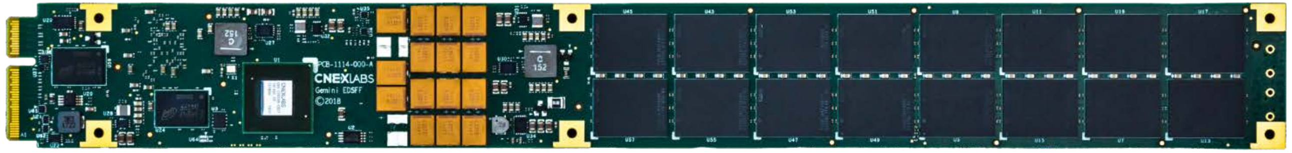
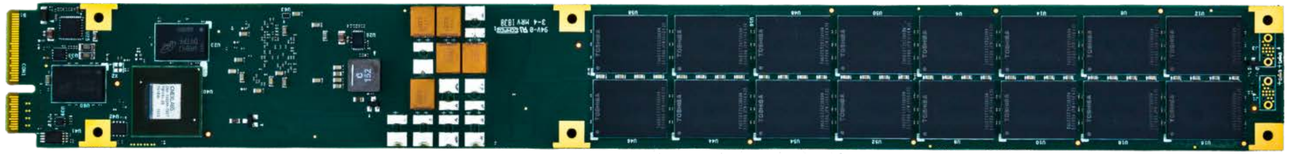
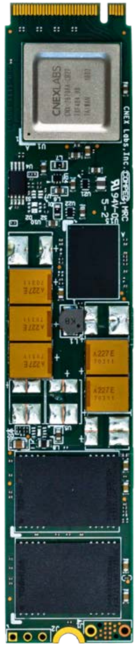
liblightnvm
The Open-Channel SSD User-Space Library

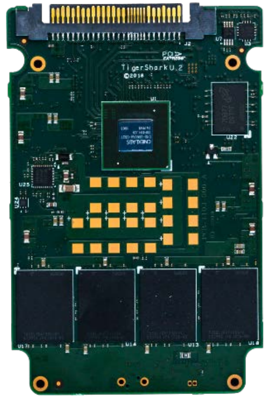
Simon A. F. Lund
CNEX Labs

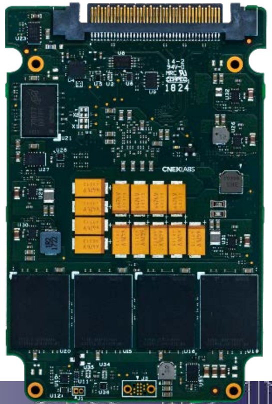
Open-Channel SSD



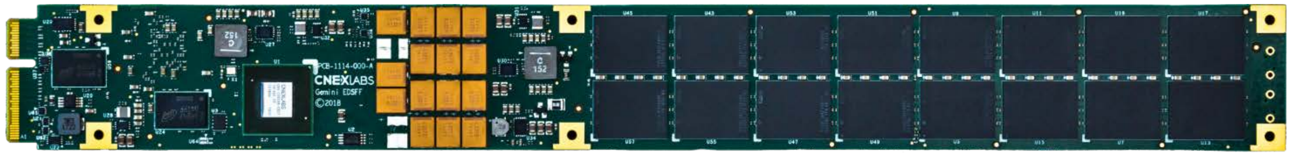
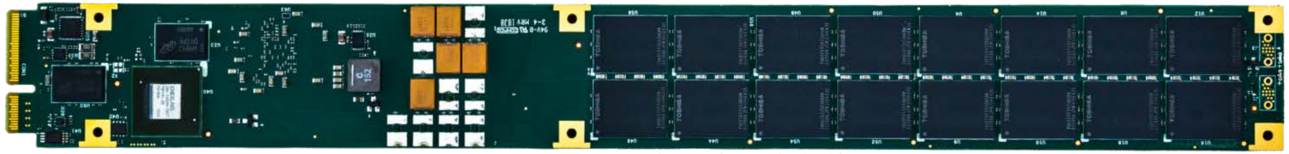
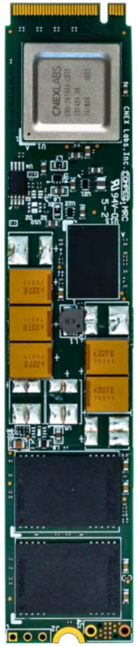
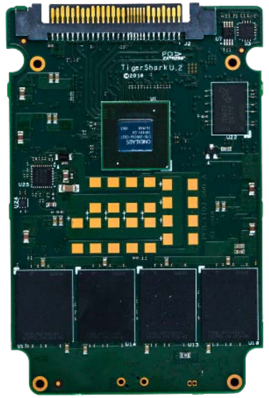


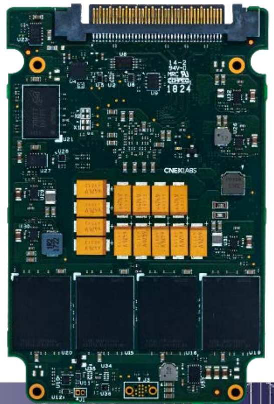






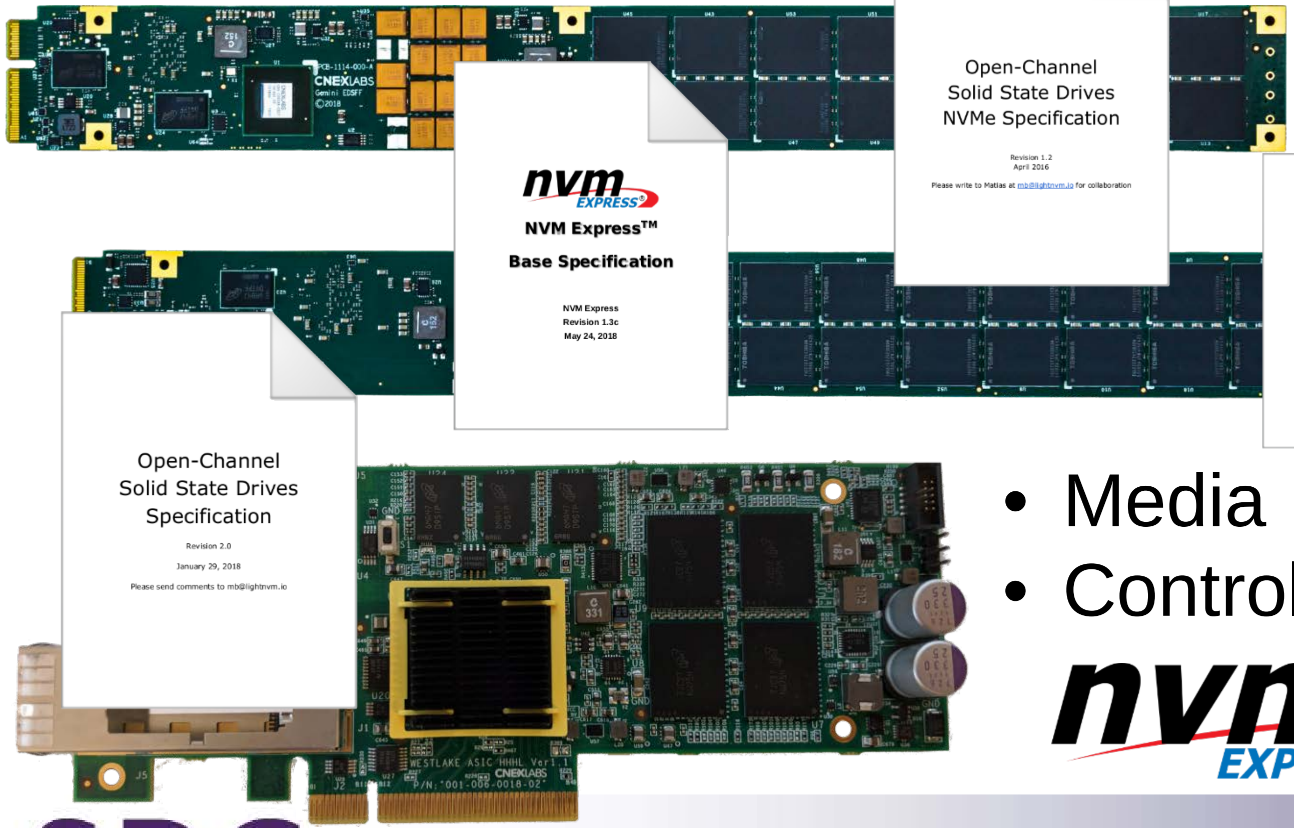
Open-Channel SSD



- Media
- Controller

Open-Channel SSD



NVM Express™

Base Specification

NVM Express
Revision 1.3c
May 24, 2018

Open-Channel
Solid State Drives
NVMe Specification

Revision 1.2
April 2016

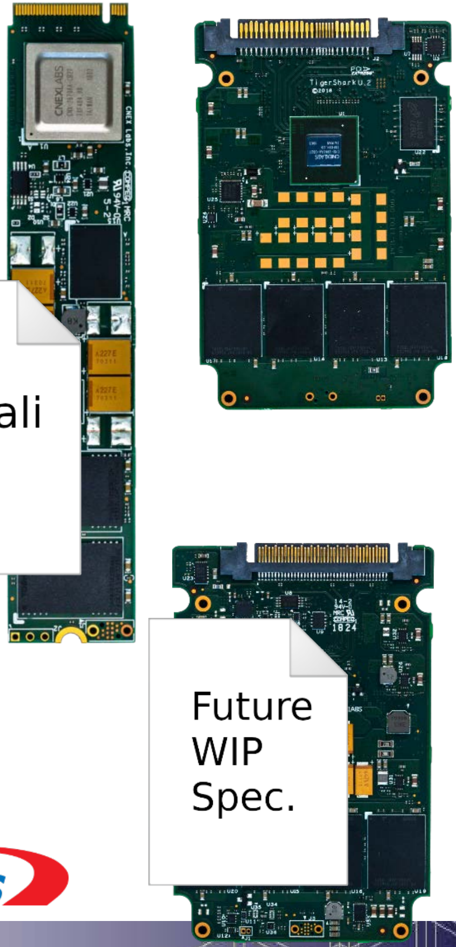
Please write to Matias at mb@lightnvm.io for collaboration

Open-Channel
Solid State Drives
Specification

Revision 2.0

January 29, 2018

Please send comments to mb@lightnvm.io

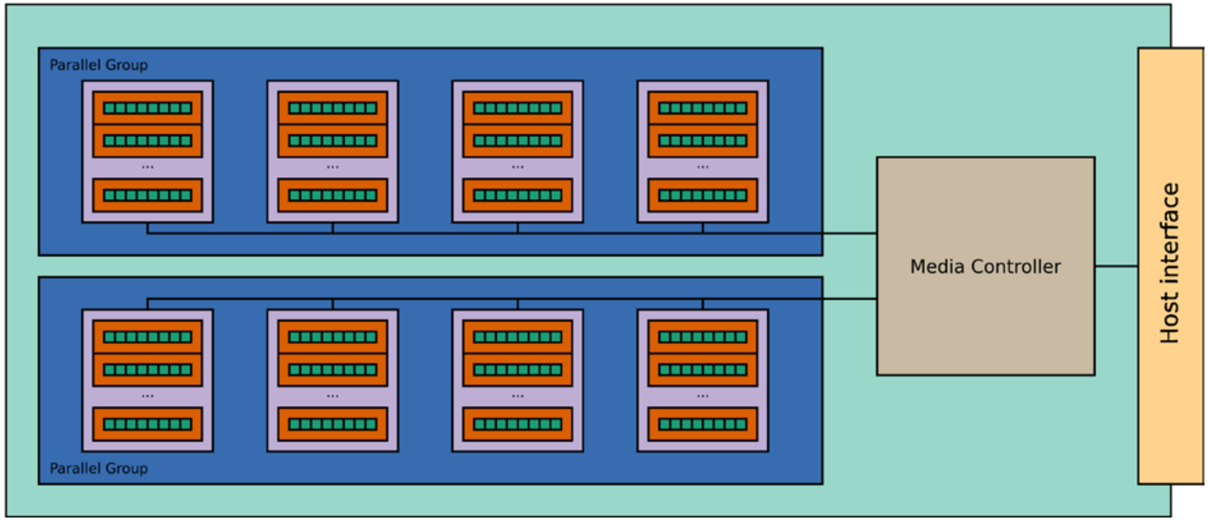


Denali

Future
WIP
Spec.

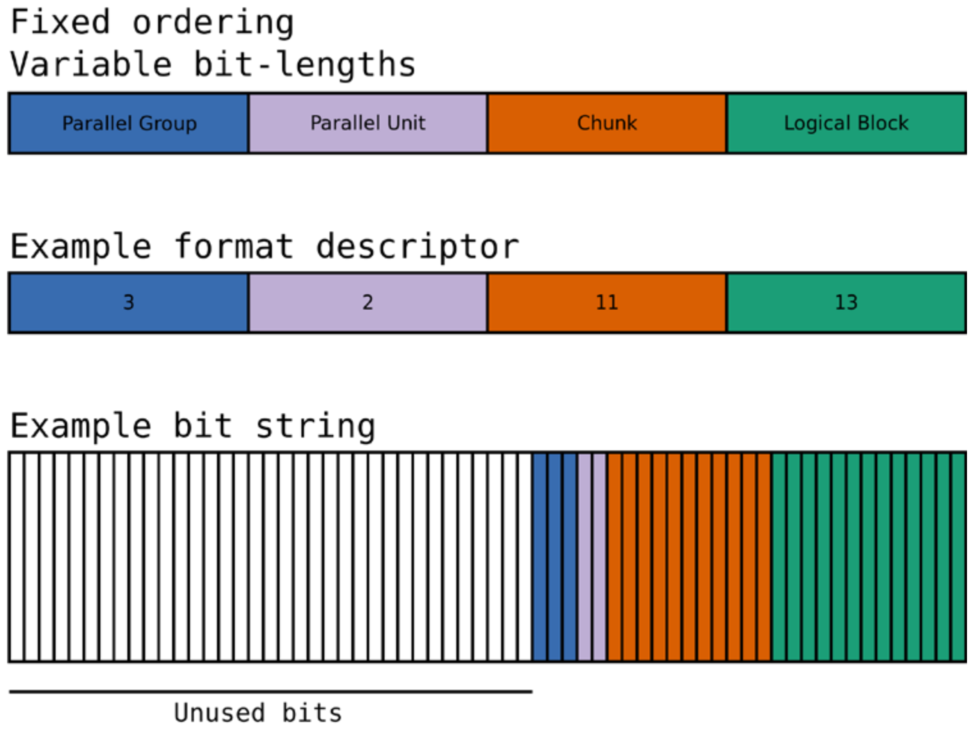
- Media
- Controller

Open-Channel SSD: Drive Model

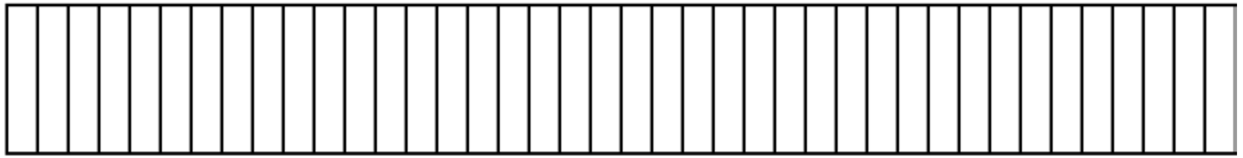


- Logical Block
- Chunk
- Parallel Unit
- Group of Parallel Units

Open-Channel SSD: Addressing

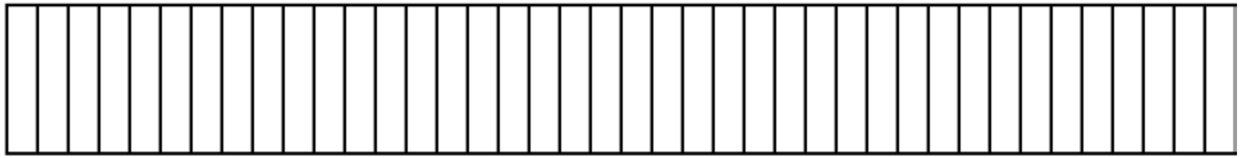


Open-Channel SSD: Chunk



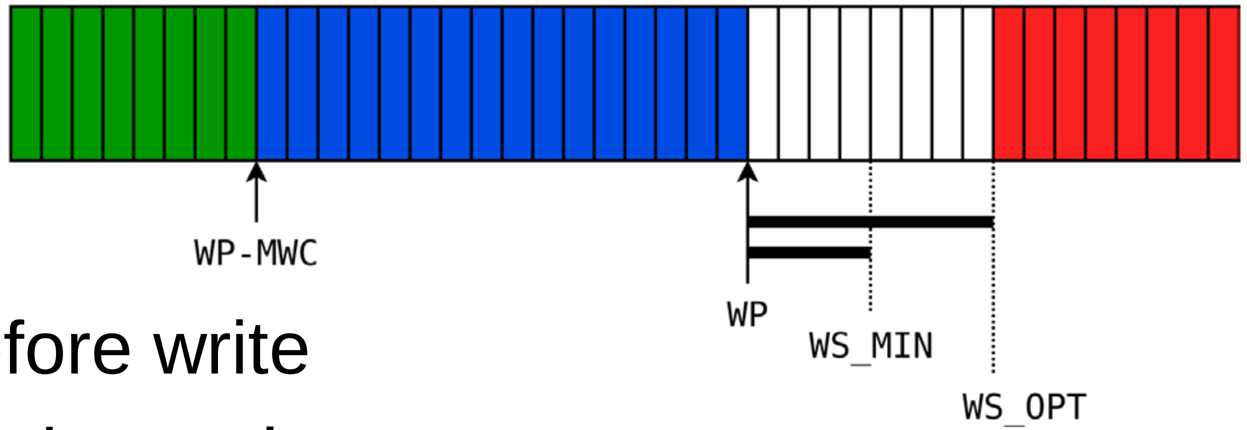
- Contains Minimal Addressable Units
  - Each unit has size in bytes e.g. 4096
  - **Nomenclature**: logical block, sector, address
- Addresses within a chunk are contiguous
  - E.g. Address range [0, naddrs-1]

Open-Channel SSD: Chunk IO Constraints



1. Erase before write
2. Write contiguously
3. Write **WS_MIN** multiple # of addresses pr. cmd
4. Read no further than address **WP – MWC**

Open-Channel SSD: Chunk IO Constraints

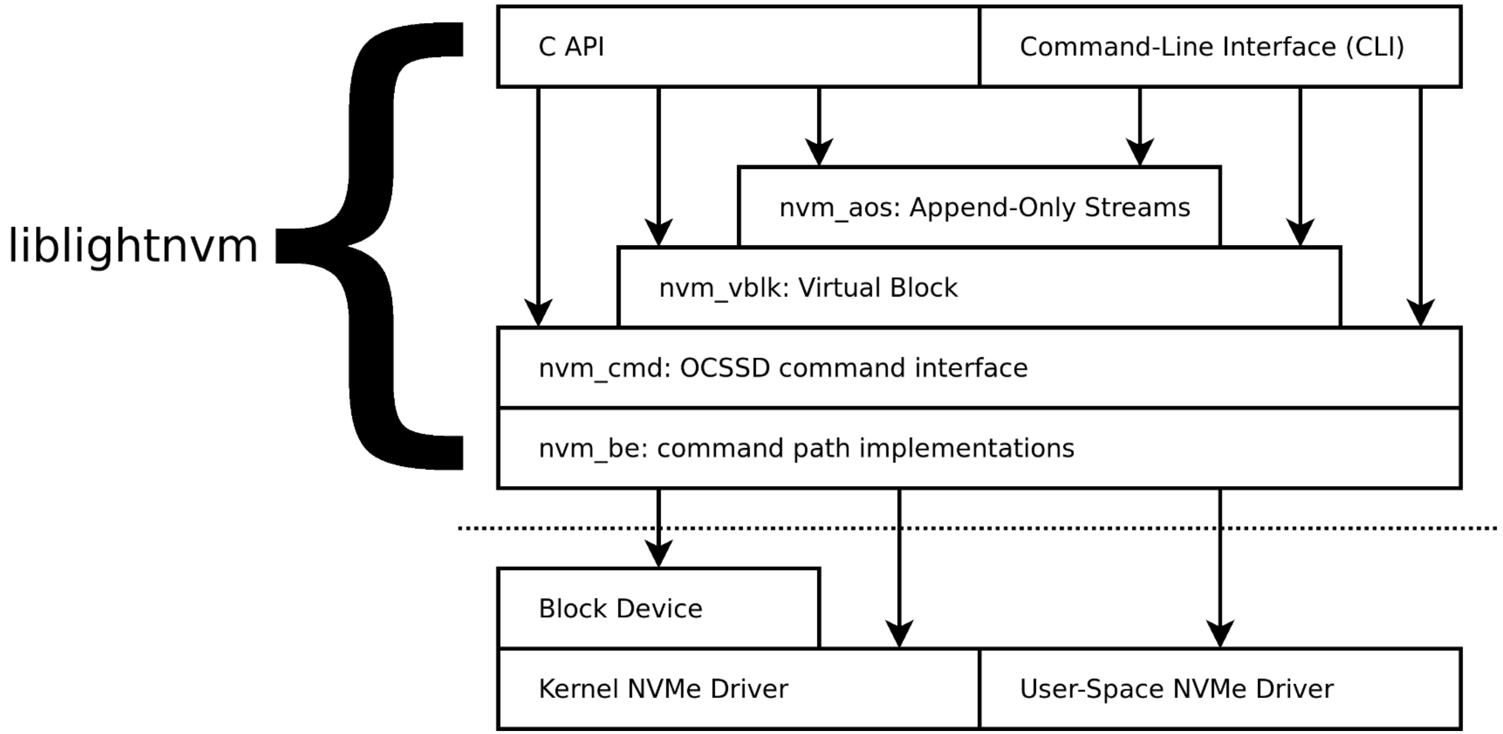


1. Erase before write
2. Write contiguously
3. Write **WS_MIN** multiple # of addresses pr. cmd
4. Read no further than address **WP – MWC**

liblightnvm: Related

- nvme-cli
  - https://github.com/linux-nvme/nvme-cli
- SPDK
  - https://github.com/spdk/spdk
- libnvme
  - https://github.com/hgst/libnvme

liblightnvm: overview



liblightnvm
C API
Command-Line Interface (CLI)
nvm_aos: Append-Only Streams
nvm_vblk: Virtual Block
nvm_cmd: OCSSD command interface
nvm_be: command path implementations
Block Device
Kernel NVMe Driver
User-Space NVMe Driver

liblightnvm: Usages

- Pure User-Space driven IO
  - Dedicated application-integration

liblightnvm: Usages

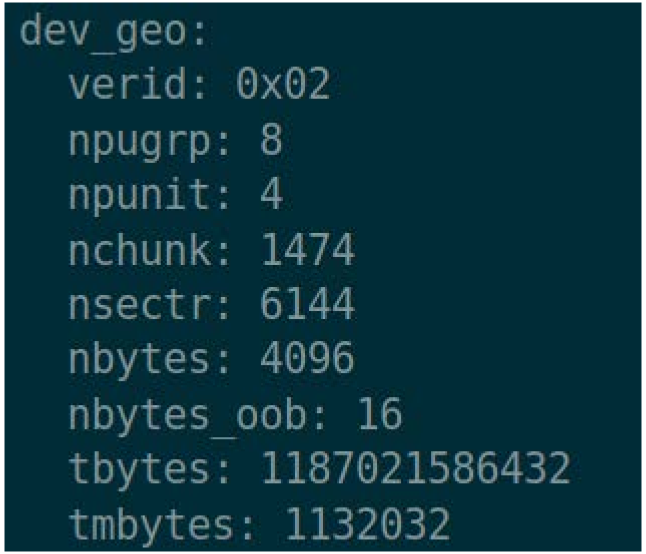
- Pure User-Space driven IO
  - Dedicated application-integration
- Hybrid Kernel and User-Space
  - PBLK + User-Space for application IO

liblightnvm: Usages

- Pure User-Space driven IO
  - Dedicated application-integration
- Hybrid Kernel and User-Space
  - PBLK + User-Space for application IO
- Tooling for Open-Channel SSD kernel services
  - File-system and FTL management and maintenance

liblightnvm: OCSSD Device Attributes

- nvm_cmd_idfy
- struct nvm_dev
  - nvm_dev_openf(ident, flags)
  - nvm_dev_get_wsopt(dev)
  - nvm_dev_get_geo(dev)
- struct nvm_geo



```
dev_geo:
  verid: 0x02
  npugrp: 8
  npunit: 4
  nchunk: 1474
  nsectr: 6144
  nbytes: 4096
  nbytes_oob: 16
  tbytes: 1187021586432
  tmbytes: 1132032
```

liblightnvm: OCSSD Device Attributes

DEMO: nvm_cmd_idfy

https://asciinema.org/a/WJJMxRKsgAq0GIbWlfhIAGZDl

DEMO: nvm_dev and nvm_geo

https://asciinema.org/a/DCr9ak5VdnClpJjjvxjKQQNIg

liblightnvm: OCSSD Media State

- nvm_cmd_rprt
  - Retrieve chunk descriptors for all chunks
  - Retrieve chunk descriptors for all chunks in a parallel unit

liblightnvm: OCSSD Media State

- nvm_cmd_rprt
  - Retrieve chunk descriptors for all chunks
  - Retrieve chunk descriptors for all chunks in a parallel unit
- nvm_cmd_rprt_arbs
  - Provides **N** arbitrary chunk addresses in the requested state in distinct parallel units

liblightnvm: OCSSD Media State

DEMO

https://asciinema.org/a/XGppr2Yjdc90fsoyLCPVCx0sc

liblightnvm: OCSSD Addressing

- struct nvm_addr
  - Geometric accessors
  - Address translation is handled by the library
  - User does not need to know about the LBAF

liblightnvm: OCSSD Addressing

- struct nvm_addr
  - Geometric accessors
  - Address translation is handled by the library
  - User does not need to know about the LBAF
- nvm_dev_gen2dev
- nvm_dev_dev2gen

liblightnvm: OCSSD Addressing

DEMO

https://asciinema.org/a/tFwlWRMq0DwwvK5oq5bCuBpty

liblightnvm: OCSSD IO Commands

- nvm_cmd_erase – Vector Reset / DSM deallocate
- nvm_cmd_write – Vector / Scalar Write
- nvm_cmd_read – Vector / Scalar Read
- nvm_cmd_copy – Vector Copy

liblightnvm: OCSSD IO Commands

DEMO

https://asciinema.org/a/iq8hoPAYpXSqY5Jgq67SrbA1Q

liblightnvm: OCSSD IO Command Options

- IO Addressing Mode (SCALAR or VECTOR)

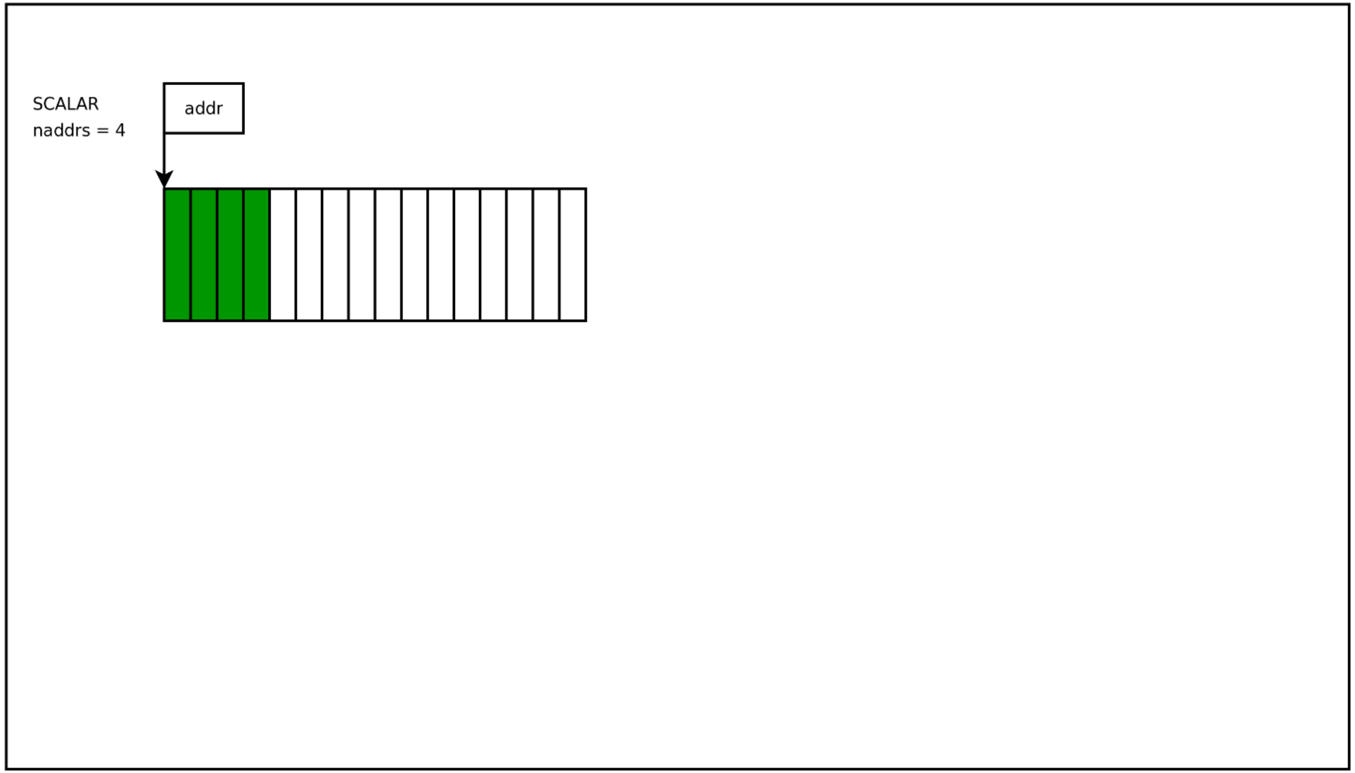
liblightnvm: OCSSD IO Command Options

- IO Addressing Mode (SCALAR or VECTOR)
- NVM_CMD_SCALAR
  - erase / write / read mapped to **NVMe** spec. defined opcodes

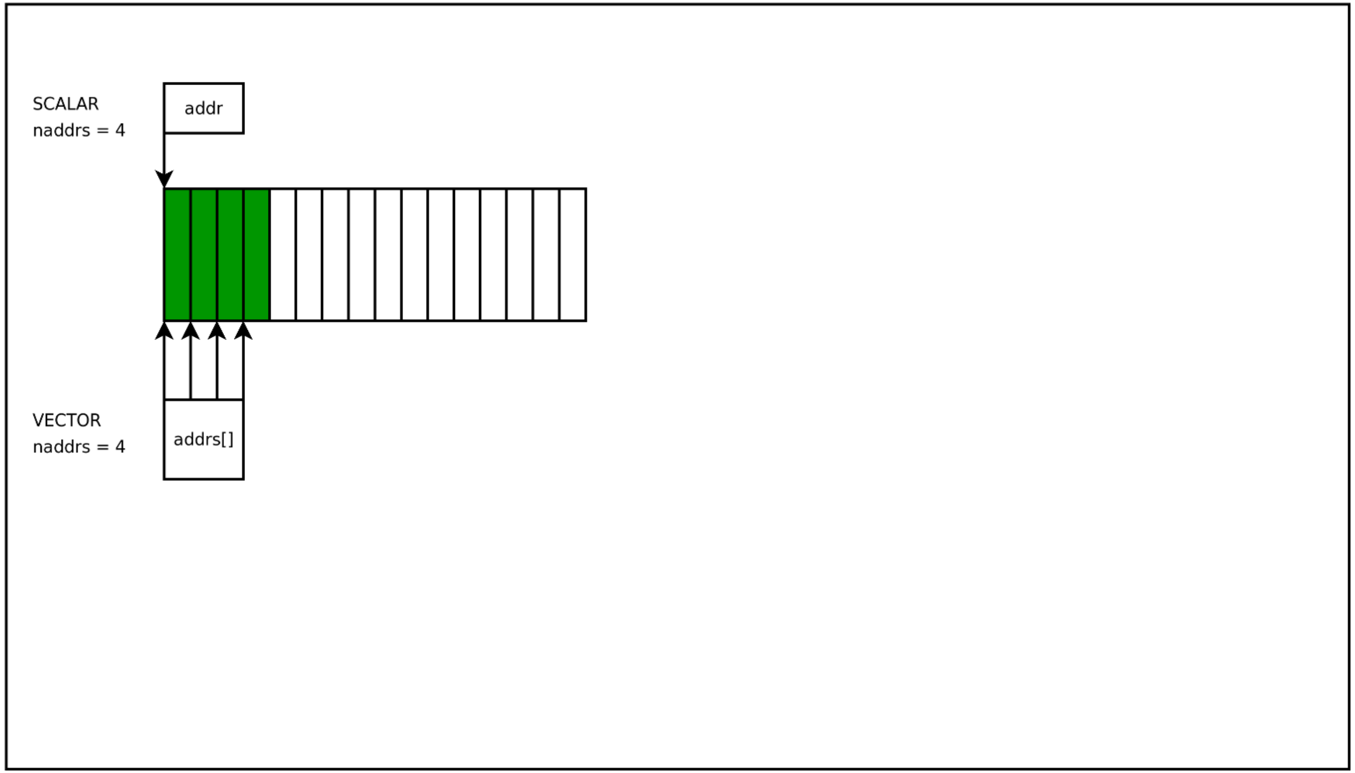
liblightnvm: OCSSD IO Command Options

- IO Addressing Mode (SCALAR or VECTOR)
- NVM_CMD_SCALAR
  - erase / write / read mapped to **NVMe** spec. defined opcodes
- NVM_CMD_VECTOR
  - erase / write / read mapped to **OCSSD** spec. defined VECTOR opcodes

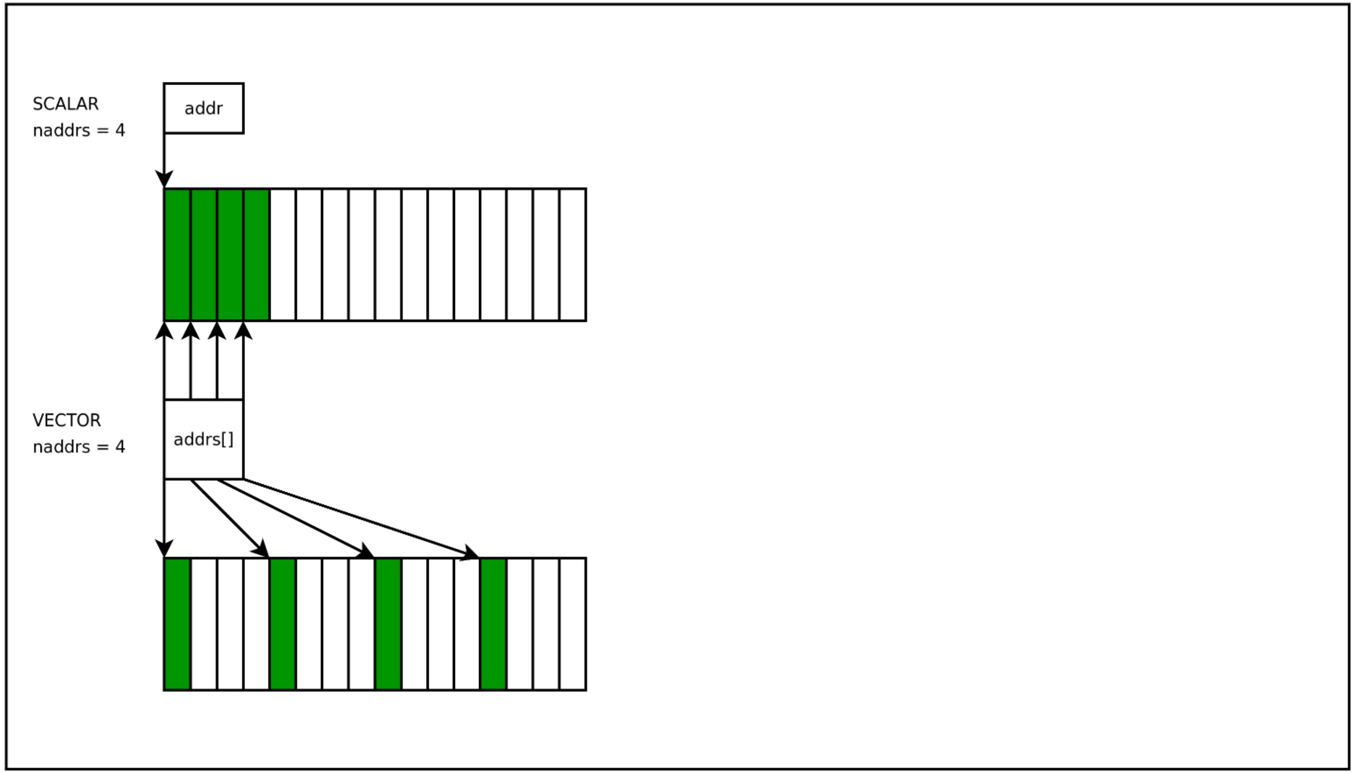
liblightnvm: OCSSD IO Command Options



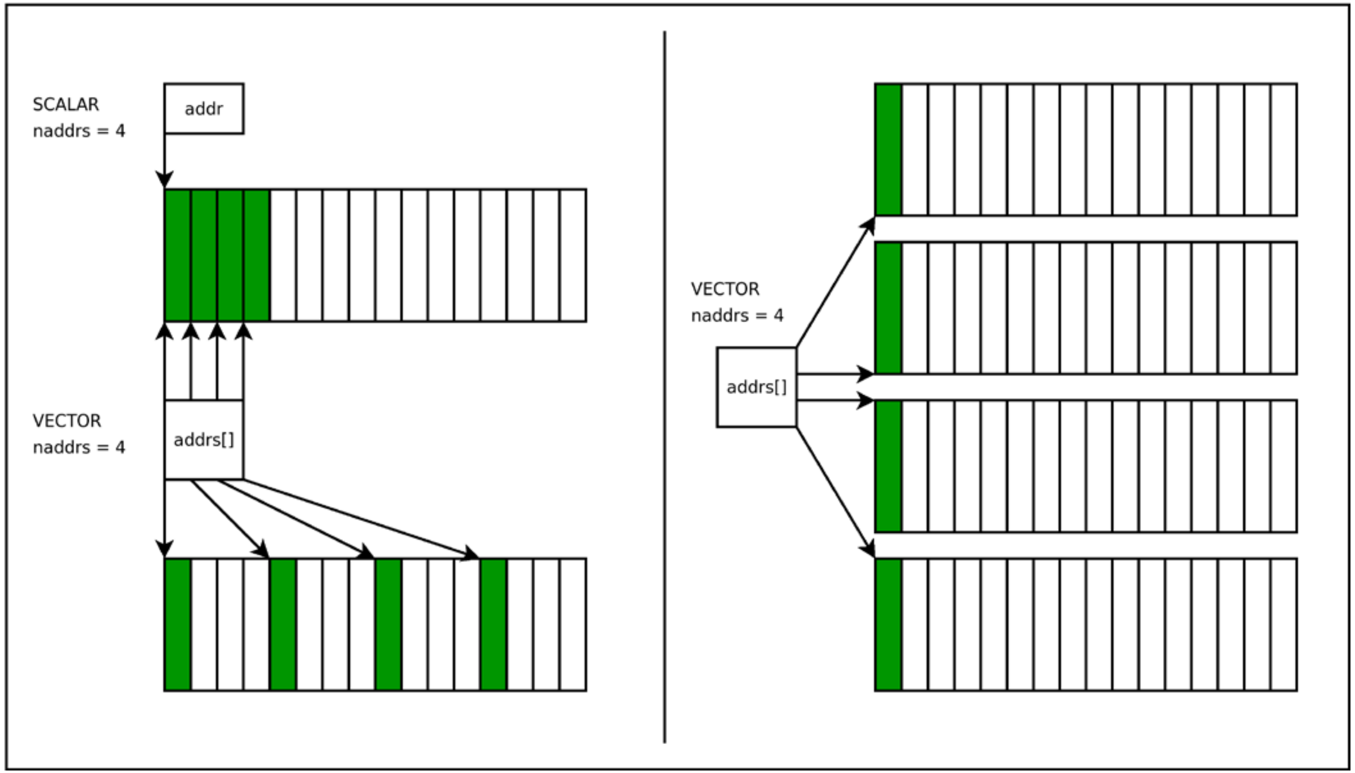
liblightnvm: OCSSD IO Command Options



liblightnvm: OCSSD IO Command Options



liblightnvm: OCSSD IO Command Options



liblightnvm: OCSSD IO Command Options

- ❒ IO Execution Mode (SYNC or ASYNC)

liblightnvm: OCSSD IO Command Options

- IO Execution Mode (SYNC or ASYNC)
- NVM_CMD_SYNC
  - Submits and blocks until **completion**

liblightnvm: OCSSD IO Command Options

- IO Execution Mode (SYNC or ASYNC)
- NVM_CMD_SYNC
  - Submits and blocks until **completion**
- NVM_CMD_ASYNC
  - Returns after **submission**
  - Callback function called upon **completion**

liblightnvm: Striping

chunk0	chunk1	chunk2	chunk3
1	13	25	37
2	14	26	38
3	15	27	39
4	16	28	40
5	17	29	41
6	18	30	42
7	19	31	43
8	20	32	44
9	21	33	45
10	22	34	46
11	23	35	47
12	24	36	48

VERT

chunk0	chunk1	chunk2	chunk3
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48

HORZ

liblightnvm: Striping

chunk0	chunk1	chunk2	chunk3
1	13	25	37
2	14	26	38
3	15	27	39
4	16	28	40
5	17	29	41
6	18	30	42
7	19	31	43
8	20	32	44
9	21	33	45
10	22	34	46
11	23	35	47
12	24	36	48

VERT

chunk0	chunk1	chunk2	chunk3
1	**2**	**3**	**4**
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48

HORZ

liblightnvm: Striping Caveat

- Constraints amplified
- Write-cache increase
  - **MWC** x **k**
- Optimal write-size
  - **WS_OPT** x **k**
- Minimal write is intact

chunk0	chunk1	chunk2	chunk3
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48

HORZ

liblightnvm: OCSSD ASYNC IO Example

DEMO

https://asciinema.org/a/8bo7Ma0DWqqZaMQRelGWDNTaf

liblightnvm: Abstractions

- Reduce the cognitive load on the OCSSD user

liblightnvm: Abstractions

- Reduce the cognitive load on the OCSSD user
- Provide traditional IO semantics
  - write(fd, *buf, count) / read(fd, *buf, count)
  - pread(fd, *buf, count, offset)

liblightnvm: Abstractions

- Reduce the cognitive load on the OCSSD user
- Provide traditional IO semantics
  - write(fd, *buf, count) / read(fd, *buf, count)
  - pread(fd, *buf, count, offset)
- Use them when you need them
  - Peel them off and take control when you don’t

liblightnvm: Virtual Block

- Encapsulates IO to a disjoint **set** of **k** chunks
- Dynamic / Runtime control of parallel units
  - **User** provisioned **set** of chunks
- **HORZ** striping on **WS_OPT** for throughput

liblightnvm: Virtual Block

- Traditional IO Semantics
  - nvm_vblk_write(*vblk, *buf, count)
  - nvm_vblk_read(*vblk, *buf, count)
  - nvm_vblk_pread(*vblk, *buf, count, offset)
- Agnostic to media and spec. variation

liblightnvm: Virtual Block

DEMO

https://asciinema.org/a/HnPa9smu8W6HoeyaqC6DavBeo

liblightnvm: Append-Only Streams

- Encapsulates IO to a disjoint **set** of **k** chunks
- Dynamic / Runtime control of parallel units
  - **Library** provisioned **set** of chunks
  - Provisioning strategy e.g. **HORZ** or **VERT**
- **HORZ** striping on **WS_OPT** for throughput
- Stream states are persistent!

liblightnvm: Append-Only Streams

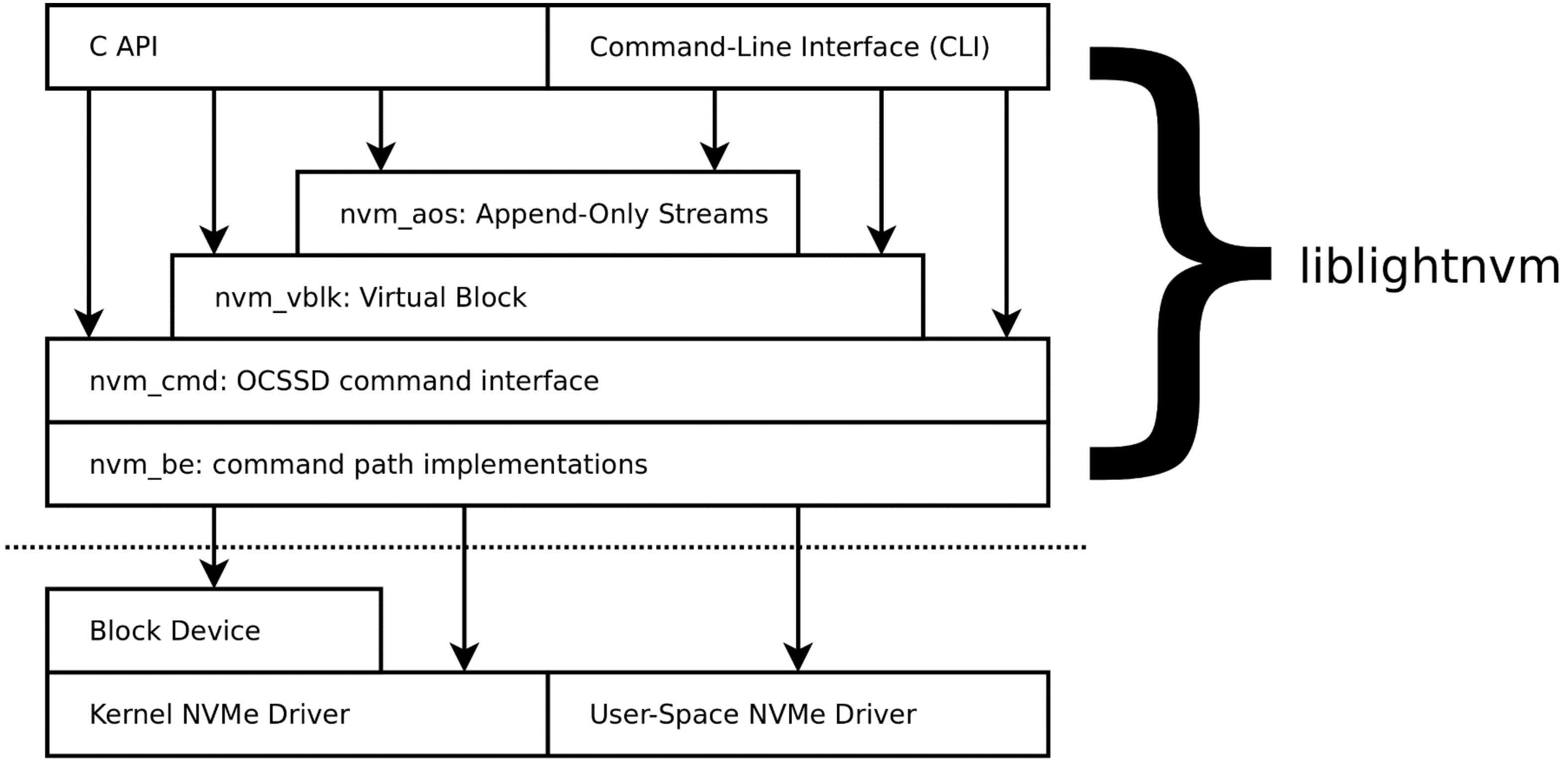
- Traditional IO Semantics
  - aos_write(sid, *buf, count)
  - aos_read(sid, *buf, count)
  - aos_pread(sid, *buf, count, read)
- Agnostic to media and spec. variation
- Encapsulates geometry and addressing

liblightnvm: Append-Only Streams

DEMO

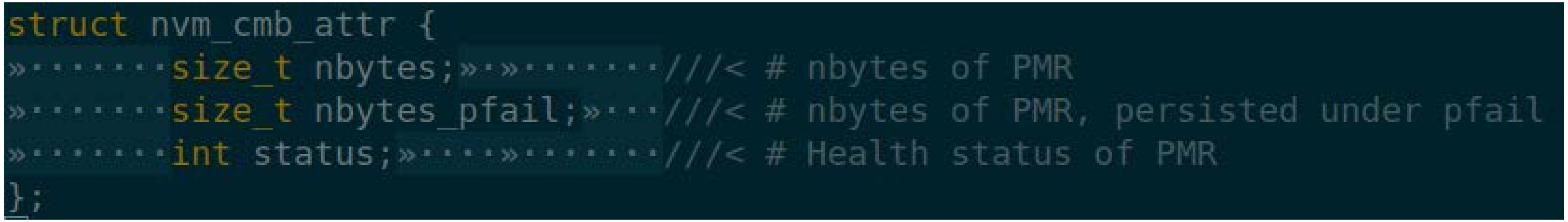
https://asciinema.org/a/ljb7fhetCKmRCd79G8cbYpaic

liblightnvm: Summary


C API
Command-Line Interface (CLI)
nvm_aos: Append-Only Streams
nvm_vblk: Virtual Block
nvm_cmd: OCSSD command interface
nvm_be: command path implementations
liblightnvm
Block Device
Kernel NVMe Driver
User-Space NVMe Driver

Roadmap: Persistent CMB interface

- Raw
  - nvm_cmb_write
  - nvm_cmd_read
- IO oriented
  - nvm_cmb_io_write
  - nvm_cmb_io_read
  - nvm_cmd_io_push



```
struct nvm_cmb_attr {
	size_t nbytes;		///< # nbytes of PMR
	size_t nbytes_pfail;	///< # nbytes of PMR, persisted under pfail
	int status;		///< # Health status of PMR
};
```

Roadmap: Spec. support

- Expand support in the evolving spec. space
  - Denali / OCSSD 2.1 / NVMe
  - Raw support via **nvm_cmd_***
  - Encapsulation in upper-level abstractions
    - Virtual Block and Append-Only interfaces

Roadmap: Related tools

- **nvm_ui**
  - Web interface for management of PBLK instances, NVMoF targets, subsystems and ports
  - Visualization of IO stats. in real-time
- **CIJOE**
  - Toolchain for QA, test, and development

Roadmap: Collaboration

- What are you missing from liblightnvm?

Roadmap: Collaboration

- What are you missing from liblightnvm?
- Regarding SGL support, would you prefer …
  - An array of buffers
  - A list of SGL segments
  - Iterator / function-pointer
  - Something else? All of them?

Thanks

SRC http://github.com/OpenChannelSSD/liblightnvm

DOC http://lightnvm.io/liblightnvm

MAIL slund@cnexlabs.com

www.linkedin.com/in/simonlund