**SPDK**

**( Storage Performance Development Kit )**

Contents

[**1.Introduction:** 5](#_Toc528933645)

[**2.Getting Started:** 5](#_Toc528933646)

[a. get source code 6](#_Toc528933647)

[b. install prerequisites 6](#_Toc528933648)

[c.build 6](#_Toc528933649)

[d.run unit tests 6](#_Toc528933650)

[**3.Concepts:** 6](#_Toc528933651)

[a.User Space Driver 6](#_Toc528933652)

[b.Interrupts 6](#_Toc528933653)

[c.Threading 7](#_Toc528933654)

[d.Memory Management: 7](#_Toc528933655)

[e.IOMMU Support 7](#_Toc528933656)

[f.Message Passing and Concurrency 7](#_Toc528933657)

[**4.Driver Modules** 7](#_Toc528933658)

[ NVMe Driver 7](#_Toc528933659)

[ I/OAT Driver 9](#_Toc528933660)

[ Virtio Driver 9](#_Toc528933661)

[**5.Application and tests:** 9](#_Toc528933662)

[ iSCSI 9](#_Toc528933663)

[ NVMe over Fabrics(SoftRoCE) 11](#_Toc528933664)

[**6.Performance:** 14](#_Toc528933665)

[ fio-plugin 14](#_Toc528933666)

[ perf 16](#_Toc528933667)

[**7.Questions:** 18](#_Toc528933668)

[**8.Appendix** 19](#_Toc528933669)

[ iscsi.conf 19](#_Toc528933670)

[ SPDK API 24](#_Toc528933671)

[ fio\_softroce.job 24](#_Toc528933672)

[**9.Reference:** 25](#_Toc528933673)

[ URL 25](#_Toc528933674)

[ Explanation 25](#_Toc528933675)

[**10.Compile SPDK for Arm:** 25](#_Toc528933676)

[ v18.04 25](#_Toc528933677)

[ v18.07 25](#_Toc528933678)

[ v18.10(master) 26](#_Toc528933679)

[**11.GCC:** 26](#_Toc528933680)

[ compile 27](#_Toc528933681)

***\* 版本修订记录 \****

|  |  |  |
| --- | --- | --- |
| ***版本号*** | ***修订时间*** | ***修订内容*** |
| *v1.0* | *2018-11-02* | *初版修订* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***\* Release Copyleft©free \****

**1.Introduction:**

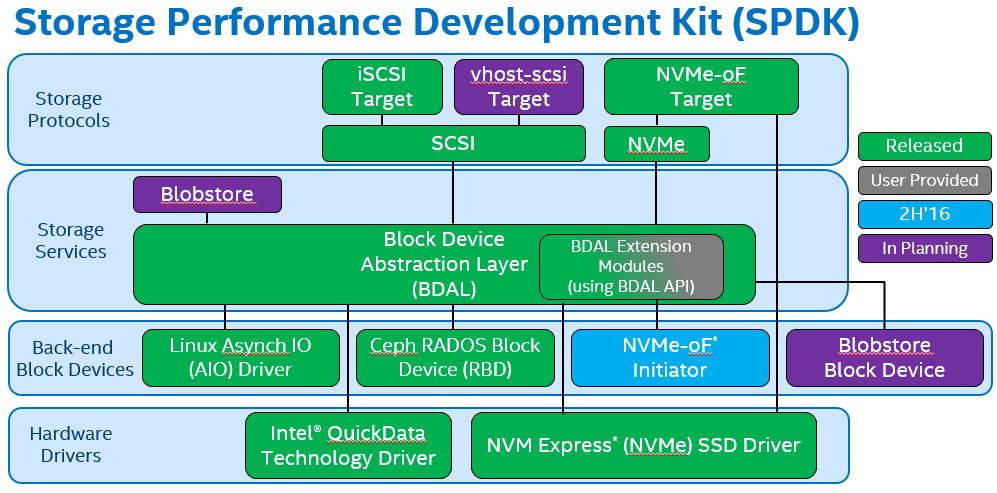
*Storage Performance Development Kit (****SPDK****)*

* Moving all of the necessary drivers into userspace, which avoids syscalls and enables zero-copy access from the application.
* Polling hardware for completions instead of relying on interrupts, which lowers both total latency and latency variance.
* Avoiding all locks in the I/O path, instead relying on message passing.

The bedrock of SPDK is a **user space**, **polled-mode**, **asynchronous**, **lockless NVMe driver**. This provides **zero-copy**, **highly parallel access directly to an SSD** from a user space application. The driver is written as a C library with a single public header.

SPDK further provides a full **block stack** as a user space library that performs many of the same operations as a block stack in an operating system. This includes unifying the interface between disparate storage devices, queueing to handle conditions such as out of memory or I/O hangs, and logical volume management.

Finally, SPDK provides **NVMe-oF**, **iSCSI**, and **vhost servers** built on top of these components that are capable of serving disks over the network or to other processes. The standard Linux kernel initiators for NVMe-oF and iSCSI interoperate with these targets, as well as QEMU with vhost. These servers can be up to an order of magnitude more CPU efficient than other implementations. These targets can be used as examples of how to implement a high performance storage target, or used as the basis for production deployments.



**2.Getting Started:**

a. get source code

|  |
| --- |
| [cpu@mon SPDK]$ git clone https://github.com/spdk/spdk  [cpu@mon SPDK]$ cd spdk/  [cpu@mon spdk]$ git submodule update --init |

b. install prerequisites

|  |
| --- |
| [cpu@mon spdk]$ sudo scripts/pkgdep.sh |

c.build

|  |
| --- |
| [cpu@mon spdk]$ ./configure  [cpu@mon spdk]$ make  # support RDMA  [cpu@mon spdk]$ ./configure --with-rdma  [cpu@mon spdk]$ make |

d.run unit tests

|  |
| --- |
| [yang@admin spdk]$ ./test/unit/unittest.sh |

**3.Concepts:**

a.User Space Driver

* a driver is software that directly controls a particular device attached to a computer
* operating systems segregate the system's virtual memory into two categories of addresses based on privilege level - kernel space and user space
* Once the device is unbound from the operating system kernel, the devices corresponding to it such as /dev/nvme0n1 will disappear.
* User space drivers utilize features in uio or vfio to map the PCI BAR for the device into the current process, which allows the driver to perform MMIO directly.

b.Interrupts

* SPDK polls devices for completions instead of waiting for interrupts.
* Operations in SPDK are almost universally asynchronous and allow the user to provide a callback on completion.
* Polling an NVMe device is fast(Intel's DDIO technologies)

c.Threading

* software can send requests to the device from multiple threads of execution in parallel without locks.

d.Memory Management:

* NVMe devices transfer data to and from system memory using Direct Memory Access (DMA);
* The NVMe 1.0 specification requires all physical memory to be describable by what is called a PRP list; memory must have the following properties:

1. The memory is broken into physical 4KiB pages, which we'll call device pages.
2. The first device page can be a partial page starting at any 4-byte aligned address. It may extend up to the end of the current physical page, but not beyond.
3. If there is more than one device page, the first device page must end on a physical 4KiB page boundary.
4. The last device page begins on a physical 4KiB page boundary, but is not required to end on a physical 4KiB page boundary.

* The NVMe 1.1 specification added support for fully flexible scatter gather lists, but the feature is optional and most devices available today do not support it.
* SPDK relies on DPDK to allocate pinned memory.

e.IOMMU Support

* All DMA operations between the PCI device and system memory are then translated through the IOMMU(input/output memory management unit) by converting the bus address to a virtual address and then the virtual address to the physical address.

f.Message Passing and Concurrency

* SPDK will often assign that data to a single thread.

**4.Driver Modules**

* NVMe Driver

1. The NVMe driver is a C library that may be linked directly into an application that provides direct, zero-copy data transfer to and from NVMe SSDs.
2. it spawns no threads and only performs actions in response to function calls from the application itself.
3. The library controls NVMe devices by directly mapping the PCI BAR into the local process and performing [MMIO(Memory-mapped IO)](#MMIO).
4. I/O is submitted asynchronously via queue pairs.
5. the library has been improved to also connect to remote NVMe devices via NVMe over Fabrics.(spdk\_nvme\_probe())

|  |
| --- |
| # bind nvme driver and allocate hugepages  [root@nvme spdk]# ./scripts/setup.sh status  Hugepages  node hugesize free / total  node0 2048kB 0 / 0  NVMe devices  BDF Numa Node Driver name Device name  0000:00:0e.0 0 nvme nvme0  I/OAT DMA  BDF Numa Node Driver Name  virtio  BDF Numa Node Driver Name Device Name  [root@nvme spdk]# ./scripts/setup.sh config  0000:00:0e.0 (80ee 4e56): nvme -> uio\_pci\_generic  [root@nvme spdk]# ./scripts/setup.sh status  Hugepages  node hugesize free / total  node0 2048kB 1024 / 1024  NVMe devices  BDF Numa Node Driver name Device name  0000:00:0e.0 0 uio\_pci\_generic -  I/OAT DMA  BDF Numa Node Driver Name  virtio  BDF Numa Node Driver Name Device Name  # run hello-world  [root@nvme spdk]# cd examples/nvme/hello\_world/  [root@nvme hello\_world]# ./hello\_world  Starting SPDK v18.10-pre / DPDK 18.05.0 initialization...  [ DPDK EAL parameters: hello\_world -c 0x1 --legacy-mem --file-prefix=spdk0 --base-virtaddr=0x200000000000 --proc-type=auto ]  EAL: Detected 1 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: WARNING! Base virtual address hint (0x20040008f000 != 0x7ff46def9000) not respected!  EAL: This may cause issues with mapping memory into secondary processes  EAL: WARNING! Base virtual address hint (0x2008000f0000 != 0x7ff46de98000) not respected!  EAL: This may cause issues with mapping memory into secondary processes  EAL: WARNING! Base virtual address hint (0x200c00151000 != 0x7ff46de37000) not respected!  EAL: This may cause issues with mapping memory into secondary processes  Initializing NVMe Controllers  EAL: PCI device 0000:00:0e.0 on NUMA socket 0  EAL: probe driver: 80ee:4e56 spdk\_nvme  Attaching to 0000:00:0e.0  Message from syslogd@localhost at Oct 9 23:20:29 ...  kernel:Disabling IRQ #22  Attached to 0000:00:0e.0  Using controller ORCL-VBOX-NVME-VER12 (VB1234-56789 ) with 1 namespaces.  Namespace ID: 1 size: 8GB  Initialization complete.  INFO: using host memory buffer for IO  Hello world! |

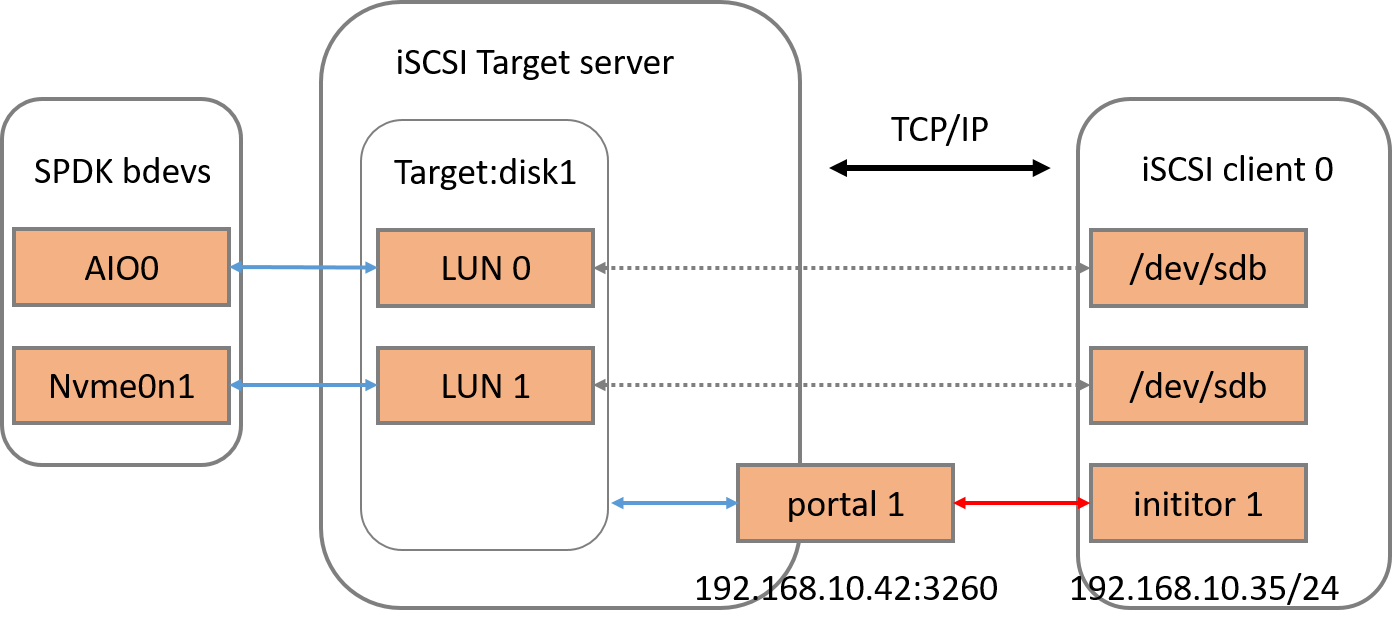
* I/OAT Driver
* Virtio Driver

The driver supports two different usage models:

* PCI : This is the standard mode of operation when used in a guest virtual machine, where QEMU has presented the virtio controller as a virtual PCI device.
* vhost-user : Can be used to connect to a vhost socket directly on the same host.

**5.Application and tests:**

* iSCSI



|  |
| --- |
| # configure target  [root@nvme spdk]# sudo scripts/setup.sh # if you have NVMe devices, you can execute this script to change the driver.  [root@nvme spdk]# cd app/iscsi\_tgt/  [root@nvme iscsi\_tgt]# cp ../../etc/spdk/iscsi.conf.in iscsi.conf # copy the configure file  [root@nvme iscsi\_tgt]# vim iscsi.conf # edit iscsi configure file and you can reference the Appendix part of this document  [root@nvme iscsi\_tgt]# ./iscsi\_tgt -c iscsi.conf # start iscsi target  [root@nvme spdk]# ../spdk/scripts/rpc.py get\_bdevs # get the device  [ {  "name": "Nvme0n1",  "aliases": [],  "product\_name": "NVMe disk",  … …  },  {  "name": "AIO0",  "aliases": [],  "product\_name": "AIO disk",  … …  },  "driver\_specific": {  "aio": {  "filename": "/dev/sdb"  }  }  }  ]  # configure initiator  [root@mon1 ~]# yum -y install iscsi-initiator-utils  [root@mon1 ~]# rpm -qa | grep iscsi  iscsi-initiator-utils-iscsiuio-6.2.0.874-7.el7.x86\_64  iscsi-initiator-utils-6.2.0.874-7.el7.x86\_64  [root@mon1 ~]# vim /etc/iscsi/initiatorname.iscsi # edit initiator name  [root@mon1 ~]# cat /etc/iscsi/initiatorname.iscsi  InitiatorName=iqn.1994-05.com.redhat:8518f439a5c3  [root@mon1 ~]# systemctl start iscsid # start the daemon  [root@mon1 ~]# systemctl status iscsid  ● iscsid.service - Open-iSCSI  Loaded: loaded (/usr/lib/systemd/system/iscsid.service; disabled; vendor preset: disabled)  Active: active (running) since Wed 2018-09-26 22:57:25 EDT; 26min ago  # Discovery and Login  # Run command on initiator  [root@admin ~]# iscsiadm -m discovery -t st -p 192.168.10.42  192.168.10.42:3260,1 iqn.2018-09.io.spdk:disk1  [root@admin ~]# iscsiadm -m node -T iqn.2018-09.io.spdk:disk1 -p 192.168.10.42 -l  Logging in to [iface: default, target: iqn.2018-09.io.spdk:disk1, portal: 192.168.10.42,3260] (multiple)  Login to [iface: default, target: iqn.2018-09.io.spdk:disk1, portal: 192.168.10.42,3260] successful.  [root@admin ~]# fdisk -l  Disk /dev/sdc: 8589 MB, 8589934592 bytes, 16777216 sectors  Units = sectors of 1 \* 512 = 512 bytes  Sector size (logical/physical): 512 bytes / 512 bytes  I/O size (minimum/optimal): 4096 bytes / 4194304 bytes  Disk /dev/sde: 8589 MB, 8589934592 bytes, 16777216 sectors  Units = sectors of 1 \* 512 = 512 bytes  Sector size (logical/physical): 512 bytes / 512 bytes  I/O size (minimum/optimal): 4096 bytes / 4194304 bytes  # Logout from the target  [root@admin ~]# iscsiadm -m node -T iqn.2018-09.io.spdk:disk1 -p 192.168.10.42 -u  Logging out of session [sid: 5, target: iqn.2018-09.io.spdk:disk1, portal: 192.168.10.42,3260]  Logout of [sid: 5, target: iqn.2018-09.io.spdk:disk1, portal: 192.168.10.42,3260] successful.  # Tunning  [root@admin ~]# echo noop > /sys/block/sdc/queue/scheduler  [root@admin ~]# echo "2" > /sys/block/sdc/queue/nomerges  [root@admin ~]# echo "1024" > /sys/block/sdc/queue/nr\_requests |

* NVMe over Fabrics(SoftRoCE)

|  |
| --- |
| # Target and Initiator  [root@rocet nvmf\_tgt]# pwd  /data/spdk/app/nvmf\_tgt  [root@rocet nvmf\_tgt]# lsmod | grep nvme  nvme\_rdma 32768 0  rdma\_cm 69632 7 rpcrdma,ib\_srpt,ib\_srp,nvme\_rdma,ib\_iser,ib\_isert,rdma\_ucm  ib\_core 278528 14 rdma\_cm,ib\_ipoib,rdma\_rxe,rpcrdma,ib\_srpt,ib\_srp,nvme\_rdma,iw\_cm,ib\_iser,ib\_umad,ib\_isert,rdma\_ucm,ib\_uverbs,ib\_cm  nvme\_fabrics 24576 1 nvme\_rdma  nvme 36864 0  nvme\_core 81920 3 nvme,nvme\_rdma,nvme\_fabrics  [root@rocet nvmf\_tgt]# lsmod | grep rdma\_rxe  rdma\_rxe 126976 0  ip6\_udp\_tunnel 16384 1 rdma\_rxe  udp\_tunnel 16384 1 rdma\_rxe  ib\_core 278528 14 rdma\_cm,ib\_ipoib,rdma\_rxe,rpcrdma,ib\_srpt,ib\_srp,nvme\_rdma,iw\_cm,ib\_iser,ib\_umad,ib\_isert,rdma\_ucm,ib\_uverbs,ib\_cm  [root@rocet nvmf\_tgt]#  # Target  [root@rocet nvmf\_tgt]# cp ../../etc/spdk/nvmf.conf.in nvmf.conf  [root@rocet nvmf\_tgt]# vim nvmf.conf # edit configuration file and reference nvmf.conf in Appendix of this document.  [root@rocet nvmf\_tgt]# ./nvmf\_tgt -c nvmf.conf  Starting SPDK v18.07 / DPDK 18.05.0 initialization...  [ DPDK EAL parameters: nvmf -c 0x1 --legacy-mem --file-prefix=spdk\_pid3775 ]  EAL: Detected 1 lcore(s)  EAL: Detected 1 NUMA nodes  EAL: Multi-process socket /var/run/dpdk/spdk\_pid3775/mp\_socket  EAL: Probing VFIO support...  app.c: 530:spdk\_app\_start: \*NOTICE\*: Total cores available: 1  reactor.c: 718:spdk\_reactors\_init: \*NOTICE\*: Occupied cpu socket mask is 0x1  reactor.c: 492:\_spdk\_reactor\_run: \*NOTICE\*: Reactor started on core 0 on socket 0  EAL: PCI device 0000:00:0e.0 on NUMA socket 0  EAL: probe driver: 80ee:4e56 spdk\_nvme  # Initiator  [root@rocei ~]# yum -y install nvme-cli  [root@rocei ~]# nvme discover -t rdma -a 192.168.10.52  Discovery Log Number of Records 1, Generation counter 4  =====Discovery Log Entry 0======  trtype: rdma  adrfam: ipv4  subtype: nvme subsystem  treq: not specified  portid: 0  trsvcid: 4420  subnqn: nqn.2016-06.io.spdk:cnode1  traddr: 192.168.10.52  rdma\_prtype: not specified  rdma\_qptype: connected  rdma\_cms: rdma-cm  rdma\_pkey: 0x0000  [root@rocei ~]# nvme connect -t rdma -n "nqn.2016-06.io.spdk:cnode1" -a 192.168.10.52 -s 4420  [root@rocei ~]# fdisk -l  Disk /dev/nvme0n1: 50 GiB, 53687091200 bytes, 104857600 sectors  Units: sectors of 1 \* 512 = 512 bytes  Sector size (logical/physical): 512 bytes / 512 bytes  I/O size (minimum/optimal): 512 bytes / 512 bytes  # Initiator fio and perf test  # fio file fio\_softroce.job append in Appendix  [root@rocei ~]# fio fio\_softroce.job  job0: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=libaio, iodepth=128  fio-3.3  Starting 1 thread  Jobs: 1 (f=1): [m(1)][100.0%][r=61.1MiB/s,w=59.0MiB/s][r=15.6k,w=15.3k IOPS][eta 00m:00s]  job0: (groupid=0, jobs=1): err= 0: pid=3737: Fri Nov 2 14:49:52 2018  read: IOPS=17.3k, BW=67.7MiB/s (70.0MB/s)(677MiB/10001msec)  slat (usec): min=3, max=2512, avg= 8.71, stdev=23.13  clat (usec): min=400, max=18418, avg=3711.75, stdev=796.46  lat (usec): min=408, max=18425, avg=3720.61, stdev=794.04  clat percentiles (usec):  | 1.00th=[ 1205], 5.00th=[ 2966], 10.00th=[ 3163], 20.00th=[ 3294],  | 30.00th=[ 3425], 40.00th=[ 3523], 50.00th=[ 3621], 60.00th=[ 3752],  | 70.00th=[ 3916], 80.00th=[ 4146], 90.00th=[ 4555], 95.00th=[ 4948],  | 99.00th=[ 5932], 99.50th=[ 6652], 99.90th=[10159], 99.95th=[11863],  | 99.99th=[16057]  bw ( KiB/s): min=62698, max=74360, per=100.00%, avg=69531.11, stdev=3181.02, samples=19  iops : min=15674, max=18590, avg=17382.74, stdev=795.34, samples=19  write: IOPS=17.3k, BW=67.6MiB/s (70.9MB/s)(676MiB/10001msec)  slat (usec): min=4, max=6789, avg= 9.24, stdev=31.20  clat (usec): min=361, max=16183, avg=3656.96, stdev=804.29  lat (usec): min=399, max=16190, avg=3666.34, stdev=802.19  clat percentiles (usec):  | 1.00th=[ 1156], 5.00th=[ 2638], 10.00th=[ 3130], 20.00th=[ 3261],  | 30.00th=[ 3392], 40.00th=[ 3490], 50.00th=[ 3589], 60.00th=[ 3720],  | 70.00th=[ 3884], 80.00th=[ 4080], 90.00th=[ 4490], 95.00th=[ 4883],  | 99.00th=[ 5735], 99.50th=[ 6390], 99.90th=[ 9372], 99.95th=[11863],  | 99.99th=[15926]  bw ( KiB/s): min=61421, max=74168, per=100.00%, avg=69489.68, stdev=3392.20, samples=19  iops : min=15355, max=18542, avg=17372.37, stdev=848.17, samples=19  lat (usec) : 500=0.01%, 750=0.01%, 1000=0.22%  lat (msec) : 2=3.24%, 4=71.90%, 10=24.52%, 20=0.10%  cpu : usr=8.35%, sys=23.85%, ctx=7572, majf=0, minf=1  IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%  submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%  complete : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.1%  issued rwts: total=173357,173083,0,0 short=0,0,0,0 dropped=0,0,0,0  latency : target=0, window=0, percentile=100.00%, depth=128  Run status group 0 (all jobs):  READ: bw=67.7MiB/s (70.0MB/s), 67.7MiB/s-67.7MiB/s (70.0MB/s-70.0MB/s), io=677MiB (710MB), run=10001-10001msec  WRITE: bw=67.6MiB/s (70.9MB/s), 67.6MiB/s-67.6MiB/s (70.9MB/s-70.9MB/s), io=676MiB (709MB), run=10001-10001msec  Disk stats (read/write):  nvme0n1: ios=171328/171110, merge=0/0, ticks=493028/486217, in\_queue=1078298, util=97.97%  [root@rocei perf]# pwd  /data/spdk/examples/nvme/perf  [root@rocei perf]# ./perf -q 16 -s 4096 -w randwrite -t 10 /dev/nvme1n1  Starting SPDK v18.07 / DPDK 18.05.0 initialization...  [ DPDK EAL parameters: perf -c 0x1 --legacy-mem --file-prefix=spdk\_pid4575 ]  EAL: Detected 1 lcore(s)  EAL: Detected 1 NUMA nodes  EAL: Multi-process socket /var/run/dpdk/spdk\_pid4575/mp\_socket  EAL: Probing VFIO support...  Initializing NVMe Controllers  EAL: PCI device 0000:00:0e.0 on NUMA socket 0  EAL: probe driver: 80ee:4e56 spdk\_nvme  Attaching to NVMe Controller at 0000:00:0e.0  Attached to NVMe Controller at 0000:00:0e.0 [80ee:4e56]  Associating ORCL-VBOX-NVME-VER12 (VB1234-56789 ) with lcore 0  Associating /dev/nvme1n1 with lcore 0  Initialization complete. Launching workers.  Starting thread on core 0  ========================================================  Latency(us)  Device Information : IOPS MB/s Average min max  /dev/nvme1n1 from core 0: 238.10 0.93 67594.83 682.64 963620.77  ORCL-VBOX-NVME-VER12 (VB1234-56789 ) from core 0: 394.60 1.54 41004.28 61.87 216086.25  ========================================================  Total : 632.70 2.47 51010.94 61.87 963620.77  # disconnect nvme device  [root@rocei perf]# nvme disconnect -d /dev/nvme1n1 |

**6.Performance:**

* fio-plugin

|  |
| --- |
| # compile fio-plugin  [yang@nvme fio]$ git clone https://github.com/axboe/fio  [yang@nvme SPDK]$ cd fio/  [yang@nvme fio]$ git checkout fio-3.3  [yang@nvme spdk]$ ./configure --with-fio=/home/yang/SPDK/fio/fio/  [yang@nvme spdk]$ make  # Usage  [root@nvme master-fio]# mkdir fio  [root@nvme fio]# cp ../spdk/etc/spdk/nvmf.conf.in nvmf.conf  [root@nvme fio]# vim nvmf.conf # just edit the [Nvme] part  [Nvme]  TransportID "trtype:PCIe traddr:0000:00:0e.0" Nvme0  RetryCount 4  TimeoutUsec 0    ActionOnTimeout None    AdminPollRate 100000  HotplugEnable No  # edit fio script  [root@nvme fio]# touch example.fio  [root@nvme fio]# vim example.fio  [global]  ioengine=spdk\_bdev  spdk\_conf=/home/yang/SPDK/master-fio/fio/nvmf.conf  thread=1  group\_reporting=1  direct=1  verify=0  time\_based=1  ramp\_time=0  runtime=60  iodepth=128  rw=randread  bs=4k  [test]  numjobs=1  filename=Nvme0n1  # run fio  [root@nvme fio]# LD\_PRELOAD=/home/yang/SPDK/master-fio/spdk/examples/bdev/fio\_plugin/fio\_plugin /home/yang/SPDK/fio/fio/fio /home/yang/SPDK/master-fio/fio/example.fio  test: (g=0): rw=randread, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T) 4096B-4096B, ioengine=spdk\_bdev, iodepth=128  fio-3.3  Starting 1 thread  Starting SPDK v18.10-pre / DPDK 18.05.0 initialization...  [ DPDK EAL parameters: fio --no-shconf -c 0x1 --legacy-mem --file-prefix=spdk\_pid4855 ]  EAL: Detected 1 lcore(s)  EAL: Detected 1 NUMA nodes  EAL: Multi-process socket /var/run/dpdk/spdk\_pid4855/mp\_socket  EAL: Probing VFIO support...  EAL: PCI device 0000:00:0e.0 on NUMA socket 0  EAL: probe driver: 80ee:4e56 spdk\_nvme  Jobs: 1 (f=1): [r(1)][100.0%][r=648MiB/s,w=0KiB/s][r=166k,w=0 IOPS][eta 00m:00s]  test: (groupid=0, jobs=1): err= 0: pid=4862: Wed Oct 10 02:59:15 2018  read: IOPS=166k, BW=649MiB/s (680MB/s)(38.0GiB/60001msec)  slat (usec): min=2, max=7657, avg= 4.67, stdev= 9.66  clat (usec): min=2, max=14540, avg=633.75, stdev=757.57  lat (usec): min=5, max=14551, avg=638.42, stdev=757.59  clat percentiles (usec):  | 1.00th=[ 9], 5.00th=[ 18], 10.00th=[ 37], 20.00th=[ 96],  | 30.00th=[ 184], 40.00th=[ 285], 50.00th=[ 383], 60.00th=[ 482],  | 70.00th=[ 603], 80.00th=[ 947], 90.00th=[ 1811], 95.00th=[ 2409],  | 99.00th=[ 3130], 99.50th=[ 3556], 99.90th=[ 4686], 99.95th=[ 5473],  | 99.99th=[ 7504]  bw ( KiB/s): min=496275, max=690693, per=98.80%, avg=656126.07, stdev=25096.54, samples=119  iops : min=124068, max=172673, avg=164031.13, stdev=6274.11, samples=119  lat (usec) : 4=0.05%, 10=1.97%, 20=3.75%, 50=7.12%, 100=7.68%  lat (usec) : 250=16.09%, 500=25.06%, 750=14.12%, 1000=5.16%  lat (msec) : 2=10.61%, 4=8.14%, 10=0.26%, 20=0.01%  cpu : usr=99.41%, sys=0.07%, ctx=5178, majf=0, minf=9  IO depths : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%, >=64=100.0%  submit : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%  complete : 0=0.0%, 4=97.5%, 8=0.5%, 16=0.4%, 32=0.4%, 64=1.3%, >=64=0.1%  issued rwt: total=9961517,0,0, short=0,0,0, dropped=0,0,0  latency : target=0, window=0, percentile=100.00%, depth=128  Run status group 0 (all jobs):  READ: bw=649MiB/s (680MB/s), 649MiB/s-649MiB/s (680MB/s-680MB/s), io=38.0GiB (40.8GB), run=60001-60001msec |

* perf

|  |
| --- |
| [root@nvme spdk]# ./scripts/setup.sh status  Hugepages  node hugesize free / total  node0 2048kB 0 / 1024  NVMe devices  BDF Numa Node Driver name Device name  0000:00:0e.0 0 uio\_pci\_generic -  I/OAT DMA  BDF Numa Node Driver Name  virtio  BDF Numa Node Driver Name Device Name  # perf options  [-q io depth]  [-o io size in bytes]  [-w io pattern type, must be one of  (read, write, randread, randwrite, rw, randrw)]  [-M rwmixread (100 for reads, 0 for writes)]  [-L enable latency tracking via sw, default: disabled]  -L for latency summary, -LL for detailed histogram  [-l enable latency tracking via ssd (if supported), default: disabled]  [-t time in seconds]  [-c core mask for I/O submission/completion.]  (default: 1)]  [-D disable submission queue in controller memory buffer, default: enabled]  [-r Transport ID for local PCIe NVMe or NVMeoF]  Format: 'key:value [key:value] ...'  Keys:  trtype Transport type (e.g. PCIe, RDMA)  adrfam Address family (e.g. IPv4, IPv6)  traddr Transport address (e.g. 0000:04:00.0 for PCIe or 192.168.100.8 for RDMA)  trsvcid Transport service identifier (e.g. 4420)  subnqn Subsystem NQN (default: nqn.2014-08.org.nvmexpress.discovery)  Example: -r 'trtype:PCIe traddr:0000:04:00.0' for PCIe or  -r 'trtype:RDMA adrfam:IPv4 traddr:192.168.100.8 trsvcid:4420' for NVMeoF  [-e metadata configuration]  Keys:  PRACT Protection Information Action bit (PRACT=1 or PRACT=0)  PRCHK Control of Protection Information Checking (PRCHK=GUARD|REFTAG|APPTAG)  Example: -e 'PRACT=0,PRCHK=GUARD|REFTAG|APPTAG'  -e 'PRACT=1,PRCHK=GUARD'  [-s DPDK huge memory size in MB.]  [-m max completions per poll]  (default: 0 - unlimited)  [-i shared memory group ID]  [root@nvme spdk]# cd examples/nvme/perf/  [root@nvme perf]# ./perf -q 128 -s 4096 -w randread -r 'trtype:PCIe traddr:0000:00:0e.0' -t 10  Starting SPDK v18.10-pre / DPDK 18.05.0 initialization...  [ DPDK EAL parameters: perf --no-shconf -c 0x1 --legacy-mem --file-prefix=spdk\_pid3013 ]  EAL: Detected 1 lcore(s)  EAL: Detected 1 NUMA nodes  EAL: Multi-process socket /var/run/dpdk/spdk\_pid3013/mp\_socket  EAL: Probing VFIO support...  Initializing NVMe Controllers  EAL: PCI device 0000:00:0e.0 on NUMA socket 0  EAL: probe driver: 80ee:4e56 spdk\_nvme  Attaching to NVMe Controller at 0000:00:0e.0  Attached to NVMe Controller at 0000:00:0e.0 [80ee:4e56]  Associating ORCL-VBOX-NVME-VER12 (VB1234-56789 ) with lcore 0  Initialization complete. Launching workers.  Starting thread on core 0  ========================================================  Latency(us)  Device Information : IOPS MB/s Average min max  ORCL-VBOX-NVME-VER12 (VB1234-56789 ) from core 0: 227018.20 886.79 563.78 5.46 9584.33  ========================================================  Total : 227018.20 886.79 563.78 5.46 9584.33  [root@nvme perf]# |
|  |

**7.Questions:**

|  |
| --- |
| **Q:**  Crypto requires NASM version 2.12.02 or newer. Please install;  **A:** build newer nasm rpm package for arm.  [cpu@mon rpmsrc]$ wget https://www.nasm.us/pub/nasm/stable/linux/nasm-2.13.03-0.fc24.src.rpm  [cpu@mon rpmsrc]$ rpm2cpio nasm-2.13.03-0.fc24.src.rpm | cpio -div  nasm-2.13.03.tar.xz  nasm.spec  1582 blocks  [cpu@mon rpmsrc]$ mkdir -p /home/cpu/NASM/aarch64  [cpu@mon rpmsrc]$ vim ~/.rpmmacros  %\_topdir /home/cpu/NASM/aarch64 # add this line  [cpu@mon rpmsrc]$ rpmdev-setuptree  [cpu@mon rpmsrc]$ mv nasm.spec ~/NASM/aarch64/SPECS/  [cpu@mon rpmsrc]$ mv nasm-2.13.03.tar.xz ~/NASM/aarch64/SOURCES/  [cpu@mon rpmsrc]$ cd ~/NASM/aarch64/SPECS/  [cpu@mon SPECS]$ ls  nasm.spec  [cpu@mon SPECS]$ sudo yum -y install asciidoc xmlto adobe-source-sans-pro-fonts adobe-source-code-pro-fonts perl-Font-TTF perl-Sort-Versions  [cpu@mon SPECS]$ rpmbuild -ba nasm.spec  [cpu@mon SPECS]$ cd ../RPMS/aarch64/  [cpu@mon aarch64]$ sudo yum -y install nasm-2.13.03-0.el7.aarch64.rpm  **Q:** log.c:40:23: fatal error: libunwind.h: No such file or directory  #include <libunwind.h>  **A:** [yang@nvme spdk]$ sudo yum -y install libunwind libunwind-devel  **Q:** EAL: Failed to create shared memory!  **A:** Privilege problem, Use super user to run spdk if possible.  **Q**: EAL: Multi-process socket /var/run/dpdk/spdk\_pid1688/mp\_socket  EAL: No free hugepages reported in hugepages-2048Kb  **A**:  [root@nvme hello\_world]# cat /proc/meminfo | grep Huge  AnonHugePages: 14336 kB  HugePages\_Total: 0  HugePages\_Free: 0  HugePages\_Rsvd: 0  HugePages\_Surp: 0  Hugepagesize: 2048 kB  [root@nvme hello\_world]# echo 1024 > /sys/kernel/mm/hugepages/hugepages-2048kB/nr\_hugepages  [root@nvme hello\_world]# cat /proc/meminfo | grep Huge  AnonHugePages: 14336 kB  HugePages\_Total: 1024  HugePages\_Free: 1024  HugePages\_Rsvd: 0  HugePages\_Surp: 0  Hugepagesize: 2048 kB  [root@nvme hello\_world]#  **Q:**[root@nvme spdk]# ./scripts/spdkcli.py help  Traceback (most recent call last):  File "./scripts/spdkcli.py", line 5, in <module>  from configshell\_fb import ConfigShell  ModuleNotFoundError: No module named 'configshell\_fb'  **A:**  [root@nvme spdk]# pip install configshell\_fb  **Q:**  CUnit/Basic.h: No such file or directory  **A:**  [root@roce spdk]# yum -y install CUnit CUnit-devel  **Q:**  fatal error: numa.h: No such file or directory  **A:**  [root@rocet spdk]# yum -y install numactl-devel  **Q:**  fatal error: libaio.h: No such file or directory  **A:**  [root@rocet spdk]# yum -y install libaio-devel  **Q:**  fatal error: uuid/uuid.h: No such file or directory  **A:**  [root@rocet spdk]# yum -y install libuuid-devel |

**8.Appendix**

* iscsi.conf

|  |
| --- |
| # iSCSI target configuration file  #  # Please write all parameters using ASCII.  # The parameter must be quoted if it includes whitespace.  #  # Configuration syntax:  # Leading whitespace is ignored.  # Lines starting with '#' are comments.  # Lines ending with '\' are concatenated with the next line.  # Bracketed ([]) names define sections  [Global]  # Shared Memory Group ID. SPDK applications with the same ID will share memory.  # Default: <the process PID>  #SharedMemoryID 0  # Disable PCI access. PCI is enabled by default. Setting this  # option will hide any PCI device from all SPDK modules, making  # SPDK act as if they don't exist.  #NoPci Yes  # Tracepoint group mask for spdk trace buffers  # Default: 0x0 (all tracepoint groups disabled)  # Set to 0xFFFF to enable all tracepoint groups.  #TpointGroupMask 0x0  # Users may activate entries in this section to override default values for  # global parameters in the block device (bdev) subsystem.  [Bdev]  # Number of spdk\_bdev\_io structures allocated in the global bdev subsystem pool.  #BdevIoPoolSize 65536  # Maximum number of spdk\_bdev\_io structures to cache per thread.  #BdevIoCacheSize 256  [iSCSI]  # node name (not include optional part)  # Users can optionally change this to fit their environment.  NodeBase "iqn.2018-09.io.spdk"  AuthFile /usr/local/etc/spdk/auth.conf  MinConnectionsPerCore 4  # Socket I/O timeout sec. (0 is infinite)  Timeout 30  # authentication information for discovery session  # Options:  # None, Auto, CHAP and Mutual. Note that Mutual infers CHAP.  DiscoveryAuthMethod Auto  #MaxSessions 128  #MaxConnectionsPerSession 2  # iSCSI initial parameters negotiate with initiators  # NOTE: incorrect values might crash  DefaultTime2Wait 2  DefaultTime2Retain 60  # Maximum amount in bytes of unsolicited data the iSCSI  # initiator may send to the target during the execution of  # a single SCSI command.  FirstBurstLength 8192  ImmediateData Yes  ErrorRecoveryLevel 0  # Users must change the PortalGroup section(s) to match the IP addresses  # for their environment.  # PortalGroup sections define which network portals the iSCSI target  # will use to listen for incoming connections. These are also used to  # determine which targets are accessible over each portal group.  # Up to 1024 portal directives are allowed. These define the network  # portals of the portal group. The user must specify a IP address  # for each network portal, and may optionally specify a port and  # a cpumask. If the port is omitted, 3260 will be used. Cpumask will  # be used to set the processor affinity of the iSCSI connection  # through the portal. If the cpumask is omitted, cpumask will be  # set to all available processors.  # Syntax:  # Portal <Name> <IP address>[:<port>[@<cpumask>]]  [PortalGroup1]  Portal DA1 192.168.10.42:3260 # edit the IP address  # Portal DA2 192.168.10.42:3260@0xF  # Users must change the InitiatorGroup section(s) to match the IP  # addresses and initiator configuration in their environment.  # Netmask can be used to specify a single IP address or a range of IP addresses  # Netmask 192.168.1.20 <== single IP address  # Netmask 192.168.1.0/24 <== IP range 192.168.1.\*  [InitiatorGroup1]  InitiatorName ANY  Netmask 192.168.10.0/24 # filter for initiator which can connect to this target  # NVMe configuration options  [Nvme]  # NVMe Device Whitelist  # Users may specify which NVMe devices to claim by their transport id.  # See spdk\_nvme\_transport\_id\_parse() in spdk/nvme.h for the correct format.  # The second argument is the assigned name, which can be referenced from  # other sections in the configuration file. For NVMe devices, a namespace  # is automatically appended to each name in the format <YourName>nY, where  # Y is the NSID (starts at 1).  TransportID "trtype:PCIe traddr:0000:00:0e.0" Nvme0 # this is the address of your NVMe devices. Use command ‘lspci’ to seek the number.  # TransportID "trtype:PCIe traddr:0000:01:00.0" Nvme1  # The number of attempts per I/O when an I/O fails. Do not include  # this key to get the default behavior.  RetryCount 4  # Timeout for each command, in microseconds. If 0, don't track timeouts.  TimeoutUsec 0  # Action to take on command time out. Only valid when Timeout is greater  # than 0. This may be 'Reset' to reset the controller, 'Abort' to abort  # the command, or 'None' to just print a message but do nothing.  # Admin command timeouts will always result in a reset.  ActionOnTimeout None  # Set how often the admin queue is polled for asynchronous events.  # Units in microseconds.  AdminPollRate 100000  # Disable handling of hotplug (runtime insert and remove) events,  # users can set to Yes if want to enable it.  # Default: No  HotplugEnable No  # Set how often the hotplug is processed for insert and remove events.  # Units in microseconds.  HotplugPollRate 0  # Users may change this section to create a different number or size of  # malloc LUNs.  # If the system has hardware DMA engine, it can use an IOAT  # (i.e. Crystal Beach DMA) channel to do the copy instead of memcpy  # by specifying "Enable Yes" in [Ioat] section.  # Offload is disabled by default even it is available.  [Malloc]  # Number of Malloc targets  NumberOfLuns 3  # Malloc targets are 128M  LunSizeInMB 128  # Block size. Default is 512 bytes.  BlockSize 4096  # Users can use offload by specifying "Enable Yes" in this section  # if it is available.  # Users may use the whitelist to initialize specified devices, IDS  # uses BUS:DEVICE.FUNCTION to identify each Ioat channel.  [Ioat]  Enable No  Whitelist 00:04.0  Whitelist 00:04.1  # Users must change this section to match the /dev/sdX devices to be  # exported as iSCSI LUNs. The devices are accessed using Linux AIO.  # The format is:  # AIO <file name> <bdev name> [<block size>]  # The file name is the backing device  # The bdev name can be referenced from elsewhere in the configuration file.  # Block size may be omitted to automatically detect the block size of a disk.  [AIO]  # AIO /dev/sdb AIO0  AIO /dev/sdb AIO0 # a block device must exist  # AIO /tmp/myfile AIO2 4096  # PMDK libpmemblk-based block device  [Pmem]  # Syntax:  # Blk <pmemblk pool file name> <bdev name>  Blk /path/to/pmem-pool Pmem0  # The Split virtual block device slices block devices into multiple smaller bdevs.  # [Split]  # Syntax:  # Split <bdev> <count> [<size\_in\_megabytes>]  # Split Malloc1 into two equally-sized portions, Malloc1p0 and Malloc1p1  # Split Malloc1 2  # Split Malloc2 into eight 1-megabyte portions, Malloc2p0 ... Malloc2p7,  # leaving the rest of the device inaccessible  # Split Malloc2 8 1  # Users should change the TargetNode section(s) below to match the  # desired iSCSI target node configuration.  # TargetName, Mapping, LUN0 are minimum required  [TargetNode1]  TargetName disk1  TargetAlias "Data Disk1"  Mapping PortalGroup1 InitiatorGroup1  AuthMethod Auto  AuthGroup AuthGroup1  # Enable header and data digest  # UseDigest Header Data  UseDigest Auto  # Use the first malloc target  # LUN0 Malloc0  # Using the first AIO target  LUN0 AIO0  LUN1 Nvme0n1  # Using the second storage target  # LUN2 AIO1  # Using the third storage target  # LUN3 AIO2  QueueDepth 128  #[TargetNode2]  # TargetName disk2  # TargetAlias "Data Disk2"  #Mapping PortalGroup1 InitiatorGroup1  # AuthMethod Auto  # AuthGroup AuthGroup1  # UseDigest Auto  # LUN0 Nvme0n1  # QueueDepth 32 |

* SPDK API
* fio\_softroce.job

|  |
| --- |
| [global]  invalidate=1  norandommap=1  thread=1  rw=randrw  runtime=10  ioengine=libaio  direct=1  bs=4096  size=1G  iodepth=128  group\_reporting  time\_based=1  [job0]  filename=/dev/nvme0n1 |

**9.Reference:**

* URL

*[1].SOURCE CODE:* [*https://github.com/spdk/spdk*](https://github.com/spdk/spdk)

*[2].ROADMAP:* [*http://spdk.io/roadmap/*](http://spdk.io/roadmap/)

*[3].SPDK fio:https://www.jianshu.com/p/3a38a4d10d94*

*[4].* Introduction to the Storage Performance Development Kit (SPDK):*https://software.intel.com/en-us/articles/introduction-to-the-storage-performance-development-kit-spdk?language=en-us&https=1*

* Explanation

*M**MIO:*

Memory-mapped I/O(MMIO) uses the same address space to address both memory and I/O devices. The memory and registers of the I/O devices are mapped to (associated with) address values. the CPU instructions used to access the memory can also be used for accessing devices(https://en.wikipedia.org/wiki/Memory-mapped\_I/O).

*PMIO:*

Port-mapped I/O(PMIO) often uses a special class of CPU instructions designed specifically for performing I/O, such as the in and out instructions. I/O devices have a separate address space from general memory, either accomplished by an extra "I/O" pin on the CPU's physical interface, or an entire bus dedicated to I/O(https://en.wikipedia.org/wiki/Memory-mapped\_I/O).

**10.Compile SPDK for Arm:**

* v18.04

|  |
| --- |
| [cpu@mon spdk-v18.04]$ git clone -b v18.04 https://github.com/spdk/spdk.git  [cpu@mon spdk-v18.04]$ cd spdk/  [cpu@mon spdk]$ git submodule update --init  [cpu@mon spdk]$ sudo scripts/pkgdep.sh  [cpu@mon spdk]$ ./configure  [cpu@mon spdk]$ make |

* v18.07

|  |
| --- |
| [cpu@mon spdk-v18.07]$ git clone -b v18.07 https://github.com/spdk/spdk.git  [cpu@mon spdk-v18.07]$ cd spdk/  [cpu@mon spdk]$ git submodule update --init  [cpu@mon spdk]$ sudo scripts/pkgdep.sh  [cpu@mon spdk]$ ./configure  [cpu@mon spdk]$ make  Q:  error: unknown field ‘ndo\_change\_mtu\_rh74’ specified in initializer  A:  [cpu@mon spdk]$ vim dpdk/kernel/linux/kni/compat.h  107 #define ndo\_change\_mtu ndo\_change\_mtu\_rh74 # comment this line  Q:  Error: constant expression required at operand 1 -- `dsb lt'  A:  [cpu@mon spdk]$ vim CONFIG  62 CONFIG\_TESTS?=n # edit to n |

* v18.10(master)

|  |
| --- |
| [cpu@mon spdk-v18.07]$ git clone https://github.com/spdk/spdk.git  [cpu@mon spdk-v18.04]$ cd spdk/  [cpu@mon spdk]$ vim scripts/pkgdep.sh  # comment these lines below  95 # make  96 # make install  [cpu@mon spdk]$ sudo scripts/pkgdep.sh  [cpu@mon spdk]$ cd ../../  [cpu@mon compile]$ mkdir dpdk/  [cpu@mon dpdk]$ git clone https://github.com/DPDK/dpdk.git  [cpu@mon dpdk]$ cd dpdk/  [cpu@mon dpdk]$ vim kernel/linux/kni/compat.h  # comment this line  107 //#define ndo\_change\_mtu ndo\_change\_mtu\_rh74  [cpu@mon dpdk]$ make config T=arm64-armv8a-linuxapp-gcc  [cpu@mon dpdk]$ make install T=arm64-armv8a-linuxapp-gcc  Q:  home/cpu/SPDK/compile/dpdk/dpdk/drivers/event/octeontx/timvf\_worker.c:88:1: error: could not split insn  A: recompile GCC for arm. |

**11.GCC:**

* compile

|  |
| --- |
| [cpu@mon GCC]$ wget http://mirrors.concertpass.com/gcc/releases/gcc-7.2.0/gcc-7.2.0.tar.gz  [cpu@mon GCC]$ tar -zxvf gcc-7.2.0.tar.gz  [cpu@mon GCC]$ cd gcc-7.2.0  [cpu@mon gcc-7.2.0]$ sudo ./contrib/download\_prerequisites  [cpu@mon gcc-7.2.0]$ ./configure --prefix=/home/cpu/GCC/gcc-6.2-linaro/ -enable-checking=release -enable-languages=c,c++ -disable-multilib  [cpu@mon gcc-7.2.0]$ make  [cpu@mon gcc-7.2.0]$ make install  Q:  error: GNAT is required to build ada  A:  [cpu@mon gcc-7.2.0]$ sudo yum install gcc-gnaty |