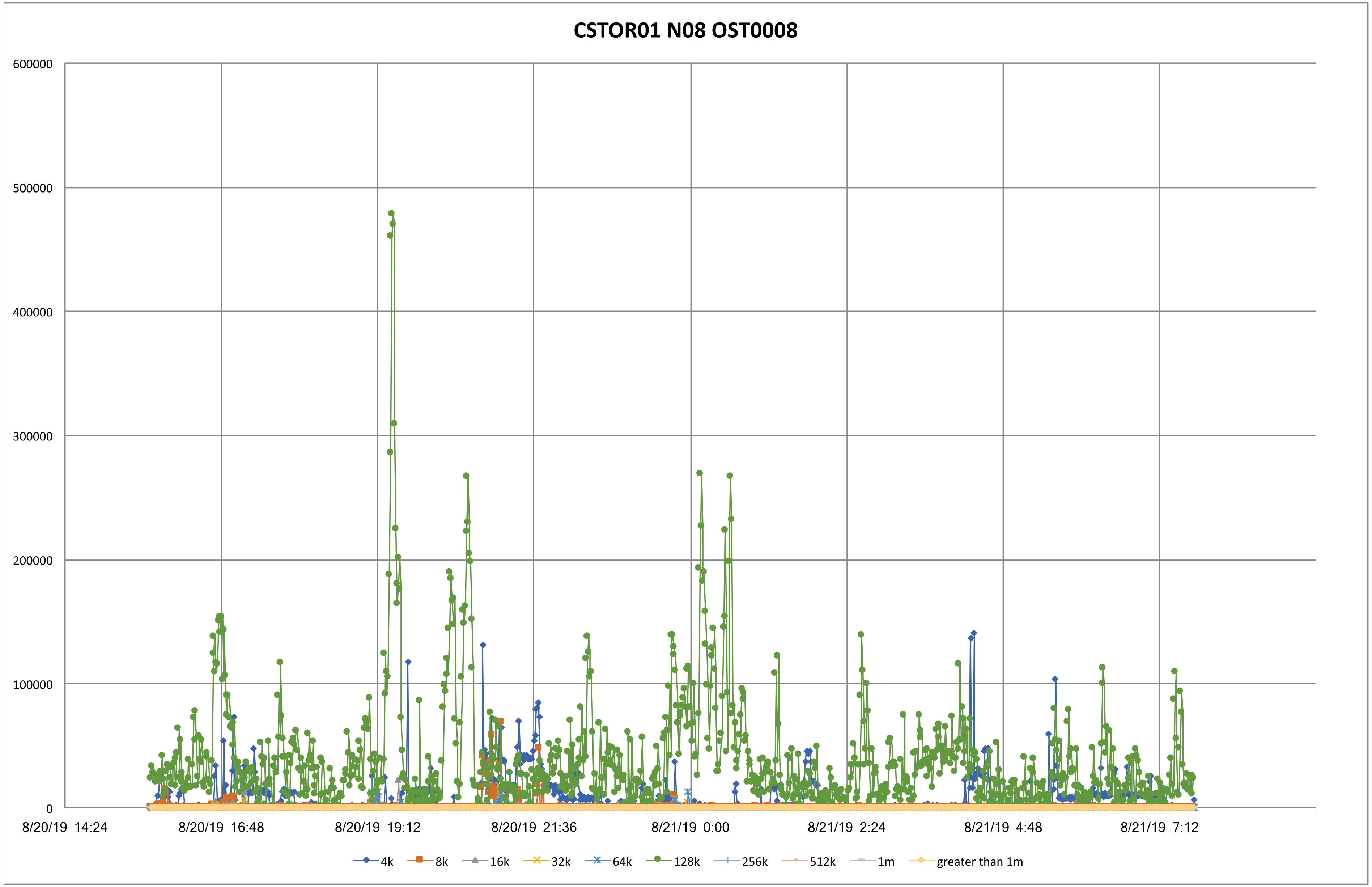
# NXD - Transparent write caching

- Up to 4 SSDs for NXD use
- Small block IO Profiler
  - Portable (outside of CS)
- Wire speed block size analyzer
  - Below breakpoint -> directly to Flash partition
  - Above breakpoint –> Standard disk I/O
  - Breakpoint user settable (4k – 128k) ..
- Asynchronous flushing to HDD tier

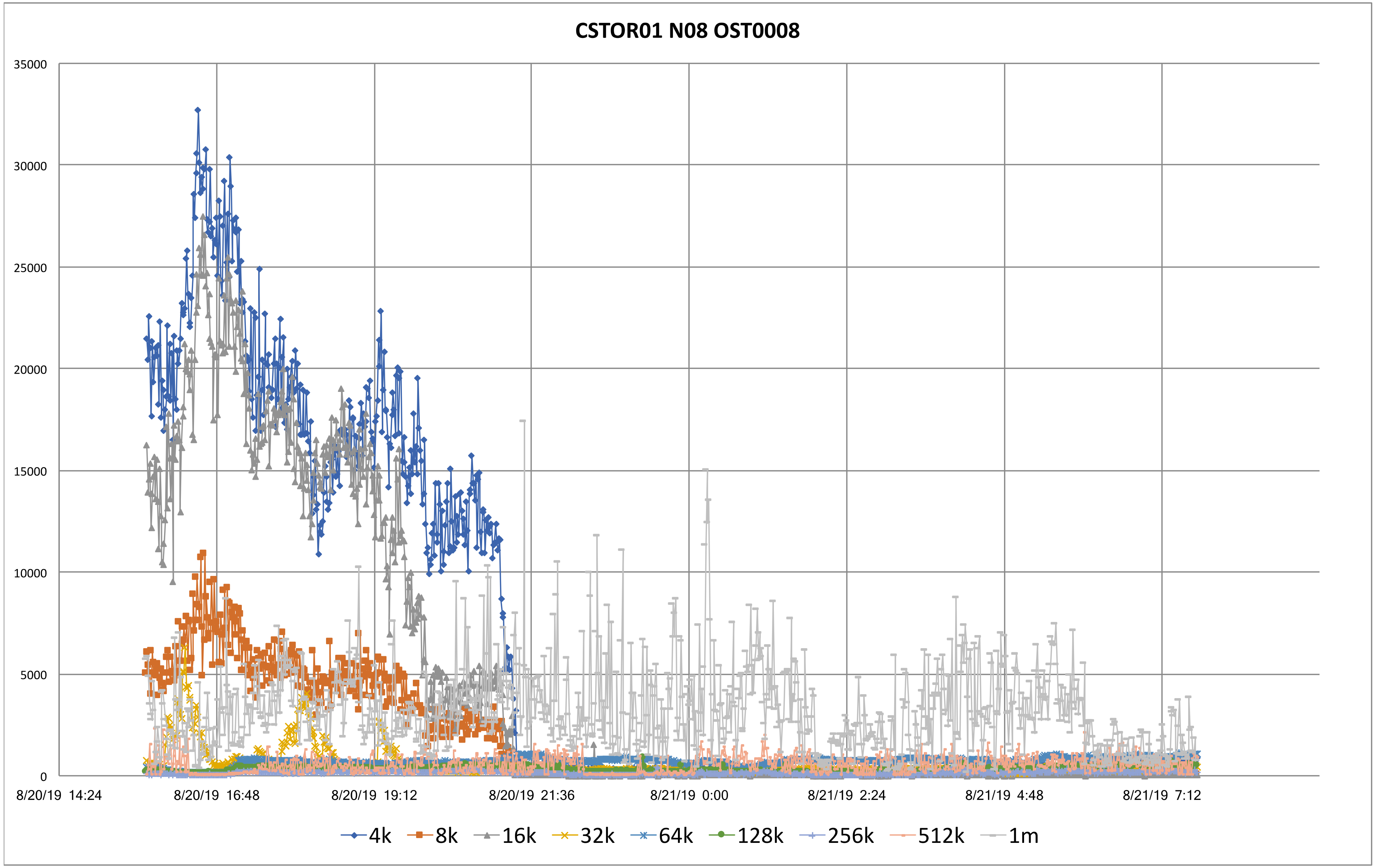


* User settable breakpoint

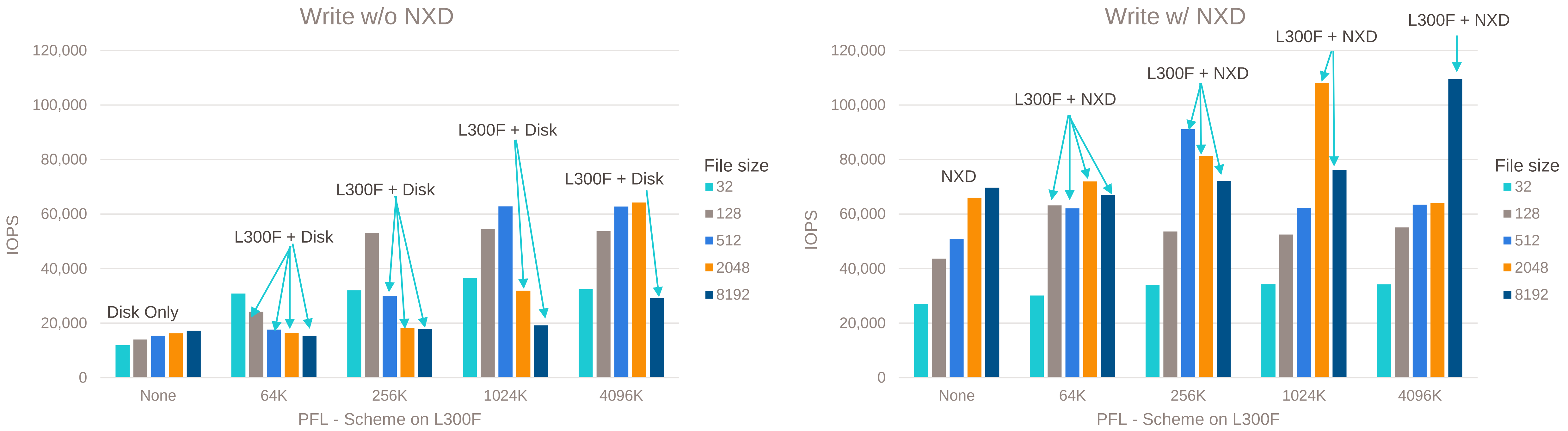
# Analysis of Read I/O



# Analysis of Write I/O



# L300F (All Flash Array) + L300N HDD w/ NXD Working Together Write IOPS Acceleration - Random

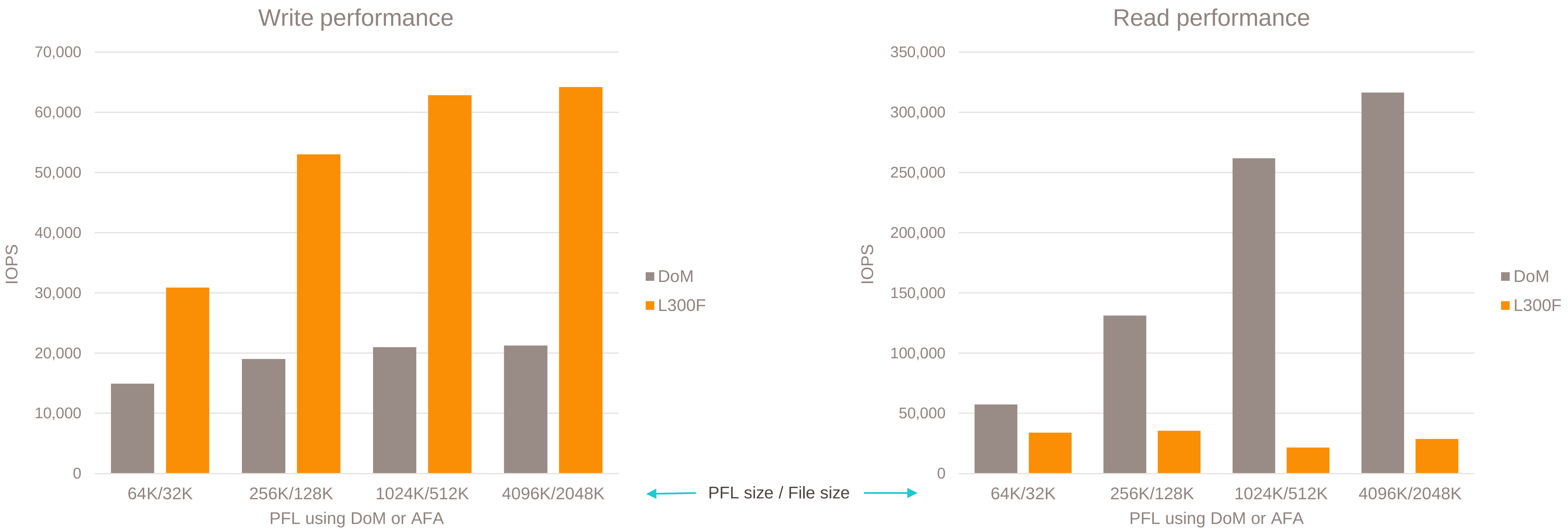


- On Average, NXD accelerated Writes with combination of L300F and L300N with NXD by 3.5x to 4.5x Speed Up
- NXD with Small File improvements using L300F with IOR Random Updates to a large File shows better performance than just using L300F

**NXD + Lustre Flash Targets accelerate IOPS for random unaligned writes workload**

# Comparing DoM vs. All Flash Array for Small File

## All file sizes < PFL segment on MDT/AFA (Direct IO)



**Read intensive Workloads – DoM is the right solution** (2-10x Gain over AFA)
**Write intensive Workloads – AFA is the right solution** (3x Gain over DoM)

# Evaluating Lustre on NVM

- Lustre is NOT optimized for NVM based targets
- New features help but …
- Tiering is now a requirement
- Differences between HDDs and NVM not well understood
  - RAID, T10DIF, TRIM/UNMAP, ….
- IOPS vs Throughput
- Standards vs implied/implemented standards
- Benchmarking vs Production (i.e. the Write Cliff issue)
- Usability at (almost) full capacity

# Write a new file system – Easy !!!!

- **FAST** (IOPS and throughput)
- Reliable
  - RAID protection or network erasure coding
  - Data integrity (T10-PI etc)
- Scalable
  - Metadata, name space, performance, ….
- Secure
  - Multitenancy
  - Encryption
  - Role based access
- Features ..
  - CoW, Snapshots, Replication, Compression
- Mature
  - Proven over time

- Usability
  - Simple to manage
- POSIX or ???
  - Support for ”traditional protocols” (NFS, S3)
- Eliminate downtime
  - Rolling Upgrades
  - Containers and Kubernetes
- Built in HSM
  - … and a back-up function
  - Disaster recovery and geo distribution
- Open Source …
- Monitoring, Analytics and AI

- And did I mention it need to be **FAST**

# So …. What's the verdict ???

Lustre still got a lot of life left !!!

But it's getting harder to continue improving it …

Replacing Lustre ??

- Parallel file system
- Object Storage
- Key Value Store

# Thank you for listening to a ramblings of a slightly deranged engineer

QUESTIONS?

✉ tpetersen@cray.com