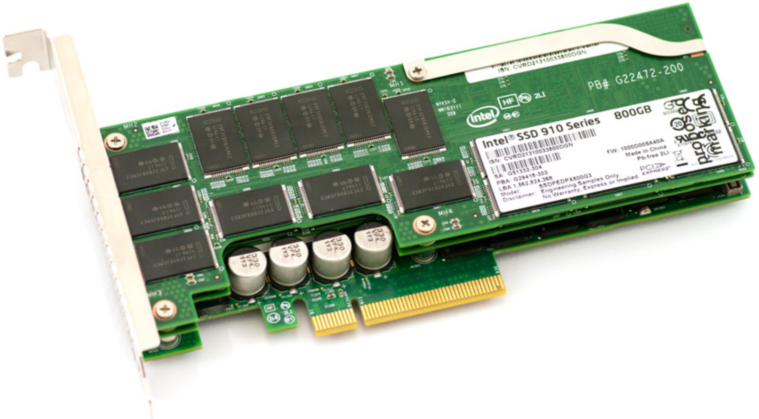
Introducing the
Non-Volatile Device Layer and LightNVM
(WIP)

Matias Bjørling, Jesper Madsen, Philippe Bonnet (IT University of Copenhagen) and Maximilian Werner Singh, G S Madhusudan (IIT Madras)

Solid State Drives

- Orders of magnitude faster than traditional hard drives
  - Thousands of IOs per second
  - Throughput measured in GB/s
  - Sub-millisecond access timings



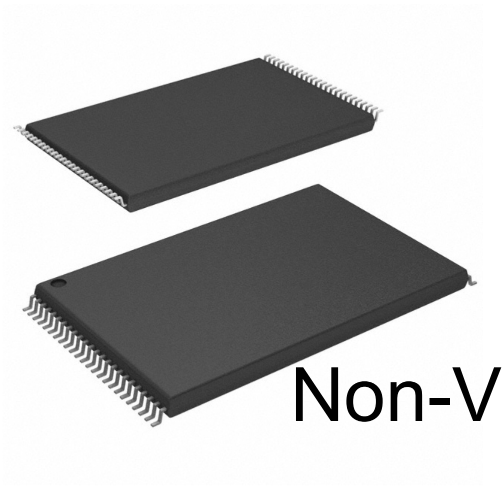
- High-performance parallel architecture
  - Tens of chips wired in parallel
  - Fast CPU and additional offload processors

Solid State Drives

- Each vendor implement their own SSD
  - No behavior model
    - Depends on history of IO's, NAND state, etc.
  - No transparency
- Narrow Interface (Read & Write)
  - Hides the read/write/erase interface of flash
  - Unpredictability
- Research requires significant hardware investments

New Indirection Layers

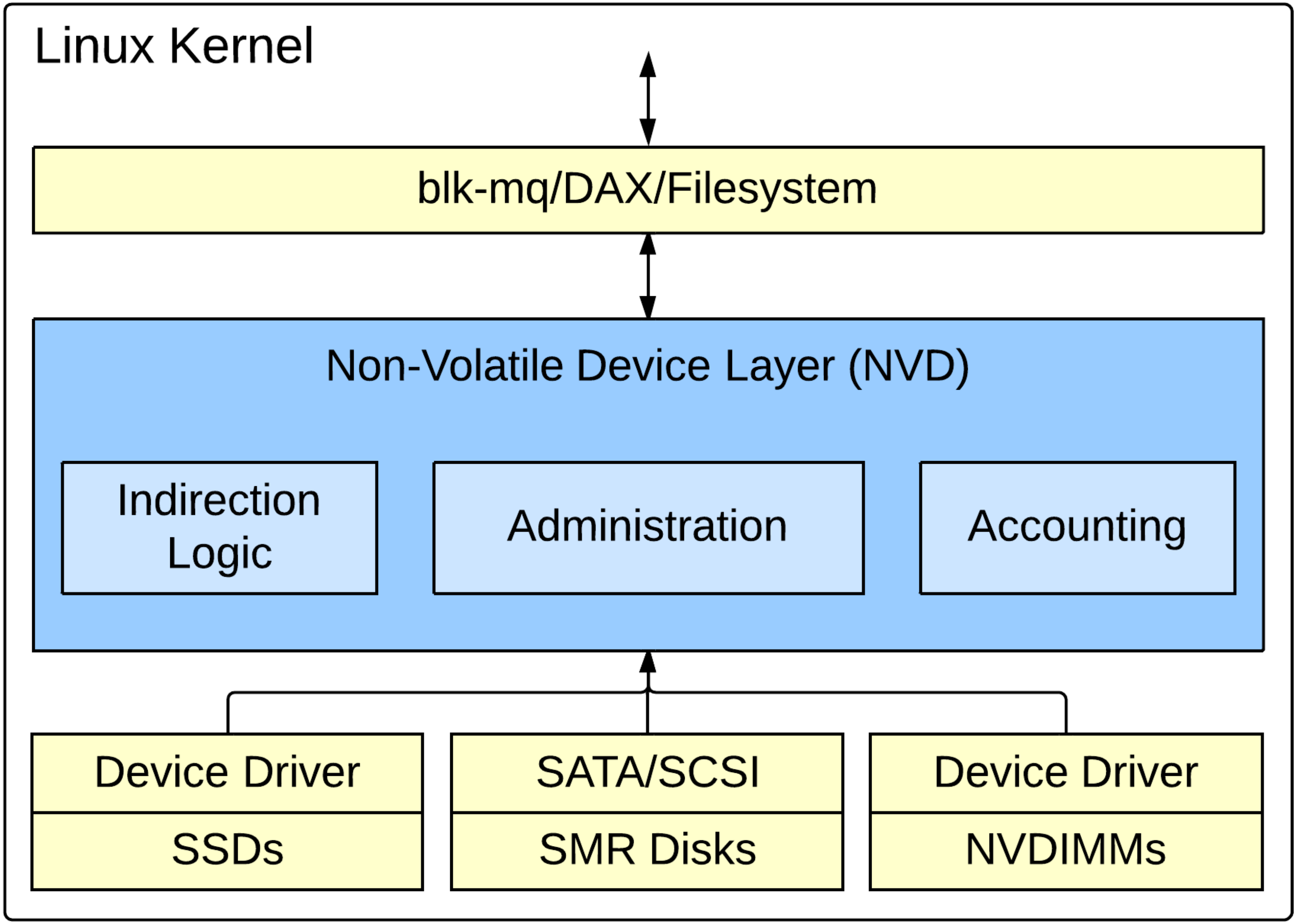
- Block and byte-addressable Non-Volatile Devices (NVD) layer
- LightNVM, a host-side “FTL” for LightNVM compatible SSDs



Non-Volatile Devices Layer

LightNVM

A home for Non-Volatile Devices logic



Linux Kernel

blk-mq/DAX/Filesystem

Non-Volatile Device Layer (NVD)

Indirection Logic

Administration

Accounting

Device Driver

SSDs

SATA/SCSI

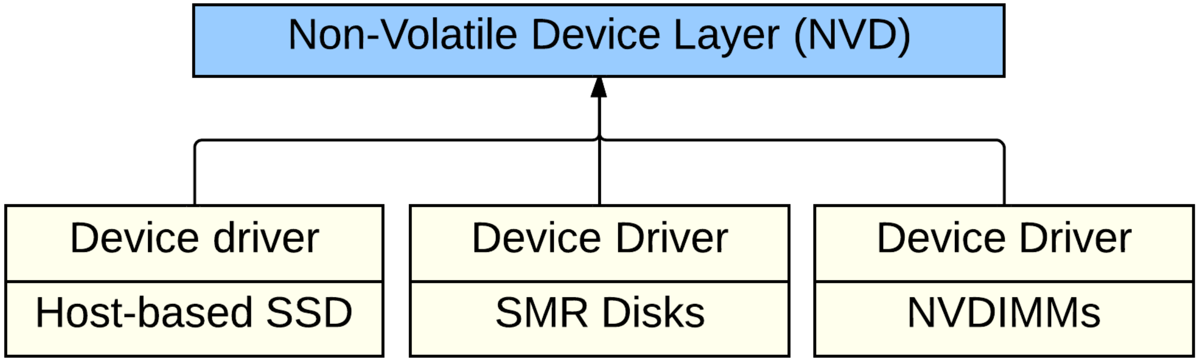
SMR Disks

Device Driver

NVDIMMs

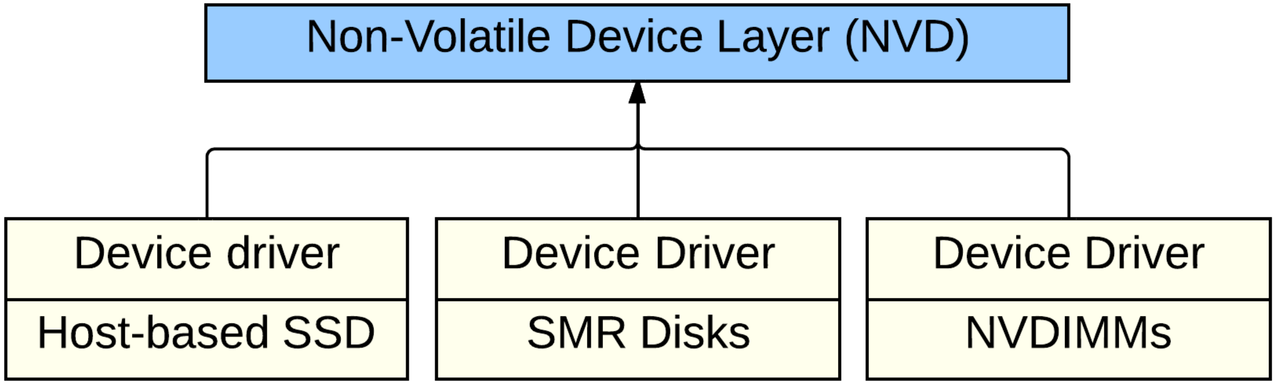
Lightweight Non-Volatile Device Layer (NVD)

- Indirection
  - Host-based Flash SSD translation layer
  - Shingled Disk Drives (SMR) translation layer
  - NVDIMM durability
- Administration
  - Formatting, etc.
  - Namespaces
- Accounting
  - Layer specific

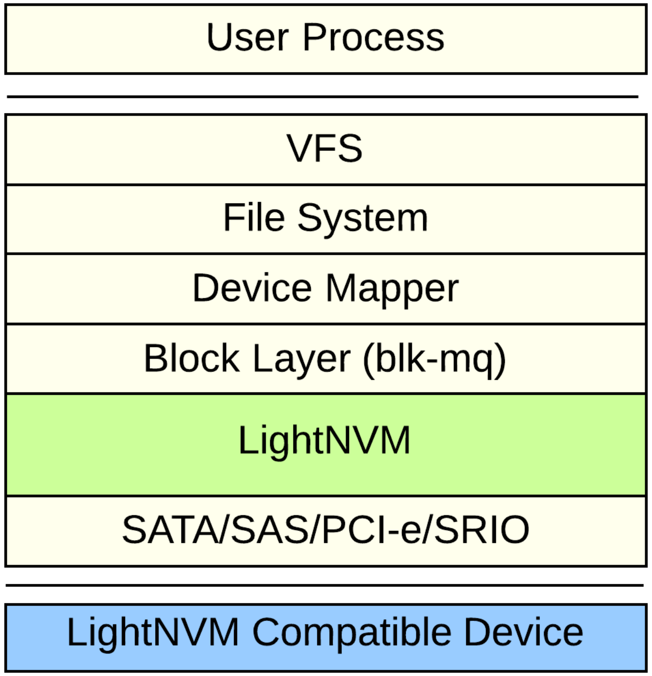


Lightweight Non-Volatile Device Layer (NVD)

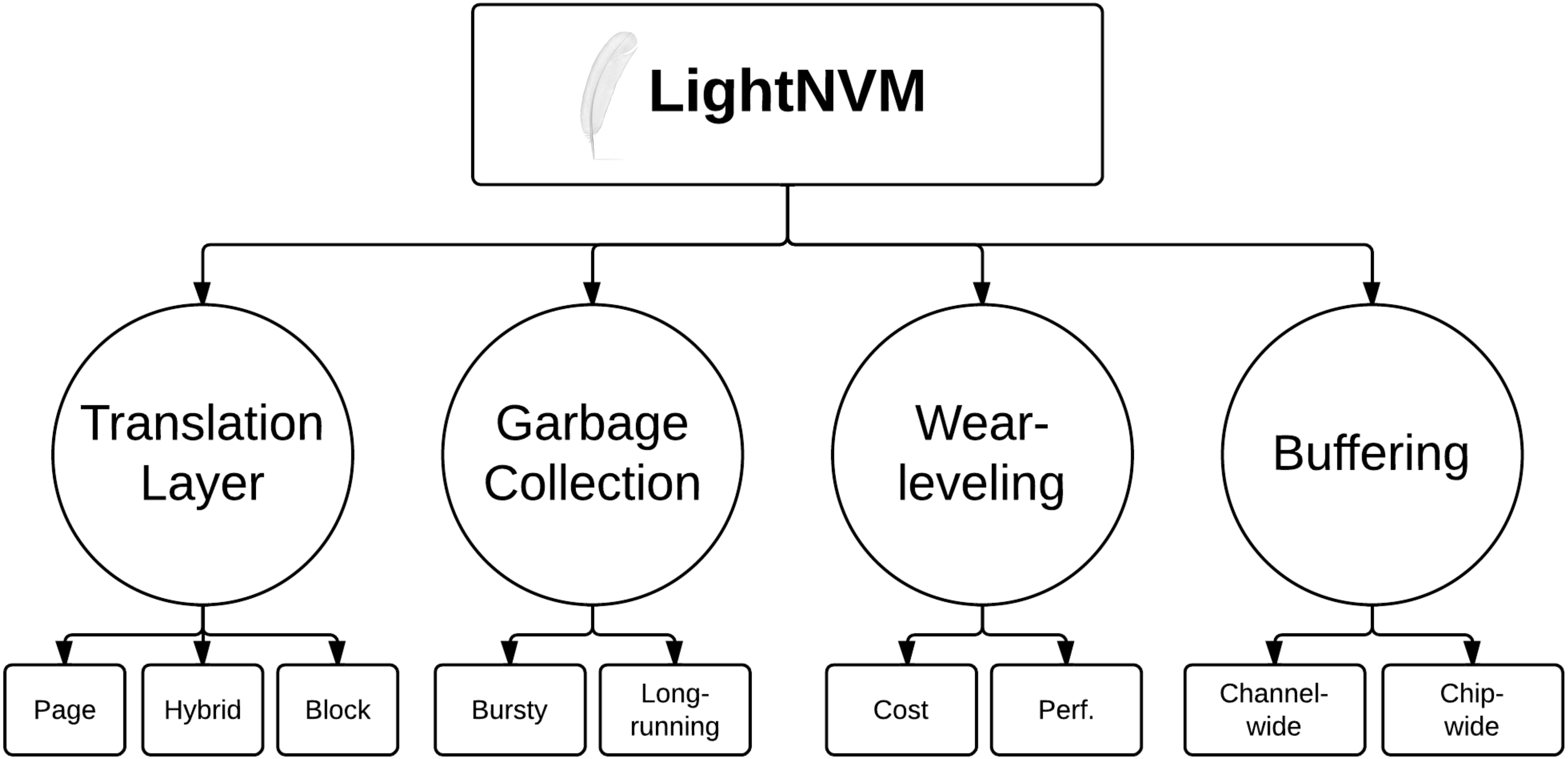
- Share common functionality
- Single registration point
- Controlled by device drivers
- Let's use LightNVM as an example



LightNVM



- A pluggable host-side “FTL”
  - Open-source
  - Predictable
  - Transparent
- Initialized on top of device drivers
- Scalable
  - >800.000 IOPS
  - 2-5us round-trip overhead (future less than 1us)


LightNVM
Translation Layer
Garbage Collection
Wear-leveling
Buffering
Page
Hybrid
Block
Bursty
Long-running
Cost
Perf.
Channel-wide
Chip-wide

Hybrid Storage Design

- FTL responsibilities is be shared between host and device. E.g. for flash controller

Responsibilities	**Host**	**On-disk**
Log. to Phy. Translation.	x	
Durability management. Disk maintain internal trans. mappings		x
Garbage collection of physical NV blocks	x	
Wear-leveling	x	x
Bad block management	x	x
Transaction/Atomic IO management	x	
Key-value IO	x	

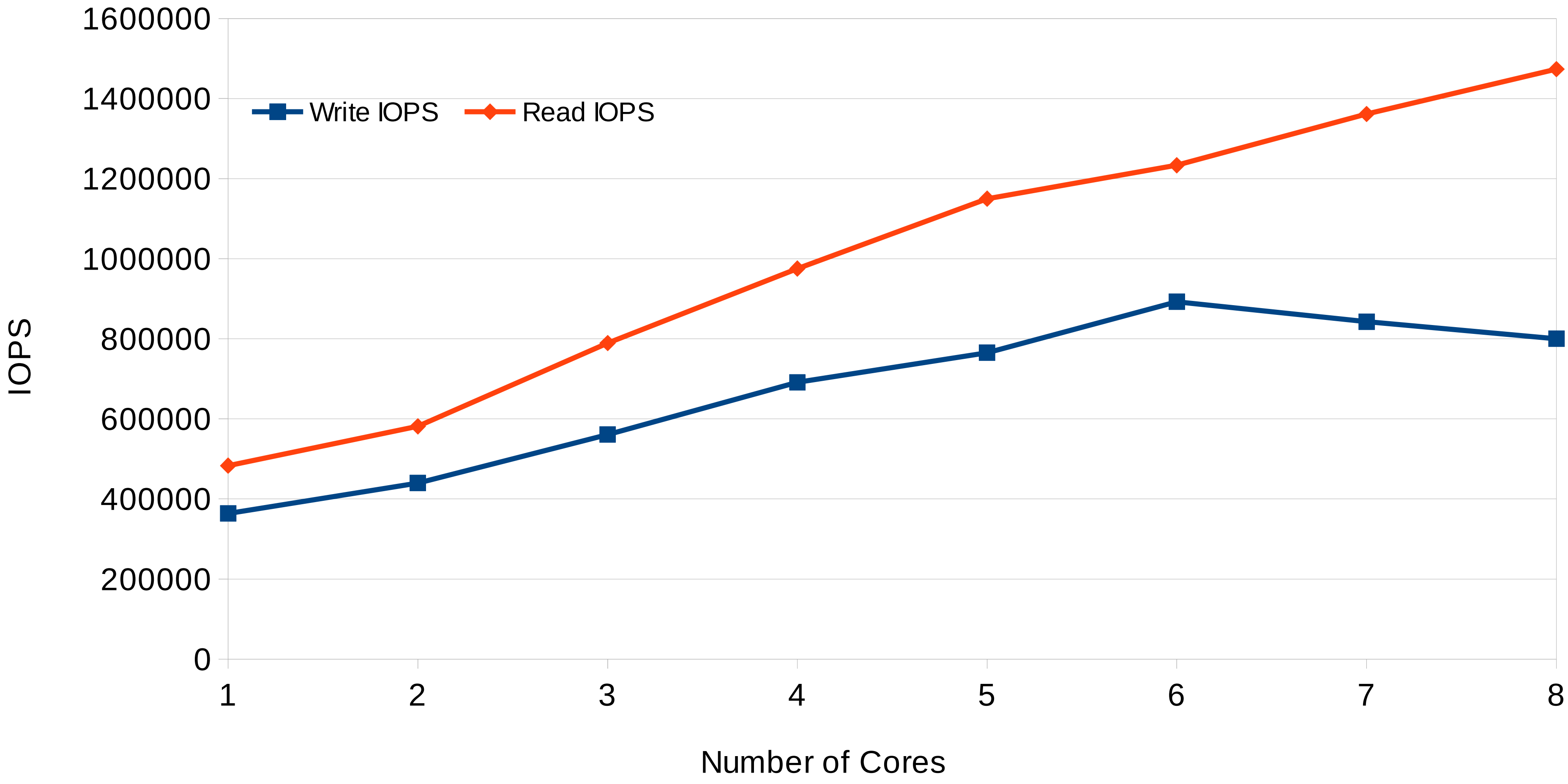
LightNVM and Hardware

- Offload critical sections
  - Non-volatile memory ECC
  - On-board capacitors
  - NV controller, etc.
- Disk exposes drive information to host
  - Number of channels, throughput, page size, channel queue depth, etc.
  - NVM type (Flash, PCM, etc.)
  - Storage interfaces, offload capabilities, etc.
- Disk expose its NV as a linear address space.

Evaluation Methodology

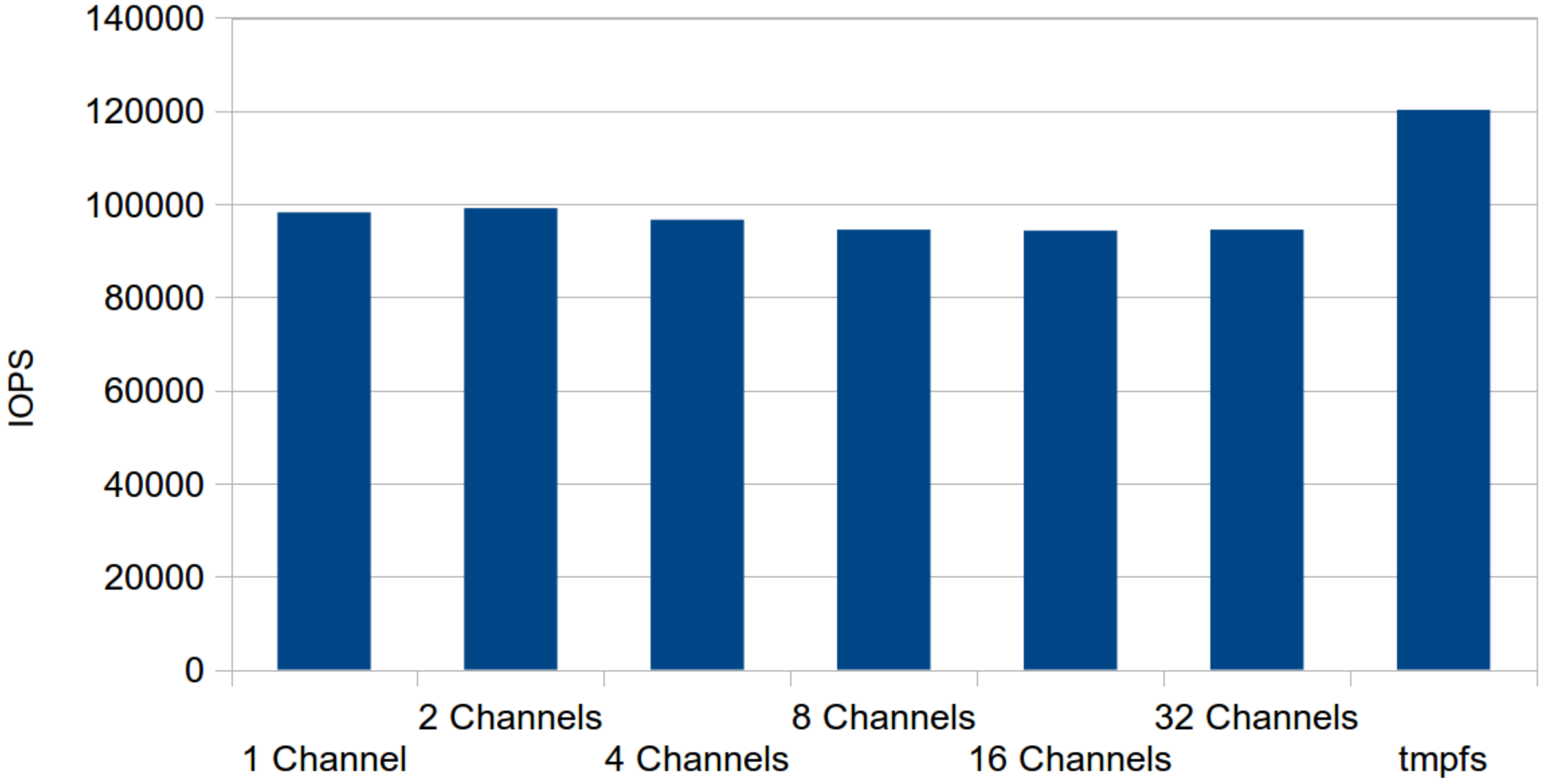
- 2CPU, Intel E5-2643, 128GB, Linux kernel 3.13
- 4K IOs
- Fio
- LightNVM configured to page-based, cost-based, and lazy GC.
- Evaluate with respect to
  - Scalability
  - Overhead
  - Timing accuracy

LightNVM – Max Performance



4K, null_blk (mq), round-robin across 4 channels.

LightNVM: Overhead Comparison



- 1QD, 4K, Random Writes. Round-robin across channels, 8GB tmpfs
- 18-21% overhead compared to tmpfs

Conclusion

- A common layer for non-volatile device logic
- LightNVM: A pluggable FTL
  - Scalable
  - Modularity: FTL, GC, wear-leveling, etc.
  - Predictability and transparency
- Patches being prepared for upstream Linux kernel