

Pynvme: test NVMe devices in Python

<https://github.com/cranechu/pynvme>

Pynvme

Pynvme is a python extension module. Users can operate NVMe SSD intuitively by Python scripts. It is designed for NVMe SSD testing with performance considered. With third-party tools, e.g. emacs, vscode and pytest, pynvme is a convenient and professional NVMe device test solution.

Pynvme in Python IDLE

```
cranechu@localhost: ~/pynvme/spdk/examples/nvme/identify
Host Read Commands: 203987588
Host Write Commands: 25207644
Controller Busy Time: 1542 minutes
Power Cycles: 45
Power On Hours: 42 hours
Unsafe Shutdowns: 19
Unrecoverable Media Errors: 28
Lifetime Error Log Entries: 1563
Warning Temperature Time: 0 minutes
Critical Temperature Time: 0 minutes
Temperature Sensor 1: 318 Kelvin (45 Celsius)

Number of Queues
=====
Number of I/O Submission Queues: 31
Number of I/O Completion Queues: 31

Namespace ID:1
Deallocate: Supported
Deallocated/Unwritten Error: Not Supported
Deallocated Read Value: Unknown
Deallocate in Write Zeroes: Not Supported
Deallocated Guard Field: 0xFFFF
Flush: Supported
Reservation: Not Supported
Namespace Sharing Capabilities: Private
Size (in LBAs): 500118192 (476M)
Capacity (in LBAs): 500118192 (476M)
Utilization (in LBAs): 500118192 (476M)
EUI64: 00080d04001b4316
Thin Provisioning: Not Supported
Per-NS Atomic Units: No
NGUID/EUI64 Never Reused: No
Number of LBA Formats: 2
Current LBA Format: LBA Format #01
LBA Format #00: Data Size: 4096 Metadata Size: 0
LBA Format #01: Data Size: 512 Metadata Size: 0

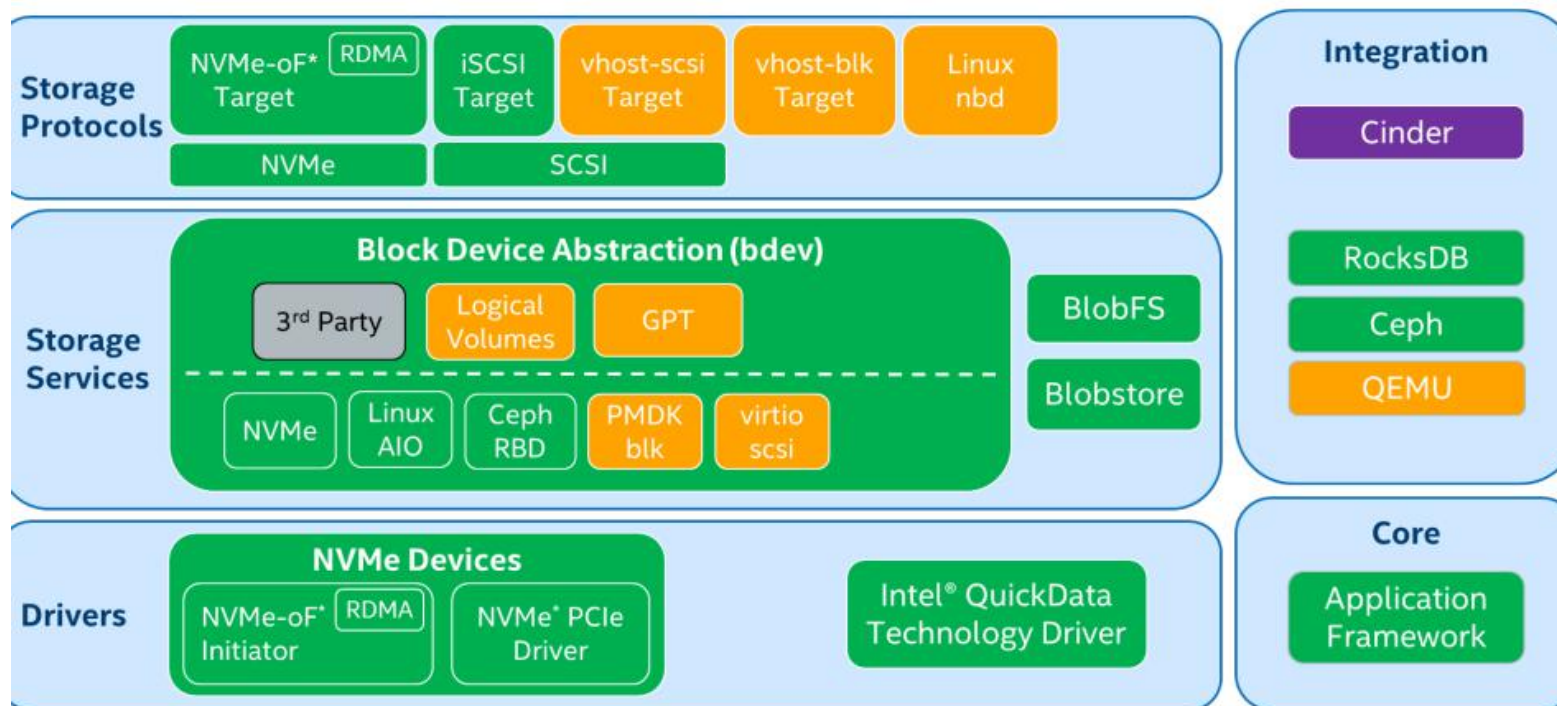
~/pynvme/~/examples/nvme/identify>

cranechu@localhost: ~/pynvme
~/pynvme> sudo python3
Python 3.7.2 (default, Jan 16 2019, 19:49:22)
[GCC 8.2.1 20181215 (Red Hat 8.2.1-6)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import nvme
Starting SPDK v19.01-pre / DPDK 18.08.0 initialization...
[ DPDK EAL parameters: pynvme_driver -c 0x2 -m 5892 --base-virtaddr=0x20
00000000 --file-prefix=spdk0 --proc-type=auto ]
EAL: Detected 4 lcore(s)
EAL: Detected 1 NUMA nodes
EAL: Auto-detected process type: PRIMARY
EAL: Multi-process socket /var/run/dpdk/spdk0/mp_socket
EAL: Probing VFIO support...
EAL: no supported IOMMU extensions found!
EAL: VFIO support could not be initialized
>>> nvme0 = nvme.Controller(b'01:00.0')
EAL: PCI device 0000:01:00.0 on NUMA socket 0
EAL: probe driver: 1179:113 spdk_nvme
nvme.pcie.c: 992:nvme.pcie.construct: *INFO*: max_completions_cap
32 num_trackers = 96
driver.c: 449:attach_cb: *INFO*: attached device 0000:01:00.0: KBG30ZMS2
, 1 namespaces, pid 3001 ADDA0102
>>> nvme0n1 = nvme.Namespace(nvme0)
driver.c: 76:memzone_reserve_shared_memory: *INFO*: create token table,
size: 2000472768
>>> eui64 = nvme0n1.id_data(127, 120)
>>> eui64
1604155579155941376
>>> eui64.to_bytes(8, byte_order='little')
... ).hex()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: to_bytes() missing required argument 'byteorder' (pos 2)
>>> eui64.to_bytes(8, byteorder='little').hex()
'00080d04001b4316'
>>> # the EUI64 got by pynvme is just the same as the spdk example. Good
...
>>> exit()
```

SPDK

pynvme

Pynvme is based on SPDK

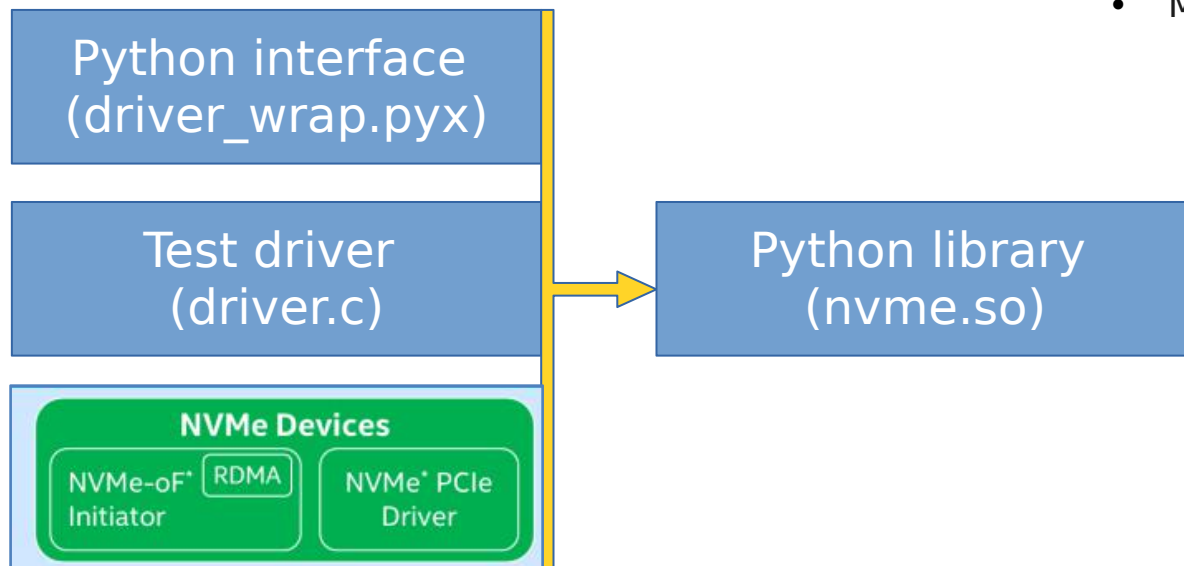


Pynvme Architecture

Build python library with

Cython:

- setup.py
- driver.c
- driver.h
- cdriver.pxd
- driver_wrap.pyx
- Makefile

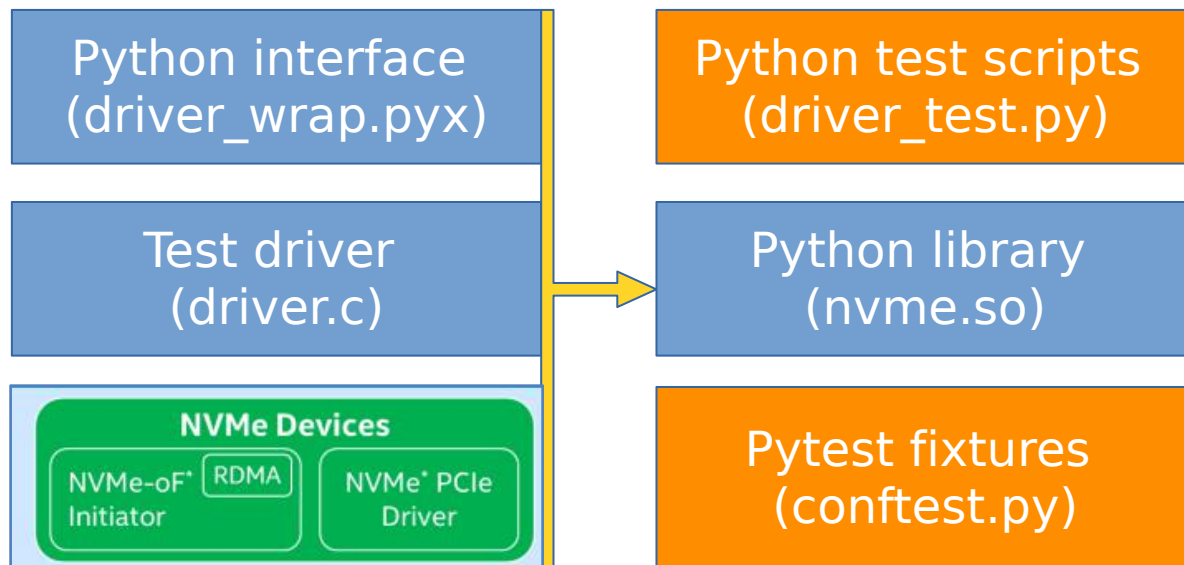


Pynvme Architecture

Organize test cases in

pytest:

- mvme.so
- pytest.ini
- conftest.py
- driver_test.py



Why Python?



✓ Many beautiful mature libraries

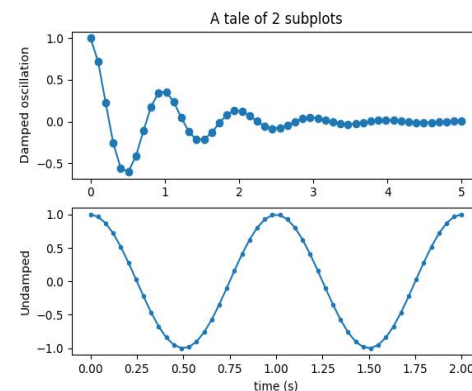
- pytest
- logging
- multiprocessing
- numpy
- matplotlib
- pydoc
- os, io, time, pytemperature, statistics, yaml, json, struct, ...

✓ Friendly IDEs for developing, debugging, and testing

- Pycharm, emacs, vscode

✓ CI: develop firmware softly













pipeline passed



Pynvme CI Status

Crane Chu > pynvme > Pipelines

All 160 Pending 1 Running 2 Finished 157 Branches Tags Run Pipeline Clear Runner Caches CI Lint

Status	Pipeline	Commit	Stages		
🔄 pending	#50853930 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	⏸		✖
🔄 running	#50846622 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	⏸	🕒 00:00:01 📅 3 hours ago	✖
🔄 running	#50842389 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	🌙	🕒 00:00:01 📅 3 hours ago	✖
✅ passed	#50835305 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	✅	🕒 00:22:26 📅 16 minutes ago	
✅ passed	#50828571 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	✅	📅 39 minutes ago	
✅ passed	#50822509 by  latest	🔒 master → 8b57310a  CI: without-isal to avoid writin...	✅	🕒 00:23:02 📅 1 hour ago	

Config and Compile

- `git clone https://github.com/cranechu/pynvme; cd pynvme`
- `git submodule update --init --recursive`
- `sudo ./spdk/scripts/pkgdep.sh`
- `sudo dnf install python3-pip -y`
- `sudo python3 -m pip install -r requirements.txt`
- `cd spdk; ./configure --without-isal; cd ..`
- `make spdk; make`

- Fedora29 is recommended.
- find script examples in `driver_test.py`.
- **find more =>** <https://github.com/cranechu/pynvme>

Test Scripts

pytest cases
are started
with test_

create qpair and
buffer for write
commands

write data,
and then read
in the callback

```
emacs
248 def test_write_identify_and_verify_with_callback(nvme0, nvme0n1):
249     id_buf = d.Buffer(4096)
250     nvme0.identify(id_buf).waitdone()
251
252     q = d.Qpair(nvme0, 20)
253     n = nvme0n1
254     read_buf = d.Buffer(4096, "read buffer")
255
256     def read_cb(cdw0, status):
257         assert id_buf[:40] == read_buf[:40]
258
259     def write_cb(cdw0, status):
260         n.read(q, read_buf, 5, 8, cb=read_cb)
261
262     n.write(q, id_buf, 5, 8, cb=write_cb).waitdone(2)
263
264     id_buf[0] += 1
265     n.write(q, id_buf, 5, 8, cb=write_cb).waitdone(2)
266     id_buf[9] = (id_buf[9] >> 1)
267     n.write(q, id_buf, 5, 8, cb=write_cb).waitdone(2)
268
55k 249: 0 UU- ~/pynvme/driver_test.py 15% -master Python [test
In: test_write_identify_and_verify_with_callback()
```

pytest fixtures

fill buffer with
identify data

callback functions
are called when
cmds are completed

the status of the
callback function
also includes the
Phase Tag bit

IOWorker

```
emacs
1066 @pytest.mark.parametrize("qcount", [1, 2, 4, 8, 15])
1067 def test_ioworker_iops_multiple_queue(nvme0n1, qcount):
1068     l = []
1069     io_total = 0
1070     for i in range(qcount):
1071         a = nvme0n1.ioworker(io_size=8, lba_align=8,
1072                             region_start=0, region_end=256*1024*8, # 1GB space
1073                             lba_random=False, qdepth=16,
1074                             read_percentage=100, time=10).start()
1075         l.append(a)
1076
1077     for a in l:
1078         r = a.close()
1079         io_total += (r.io_count_read+r.io_count_write)
1080
1081     logging.info("Q %d IOPS: %dK" % (qcount, io_total/10000))
1082
55k 1068: 0 UU- ~/pynvme/driver_test.py 60% -master Python [test_ioworker_iop
In: test_ioworker_iops_multiple_queue()
```

define IO patterns
in ioworker's
parameter list

send IO in a
separated process

wait ioworkers till
finish, and collect
result data

Fixtures of Pynvme

- create/delete test objects. in conftest.py:
 - nvme0
 - nvme0n1
 - pcie
 - ...
- parametrize of tests
 - @pytest.mark.parametrize("qcount", [1, 2, 4, 8, 15])
 - @pytest.mark.parametrize("repeat", range(10))
- test control
 - @pytest.mark.skip("nvme over tcp")
- doc: <https://docs.pytest.org/en/latest/fixture.html>

Test Files

import pytest

import nvme

test functions

test file name

```
emacs
34 # -*- coding: utf-8 -*-
35
36
37 import os
38 import time
39 import pytest
40 import logging
41 import warnings
42
43 import nvme as d
44
45 def test_create_device(nvme0, nvme0n1):
46     assert nvme0 is not None
47
48 def test_create_device_invalid():
49     with pytest.raises(d.NvmeEnumerateError):
50         nvme1 = d.Controller(b"00:00.0")
51
55k 35: 0 UU-x-~/pynvme/driver_test.py 3% -master Python
```

pytest collects test files and cases before execution

Pytest Execution



- `"""The pytest framework makes it easy to write small tests, yet scales to support complex functional testing for applications and libraries."""`
- `"""pytest fixtures offer dramatic improvements over the classic xUnit style of setup/teardown functions"""`
- `pytest test_mod.py # run tests in a module`
- `pytest testing/ # run tests in a directory`
- `pytest test_mod.py::test_func # run a specific test case`
- `pytest test_mod.py -s # run tests without log capturing`
- <https://media.readthedocs.org/pdf/pytest/latest/pytest.pdf>

Visual Studio Code



- VSCode is a popular IDE. It has a mature python extension, which supports pytest.
 - root user is not recommended by vscode
 - it is required for user to run sudo without a password.
 - *sudo visudo*
- Pynvme also provides an extension to monitor device status and cmdlog in every qpair.
 - *code --install-extension pynvme-console-0.0.1.vsix*
- *make setup; code .*

Pynvme in VSCode

qpairs

test items

cmdlog

test log

test scripts

```
1 | 2
2 | 2
3 | 2
4 | 3
5 | 2
6 | 2
7 | 2
8 | 2
9 | 2
10 | 2
11 | 2
12 | 2
13 | 20
14 | 13
15 | 14
16 | 15
17 | 16
18 | 17
19 | 18
20 | 19
21 | 20
22 | 21
23 | 22
24 | 23
25 | 24
26 | 25
27 | 26
28 | 27
29 | 28
30 | 29
31 | 30
32 | 31
33 | 32
34 | 33
35 | 34
36 | 35
37 | 36
38 | 37
39 | 38
40 | 39
41 | 40
42 | 41
43 | 42
44 | 43
45 | 44
46 | 45
47 | 46
48 | 47
49 | 48
50 | 49
51 | 50
52 | 51
53 | 52
54 | 53
```

```
482 nvme0.getlogpage(0x81, buf, 20).waitdone()
483 while buf.data(3, 2) & 0x7 != 1: # sanitize is no
484     logging.info("sanitize progress %d%%" % (buf.d
485     nvme0.getlogpage(0x81, buf, 20).waitdone()
486     time.sleep(1)
487
488 logging.info("verify data after sanitize")
489 q = d.Qpair(nvme0, 8)
490 nvme0n1.read(q, buf, 11, 1).waitdone()
491 assert buf[0] == 0
492
493 @pytest.mark.parametrize("nsid", [0, 1, 0xffffffff])
494 def test_dst_short(nvme0, nsid):
495     nvme0.dst(1, nsid).waitdone()
496
497     # check dst log page till no dst in progress
498     buf = d.Buffer(4096)
499     nvme0.getlogpage(0x6, buf, 32).waitdone()
500     while buf[0]:
501         logging.info("current dst progress percentage:
502         time.sleep(1)
503         nvme0.getlogpage(0x6, buf, 32).waitdone()
504
505     Run Test | Debug Test
506 def test_dst_extended(nvme0):
507     nvme0.dst(2).waitdone()
508
509     # check dst log page till no dst in progress
510     buf = d.Buffer(4096)
511     nvme0.getlogpage(0x6, buf, 32).waitdone()
512     while buf[0]:
513         logging.info("current dst progress percentage:
514         time.sleep(1)
515         nvme0.getlogpage(0x6, buf, 32).waitdone()
516
517     Run Test | Debug Test
518 def test_write_uncorrectable(nvme0, nvme0n1):
519     buf = d.Buffer(4096)
520     q = d.Qpair(nvme0, 8)
521     logging.info("read uncorrectable")
522     nvme0n1.write_uncorrectable(q, 0, 8).waitdone()
523     with pytest.warns(UserWarning, match="ERROR status
524     nvme0n1.read(q, buf, 0, 8).waitdone()
525
526     logging.info("read normal data")
527     nvme0n1.write(q, buf, 0, 8).waitdone()
528     nvme0n1.read(q, buf, 0, 8).waitdone()
529     logging.info("read uncorrectable")
```

```
[2019-03-19 17:02:09.853] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:10.855] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:11.856] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:12.857] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:13.858] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:14.860] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:15.861] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:16.862] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:17.863] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:18.864] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:19.865] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:20.866] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:21.870] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:22.871] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:23.872] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:24.873] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:25.874] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:26.875] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:27.876] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:28.877] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:29.878] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:30.880] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:31.880] INFO test_dst_extended(514): current dst progress percentage: 81%
PASSED [100%]
----- live log teardown -----
[2019-03-19 17:02:32.882] INFO script(33): test duration: 94.116 sec

===== 1 passed in 96.28 seconds =====
===== test session starts =====
platform linux -- Python 3.7.2 -- pytest 4.0.1 -- py-1.7.0, pluggy-0.8.1 -- /usr/bin/python3
cachedir: .pytest_cache
rootdir: /home/cranachu/pynvme, inifile: pytest.ini
plugins: cov-2.6.1
collecting ... collected 1 item

driver_test.py::test_dst_extended
----- live log setup -----
[2019-03-19 17:02:42.611] INFO pcidriver(19): running tests on OUT 01:00:00:00:00:00
----- live log call -----
[2019-03-19 17:02:44.636] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:45.637] INFO test_dst_extended(514): current dst progress percentage: 54%
[2019-03-19 17:02:46.638] INFO test_dst_extended(514): current dst progress percentage: 54%
[2019-03-19 17:02:47.640] INFO test_dst_extended(514): current dst progress percentage: 54%
[2019-03-19 17:02:48.641] INFO test_dst_extended(514): current dst progress percentage: 54%
[2019-03-19 17:02:49.642] INFO test_dst_extended(514): current dst progress percentage: 63%
[2019-03-19 17:02:50.643] INFO test_dst_extended(514): current dst progress percentage: 63%
[2019-03-19 17:02:51.644] INFO test_dst_extended(514): current dst progress percentage: 63%
[2019-03-19 17:02:52.645] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:53.647] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:54.648] INFO test_dst_extended(514): current dst progress percentage: 81%
[2019-03-19 17:02:55.649] INFO test_dst_extended(514): current dst progress percentage: 81%
```


Welcome to use and contribute

The screenshot shows the GitHub repository page for `cranechu / pynvme`. At the top, there are buttons for `Unwatch` (4), `Unstar` (5), and `Fork` (2). Below these are tabs for `Code`, `Issues` (0), `Pull requests` (0), `Projects` (0), `Wiki`, `Insights`, and `Settings`. The repository description is "test NVMe devices in Python. Find more ==> <https://github.com/cranechu/pynvme/bl...>". There are tags for `nvme`, `ssd`, `driver`, `spdk`, `test`, `python`, and `linux`, along with a `Manage topics` link. A summary bar shows `62 commits`, `1 branch`, `2 releases`, `1 contributor`, and `BSD-3-Clause` license. Below this is a bar with `Branch: master`, `New pull request`, `Create new file`, `Upload files`, `Find file`, and `Clone or download`. The commit history table is as follows:

Commit	Message	Time
<code>cranechu</code> CI: without-isal to avoid writing /tmp		Latest commit 8b5731e 5 days ago
<code>snippets/python-mode</code>	build: pre-release pypi	2 months ago
<code>spdk @ e3e35a8</code>	spdk: update to v19.01, but DPDK still use spdk-18.08	5 days ago
<code>.gitignore</code>	doc: update and simplified	5 days ago
<code>.gitlab-ci.yml</code>	CI: without-isal to avoid writing /tmp	5 days ago
<code>.gitmodules</code>	pynvme: initial code	2 months ago
<code>LICENSE</code>	pynvme: initial code	2 months ago
<code>Makefile</code>	spdk: update to v19.01, but DPDK still use spdk-18.08	5 days ago
<code>README.md</code>	CI: without-isal to avoid writing /tmp	5 days ago

<https://github.com/cranechu/pynvme>