Frankenstein’s Disk Drive

Chuck Tuffli
chuck@freebsd.org

BSDCan Conference
University of Ottawa, Ontario
May 17-18, 2019


Charles Ogle
TIONARY —
DO NOT USE THIS BRAIN!
- ABNORMAL —

The Impetus

Michael Dexter
@michaeldexter

Anyone seen an NVMe card enter read only mode and freak out with unexplained write activity that paralyzes the rest of the system?

9:23 AM · Sep 13, 2018 · Twitter for Android

2 Retweets **2** Likes

The Disclaimer

- Discussion here is *NOT A CRITIQUE*
- Bugs are normal
- All handled the abuse well

Testing Drivers

“if error”

```
static void
nvme_qpair_complete_tracker(struct nvme_qpair *qpair, struct nvme_tracker *tr,
    struct nvme_completion *cpl, error_print_t print_on_error)
{
        struct nvme_request    *req;
        boolean_t              retry, error;

        req = tr->req;
        error = nvme_completion_is_error(cpl);

        if (error) {
                ??????
        }
        ...
}
```

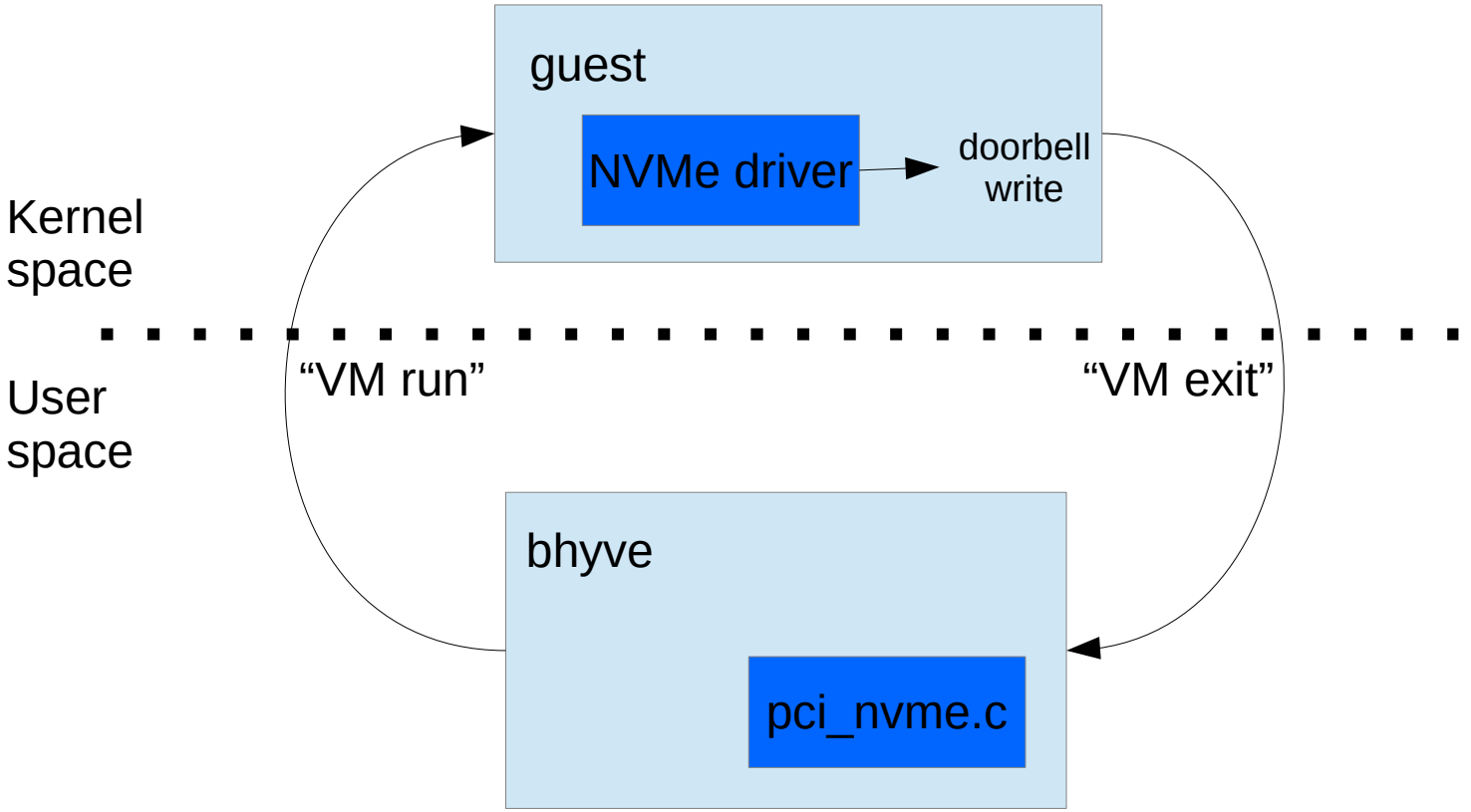
The Usual

- Add “error modes” to driver
  - It can (accidentally) go boom
  - Modes are static
  - Kernel == pain
- “Mocking”
  - Can control stimulus
  - Environment is different

The Alternative

- Virtual hardware
  - bhyve emulated device
- Customized firmware
  - User defined plug-ins

bhyve NVMe™ Emulation



NVMe 101

NVMe Queues

- Producer-consumer queue
- Host memory + head / tail registers
  - Produce to tail
  - Consume from head
  - Infer work to do if (head != tail)
- Allocate in pairs for bi-directional messaging
  - Submission Queue : host to device
  - Completion Queue : device to host

Queue Creation

Completion Queue (CQ)

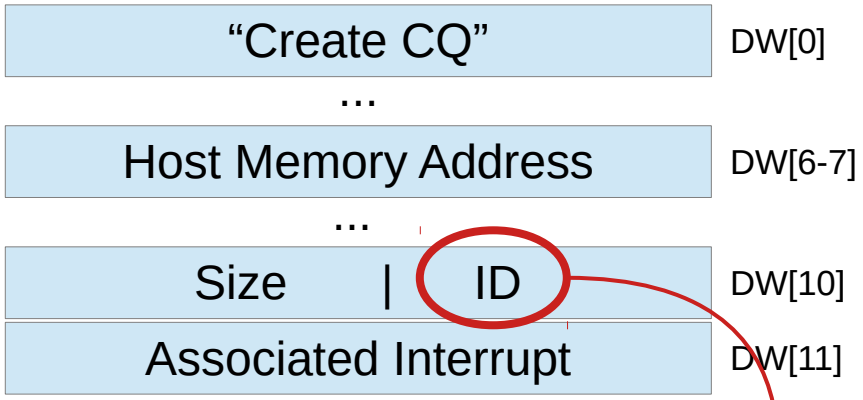
“Create CQ”	DW[0]
...	
Host Memory Address	DW[6-7]
...	
Size | ID	DW[10]
Associated Interrupt	DW[11]

Submission Queue (SQ)

“Create SQ”	DW[0]
...	
Host Memory Address	DW[6-7]
...	
Size | ID	DW[10]
Associated CQ	DW[11]

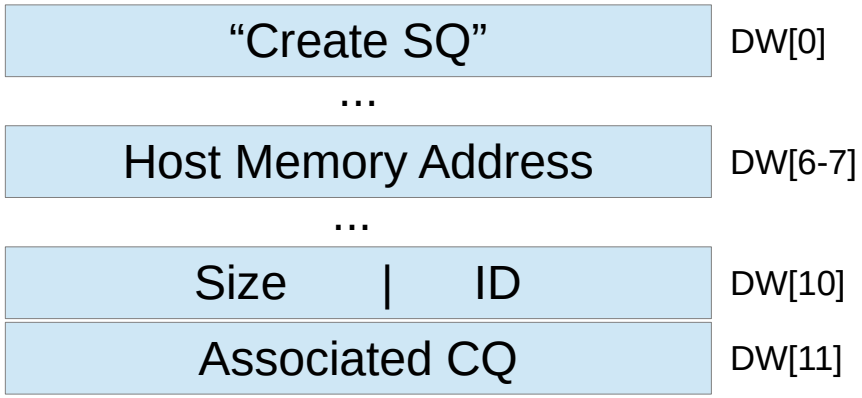
Queue Creation

Completion Queue (CQ)



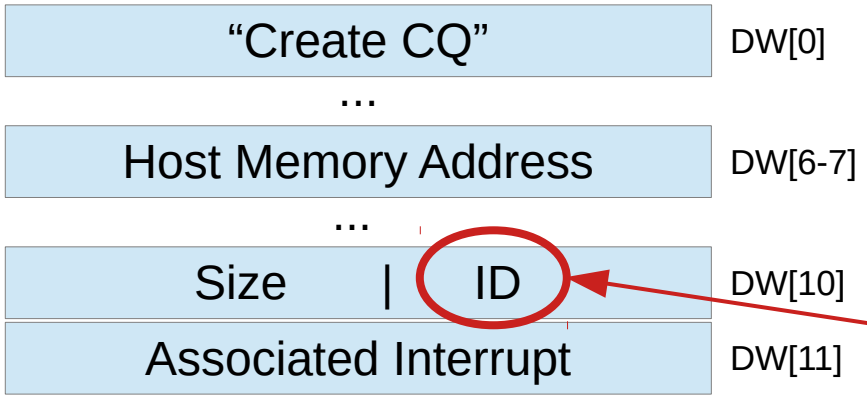
Register
0x10*xx*

Submission Queue (SQ)



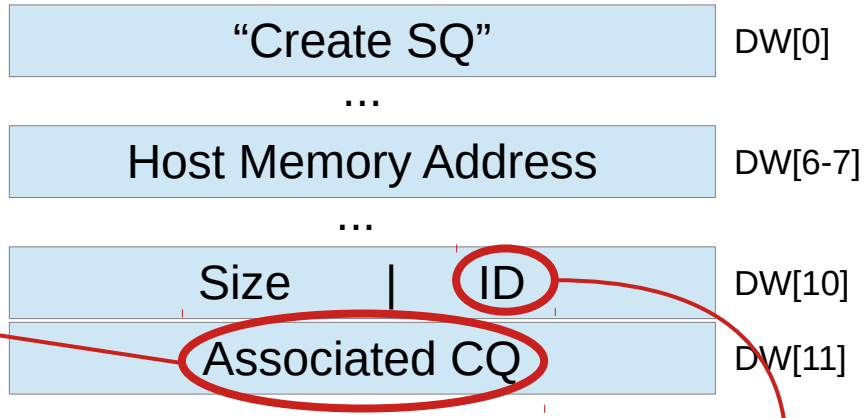
Queue Creation

Completion Queue (CQ)



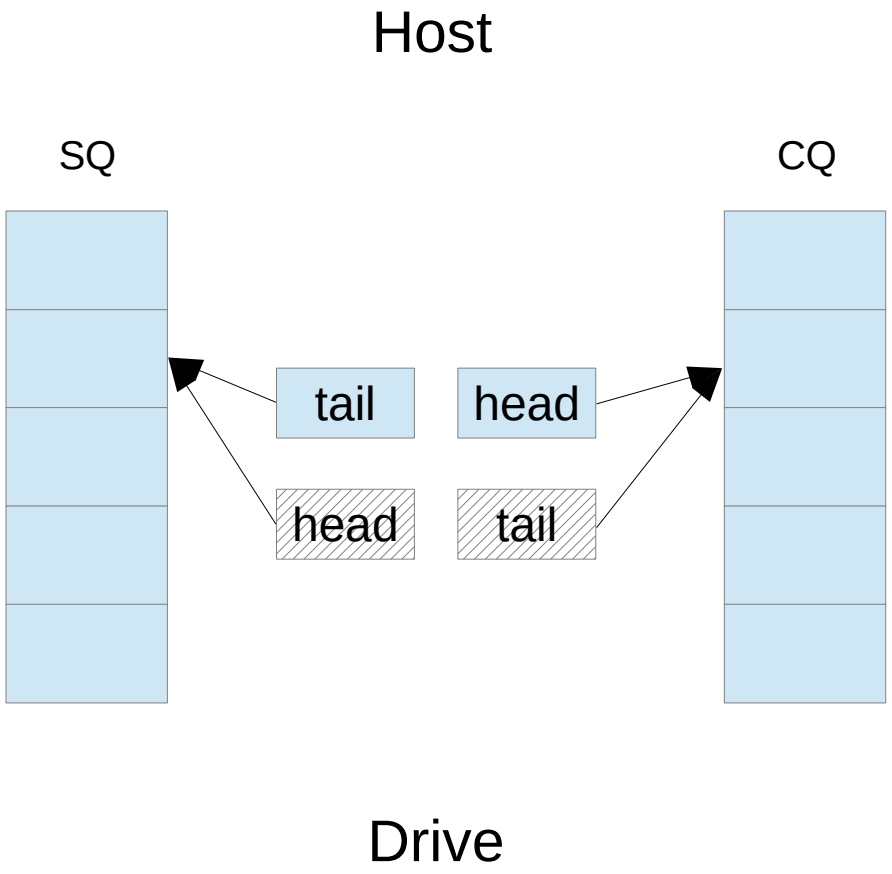
Register 0x10*xx*

Submission Queue (SQ)

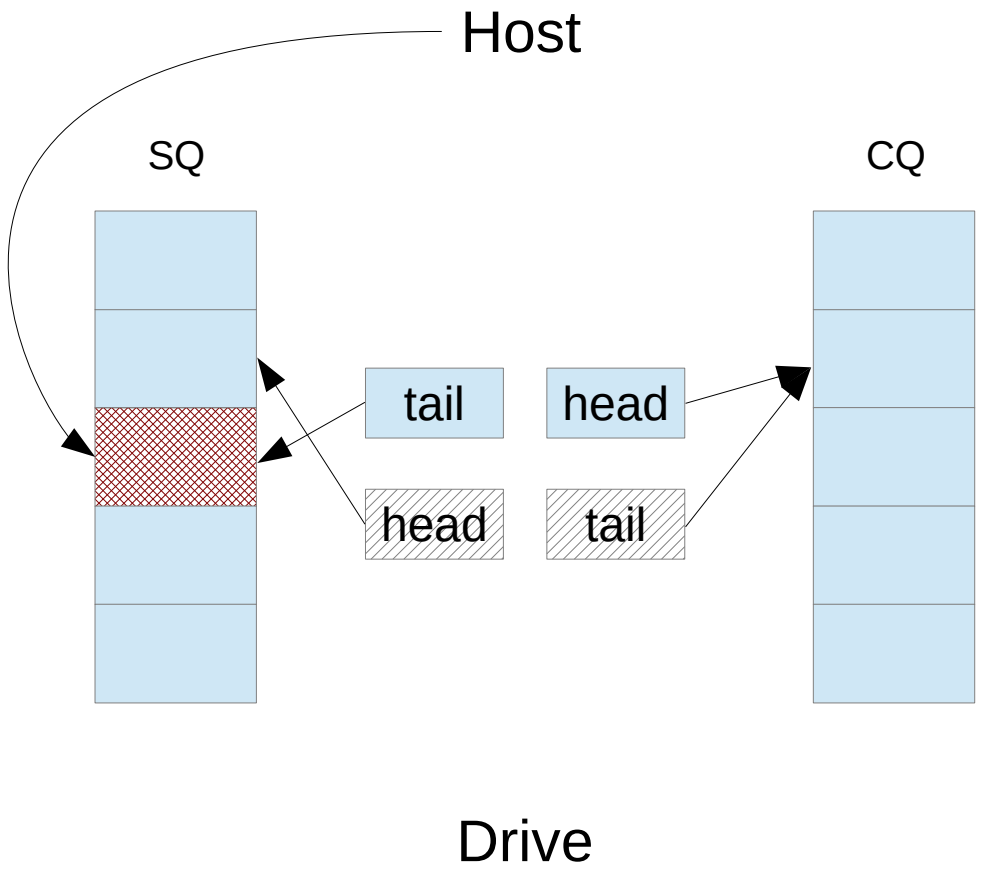


Register 0x10*xx*

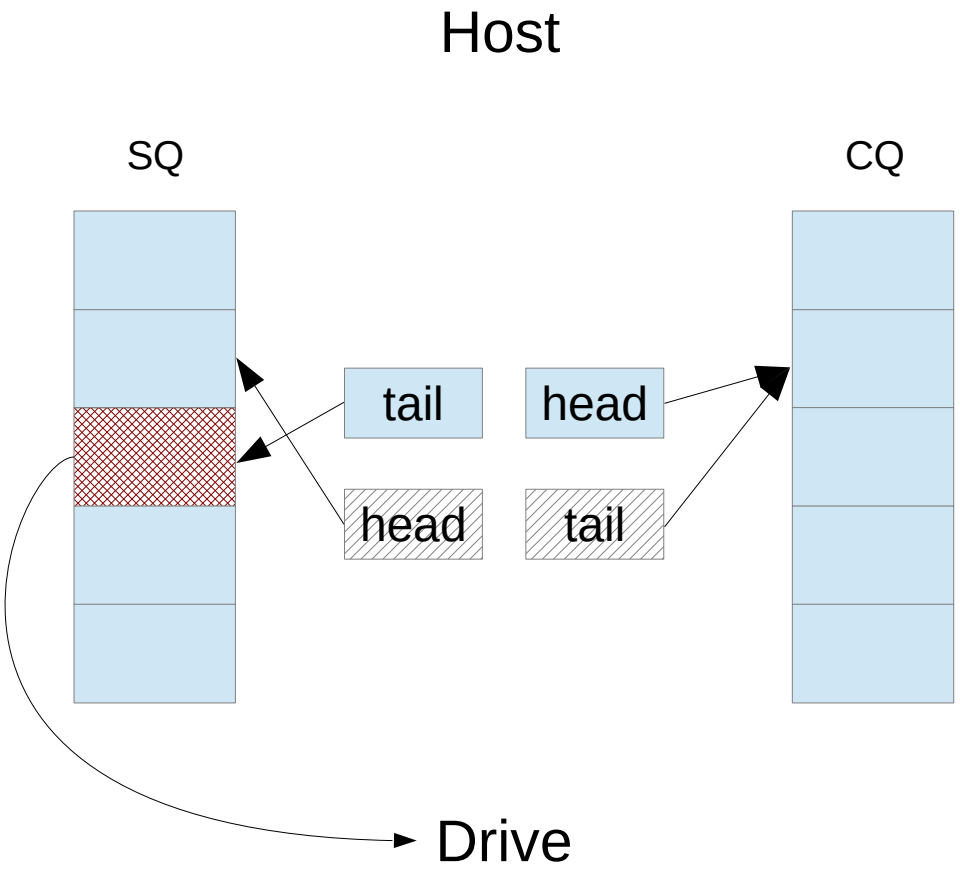
Command / Completion



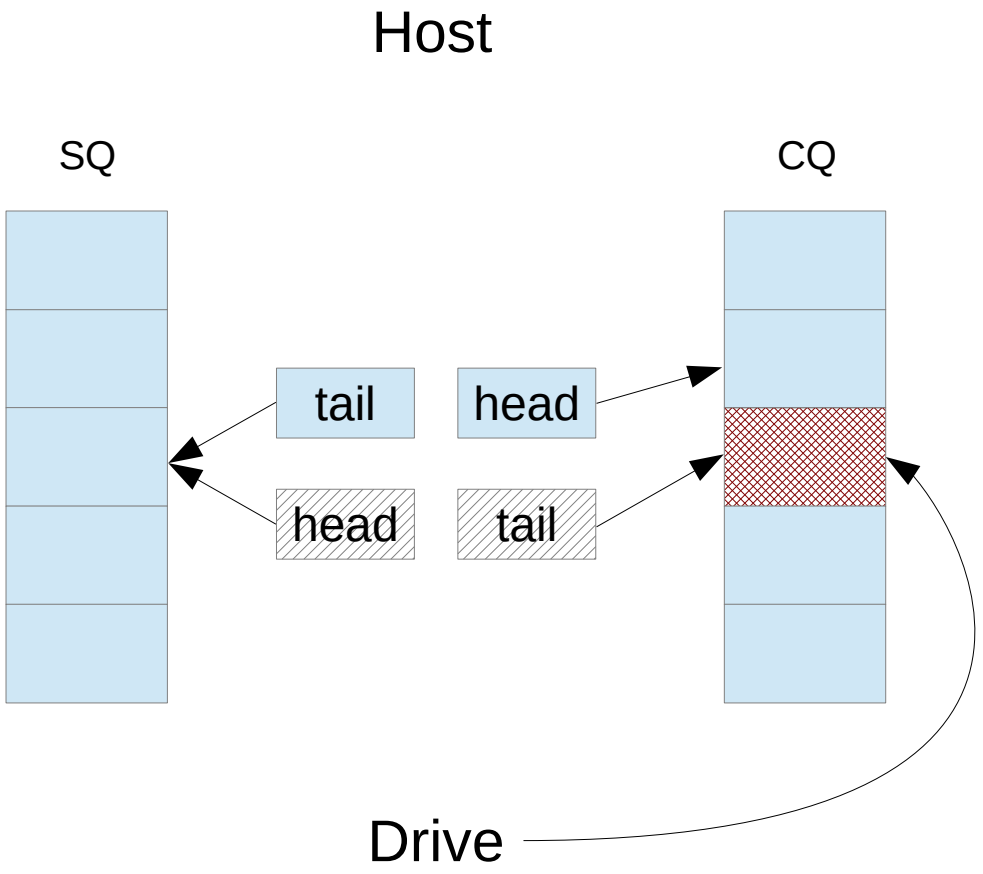
Command / Completion



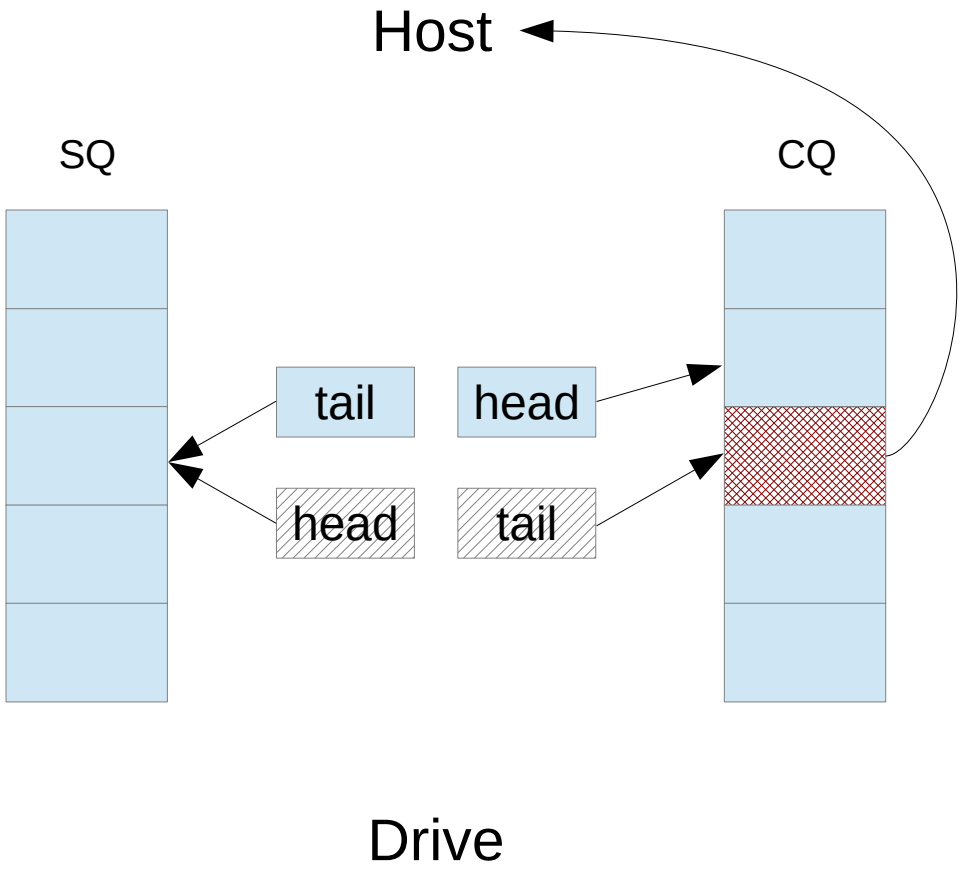
Command / Completion



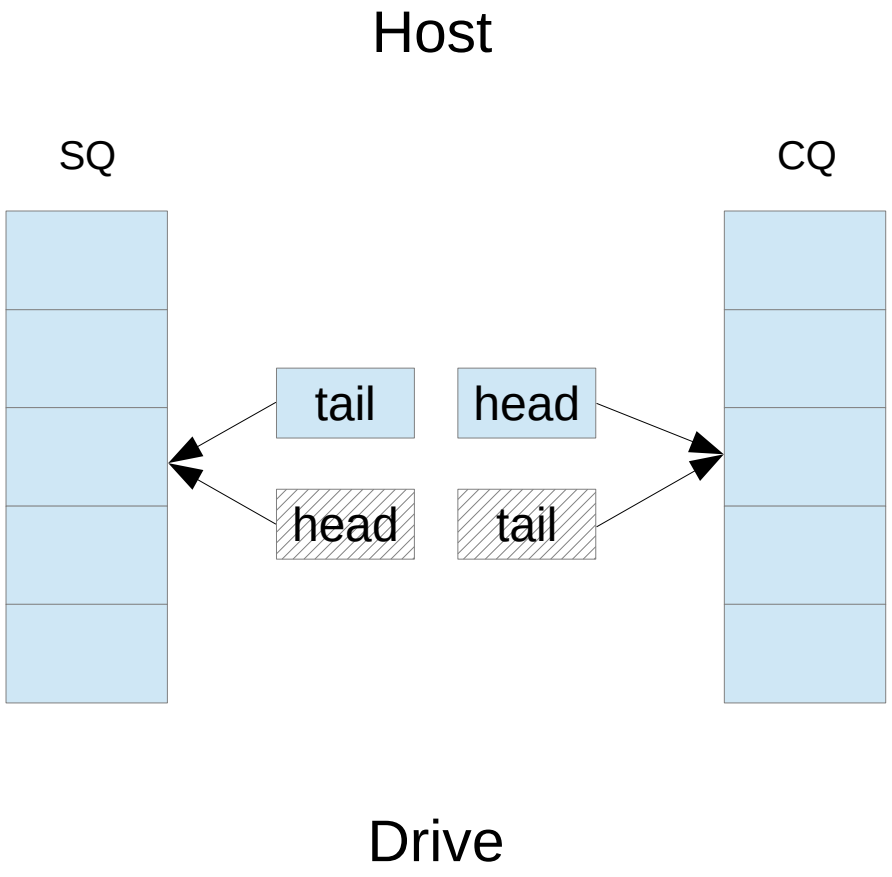
Command / Completion



Command / Completion



Command / Completion



NVMe “Pipeline”

- Classic RISC pipeline
  - Instruction fetch
  - Instruction decode
  - Execute
  - Memory access
  - Writeback
- NVMe controller processing
  - SQ entry fetch
  - Operation code decode
  - Execute operation
  - Writeback CQ entry
- Plug-in access at each “stage”

The Plug-in

- Approach similar to DTrace’s SDT provider
- Plug-in provided to bhyve via shared library
- Tap name tuple

“nvme:admin:decode”

Module Component Stage

Plug-in API

- bhyve expects setup() and teardown()
- Provides way to attach / detach taps

```
int  plugin_tap_attach(const char *name, void *cb);
int  plugin_tap_detach(const char *name);

#include "plugin.h"

static DECL_NPLUGIN_ADMIN_DECODE(admin_cmd);

int
setup(void)
{
        ...
        plugin_tap_attach("nvme:admin:decode", admin_cmd);
        ...
}
```

The Tap

```
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:
     ...
```

Read-only Failure

The Log Message

```
Sep 11 09:48:49 xxxxxxxx nvme1: async event occurred (log page id=0x2)
Sep 11 09:48:49 xxxxxxxx nvme1: async event occurred (log page id=0x2)
Sep 11 09:48:49 xxxxxxxx nvme1: media placed in read only mode
Sep 11 09:48:49 xxxxxxxx nvme1: async event occurred (log page id=0x2)
Sep 11 09:48:49 xxxxxxxx nvme1: media placed in read only mode
Sep 11 09:48:49 xxxxxxxx nvme1: media placed in read only mode
```

No Writes

```
static DECL_NPLUGIN_IO_DECODE(io_ro_fail);

int setup(void)
{
     plugin_tap_attach("nvme:io:decode", io_ro_fail);
     return (0);
}

int teardown(void)
{
     return (0);
}

static size_t n_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)
{
     if (cmd->opc == NVME_OPC_WRITE) {
          if (n_writes) n_writes--;
          else {
               NVME_STATUS_SET(cmp->status,
                         NVME_SCT_GENERIC, NVME_SC_SUCCESS);
               return (1);
          }
     }
     return (0);
}
```

No Writes + Errors

```
static DECL_NPLUGIN_IO_DECODE(io_ro_fail);

int setup(void)
{
     plugin_tap_attach("nvme:io:decode", io_ro_fail);
     return (0);
}

int teardown(void)
{
     return (0);
}

static size_t n_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)
{
     if (cmd->opc == NVME_OPC_WRITE) {
          if (n_writes) n_writes--;
          else {
               NVME_STATUS_SET(cmp->status,
                         NVME_SCT_COMMAND_SPECIFIC, NVME_SC_ATTEMPTED_WRITE_TO_RO_PAGE);
               return (1);
          }
     }
     return (0);
}
```

Yay, errors!

```
root@freebsd:~ # nvme0: WRITE sqid:2 cid:126 nsid:1 lba:128120 len:16
nvme0: WRITE TO RO PAGE (01/82) sqid:2 cid:126 cdw0:0
nvme0: WRITE sqid:1 cid:126 nsid:1 lba:528 len:16
nvme0: WRITE TO RO PAGE (01/82) sqid:1 cid:126 cdw0:0
…
nvme0: WRITE TO RO PAGE (01/82) sqid:2 cid:126 cdw0:0
nvme0: WRITE sqid:1 cid:127 nsid:1 lba:23070736 len:16
nvme0: WRITE TO RO PAGE (01/82) sqid:1 cid:127 cdw0:0

root@freebsd:~ # zpool status -v
  pool: sparks
 state: DEGRADED
status: One or more devices has experienced an error resulting in data
        corruption.  Applications may be affected.
action: Restore the file in question if possible.  Otherwise restore the
        entire pool from backup.
   see: http://illumos.org/msg/ZFS-8000-8A
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        sparks      DEGRADED     0     0     0
          ada1      ONLINE       0     0     0
        logs
          nvd0      FAULTED      0     4     0  too many errors

errors: Permanent errors have been detected in the following files:

        sparks/zv1:<0x0>
```

Temperamental

```
static DECL_NPLUGIN_IO_DECODE(io_ro_fail);

int setup(void)
{
     plugin_tap_attach("nvme:io:decode", io_ro_fail);
     srandom(time(NULL));
     return (0);
}

int teardown(void)
{
     return (0);
}

static bool is_readonly = false;

static int io_ro_fail(uint32_t bdf, struct nvme_command *cmd, struct nvme_completion *cmp,
          uint32_t sqid)
{
     if (cmd->opc == NVME_OPC_WRITE) {
          if (is_readonly) {
               NVME_STATUS_SET(cmp->status,
                         NVME_SCT_COMMAND_SPECIFIC, NVME_SC_ATTEMPTED_WRITE_TO_RO_PAGE);
               return (1);
          }
          else
               is_readonly = (random() % 100) > 42;
     }
     return (0);
}
```

Reincarnation

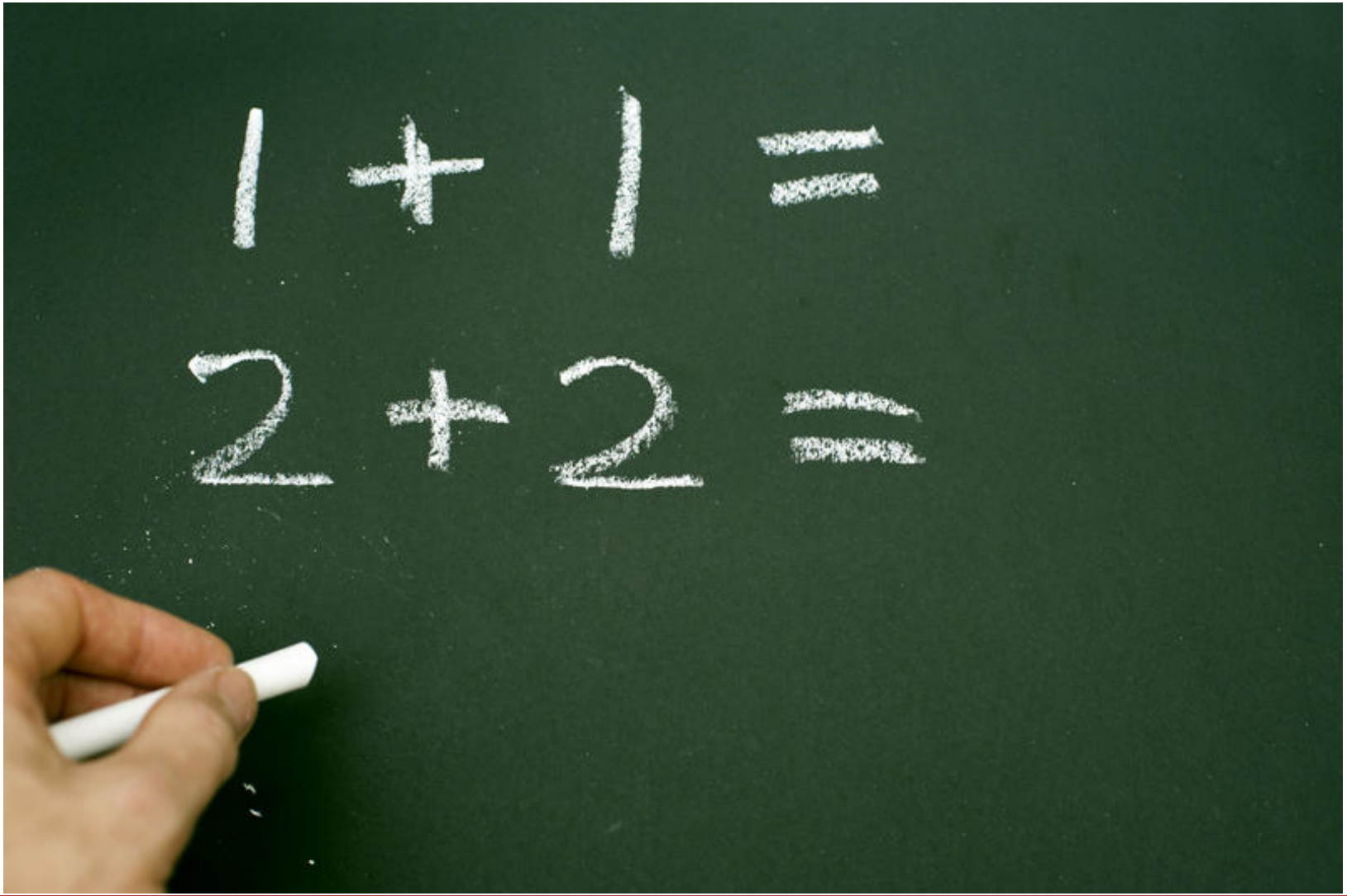
```
#define MAX_WRITES 5000 /* Fail after 5,000 Write commands */

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);
}
```


1 + 1 =
2 + 2 =

Sum Command

```
     plugin_tap_attach("nvme:admin:writeback", sum);
...
static int sum(uint32_t bdf, struct nvme_completion *cmp, uint32_t sqid)
{
     struct nvme_command *cmd = find_cmd(sqid, cmp->cid);


     if (cmd == NULL) {
          printf("%s: cache miss for CID=%04x\r\n", __func__, cmp->cid);
          return (0);
     }


     if (cmd->opc == 0x80) {
          uint32_t sum;


          sum = cmd->cdw10 + cmd->cdw11;
          cmp->cdw0 = sum;

          NVME_STATUS_SET(cmp->status, NVME_SCT_GENERIC, NVME_SC_SUCCESS);
          return (1);
     }
     return (0);
}


root@freebsd:~ # ./add /dev/nvme0 10 5
10 + 5 = 15
root@freebsd:~ #
```


OBSERVE
AND
REPORT

The Trace Plug-in

```
tbuf = calloc(MAX_E, sizeof(struct trc_ent));

out = fopen("/tmp/ptrace", "a");

plugin_tap_attach("pci:regread:writeback", cfgrd);
plugin_tap_attach("pci:regwrite:decode", cfgwr);
plugin_tap_attach("pci:msix:writeback", msix);
plugin_tap_attach("nvme:regread:writeback", regrd);
plugin_tap_attach("nvme:regwrite:decode", regwr);
plugin_tap_attach("nvme:msixread:writeback", msixrd);
plugin_tap_attach("nvme:msixwrite:decode", msixwr);
plugin_tap_attach("nvme:admin:decode", admin_cmd);
plugin_tap_attach("nvme:admin:writeback", cpl);
plugin_tap_attach("nvme:io:decode", io_cmd);
plugin_tap_attach("nvme:io:writeback", cpl);
```

OpenBSD Boot

Index Time PCI Event

```
[ 586] 1556804537:200909267 00:06.0 REG 0014 <- 00460001
[ 587] 1556804537:200915971 00:06.0 REG 001c -> 00000001
[ 588] 1556804537:200925803 00:06.0 REG 1000 <- 00000001
[ 589] 1556804537:200926382 00:06.0 ADM SQ=0 CID=0000 OPC=Identify NSID=0 prp1=bfb24000 prp2=0 CDW10=00000001
[ 590] 1556804537:200928002 00:06.0 CPL SQ=0 CID=0000 P=1 STS=0x0000 CDW0=0xddbebac0
[ 591] 1556804537:200935816 00:06.0 REG 1004 <- 00000001
[ 592] 1556804537:202818938 00:06.0 REG 1000 <- 00000002
[ 593] 1556804537:202819148 00:06.0 ADM SQ=0 CID=0000 OPC=Create IO CQ NSID=0 prp1=bfb25000 prp2=0 CDW10=007f0001
[ 594] 1556804537:202819551 00:06.0 CPL SQ=0 CID=0000 P=1 STS=0x0000 CDW0=0xddbebac0
[ 595] 1556804537:202829033 00:06.0 REG 1004 <- 00000002
[ 596] 1556804537:202836171 00:06.0 REG 1000 <- 00000003
[ 597] 1556804537:202836260 00:06.0 ADM SQ=0 CID=0000 OPC=Create IO SQ NSID=0 prp1=bfbfc000 prp2=0 CDW10=007f0001
[ 598] 1556804537:202836774 00:06.0 CPL SQ=0 CID=0000 P=1 STS=0x0000 CDW0=0xddbebac0
[ 599] 1556804537:202843934 00:06.0 REG 1004 <- 00000003
[ 600] 1556804537:202854092 00:06.0 REG 0010 <- 00000001
[ 601] 1556804537:204672931 00:06.0 REG 1000 <- 00000004
[ 602] 1556804537:204673170 00:06.0 ADM SQ=0 CID=0000 OPC=Identify NSID=0x1 prp1=bfb22000 prp2=0 CDW10=00000000
[ 603] 1556804537:204674262 00:06.0 CPL SQ=0 CID=0000 P=1 STS=0x0000 CDW0=0xddbebac0
[ 604] 1556804537:204683593 00:06.0 REG 1004 <- 00000004
[ 605] 1556804538:517281174 00:06.0 REG 1008 <- 00000001
[ 606] 1556804538:517281875 00:06.0 IO  SQ=1 CID=0000 OPC=Read NSID=0x1 LBA=0 prp1=bffc7000 prp2=0
[ 607] 1556804538:517388326 00:06.0 CPL SQ=1 CID=0000 P=1 STS=0x0000 CDW0=0
```

The Wireshark

dump_nvmeb.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Protocol	Info
444	2.188943	0000:00:05.0	NVMe/b	Mem Reg Write
445	2.188944	0000:00:05.0	NVMe	Command
446	2.188956	0000:00:05.0	NVMe	Completion
447	2.188981	0000:00:05.0	NVMe/b	Mem Reg Write
448	2.189015	0000:00:05.0	NVMe/b	Mem Reg Write
449	2.189016	0000:00:05.0	NVMe	Command
450	2.189025	0000:00:05.0	NVMe	Completion
451	2.189043	0000:00:05.0	NVMe/b	Mem Reg Write
452	2.195920	0000:00:05.0	NVMe/b	Mem Reg Write
453	2.195922	0000:00:05.0	NVMe	Command
454	2.195931	0000:00:05.0	NVMe	Completion
455	2.195951	0000:00:05.0	NVMe/b	Mem Reg Write
456	2.195971	0000:00:05.0	NVMe/b	Mem Reg Write
457	2.195971	0000:00:05.0	NVMe	Command
458	2.195978	0000:00:05.0	NVMe	Completion
459	2.195990	0000:00:05.0	NVMe/b	Mem Reg Write
460	40.569418	0000:00:05.0	NVMe/b	Mem Reg Write
461	40.569423	0000:00:05.0	NVMe	Command
462	40.569484	0000:00:05.0	NVMe	Completion
463	40.569530	0000:00:05.0	NVMe/b	Mem Reg Write
464	56.950298	0000:00:05.0	NVMe/b	Mem Reg Read
465	56.950311	0000:00:05.0	NVMe/b	Mem Reg Write
466	56.950318	0000:00:05.0	NVMe/b	Mem Reg Read

```
▸ Frame 460: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
▾ NVMe/bhyve, Mem Reg
    NVMe/bhyve Record Type: Mem Reg (1)
    NVMe/bhyve PCI Domain: 0
    NVMe/bhyve PCI Bus: 0
    NVMe/bhyve PCI Device: 5
    NVMe/bhyve PCI Function: 0
    NVMe/bhyve Register Length: 4
    NVMe/bhyve Register Address: 0x0000000000001010 SQ 2 Tail Doorbell
    NVMe/bhyve Register Value: 0x0000000000000021
```

```
0000  01 00 00 00 00 00 05 00  04 00 00 00 00 00 00 00   ........ ........
0010  10 10 00 00 00 00 00 00  21 00 00 00 00 00 00 00   ........ !.......
```

dump_nvmeb.pcap Packets: 466 · Displayed: 466 (100.0%) Profile: Default

The Wireshark

dump_nvmeb.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Protocol	Info
444	2.188943	0000:00:05.0	NVMe/b	Mem Reg Write
445	2.188944	0000:00:05.0	NVMe	Command
446	2.188956	0000:00:05.0	NVMe	Completion
447	2.188981	0000:00:05.0	NVMe/b	Mem Reg Write
448	2.189015	0000:00:05.0	NVMe/b	Mem Reg Write
449	2.189016	0000:00:05.0	NVMe	Command
450	2.189025	0000:00:05.0	NVMe	Completion
451	2.189043	0000:00:05.0	NVMe/b	Mem Reg Write
452	2.195920	0000:00:05.0	NVMe/b	Mem Reg Write
453	2.195922	0000:00:05.0	NVMe	Command
454	2.195931	0000:00:05.0	NVMe	Completion
455	2.195951	0000:00:05.0	NVMe/b	Mem Reg Write
456	2.195971	0000:00:05.0	NVMe/b	Mem Reg Write
457	2.195971	0000:00:05.0	NVMe	Command
458	2.195978	0000:00:05.0	NVMe	Completion
459	2.195990	0000:00:05.0	NVMe/b	Mem Reg Write
460	40.569418	0000:00:05.0	NVMe/b	Mem Reg Write
461	40.569423	0000:00:05.0	NVMe	Command
462	40.569484	0000:00:05.0	NVMe	Completion
463	40.569530	0000:00:05.0	NVMe/b	Mem Reg Write
464	56.950298	0000:00:05.0	NVMe/b	Mem Reg Read
465	56.950311	0000:00:05.0	NVMe/b	Mem Reg Write
466	56.950318	0000:00:05.0	NVMe/b	Mem Reg Read

```
▸ Frame 461: 80 bytes on wire (640 bits), 80 bytes captured (640 bits)
▸ NVMe/bhyve, Command
▾ NVM Express (Cmd)
    Opcode: 0x01 Write
    .... ..00 = Fuse Operation: 0x0
    ..00 00.. = Reserved: 0x0
    00.. .... = PRP Or SGL: 0x0
    Command ID: 0x007f
    Namespace Id: 0x00000001
    Reserved: 0000000000000000
    Metadata Pointer: 0x0000000000000000
  ▸ SGL1
    Start LBA: 0x0000000000002000
    Absolute Number of Logical Blocks: 0x0008
    .... ..00 0000 0000 = Reserved: 0x000
  ▸ .... .0.. .... .... = Protection info fields: 0x0
    .0.. .... .... .... = Force Unit Access: 0x0
    0... .... .... .... = Limited Retry: 0x0
    Expected Initial Logical Block Reference Tag: 0x00000000
    Expected Logical Block Application Tag Mask: 0x0000
    Expected Logical Block Application Tag: 0x0000
  ▸ DSM Flags
    Reserved: 000000
```

```
0000  02 00 00 00 00 00 05 00  02 00 00 00 00 00 00 00   ........ ........
0010  01 00 7f 00 01 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 a0 64 06 00 00 00 00   ........ ..d.....
0030  00 00 00 00 00 00 00 00  00 20 00 00 00 00 00 00   ........ . ......
```

dump_nvmeb.pcap Packets: 466 · Displayed: 466 (100.0%) Profile: Default

The Wireshark

dump_nvmeb.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Protocol	Info
444	2.188943	0000:00:05.0	NVMe/b	Mem Reg Write
445	2.188944	0000:00:05.0	NVMe	Command
446	2.188956	0000:00:05.0	NVMe	Completion
447	2.188981	0000:00:05.0	NVMe/b	Mem Reg Write
448	2.189015	0000:00:05.0	NVMe/b	Mem Reg Write
449	2.189016	0000:00:05.0	NVMe	Command
450	2.189025	0000:00:05.0	NVMe	Completion
451	2.189043	0000:00:05.0	NVMe/b	Mem Reg Write
452	2.195920	0000:00:05.0	NVMe/b	Mem Reg Write
453	2.195922	0000:00:05.0	NVMe	Command
454	2.195931	0000:00:05.0	NVMe	Completion
455	2.195951	0000:00:05.0	NVMe/b	Mem Reg Write
456	2.195971	0000:00:05.0	NVMe/b	Mem Reg Write
457	2.195971	0000:00:05.0	NVMe	Command
458	2.195978	0000:00:05.0	NVMe	Completion
459	2.195990	0000:00:05.0	NVMe/b	Mem Reg Write
460	40.569418	0000:00:05.0	NVMe/b	Mem Reg Write
461	40.569423	0000:00:05.0	NVMe	Command
462	40.569484	0000:00:05.0	NVMe	Completion
463	40.569530	0000:00:05.0	NVMe/b	Mem Reg Write
464	56.950298	0000:00:05.0	NVMe/b	Mem Reg Read
465	56.950311	0000:00:05.0	NVMe/b	Mem Reg Write
466	56.950318	0000:00:05.0	NVMe/b	Mem Reg Read

```
▸ Frame 462: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
▸ NVMe/bhyve, Completion
▾ NVM Express (Cqe)
    [Cmd in: 0]
    [Cmd Latency: 0.000 ms]
    Cmd specific Status: 0x0000000000000000
    SQ Head Pointer: 0x0021
    Reserved: 0x0002
    Command ID: 0x007f
    0000 0000 0000 000. = Status: 0x0000
    .... .... .... ...1 = Reserved: 0x1
```

```
0000  03 00 00 00 00 00 05 00  02 00 00 00 00 00 00 00   ........ ........
0010  00 00 00 00 00 00 00 00  21 00 02 00 7f 00 01 00   ........ !.......
```

dump_nvmeb.pcap Packets: 466 · Displayed: 466 (100.0%) Profile: Default

The Wireshark

dump_nvmeb.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... +

No.	Time	Source	Protocol	Info
444	2.188943	0000:00:05.0	NVMe/b	Mem Reg Write
445	2.188944	0000:00:05.0	NVMe	Command
446	2.188956	0000:00:05.0	NVMe	Completion
447	2.188981	0000:00:05.0	NVMe/b	Mem Reg Write
448	2.189015	0000:00:05.0	NVMe/b	Mem Reg Write
449	2.189016	0000:00:05.0	NVMe	Command
450	2.189025	0000:00:05.0	NVMe	Completion
451	2.189043	0000:00:05.0	NVMe/b	Mem Reg Write
452	2.195920	0000:00:05.0	NVMe/b	Mem Reg Write
453	2.195922	0000:00:05.0	NVMe	Command
454	2.195931	0000:00:05.0	NVMe	Completion
455	2.195951	0000:00:05.0	NVMe/b	Mem Reg Write
456	2.195971	0000:00:05.0	NVMe/b	Mem Reg Write
457	2.195971	0000:00:05.0	NVMe	Command
458	2.195978	0000:00:05.0	NVMe	Completion
459	2.195990	0000:00:05.0	NVMe/b	Mem Reg Write
460	40.569418	0000:00:05.0	NVMe/b	Mem Reg Write
461	40.569423	0000:00:05.0	NVMe	Command
462	40.569484	0000:00:05.0	NVMe	Completion
463	40.569530	0000:00:05.0	NVMe/b	Mem Reg Write
464	56.950298	0000:00:05.0	NVMe/b	Mem Reg Read
465	56.950311	0000:00:05.0	NVMe/b	Mem Reg Write
466	56.950318	0000:00:05.0	NVMe/b	Mem Reg Read

```
Frame 463: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
NVMe/bhyve, Mem Reg
    NVMe/bhyve Record Type: Mem Reg (1)
    NVMe/bhyve PCI Domain: 0
    NVMe/bhyve PCI Bus: 0
    NVMe/bhyve PCI Device: 5
    NVMe/bhyve PCI Function: 0
    NVMe/bhyve Register Length: 4
    NVMe/bhyve Register Address: 0x0000000000001014 CQ 2 Head Doorbell
    NVMe/bhyve Register Value: 0x0000000000000021
```

```
0000  01 00 00 00 00 00 05 00  04 00 00 00 00 00 00 00   ........ ........
0010  14 10 00 00 00 00 00 00  21 00 00 00 00 00 00 00   ........ !.......
```

dump_nvmeb.pcap Packets: 466 · Displayed: 466 (100.0%) Profile: Default

The Future

- All the bugs!
- Access data too
- Allow asynchronous command / completion
- “Events”
- Safety
- <Your idea goes here>

Questions ?

chuck@FreeBSD.org | people.freebsd.org/~chuck/doc/frankendrive

Slides and notes in directory