

# Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

# Zoned Emulation Support for QEMU

Presenter: Sam Li



LINUX PLUMBERS CONFERENCE

Vienna, Austria  
Sept. 18-20, 2024

# Collaborators

- Damien Le Moal
- Stefan Hajnoczi
- Dmitry Fomichev
- Hannes Reinecke
- The QEMU community



LINUX PLUMBERS CONFERENCE

Vienna, Austria  
Sept. 18-20, 2024

# 1 State of zoned storage in QEMU

- Virtio-scsi -> attach a zoned device (e.g. ZBC or ZAC HDD) to QEMU
- Virtio-blk emulation -> attach a zoned device or a qcow2 image file to QEMU
- PCI device passthrough -> attach an NVMe PCI device to QEMU
- NVMe device emulation -> ZNS emulation



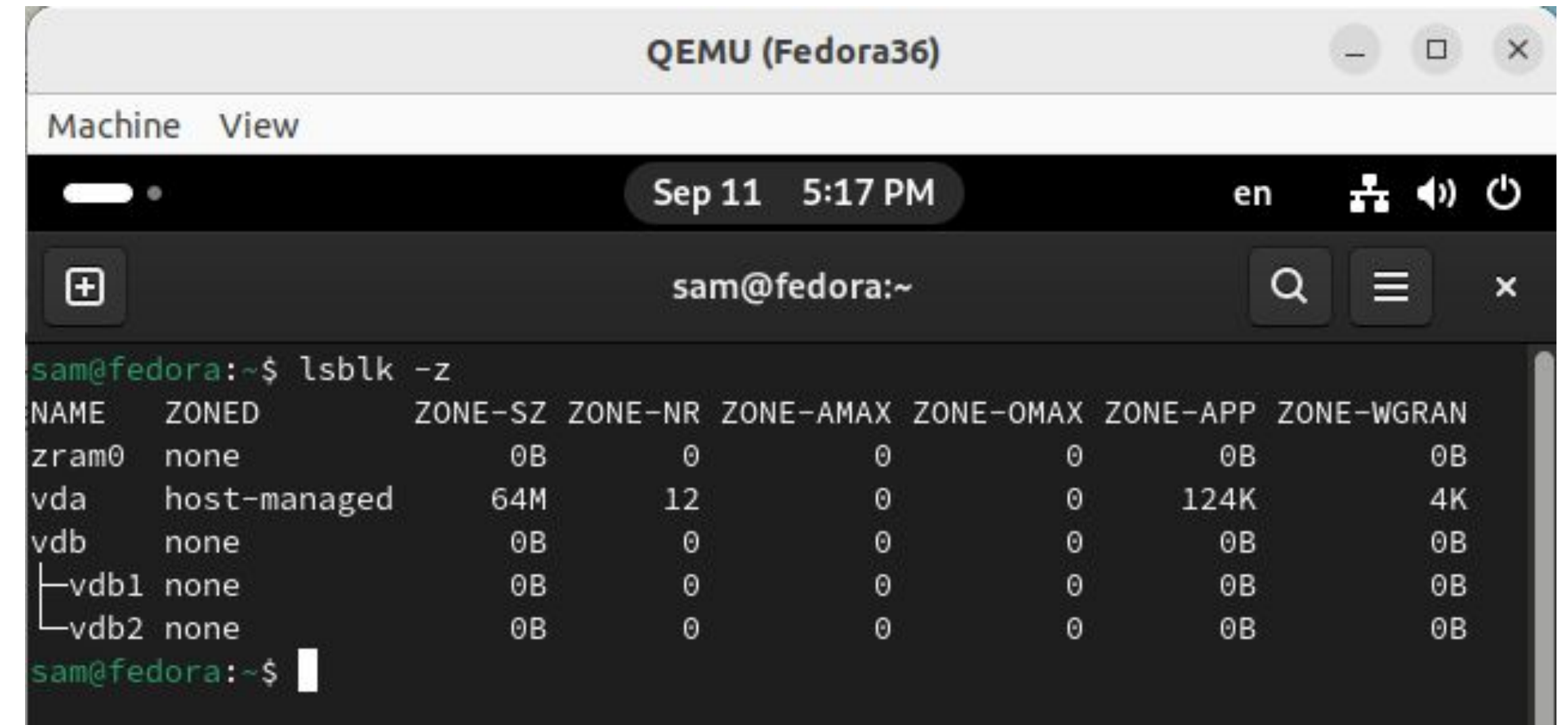
# How to play with the emulated zoned device?

Environment:

- Qemu: v8.1.0 supports zoned device via virtio-blk emulation
- Linux: suggested version > v6.3-rc1

Steps:

1. Create a zoned block device on the host
2. Boots a VM
3. Check if the zoned device is in the guest



```
QEMU (Fedora36)
Machine View
Sep 11 5:17 PM
en
sam@fedora:~
sam@fedora:~$ lsblk -z
NAME        ZONED          ZONE-SZ  ZONE-NR  ZONE-AMAX  ZONE-OMAX  ZONE-APP  ZONE-WGRAN
zram0      none              0B         0         0           0         0B         0B
vda        host-managed    64M        12         0           0        124K         4K
vdb        none              0B         0         0           0         0B         0B
├─vdb1     none              0B         0         0           0         0B         0B
└─vdb2     none              0B         0         0           0         0B         0B
sam@fedora:~$
```



```
root@fedora:/home/sam
[ 1.548557] Console: switching to colour dummy device 80x25
[ 1.549599] [drm] features: -virgl +edid -resource_blob -host_visible
[ 1.549600] [drm] features: -context_init
[ 1.551228] [drm] number of scanouts: 1
[ 1.551231] [drm] number of cap sets: 0
[ 1.555743] virtio_blk virtio1: [vda] 196608 4096-byte logical blocks (805
MB/768 MiB)
[ 1.555783] [drm] Initialized virtio_gpu 0.1.0 0 for 0000:00:01.0 on minor
0
```



# Store files to a zoned storage using btrfs on a QEMU VM

1. Open a QEMU VM
2. Operations on the VM:
  - \$ `mkfs.btrfs -f /dev/vdb`
  - \$ `mount -t btrfs /dev/vdb /mnt`
  - \$ `btrfs subvolume create /mnt/zoned`
  - \$ `echo "echo hello zbd" > hello.txt`
  - \$ `mv hello.txt /mnt/zoned`
3. Shut down the vm and restart it
4. mount the btrfs again and check if the hello.txt file is there

```
root@fedora:/home/sam/zbd
Devices:
  ID      SIZE  ZONES  PATH
  1      768.00MiB  12  /dev/vdb

root@fedora:/home/sam/zbd# lsblk -f
NAME      FSTYPE FSVER LABEL UUID                                 FSAVAIL FSUSE% MOUNTPO
INTS
zram0
vda       btrfs          6399cf3b-d1b1-4703-8015-6bba6c8f1819  511M    1% /mnt
vdb       btrfs          5ff3cd32-35c8-4e5c-b7a0-c937a3f5e6f1
vdc
├─vdc1 xfs          f9bf9fa3-da18-42a4-b648-368bcb5aefda  74.5G   22% /
└─vdc2 swap      1          3451cbe6-c3de-46e3-bbab-b20c04203b9a
root@fedora:/home/sam/zbd# blkzone report /dev/vdb
start: 0x000000000, len 0x020000, cap 0x020000, wptr 0x000008 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000020000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000040000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000060000, len 0x020000, cap 0x020000, wptr 0x000120 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000080000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x0000a0000, len 0x020000, cap 0x020000, wptr 0x000040 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x0000c0000, len 0x020000, cap 0x020000, wptr 0x000040 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x0000e0000, len 0x020000, cap 0x020000, wptr 0x0001a0 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000100000, len 0x020000, cap 0x020000, wptr 0x0001a0 reset:0 non-seq:0, zc
ond: 2(oi) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000120000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000140000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
start: 0x000160000, len 0x020000, cap 0x020000, wptr 0x000000 reset:0 non-seq:0, zc
ond: 1(em) [type: 2(SEQ_WRITE_REQUIRED)]
root@fedora:/home/sam/zbd#
```

## 2 Config: two block backends to pick from

1. Null\_blk device: modprobe null\_blk nr\_devices=1 zoned=1

[https://zonedstorage.io/docs/getting-started/zbd-emulation#zoned-block-device-emulation-with-null\\_blk](https://zonedstorage.io/docs/getting-started/zbd-emulation#zoned-block-device-emulation-with-null_blk)

2. A qcow2 file with zoned format

```
jli@groves:~/Desktop/infra/qemu$ ./build/qemu-img create -f qcow2 zbc.qcow2
-o size=768M -o zone.size=64M -o zone.capacity=64M -o zone.conventional_zones=0 -o zone.max_append_bytes=4096 -o zone.max_open_zones=6 -o zone.max_active_zones=8 -o zone.mode=host-managed
Formatting 'zbc.qcow2', fmt=qcow2 cluster_size=65536 extended_l2=off compression_type=zlib zone.mode=host-managed zone.size=67108864 zone.capacity=67108864 zone.conventional_zones=0 zone.max_append_bytes=4096 zone.max_active_zones=8 zone.max_open_zones=6 size=805306368 lazy_refcounts=off refcount_bits=16
jli@groves:~/Desktop/infra/qemu$
```



# QEMU Command line

1. From the doc, to expose the host's zoned block device through virtio-blk, the command line can be:

```
-blockdev node-name=drive0,driver=host_device,filename=/dev/nullb0,cache.direct=on \  
-device virtio-blk-pci,drive=drive0 \  
\
```

2. To expose the qcow2 file with zoned format through virtio-blk, the command line can be:

```
-blockdev node-name=drive1,driver=qcow2,file.driver=file,file.filename=test.qcow2 \  
\
```

3. To expose the qcow2 file as an emulated zns drive, the command line can be:

```
-drive file=${znsimg},id=nvmezns0,format=qcow2,if=none \  
-device nvme-ns,drive=nvmezns0,bus=nvme0,nsid=1,uuid=xxx \  
\
```





# 3 Develop, test & debug

Test suits for zbd

- Qemu-io or qemuio-tests (host)  
`$ tests/qemu-iotests/check [<test-case>]`
- [zonefs-tools](#)  
`$ tests/zonefs-tests.sh /dev/vda`
- [fio/test-zbd](#), [blktests](#)
- dd (zone append), blkzone commands  
`$ dd if=/dev/zero of=/mnt/seq/0  
oflag=direct,append bs=4096 count=1  
conv=notrunc`

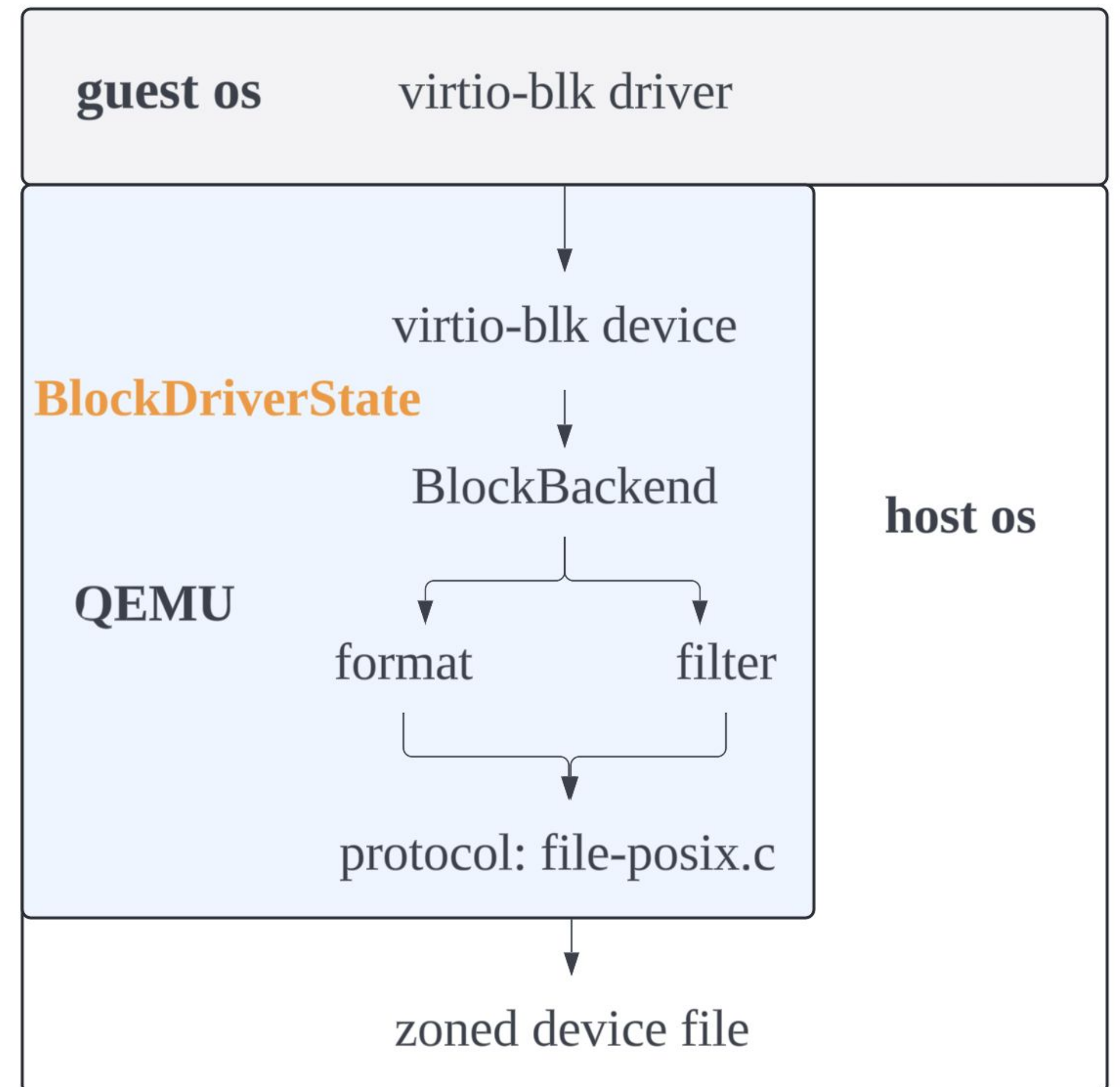
Debugging tools

- address sanitizer (config with `--enable-sanitizers`) or valgrind (host)
- Gdb (bt) + coredump debug control (host)  
`$ coredumpctl debug`
- ftrace/blktrace
- strace

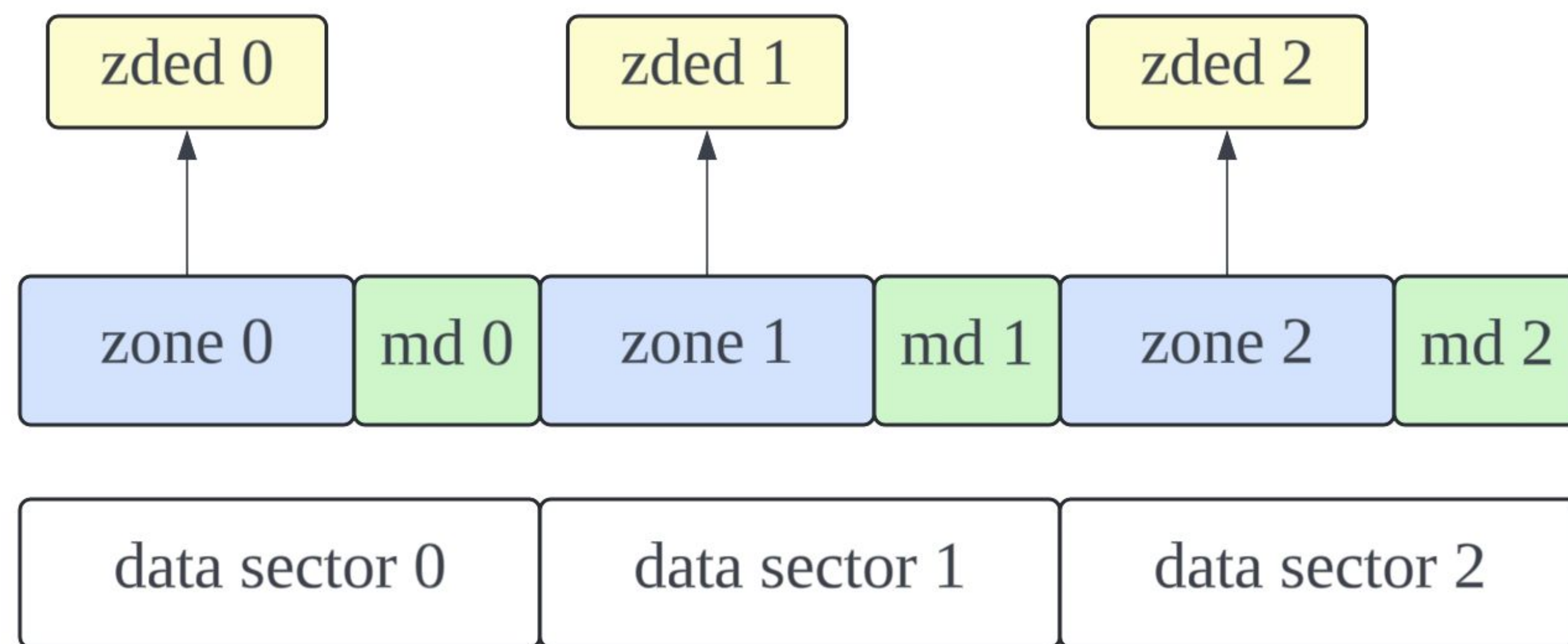


# 4 Virtio-blk: zoned emulation

- Zoned models: conventional, host-managed, host-aware
- Zone management command: report, open, close, finish
- Zone append: uses write pointer emulation  
|zone type (1)| write pointer (63)|



# 5 QCow2: full emulation

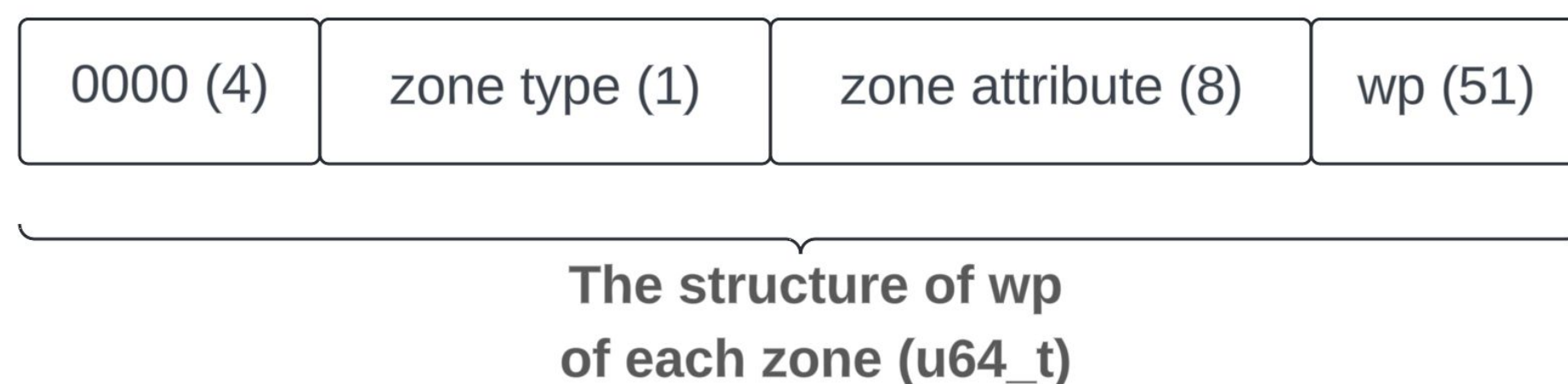


**in memory data buffer**





# 6 Persistent states for ZNS emulation



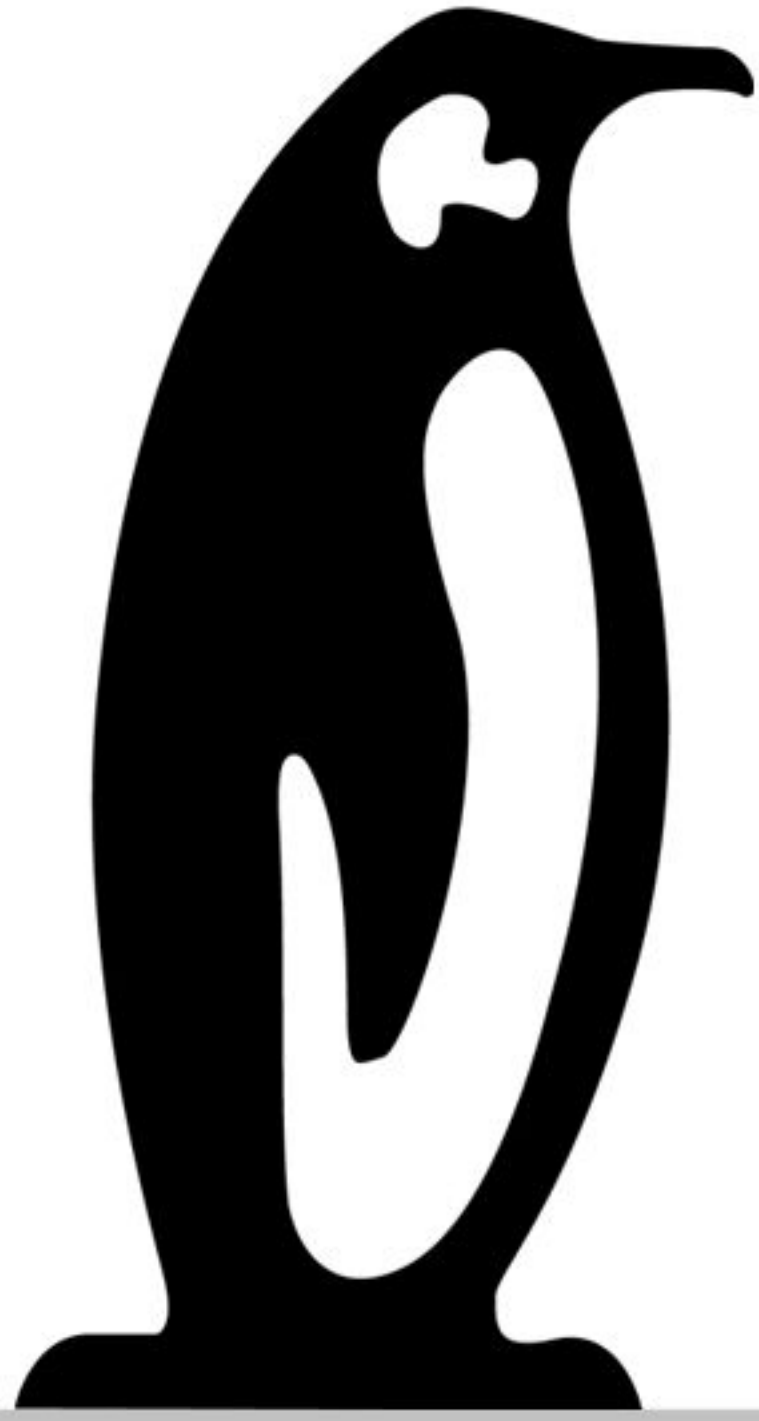
- Address translation
  - Nvme\_blk\_zone\_append
  - > dma\_blk\_zone\_append (DMA ops)
  - > blk\_aio\_zone\_append (block layer API)
- Zone attributes
  - ZRWA (zone random write area)
  - ZDED (zone descriptor extension data)



# Contributions

- Added zoned storage APIs to the block layer
- Implemented zoned storage support in virtio-blk emulation
- Add full zoned storage emulation to qcow2 driver (ongoing)
- Add persistence to NVMe ZNS emulation (ongoing)





# Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

