Open-Channel SSDs

Philippe Bonnet
phbo@itu.dk
IT University of Copenhagen

Joint work with
Matias Bjørling, Javier Gonzalez (CNEX Labs)
Ivan Luiz Piccoli, Carla Villegas, Björn Jonssón (IT University of Copenhagen)
Luc Bouganim (INRIA)

L.Bouganim et al. CIDR 2011

MINIMAL FTL: TAKE THE FTL OUT OF THE EQUATION!

Pros

Maximal performance for

- SR, RR, SW
- Semi-Random Writes

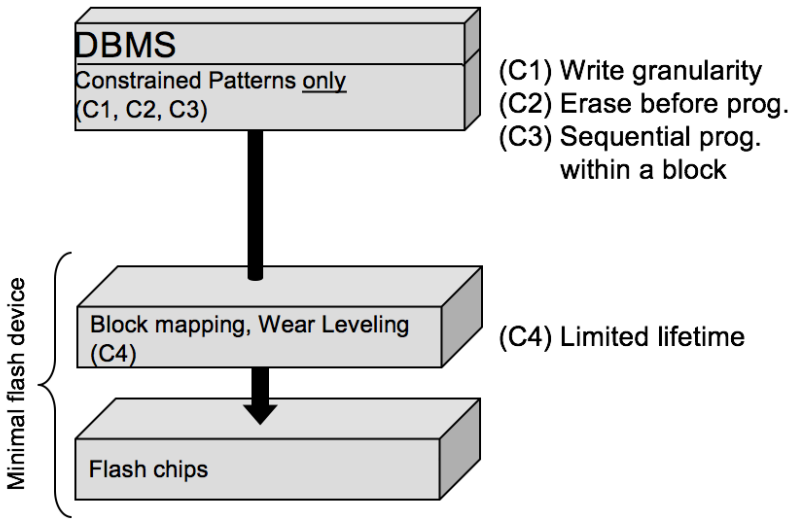
Maximal control for the DBMS

Cons

All complexity is handled by the DBMS

All IOs must follow C1-C3

- The whole DBMS must be rewritten
- The flash device is dedicated



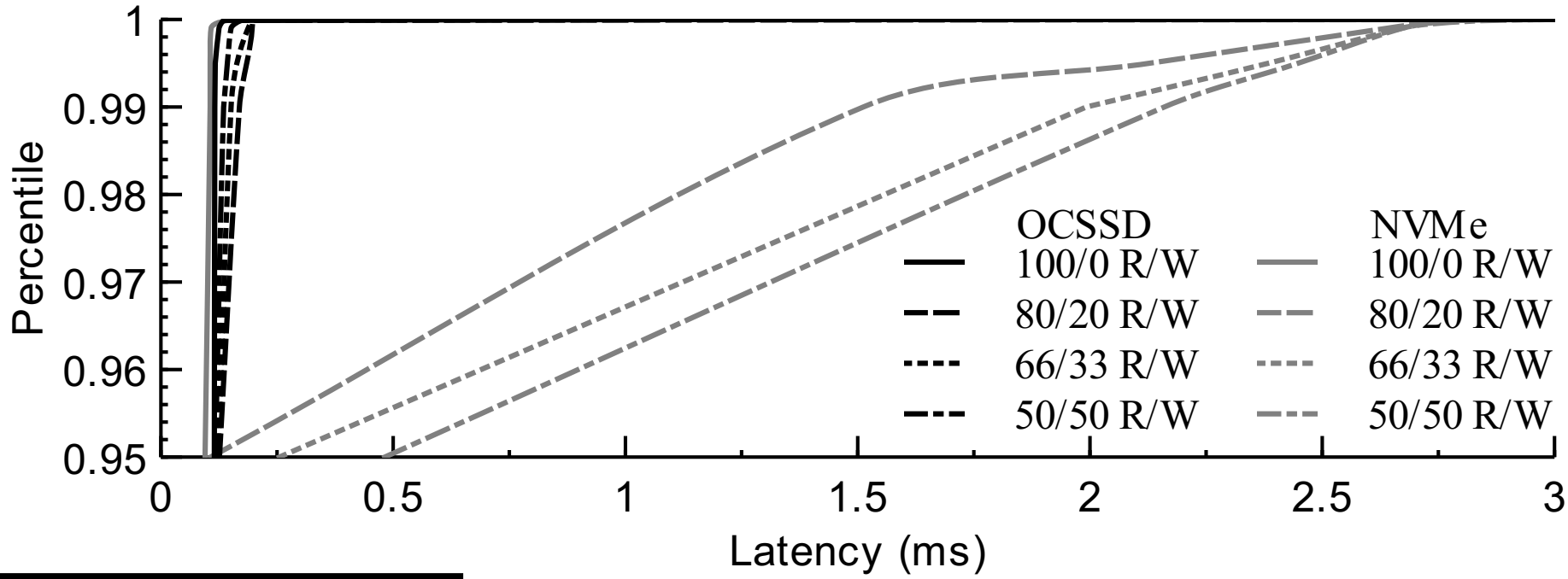
FAST 2017


usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION
LightNVM: The Linux Open-Channel SSD Subsystem
Matias Bjørling, CNEX Labs, Inc. and IT University of Copenhagen; Javier Gonzalez, CNEX Labs, Inc.; Philippe Bonnet, IT University of Copenhagen
https://www.usenix.org/conference/fast17/technical-sessions/presentation/bjorling

Predictable Reads?

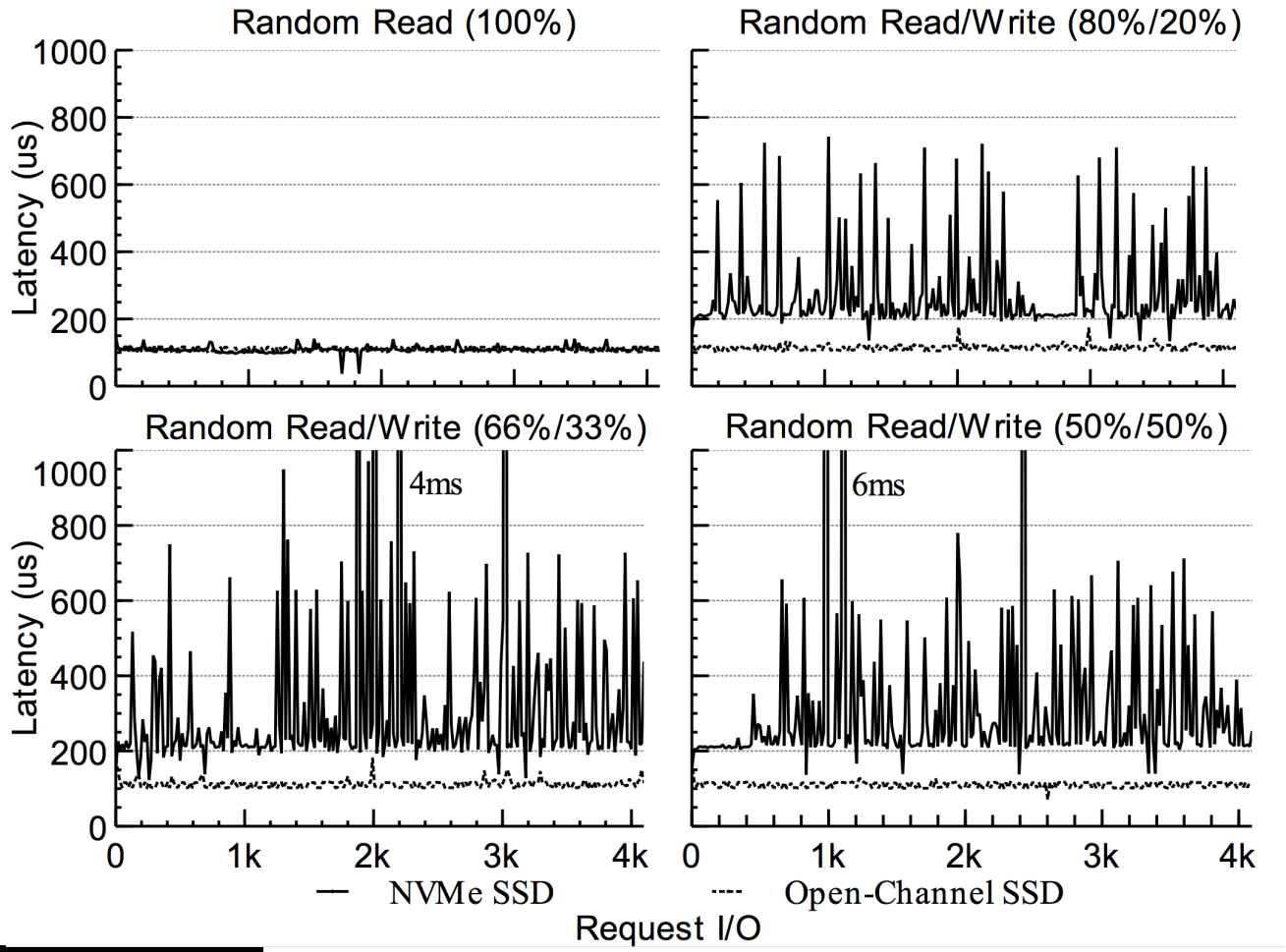
M.Bjørling et al. FAST 2017

Open-Channel SSD: CNEX Labs Westlake SDK
Concurrent 4K Reads I 64K Writes



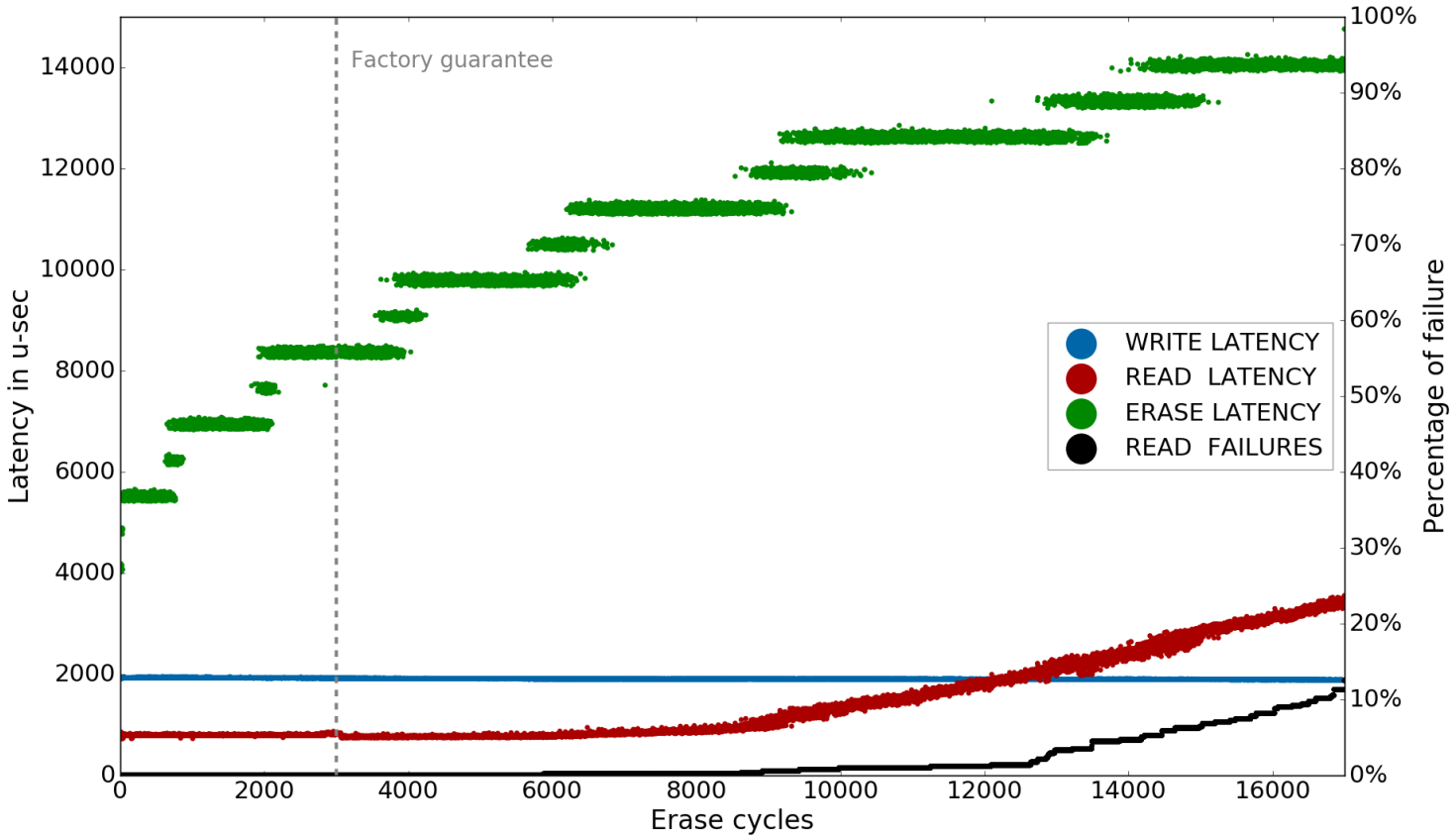
Predictable Reads?

M.Bjørling et al. FAST 2017



Anti latency-binding

I.Picoli et al. APSYS 2017



Open-Channel SSD: DFC Card + OX controller

uFlip-OC μOC_1: Loop (Erase; Write 0-511; Read 0-511)

LightNVM

Taking control of SSDs with LightNVM

LightNVM Documentation Publications Hardware Get Involved

Open-Channel SSDs

I/O Isolation

Enable I/O isolation between tenants by allocating your SSD into separate parallel units.

Predictable Latency

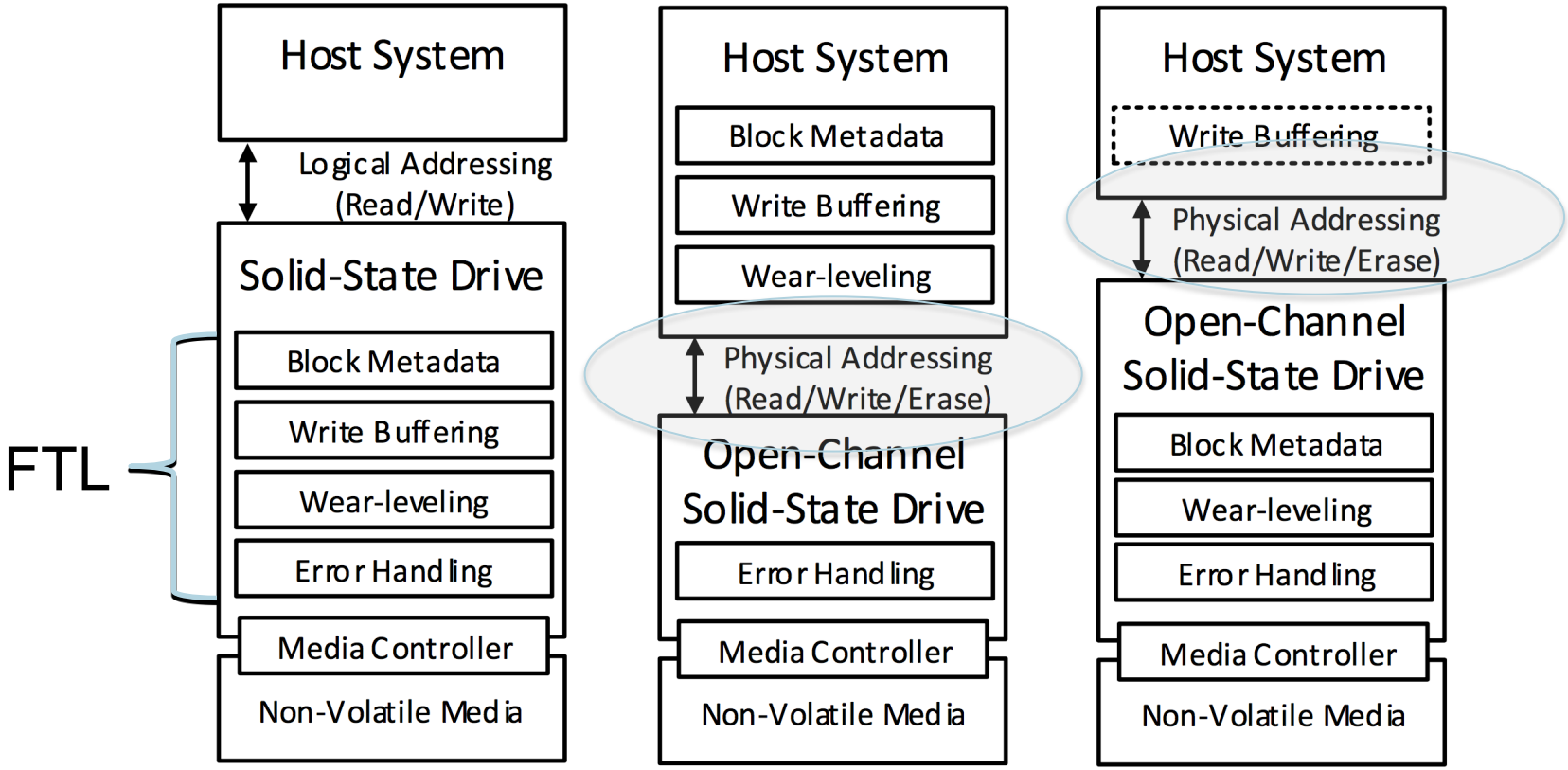
No more guessing when an IO completes. You know which parallel unit is accessed on disk.

Non-Volatile Memory

Manage your non-volatile memory as a block device, through a file-system or inside your application.

lightnvm.io

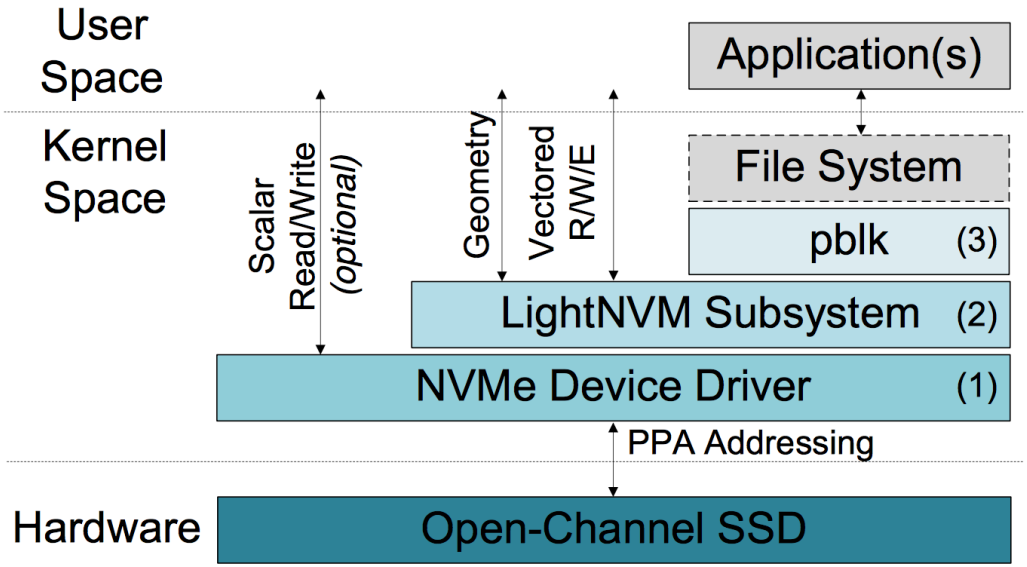
Open-Channel SSDs: Design Space



LightNVM separates (*application-customisable*) **front-end SSD management** from (*media-specific*) **back-end SSD management**.

LightNVM

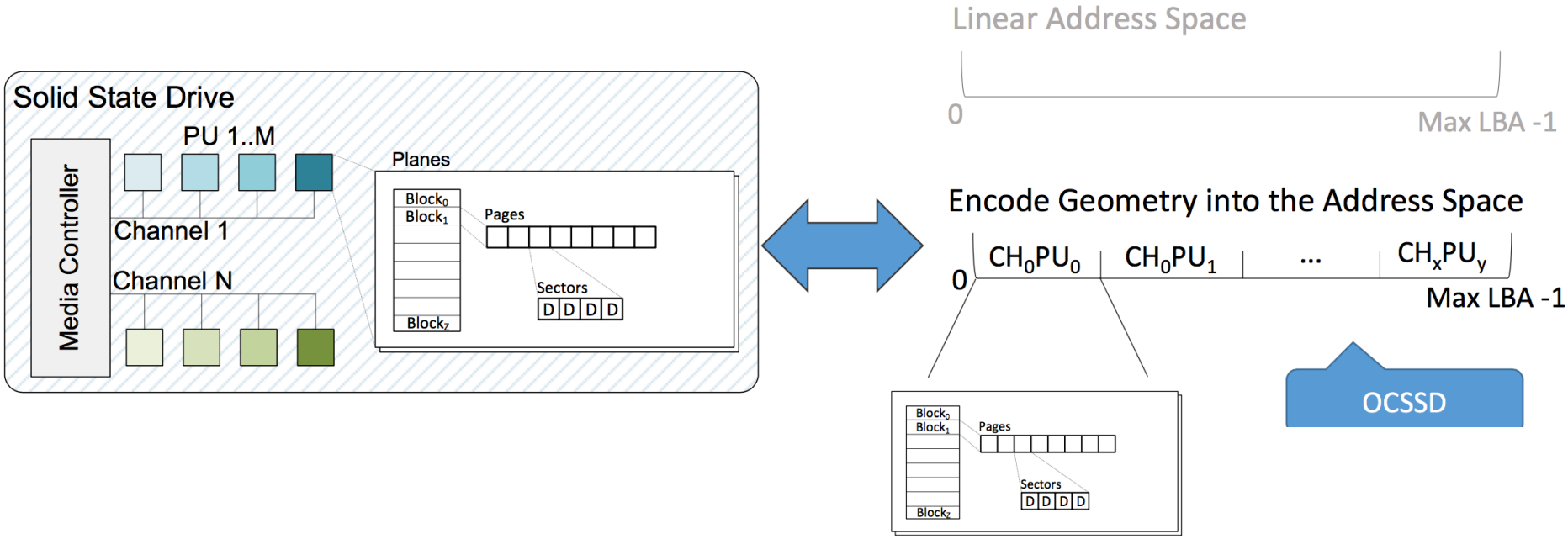
(1) NVMe Device driver
Detection of Open-Channel SSDs
Implements PPA Interface
(2) LightNVM Subsystem
Core SSD management functionality
(3) High Level I/O Interface
Pblk – block device via full-fledge FTL
liblightnvm – access to core
SSD management from user-space



With LightNVM, a system in user space can fully control data placement and I/O scheduling across multiple open-channel SSDs

PPA Address Space

Channels -> Parallel Units -> Planes -> Blocks -> Pages -> Sectors



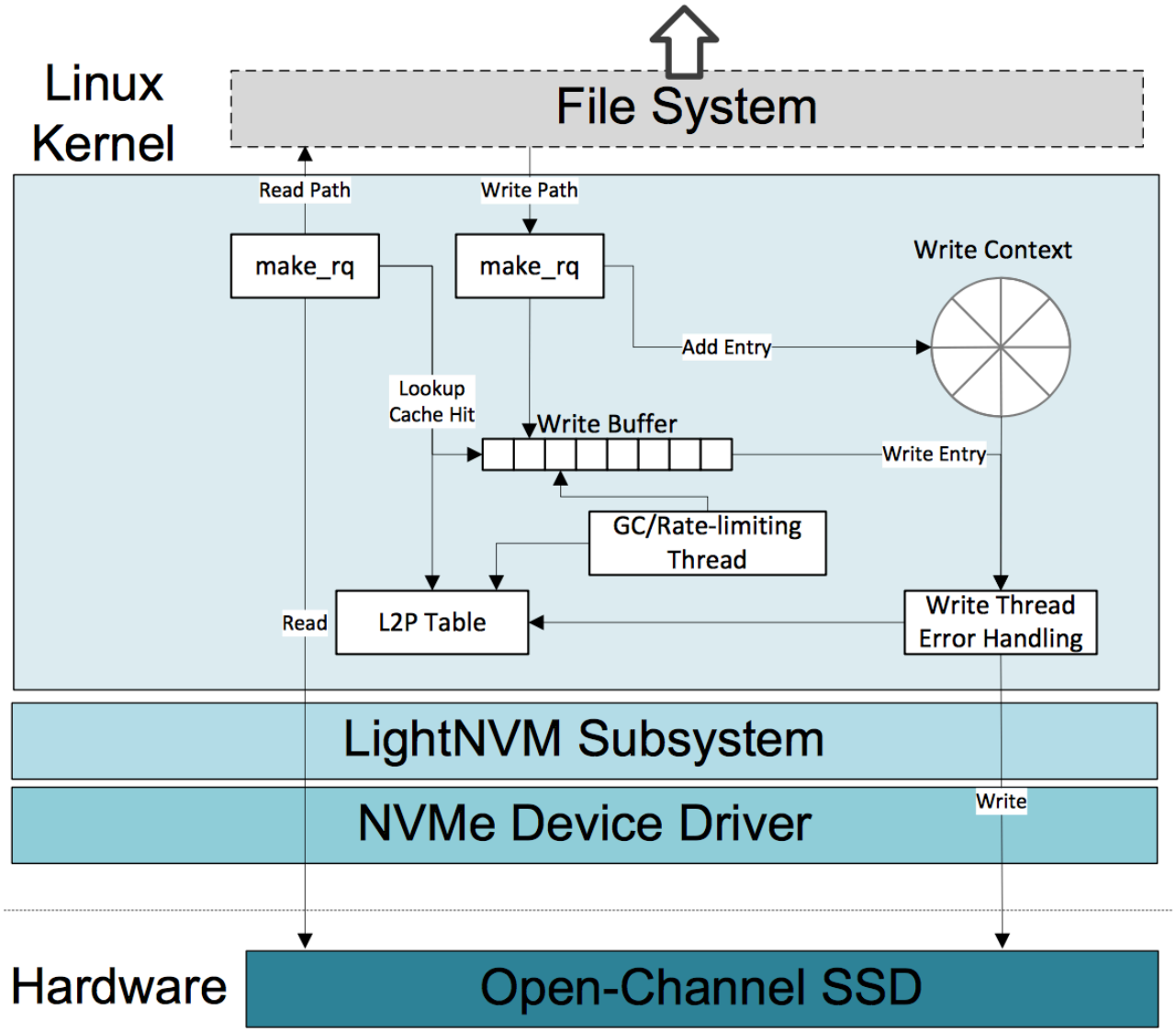
PPA provides (i) a hierarchical address space, based on *SSD intrinsic parallelism (channels and PUs) and media characteristics, and (ii) vectored I/Os.*

The Linux FTL

Pblk

http://lightnvm.io/pblk-tools/

FTL are transactional systems.
[Sang Lyul Min et al, 2002]



Adoption

Radian Memory Systems

RMS-325

12TB Flash
12GB User NVRAM
NVMe PCIe x8 Gen3 Edge Card

CNEX LABS PARTNERS WITH MICROSOFT TO BOOST STORAGE PERFORMANCE FOR THE CLOUD WITH OPEN-CHANNEL SSDS

August 8th, 2017 by CNEX Labs

Programming the Storage Controller

Put Everything in Future (Disk) Controllers (it's not "if", it's "when?")

Jim Gray

http://www.research.Microsoft.com/~Gray

Acknowledgements:
Dave Patterson explained this to me a year ago
Kim Keeton
Erik Riedel
Catharine Van Ingen
} **Helped me sharpen these arguments**

1

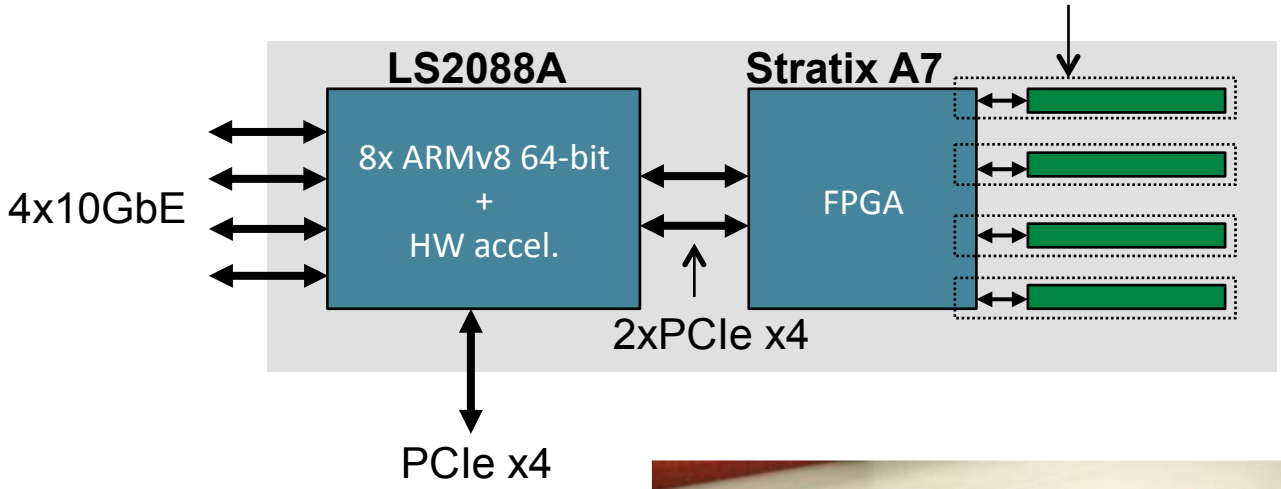
Basic Argument for x-Disks

- Future disk controller is a super-computer.
  - 1 bips processor
  - 128 MB dram
  - 100 GB disk plus one arm
- Connects to SAN via high-level protocols
  - RPC, HTTP, DCOM, Kerberos, Directory Services,….
  - Commands are RPCs
  - management, security,….
  - Services file/web/db/… requests
  - Managed by general-purpose OS with good dev environment
- Move apps to disk to save data movement
  - need programming environment in controller

Jim Gray, NASD Talk, 6/8/98
http://jimgray.azurewebsites.net/jimgraytalks.htm

DragonFire Card (DFC)

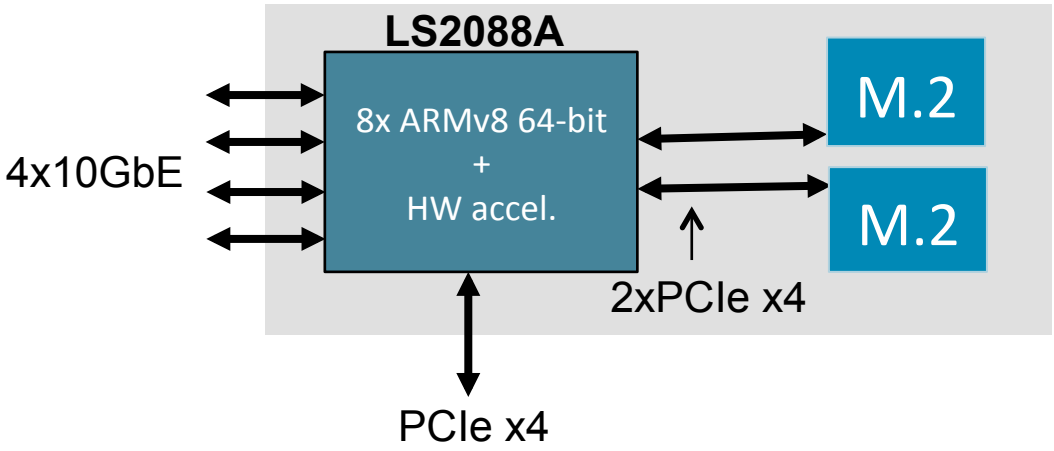
https://github.com/DFC-OpenSource

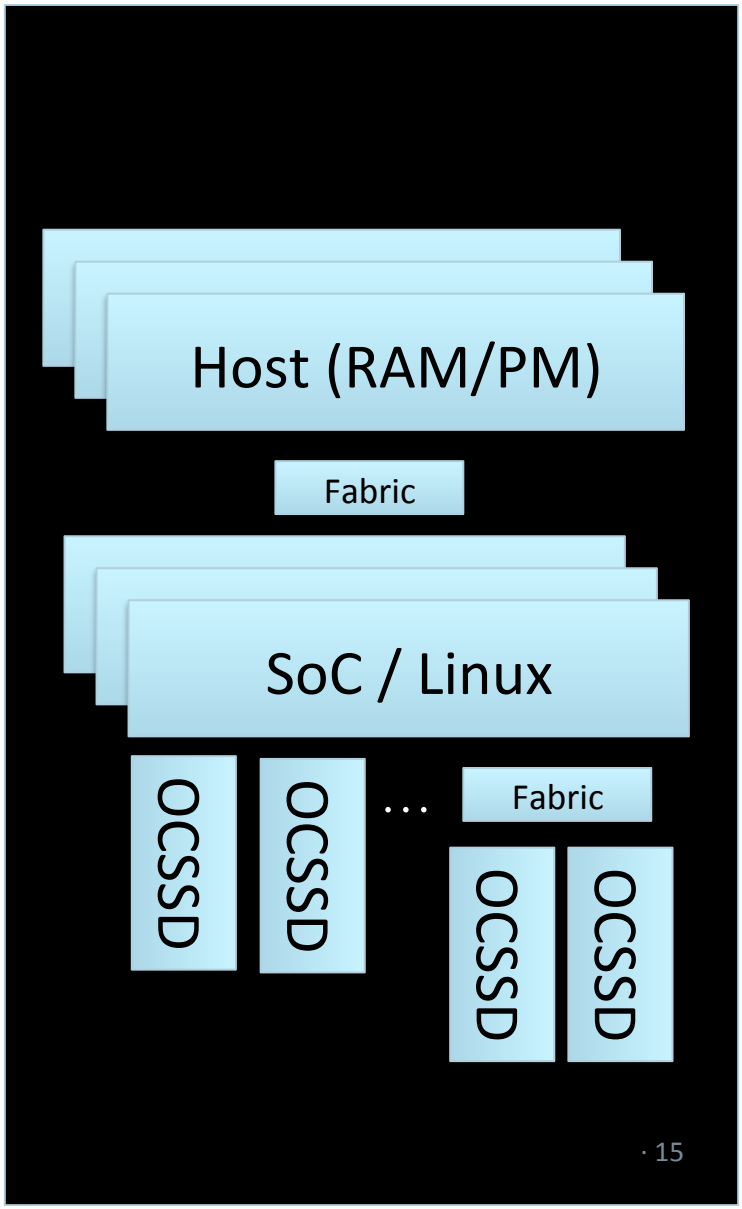


Non-Volatile Memory
(240-pin UDIMM)

IT UNIVERSITY OF COPENHAGEN

DFC M.2 Carrier





Programming the Storage Controller

1. What is “everything”?
   - Store abstraction(s) embedded on storage controller?
   - How much application logic is pushed to the storage controller?
     - Embedded run-time (Rules / DSL); binary; micro-services
2. Storage controller in charge of front-end SSD management and application services
   - OX controller as a framework for programming storage controller
   - Host-SSD protocol?
     - PCIe / Ethernet / NextGen Fabric
     - NVMe/REST/..
3. Streamlining the data path
   - Towards system-wide latency binding
   - OCSSD: No warrantee SSD, HW-based read path
   - Hardware acceleration on the SoC

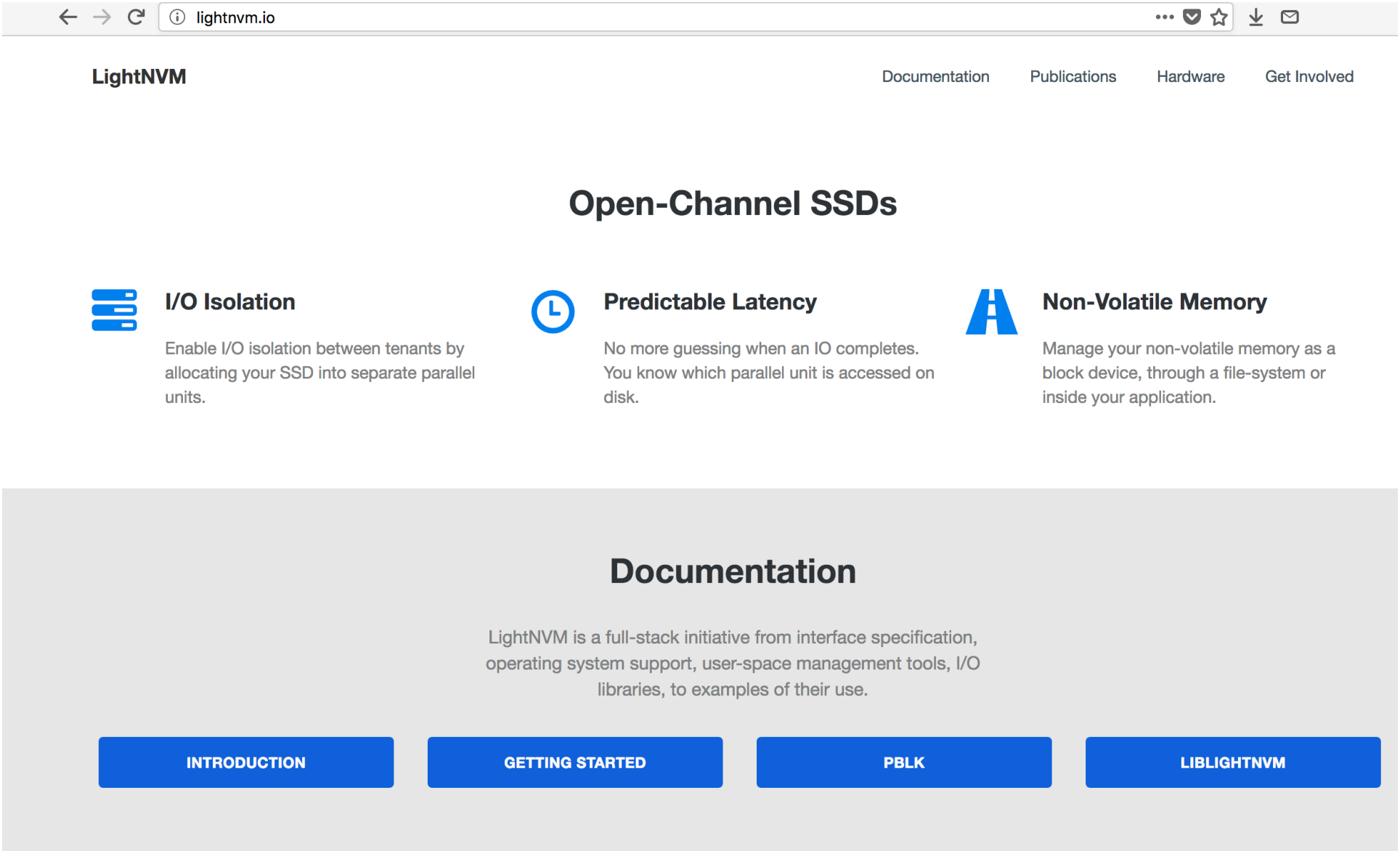
Conclusion

LightNVM separates back-end and front-end SSD management in order to get predictable read latency.

Input from this community needed to standardize open-channel SSD interface.

The time for programming the storage controller is now.
The DFC is an ideal platform for exploring this design space.

Join the DFC community!


lightnvm.io
LightNVM
Documentation
Publications
Hardware
Get Involved
Open-Channel SSDs
I/O Isolation
Enable I/O isolation between tenants by allocating your SSD into separate parallel units.
Predictable Latency
No more guessing when an IO completes. You know which parallel unit is accessed on disk.
Non-Volatile Memory
Manage your non-volatile memory as a block device, through a file-system or inside your application.
Documentation
LightNVM is a full-stack initiative from interface specification, operating system support, user-space management tools, I/O libraries, to examples of their use.
INTRODUCTION
GETTING STARTED
PBLK
LIBLIGHTNVM