OX-App: A Framework for Application-specific FTLs on Tiered Storage

Ivan Luiz Picoli
Ph.D. Fellow

Near-data Processing

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

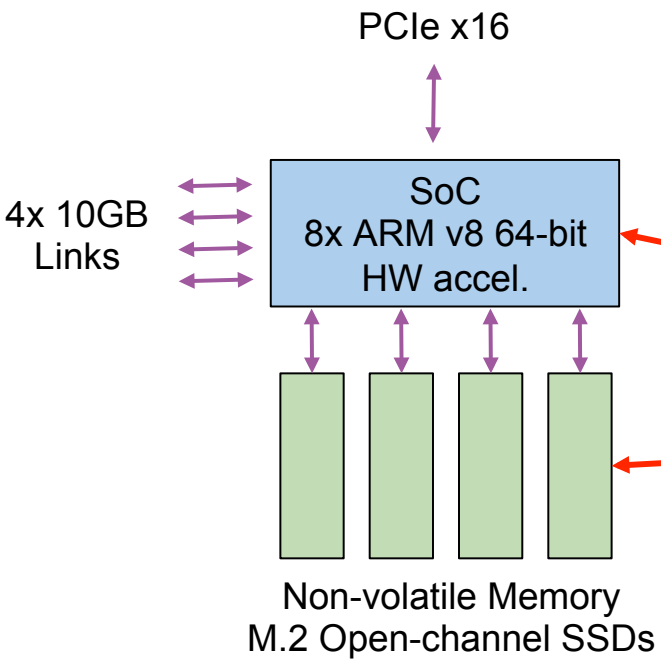
1

Basic Argument for x-Disks

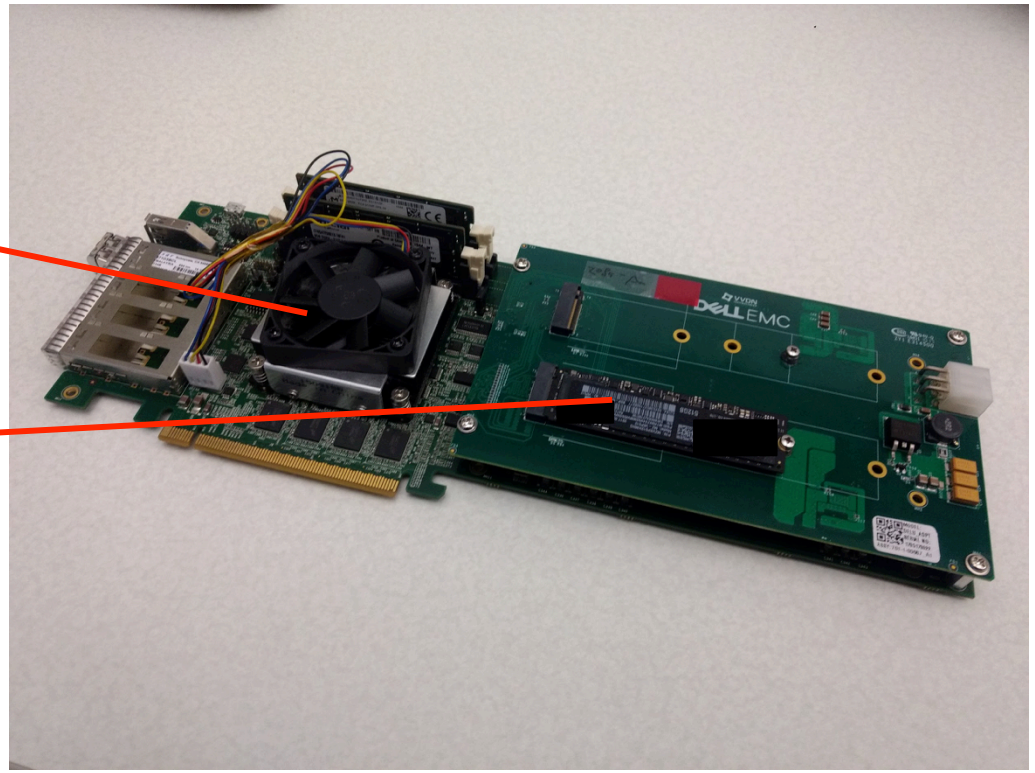
- Future disk controller is a super-computer.
  - 1 bips processor
  - 128 MB dram
  - 100 GB disk plus one arm
- Connects to SAN via high-level protocols
  - RPC, HTTP, DCOM, Kerberos, Directory Services,….
  - Commands are RPCs
  - management, security,….
  - Services file/web/db/… requests
  - Managed 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

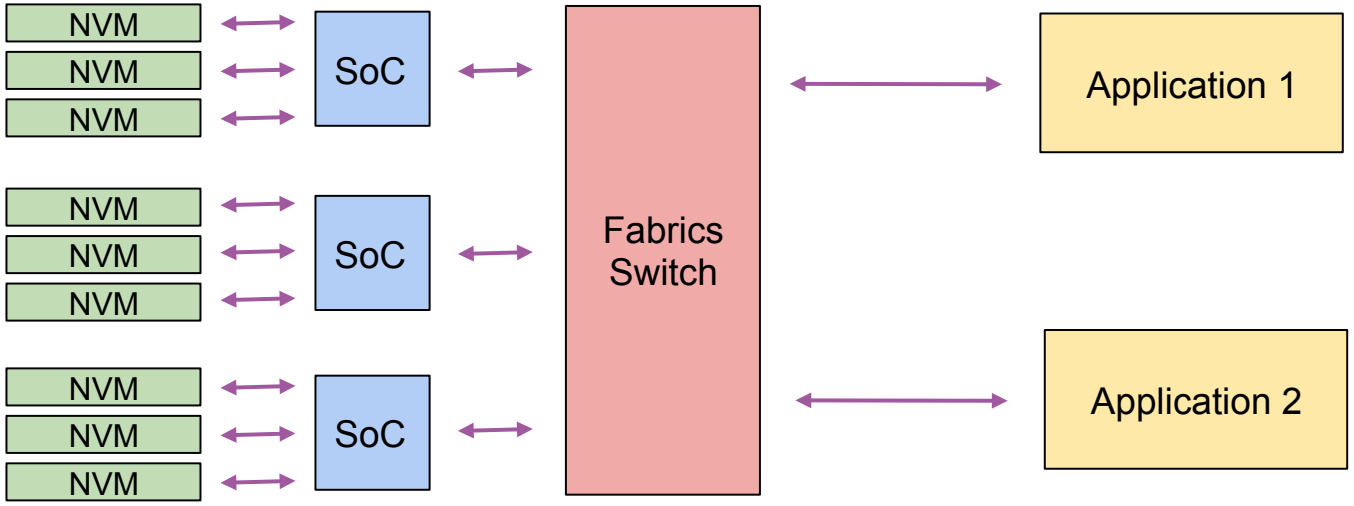
The time is now!



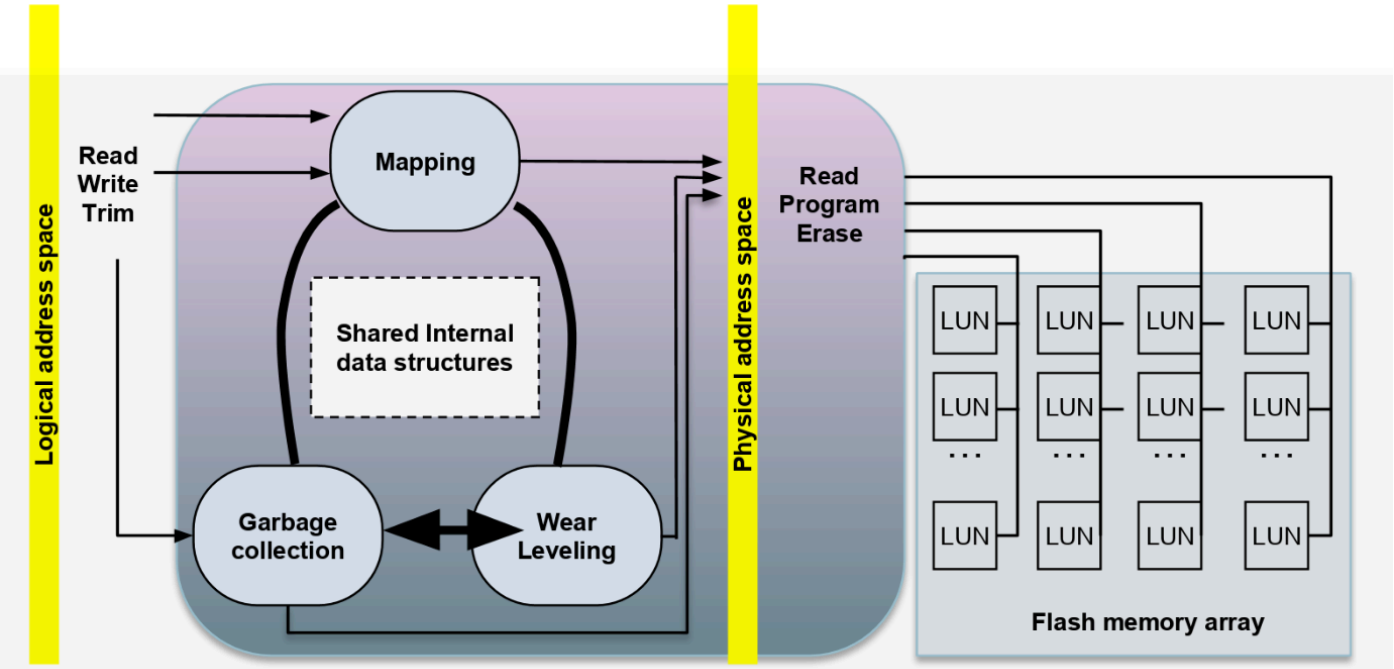
Dragon Fire Card



Tiered Storage

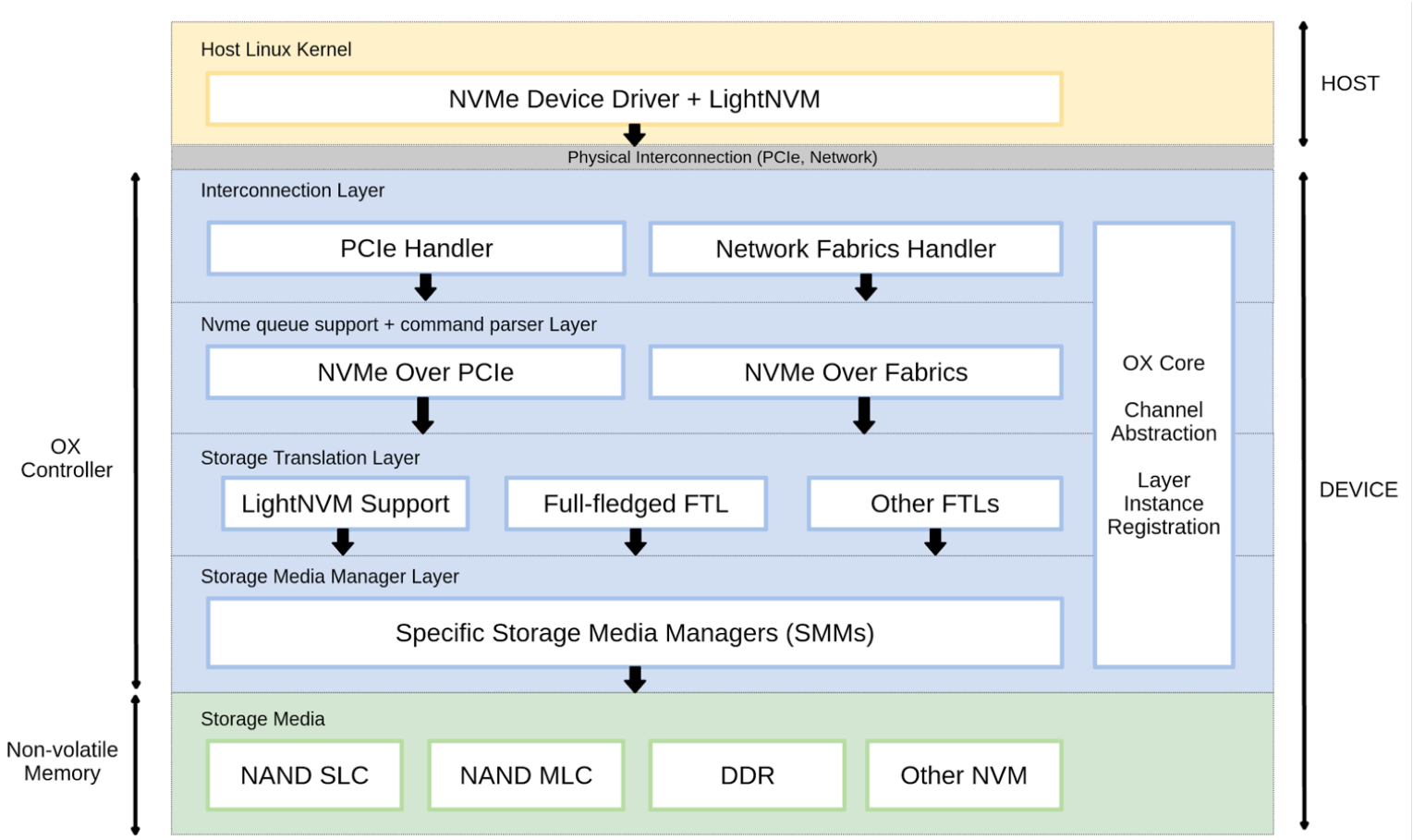


SSD Controller - Flash Management is essential

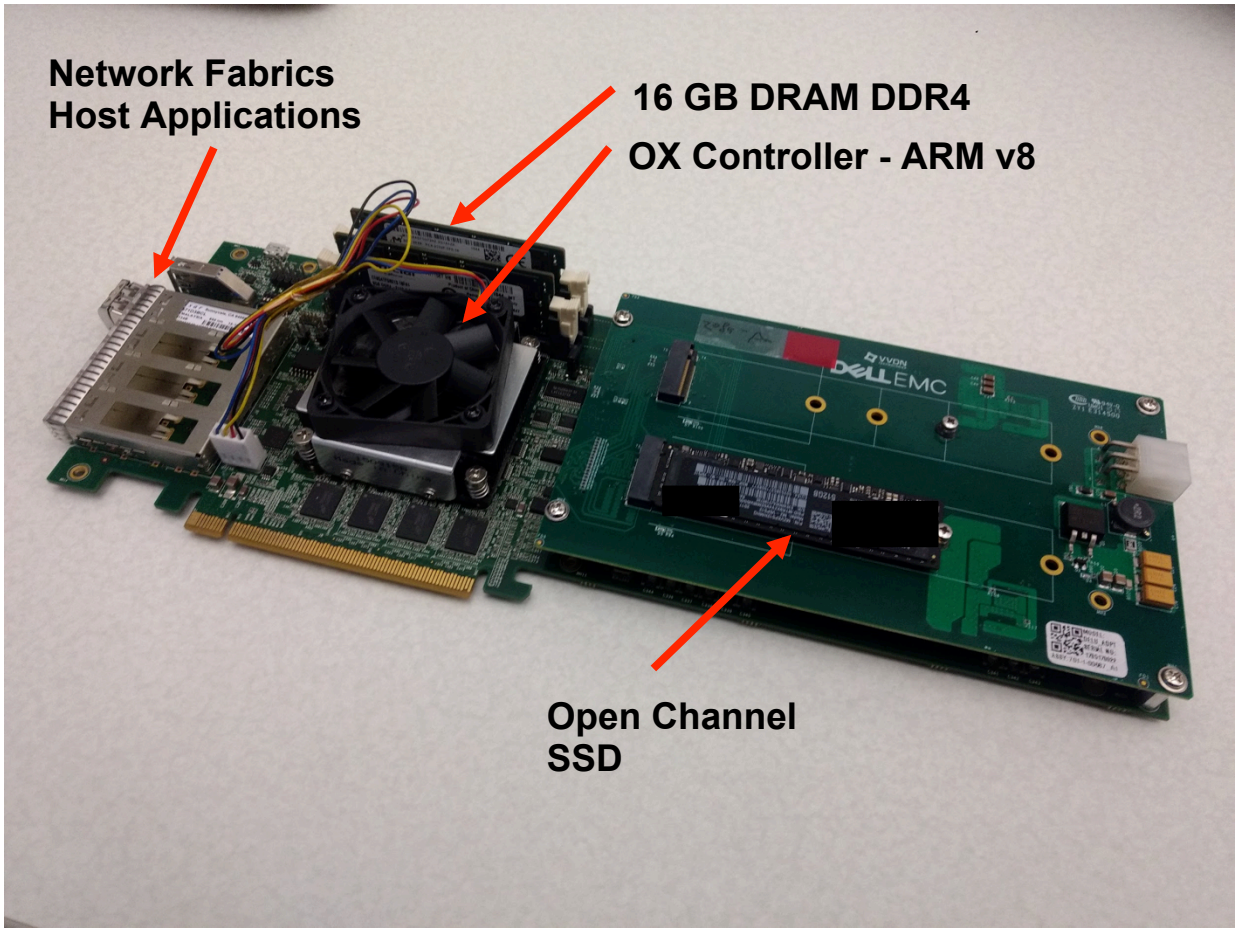


Flash Translation Layer (FTL)
Implemented on SSD controller or SSD driver

OX Controller



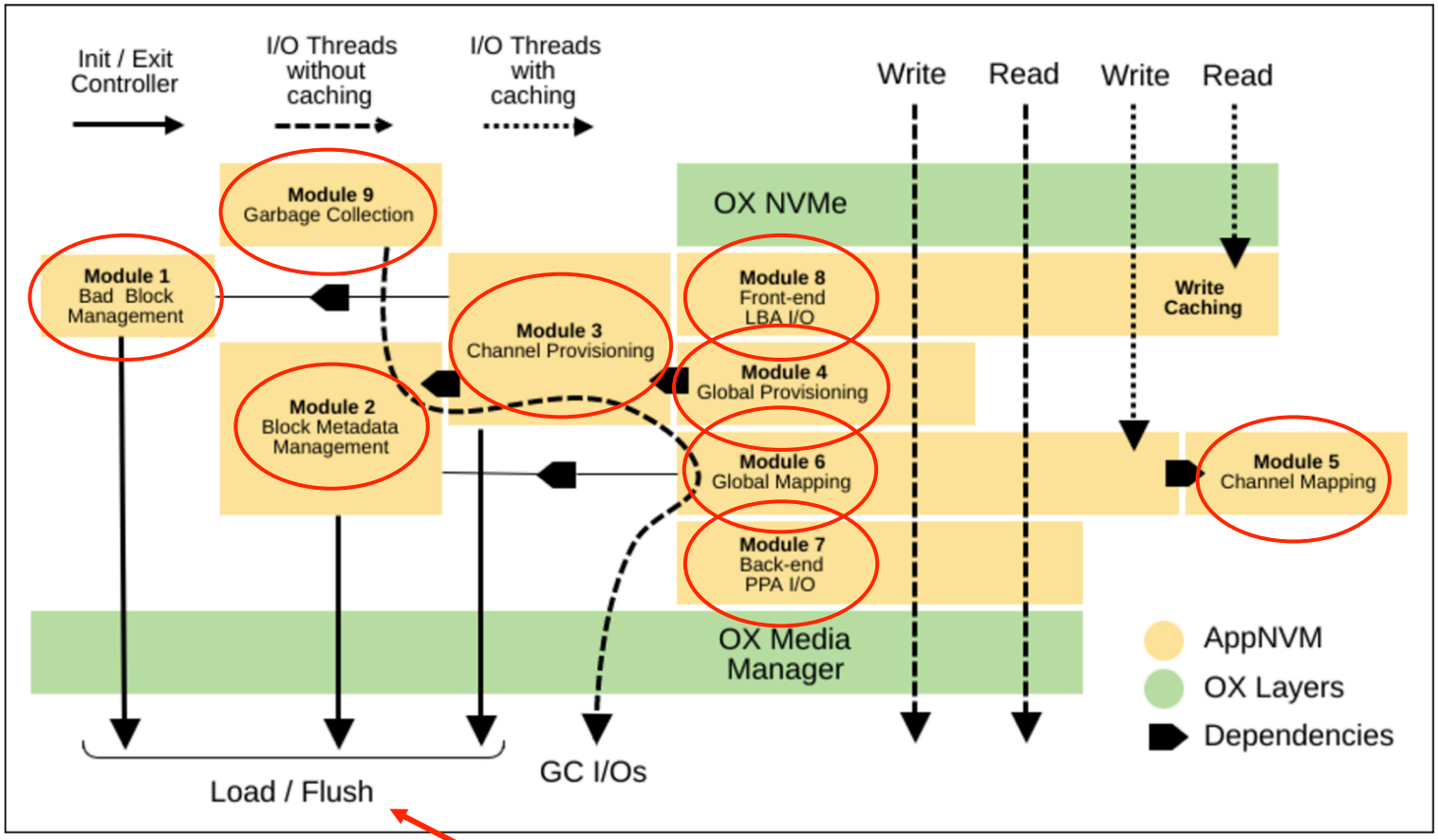
Architecture Overview



OX-App: A Framework for Application-specific FTLs

- Built on top of OX Controller, in the FTL layer;
- **FTL support:** It provides an interface to develop FTLs;
  - 11 modules with predefined interfaces;
- **Near-data processing:** It allows custom NVMe commands to be processed into the SSD controller;

OX-App FTL Modules



In progress: Log Management and Recovery

OX-Block

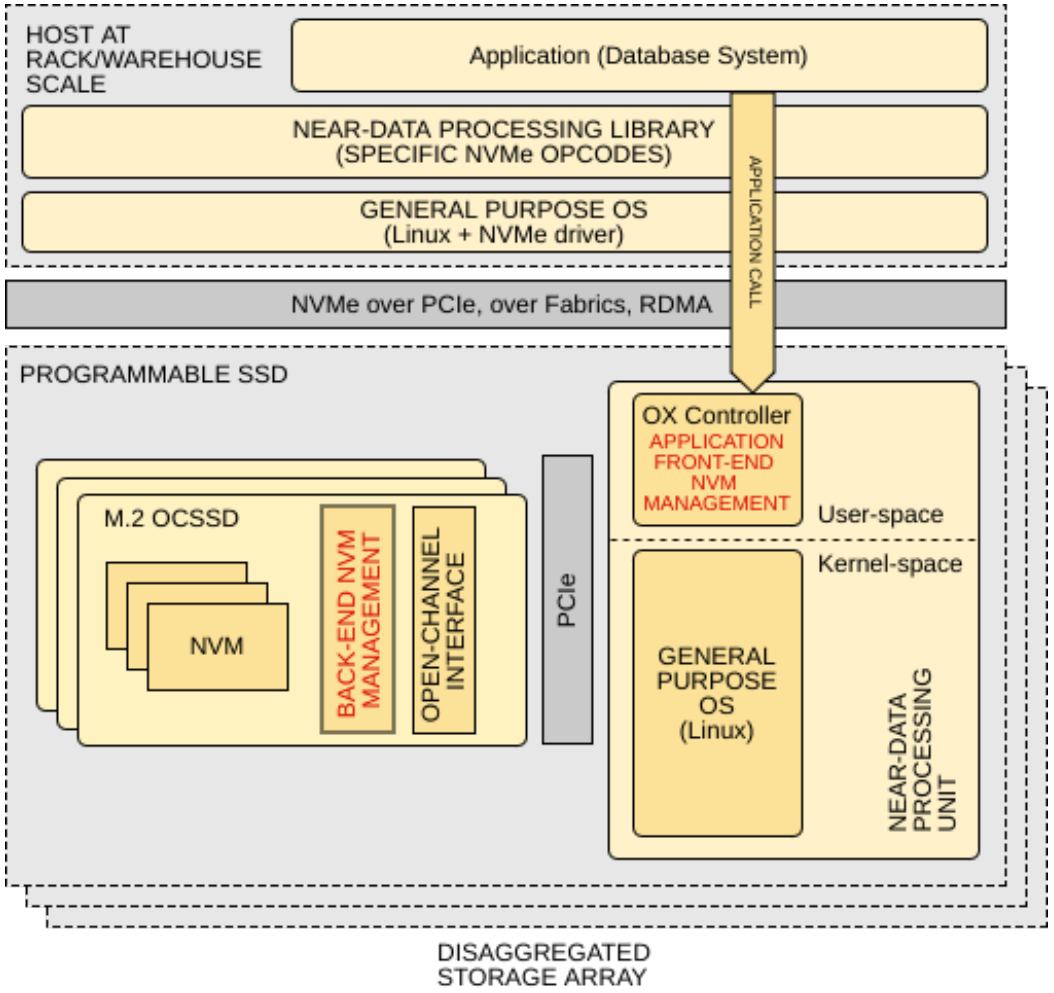
- Maintain the state and metadata of each block during the device lifetime;
- Manage a 4KB-granularity mapping table and map logical-physical addresses;
- Guarantee integrity and recovery of block metadata and mapping table after power-off;
- GC is performed per channel, where several channels might run the GC in parallel to limit the write speed;
- User writes are not allowed in channels running GC;
- Handle write and erase errors;

Near-Data Processing

Back-end NVM Mgmt:
- ECC;
- Data retention;
- RAID;
- Block metadata;

Front-end NVM Mgtm:
- Wear-leveling;
- L2P translation;
- Garbage collection;
- Write-caching;

Application functions:
- Checkpointing;
- Access Methods;
- Log Management;
- Filtering;
- Other I/O rules;



Repository and contact

OX Controller:
https://github.com/DFC-OpenSource/ox-ctrl/

OX, OX-App and OX-Block Documentation:
https://github.com/DFC-OpenSource/ox-ctrl/wiki

Ivan Luiz Picoli
ivpi@itu.dk