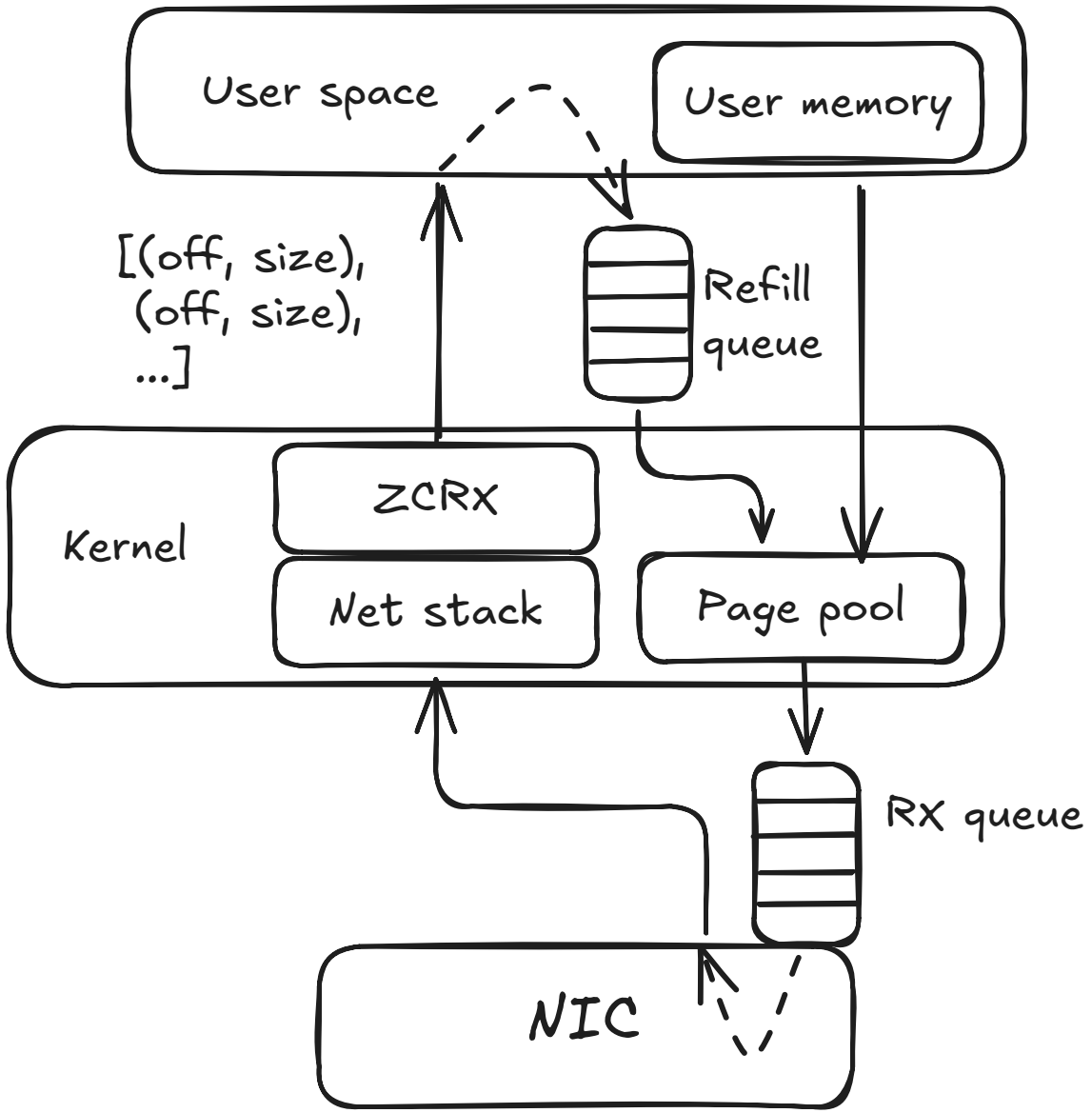
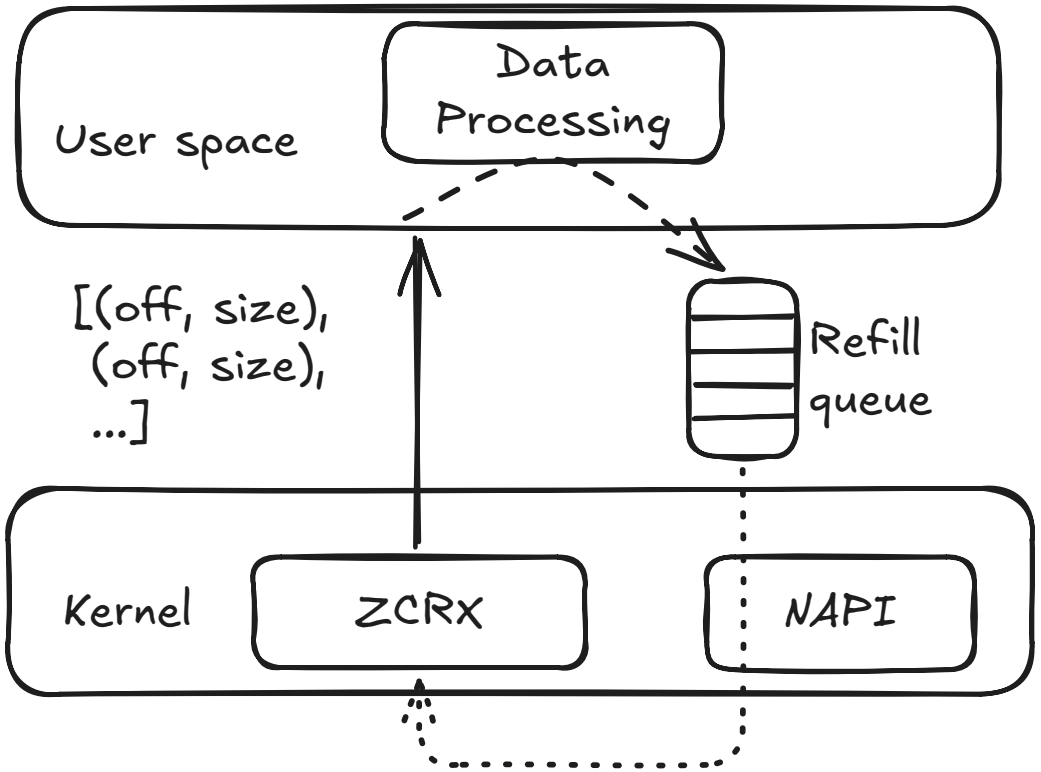
io_uring ZCRX: Progress and Next Steps

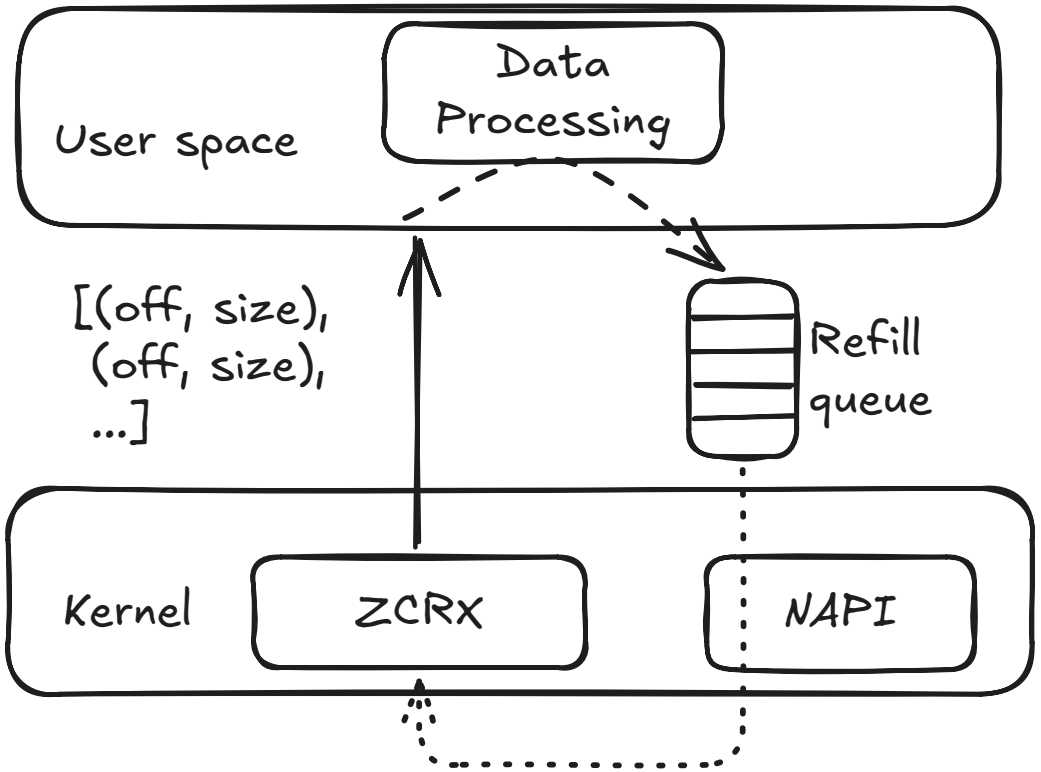
netdev conf 0x1A

Pavel Begunkov


User space
User memory
[(off, size),
(off, size),
...]
Refill
queue
Kernel
ZCRX
Net stack
Page pool
RX queue
NIC



- Refill queue is polled *asynchronously*
- `ZCRX_CTRL_FLUSH_RQ`, if full
- Helps with bursts, but slower
- Proper queue sizing is still important



- User processing may take time
- Area size should account for that
- Low memory notification
  - `ZCRX_NOTIF_NO_BUFFERS`
  - `ZCRX_CTRL_ARM_NOTIFICATION`
- Adding more memory
  - `ZCRX_CTRL_ADD_AREA`
- Fall back to kernel allocations?

Recieve queue sharing

- Ideally, 1 *Rx queue* : 1 *zcrx* : 1 *CPU*
  - Too many *RX queues*
  - Memory over provisioning
- `ZCRX_CTRL_EXPORT` + `ZCRX_REG_IMPORT`
  - sharing zcrx, 1 *Rx queue* : 1 zcrx : N *io_uring* (*CPUs*)
- Multiple **Refill Queues** in the future?

- **dma-buf** support landed May last year
- Copy fall back notifications
- Probing with `IO_URING_QUERY_ZCRX[_NOTIF]`
  - header size probing
  - feature support
- Pure copy fallback mode with `ZCRX_REG_NODEV`

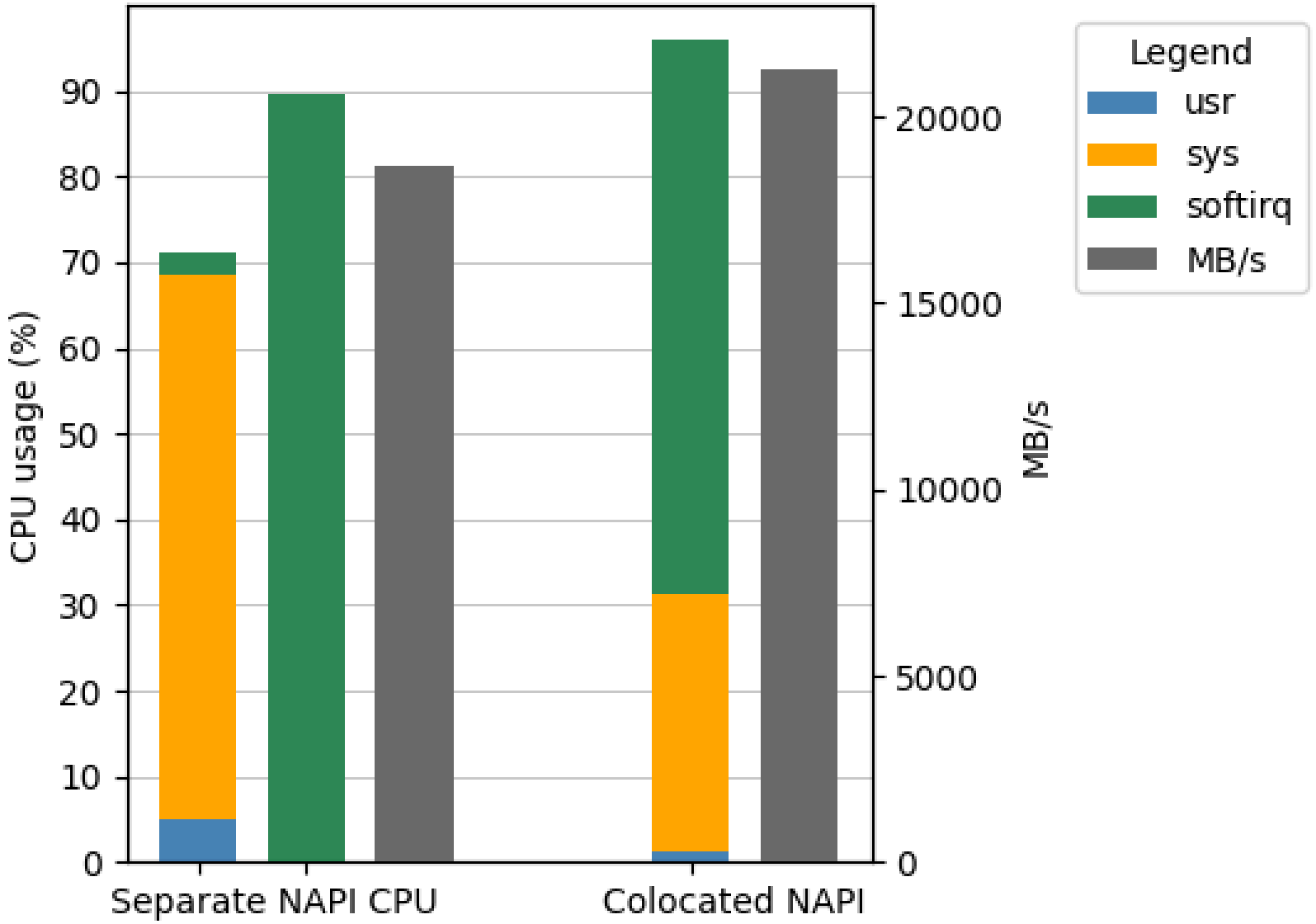
Large page support

- *NIC* can use large contiguous chunks
  => physically contiguous user memory (i.e. *huge pages*)
- Improves kernel and user processing
- Recommended with hardware *GRO*

Maxed out 200 Gbit/s NIC, **4KB** vs **32KB**:

```
     CPU    %usr   %nice    %sys %iowait    %irq   %soft   %idle
4K   |  0    1.53    0.00   27.78    2.72    1.31   66.45    0.22
32K  |  0    0.69    0.00    8.26   31.65    1.83   57.00    0.57
```

User and NAPI CPU sharing



Contention

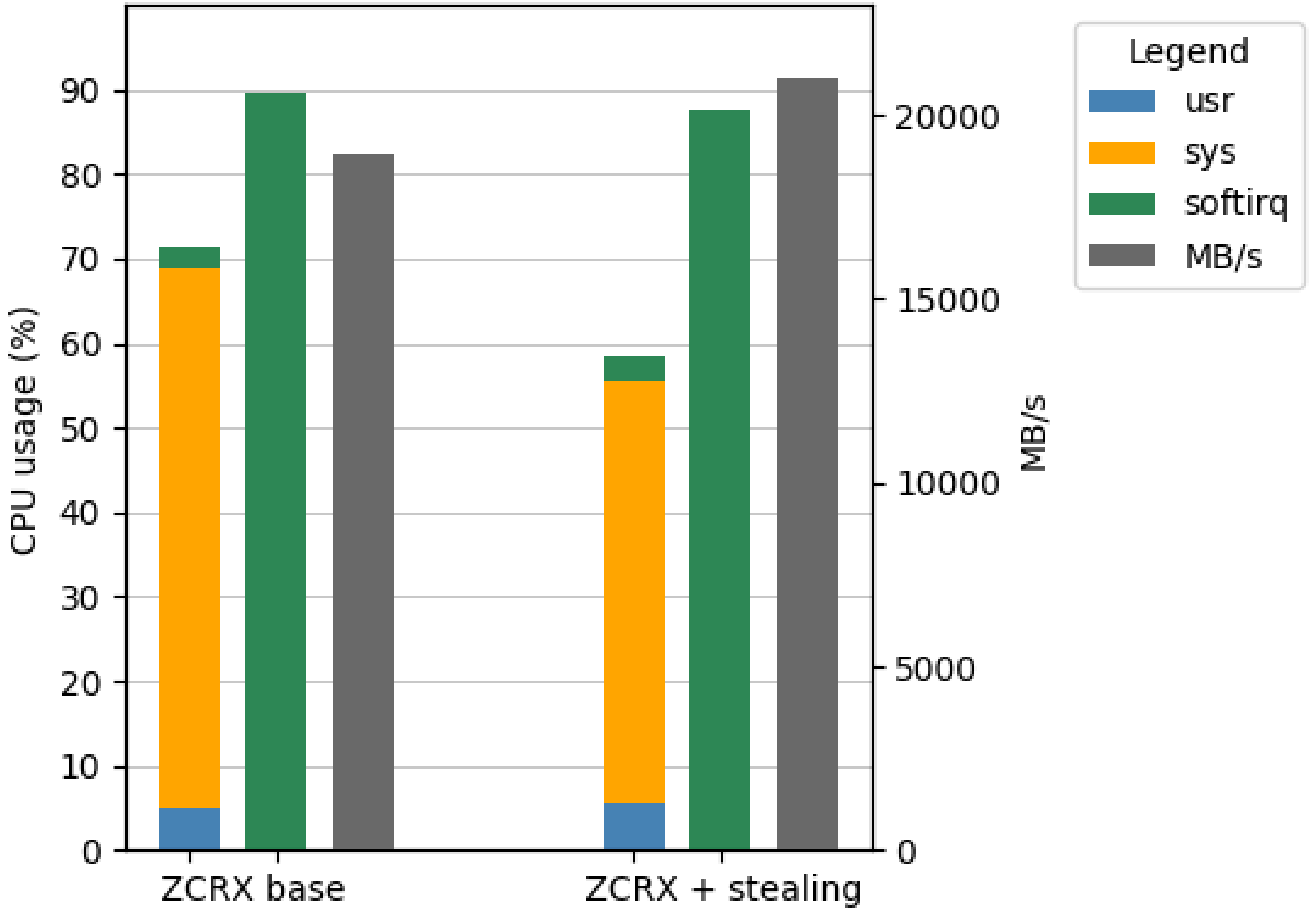
High contention when user space and *NAPI* run on different CPUs.

- Lots of refcounting, "*user*" and `net_iov`
- Extra refs => no benefits from `skb_attempt_defer_free()`
- Steal `net_iov` references from `sk_buff` frags
- move "*user*" referencing to NAPI by stealing `sk_buff`

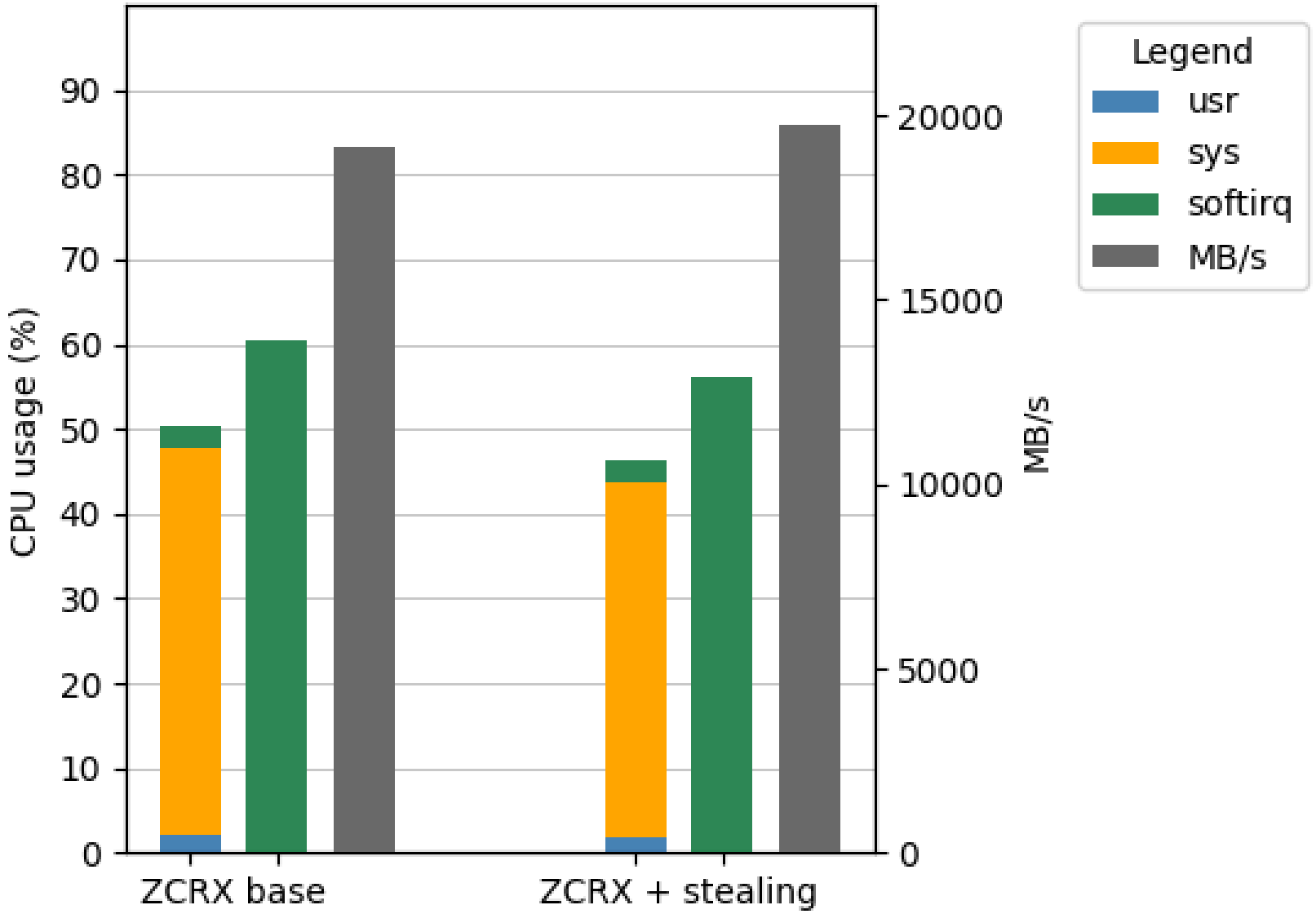
Improved refill queue handling

- The user space should be mindful as well

NAPI on separate CPUs, 4K NIC page size



NAPI on separate CPUs, 32K NIC page size



What's next

- Device memory tx is work in progress, early prototype is out
- UDP / other protocols
- Experimenting with NAPI busy polling
- Many discontiguous segments
  - Terrible for user space as well
  - Embracing direct data placement (RDMA style)?

benchmark: liburing/examples/zcrx:
url: https://github.com/isilence/liburing/tree/zcrx/test-skb-steal
git: https://github.com/isilence/liburing.git zcrx/test-skb-steal

Dynamic provisioning:
https://lore.kernel.org/io-uring/cover.1783616211.git.asml.silence@gmail.com/

TX prototype:
https://lore.kernel.org/io-uring/cover.1783614400.git.asml.silence@gmail.com/

Refcounting optimisations:
https://lore.kernel.org/io-uring/cover.1783619193.git.asml.silence@gmail.com/