Flash Memory Summit

Test Fundamental Data Structures of NVMe SSD

Crane Chu <cranechu@pynv.me>
GENG YUN Technology Pte. Ltd.

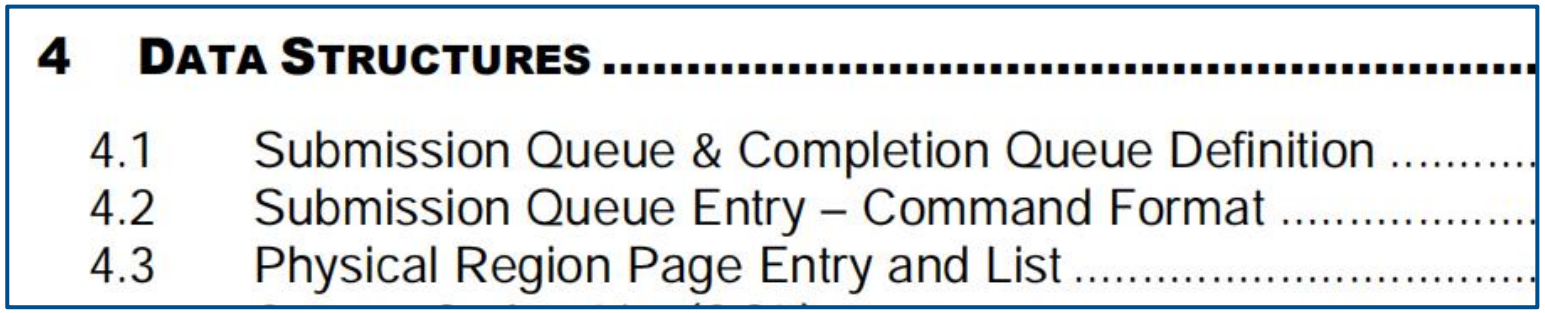
35 YEARS OF NAND FLASH MEMORY INNOVATION

Existing Test Tools

- nvme-cli
  - sudo nvme read /dev/nvme0n1 -n 1 -s 0 -c 0 -z 512
- fio
  - sudo fio --filename=/dev/nvme0n1 --direct=1 --rw=randread --bs=4k --ioengine=libaio --iodepth=256 --runtime=10 --numjobs=1 --time_based --name=demo
- with these tools, we can send admin commands, IO commands, and test most of the features defined in the NVMe specifications.
- But ...

Data Structures in NVMe

- IOSQ/IOCQ
- SQE/CQE
- PRP/PRPList



4 DATA STRUCTURES .....

4.1 Submission Queue & Completion Queue Definition .....
4.2 Submission Queue Entry – Command Format .....
4.3 Physical Region Page Entry and List .....

- They are all abstracted by NVMe device driver, thus it is difficult for existing test tools to touch them!

- But they are fundermental pieces of NVMe specifications, and have to be tested inside out.

PyNVMe3

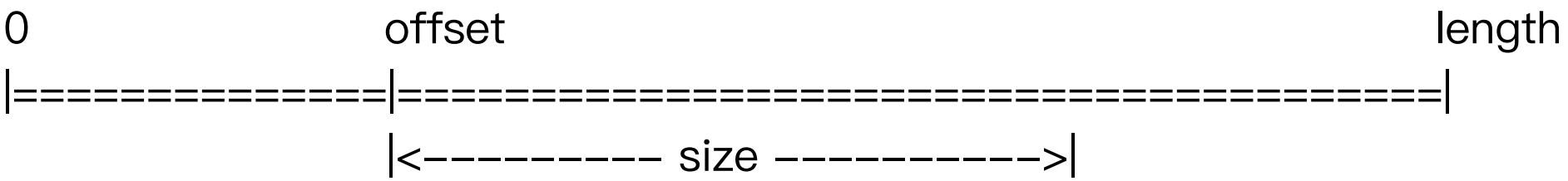
- PyNVMe3 is a Python module implemented a user space NVMe device driver, with a set of Python API.

```
def test_read_lba0(nvme0n1, qpair, buf):
    nvme0n1.read(qpair, buf, lba=0).waitdone()
```


```
def test_read_random(nvme0n1):
    nvme0n1.ioworker(io_size=8, qdepth=256,
                     read_percentage=100, time=10,
                     lba_random=True).start().close()
```


Buffer

- a single chunk of physical-contiguous memory, DMA-safe.



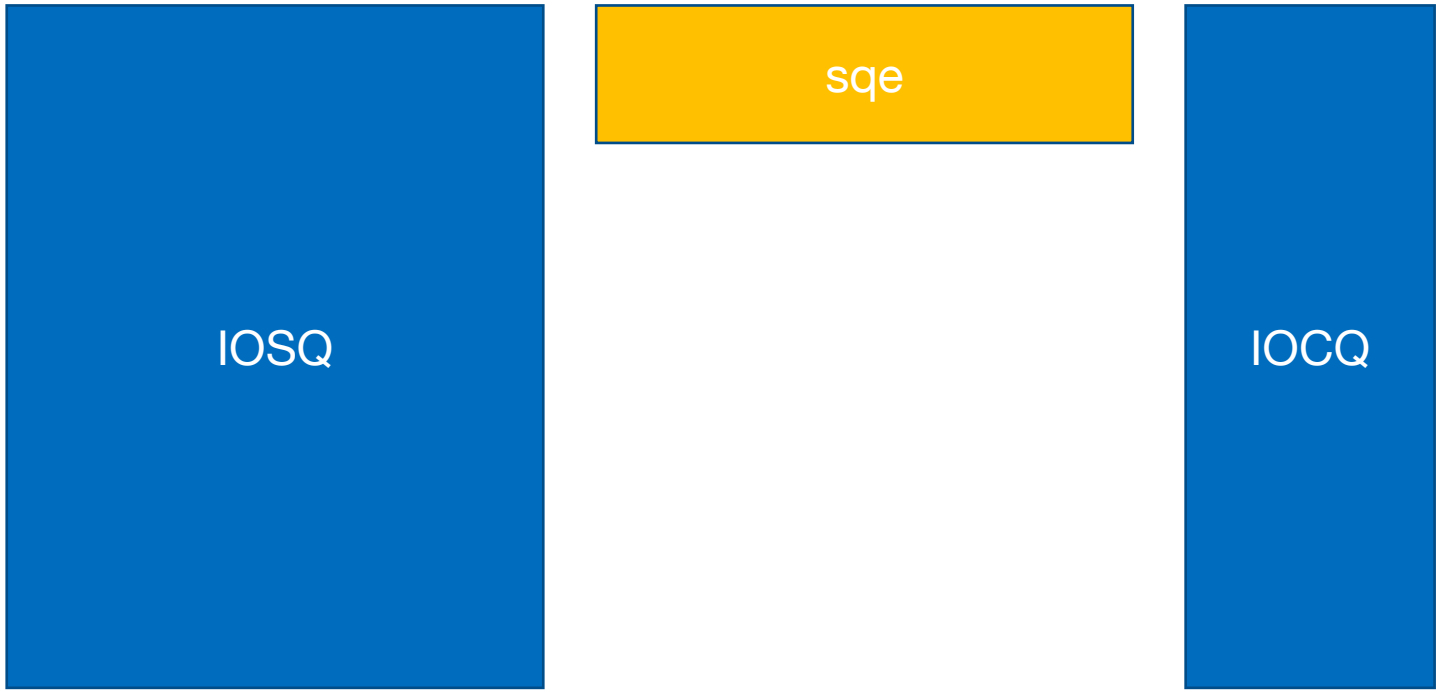
- we can put IOSQ/IOCQ, SQE/CQE, and PRP/PRPList into Buffer.

IOSQ/IOCQ


```
cq = IOCQ(nvme0, qid=1, qsize=256, prp1=Buffer(16*256))
sq = IOSQ(nvme0, qid=1, qsize=256, prp1=Buffer(64*256), cq=cq)
```

SQE: Submission Queue Entry

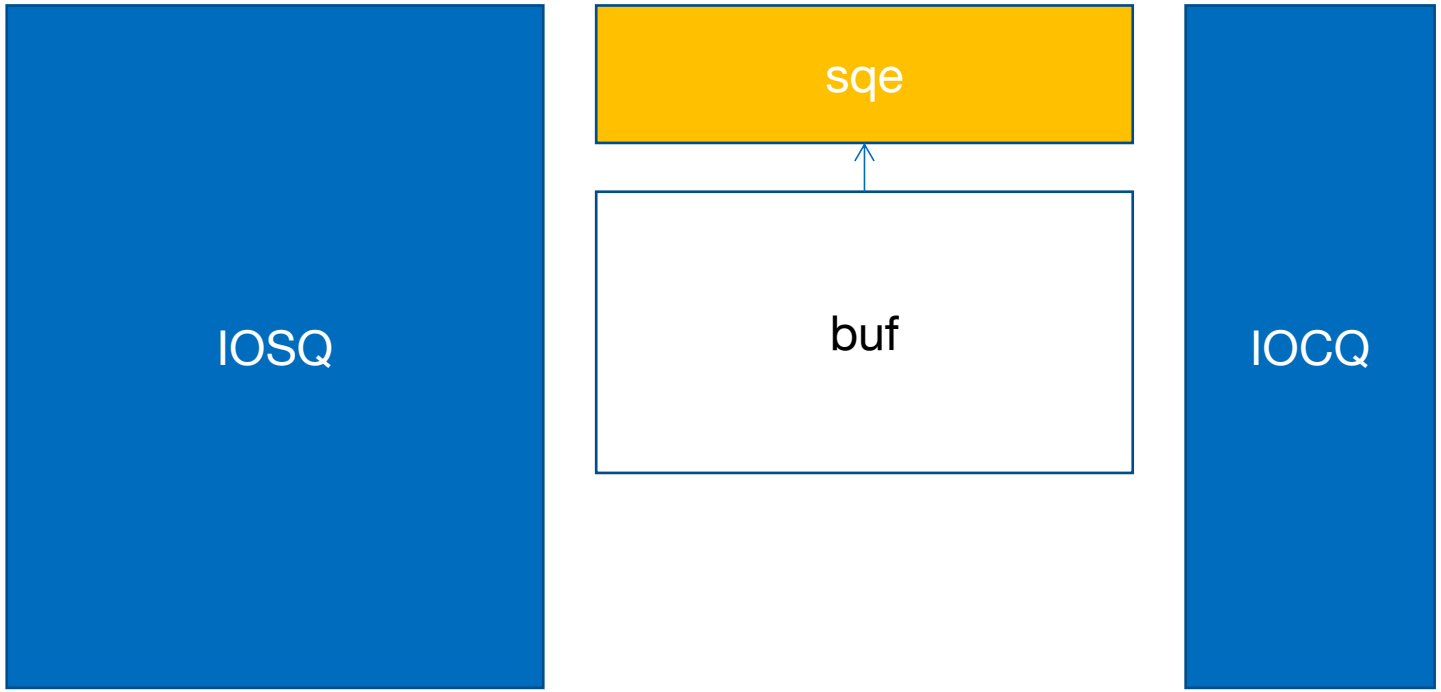

```
sqe = SQE(2, 1)  # list dwords as parameters: dword0, dword1
sqe[12] = 7      # dword 12
```



PRP of data buffer

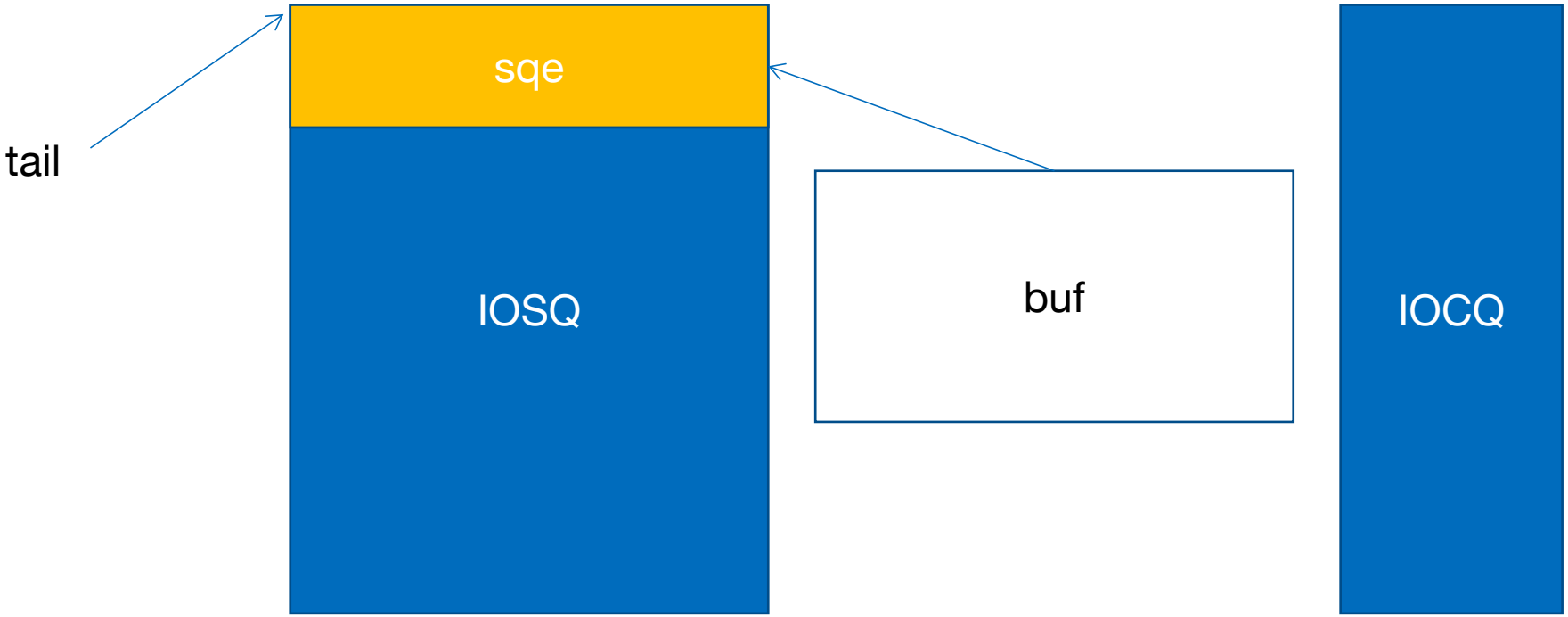
buf = Buffer(4096)

sqe.prp1 = buf



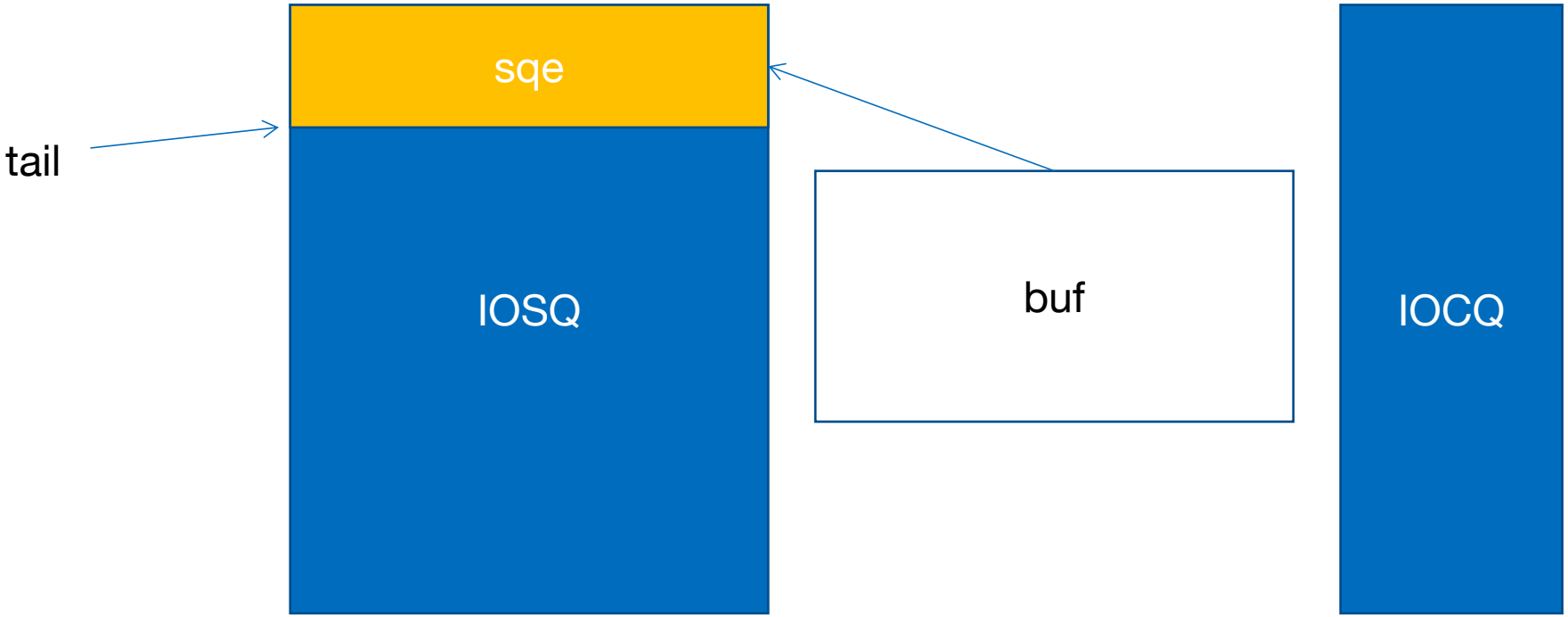
Fill SQE into IOSQ

sq[0] = cqe



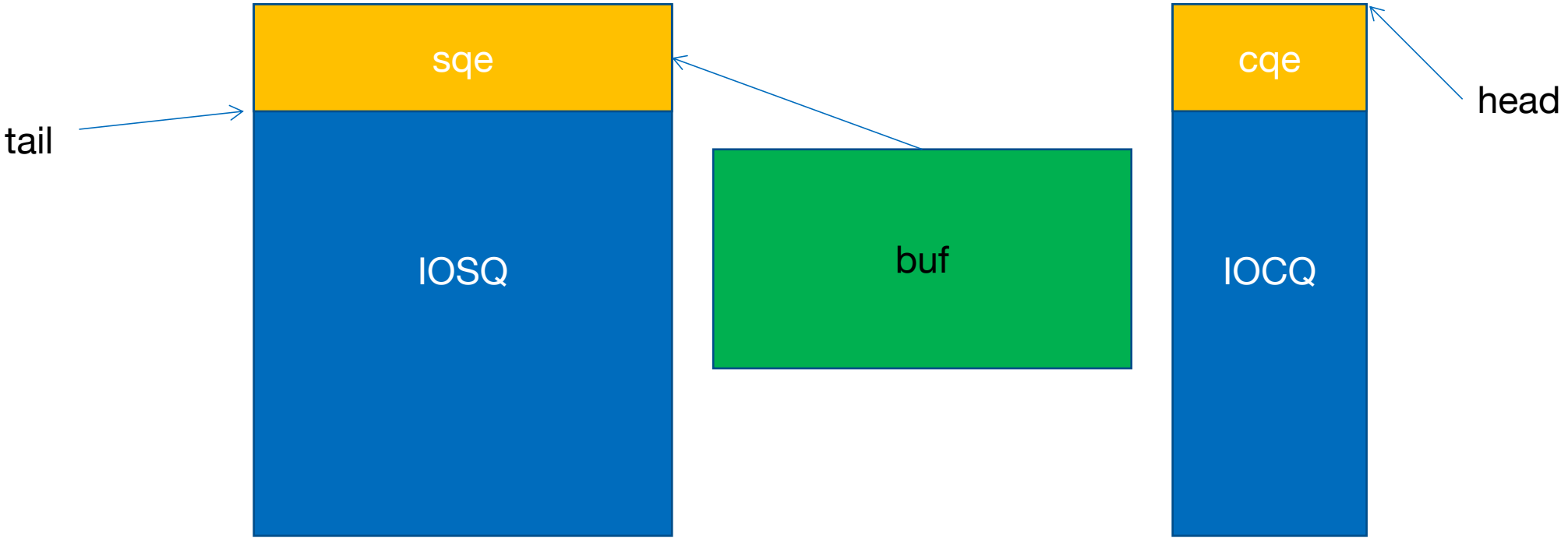
Update doorbell of IOSQ

sq.tail = 1



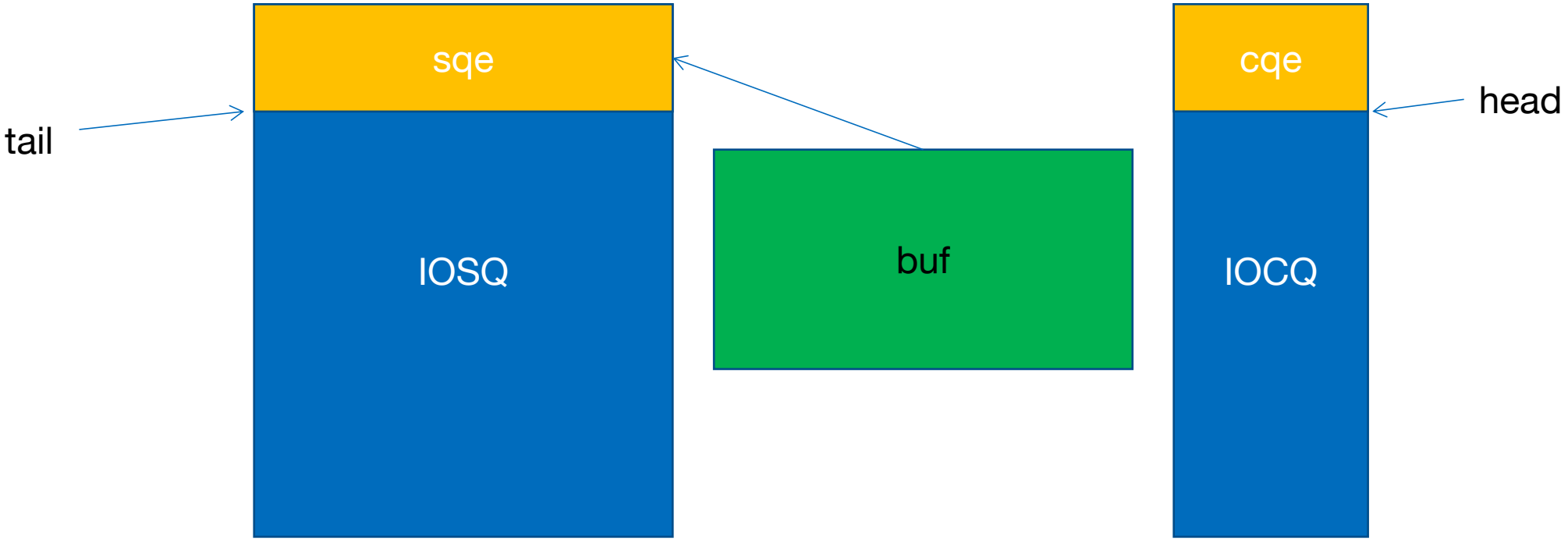
Monitor new CQE in IOCQ

while cq[0].p == 0: pass



Update doorbell of IOCQ

cq.head = 1



Delete IOSQ/IOCQ

sq.delete()

cq.delete()

Now we can test NVMe in another level


```python
import pytest
from nvme import IOCQ, IOSQ, SQE, Buffer


def test_fms22_demo(nvme0):
    cq = IOCQ(nvme0, qid=1, qsize=256, prp1=Buffer(16*256))
    sq = IOSQ(nvme0, qid=1, qsize=256, prp1=Buffer(64*256), cq=cq)

    sqe = SQE(2, 1)
    sqe[12] = 7
    buf = Buffer(4096)
    sqe.prp1 = buf
    sq[0] = sqe
    sq.tail = 1

    while cq[0].p == 0: pass
    cq.head = 1

    sq.delete()
    cq.delete()
```


Thanks!

Welcome to our Booth #1057

for more interesting tests!

