# NVMe Emulation in bhyve

Chuck Tuffli
chuck@freebsd.org
bhyvecon Ottawa, Ontario 2019

# bhyve NVMe emulation

```
bhyve -s 00:05:00,nvme,/dev/zvol/t/z1
```


Device path options:

- File
- Block device (physical or ZVOL)
- RAM

# bhyve NVMe emulation

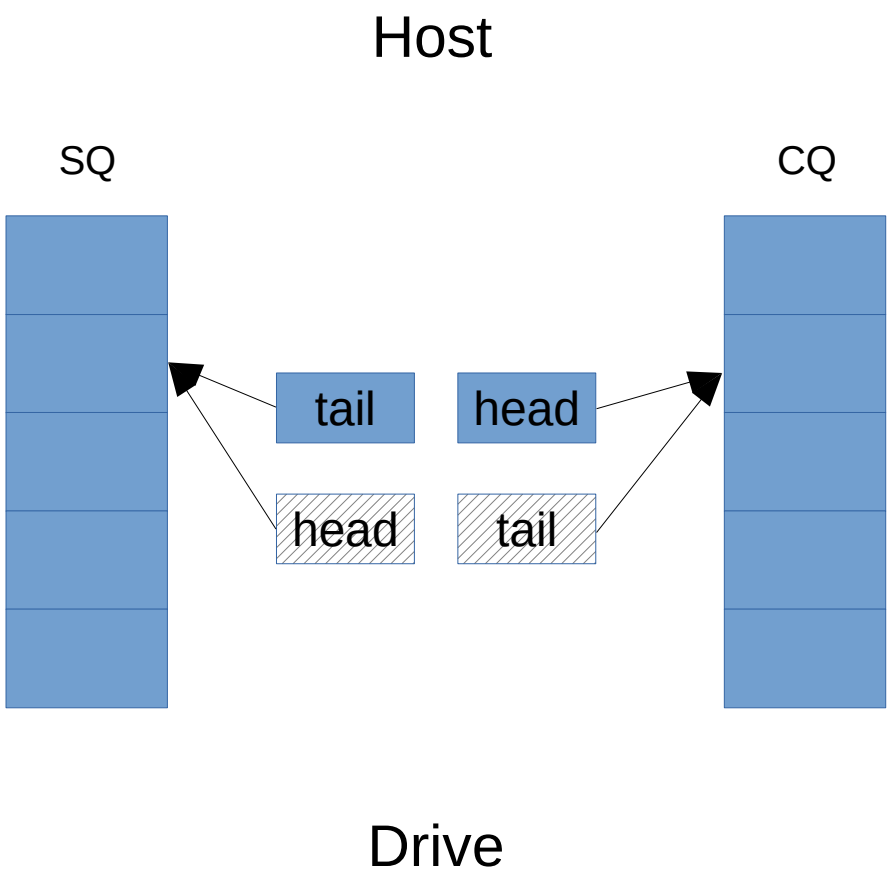
```
bhyve -s 00:05:00,nvme,/dev/zvol/t/z1
      -s 00:06:00,nvme,/dev/zvol/t/z2
      -s 00:07:00,nvme,/dev/zvol/t/z3
      -s 00:08:00,nvme,/dev/zvol/t/z4
```

# Additional NVMe Configuration

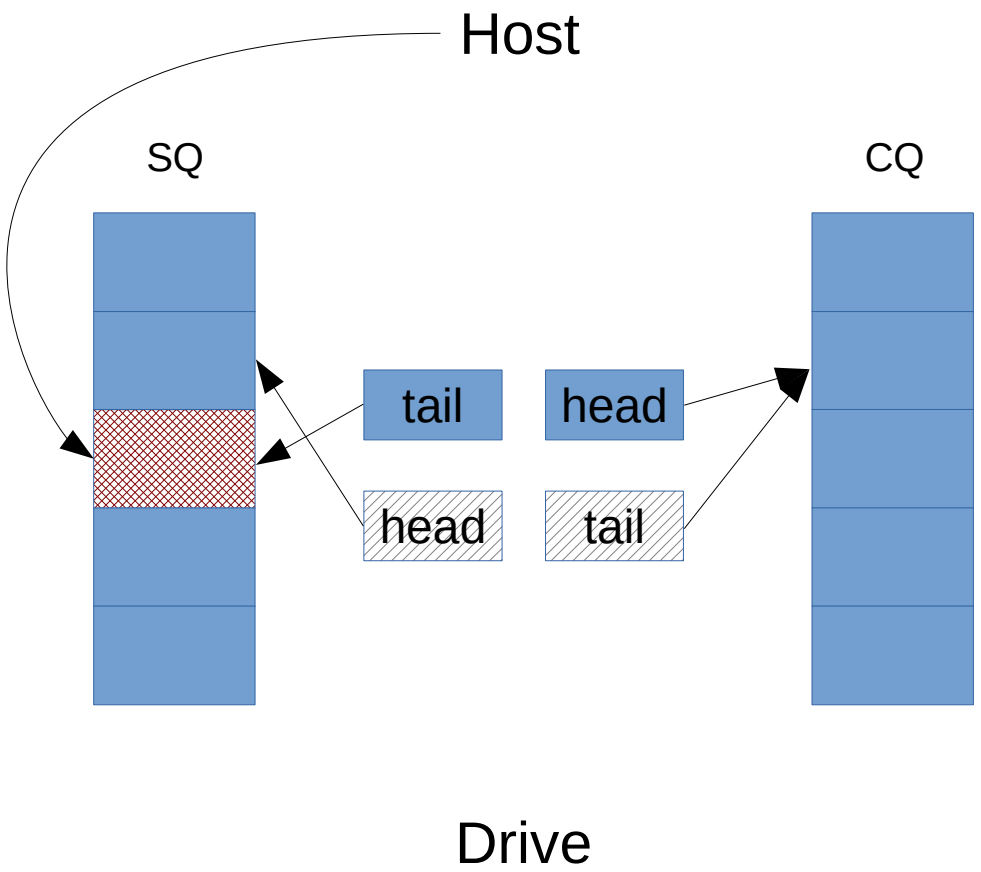
- *maxq* Max number of queues.
- *qsz* Max elements in each queue.
- *ioslots* Max number of concurrent I/O requests.
- *sectsz* Sector size (defaults to block sector size).
- *ser* Serial number with maximum 20 characters.

```
-s 00:05:00,nvme,/dev/zvol/t/z1,maxq=64,qsz=256,ser=bada55
```

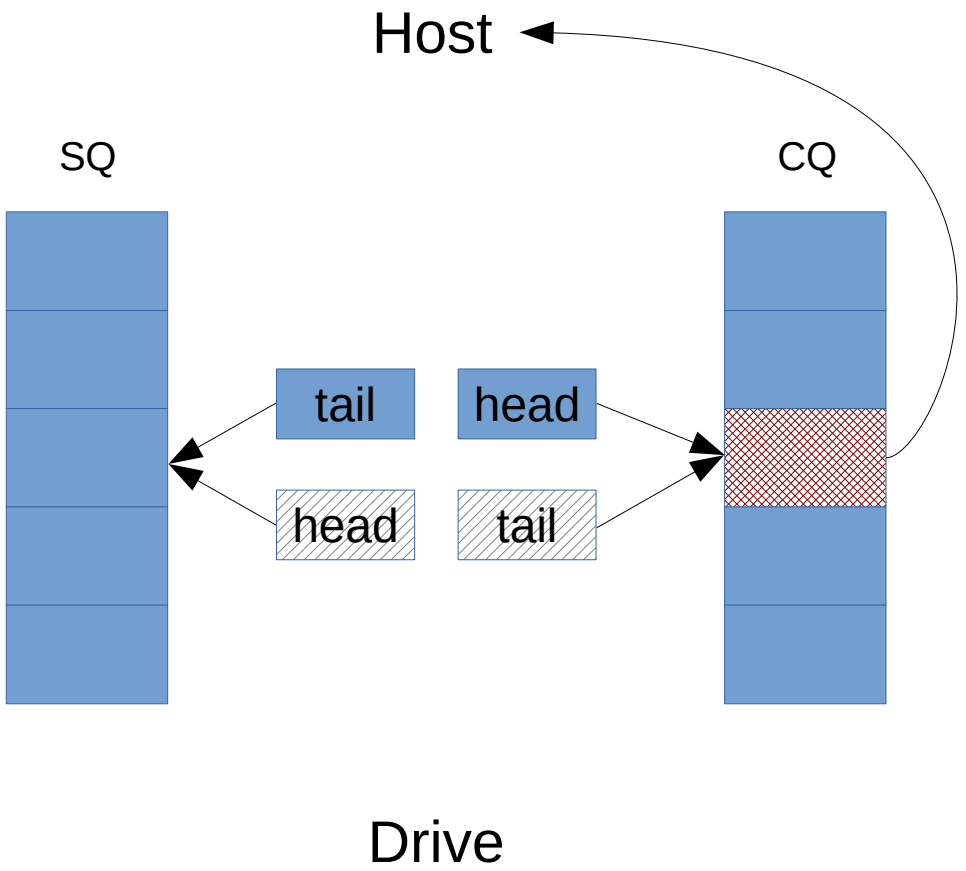
# Host / Drive Communication



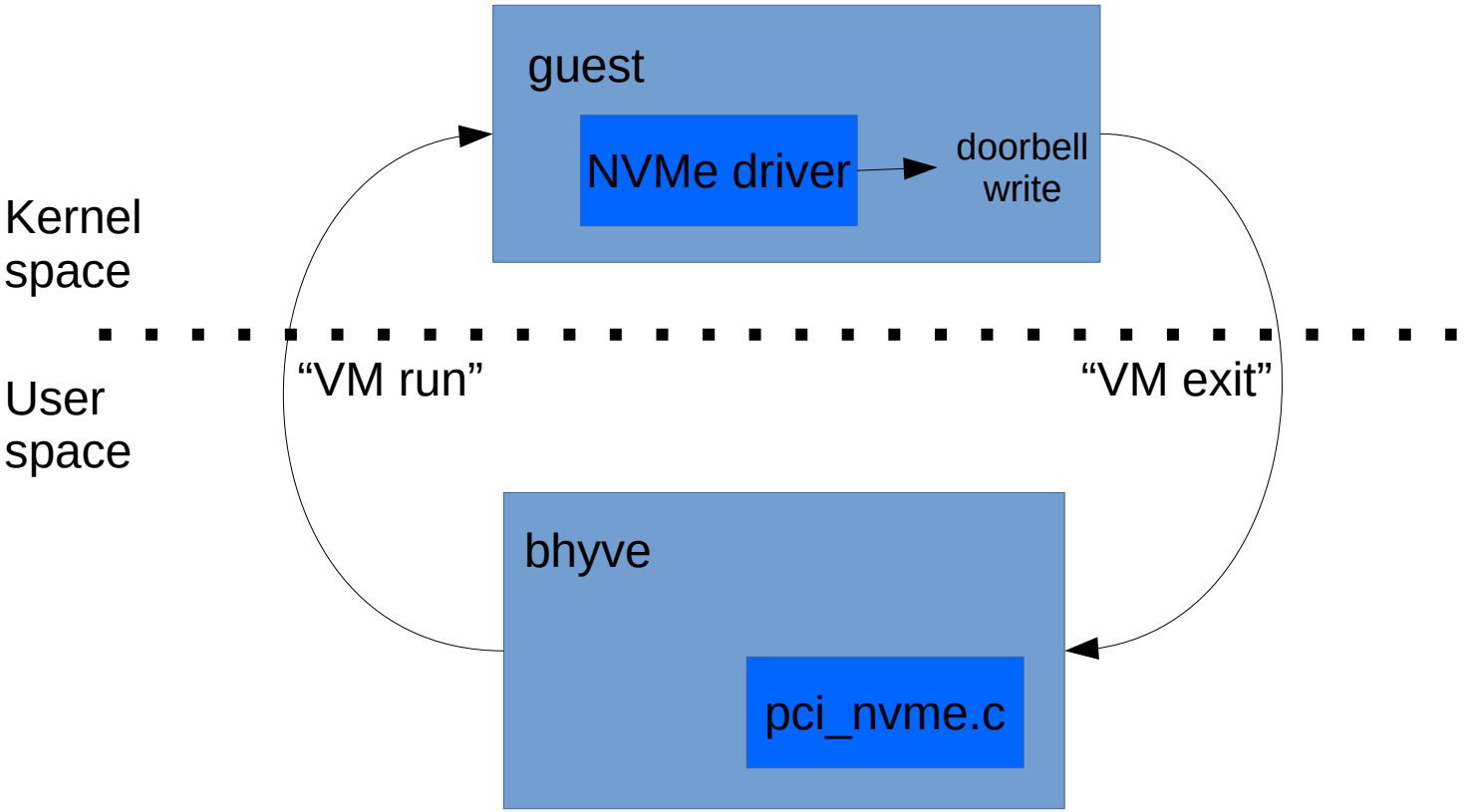
# Host / Drive Communication



# Host / Drive Communication



# bhyve Device Emulation



# Plug-in Framework

```
plugin_tap_t nvme_plugin_admin_decode[1] = {
     [0] = {
          .name =   "nvme:admin:decode",
          .cb =     NULL,
          .enable = false
     }
};
PLUGIN_TAP_SET(nvme_plugin_admin_decode);

static void
pci_nvme_handle_admin_cmd(struct pci_nvme_softc* sc, uint64_t value)
{
     ...
     while (sqhead != atomic_load_acq_short(&sq->tail)) {
          cmd = &(sq->qbase)[sqhead];
          compl.status = 0;

          if (nvme_plugin_admin_decode->enable) {
               nvme_plugin_admin_decode_callback_t cb = nvme_plugin_admin_decode->cb;
               cb(NVME_BDF(), cmd, &compl, 0);
          }

          switch (cmd->opc) {
          case NVME_OPC_DELETE_IO_SQ:
     ...
```

# The Plug-in

```
#define MAX_WRITES 5000 /* Fail after 5,000 Write commands */

static int
io_ro_fail(uint32_t bdf, struct nvme_command *cmd, struct nvme_completion *cmp, uint32_t sqid)
{
     int rc = 0;


     if (cmd->opc == NVME_OPC_WRITE) {
          if (n_writes >= MAX_WRITES) {
               if (n_writes > (MAX_WRITES + 10)) n_writes = 0;


               NVME_STATUS_SET(cmp->status,
                         NVME_SCT_COMMAND_SPECIFIC,
                         NVME_SC_ATTEMPTED_WRITE_TO_RO_PAGE);
               rc = 1;
          }
          n_writes++;
     }


     return (rc);
}
```

# The Bugs

- Return all of the completion
- Uh, is that zero’s based or one’s?
- Wait, you actually check that?
- Is that my memory or yours?

# New and Shiny

- PCIe Capability Registers

```
nda0: nvme version 1.3 x63 (max x63) lanes PCIe Gen15 (max Gen15) link
```

- EUI64 (IEEE Extended Unique Identifier)

```
Boot0001* EFI Misc Device PciRoot(0x0)/Pci(0x4,0x0)/NVMe(0x1,00-00-00-00-00-00-00-00)
Boot0002* EFI Misc Device 1 PciRoot(0x0)/Pci(0x5,0x0)/NVMe(0x1,00-00-00-00-00-00-00-00)
Boot0003* EFI Misc Device 2 PciRoot(0x0)/Pci(0x6,0x0)/NVMe(0x1,00-00-00-00-00-00-00-00)
Boot0004* EFI Misc Device 3 PciRoot(0x0)/Pci(0x7,0x0)/NVMe(0x1,00-00-00-00-00-00-00-00)
```