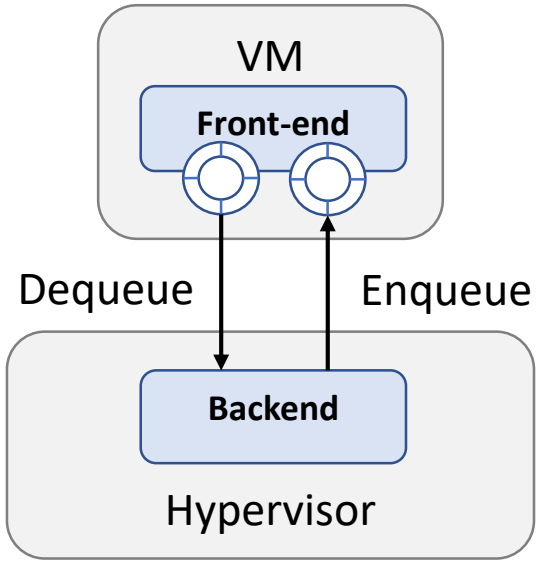
Asynchronous CBDMA Enqueue Framework for vHost-User

Jiayu Hu, Intel

VirtIO

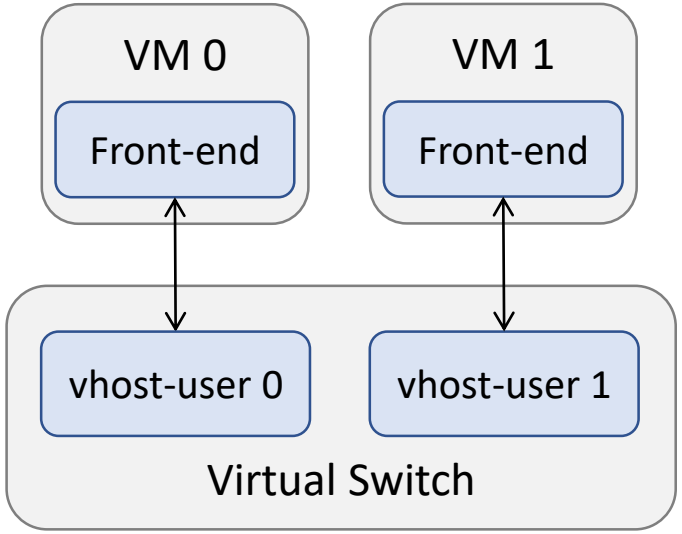
- Para-virtual I/O is a virtualization technique to enhance VM I/O performance.
- VirtIO is a standard of para-virtual I/O, which consists of VirtIO front-end in VM and backend in hypervisor.
- Back-end communicates with front-end by **copying packet buffers** between hypervisor's and VM's memory.



vHost-User

- DPDK provides efficient user-space backend device, called vhost-user.

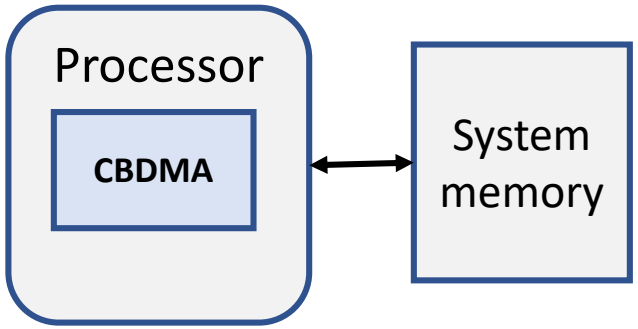
- vHost-user is widely used in virtual switches, like OVS.



Copying large bulk of data *inside vhost-user becomes a* ***hotspot****.*

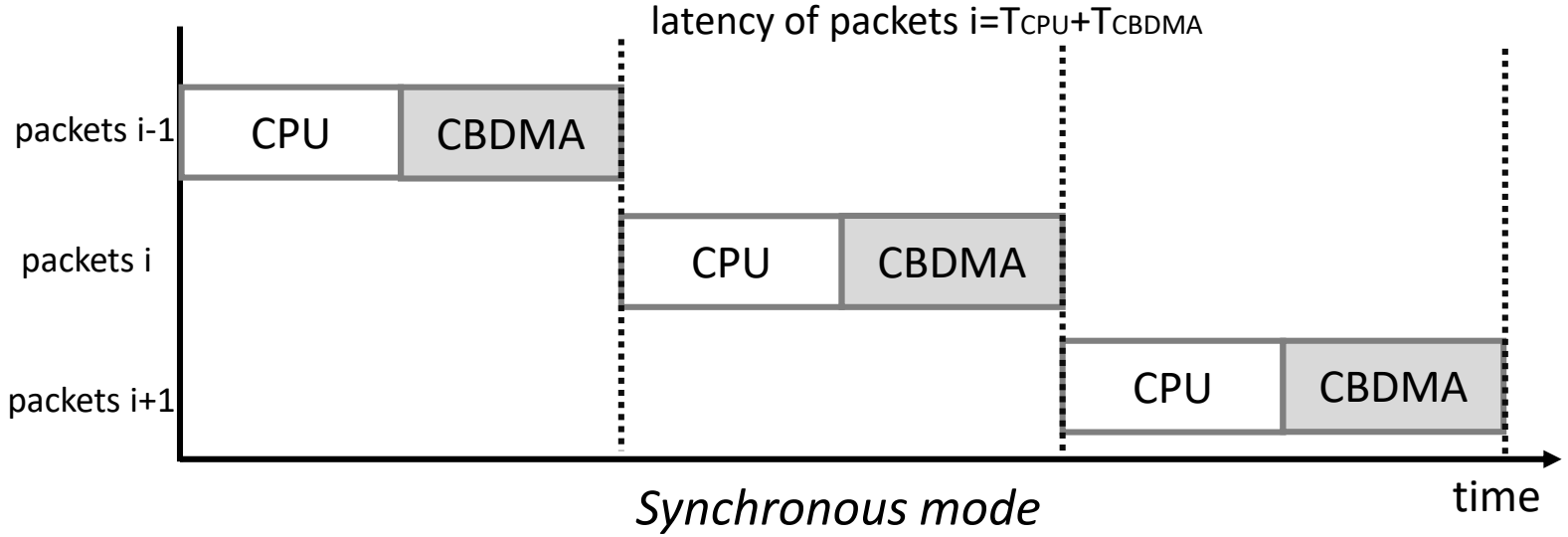
Crystal Beach DMA

- Crystal Beach DMA (CBDMA) is a DMA engine in the processor, which is extremely efficient in performing memory copy operations.
- No CPU intervention during data transfer.

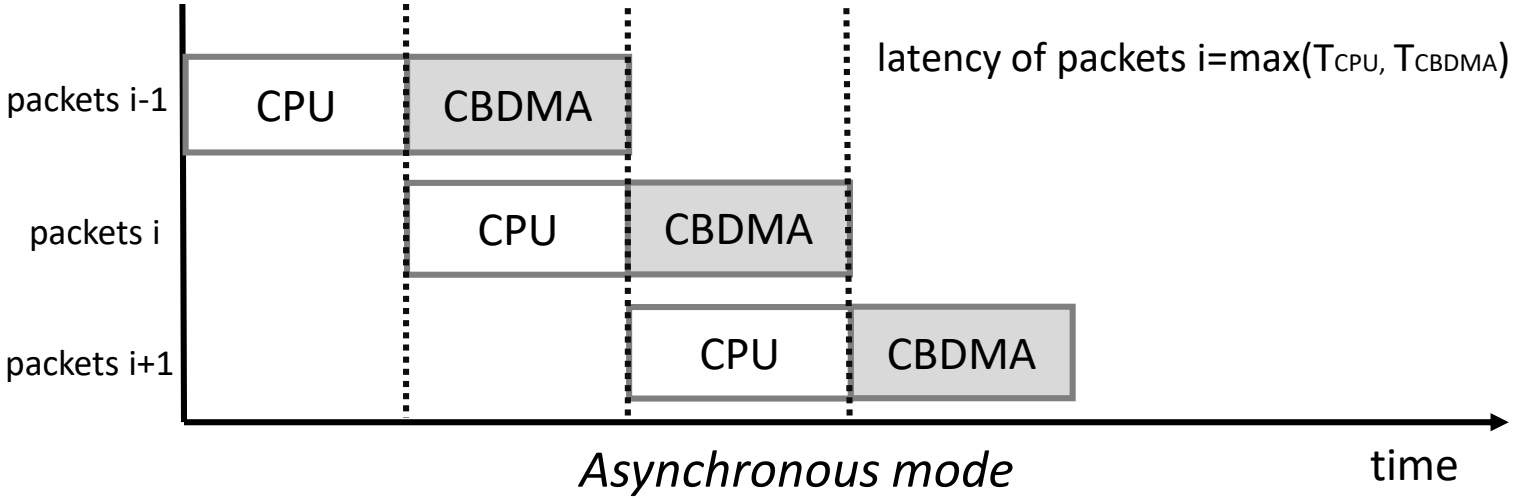


- There are two modes of offloading memory copy to the CBDMA:
  - Synchronous mode: the CPU submits copy jobs to the CBDMA and waits for completion.
  - Asynchronous mode: the CPU immediately returns as soon as submits copy jobs to the CBDMA, without waiting for completion.

Synchronous vs. Asynchronous



Synchronous mode



Asynchronous mode

- Asynchronous mode can **save precious CPU cycles** and **hide CBDMA copy overhead** in executing CPU logics.

Offload memory copy of enqueue operation to the CBDMA asynchronously

Challenges

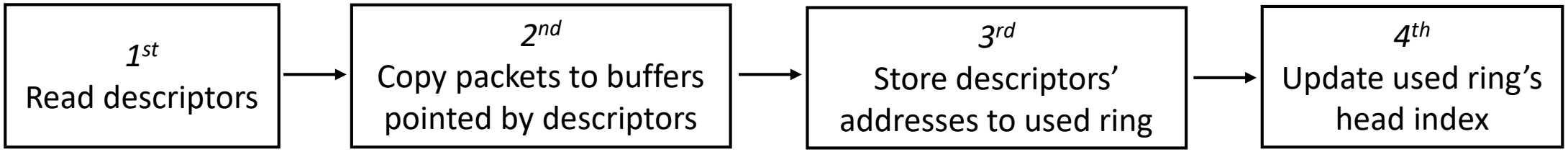
- How to fully utilize the CBDMA.
  - CBDMA performance is significantly influenced by the copy buffer length.
- How the CPU and the CBDMA cooperate to perform enqueue operation efficiently.
- Conflict with enqueue API semantics
  - Enqueue API releases the ownership of user buffers as soon as it finishes.
  - However, CBDMA copy is asynchronous with CPU operations. The CBDMA may still be copying packets when enqueue API returns.

Solutions of Addressing Challenges

- **Adaptively assign workloads to the CPU and the CBDMA, according to the copy length.**
  - Please refer to https://www.dpdk.org/wp-content/uploads/sites/35/2018/12/JiayuHu_Accelerating_paravirtio_with_CBDMA.pdf
- **Asynchronous CPU and CBDMA enqueue pipeline**
- **Provide a new PMD, vhost-ioat PMD, for CBDMA-accelerated backend.**
  - In enqueue operation, packets' mbufs that are completed copy by the CBDMA are freed inside the vhost-ioat PMD, without returning to users.

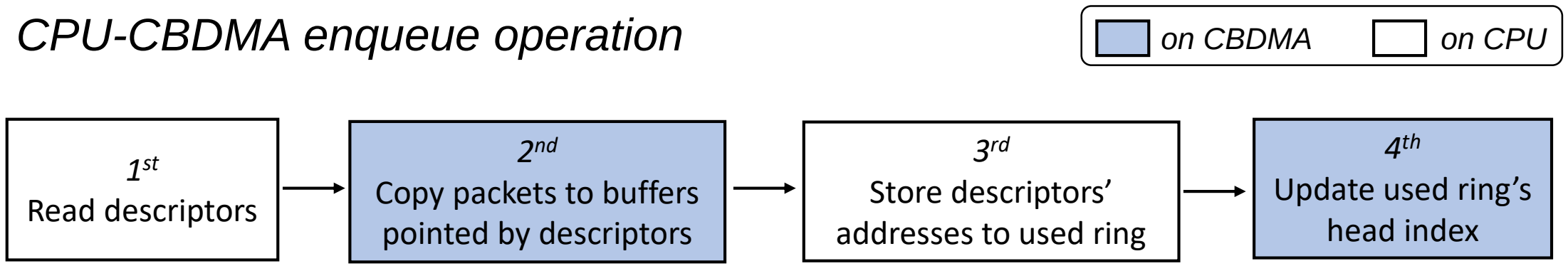
Asynchronous CPU-CBDMA Enqueue Pipeline

- Vring enqueue operation has four steps:



  - Back-end notifies front-end of enqueued packets by updating head index of used ring.
  - The execution of the 2nd and 3rd steps can be out-of-order.
- **As the CBDMA is inefficient in copying small packets, we assign the 3rd step to the CPU and the 2nd step to the CBDMA.**
- **We assign the 4th step to the CBDMA to guarantee predictable latency to front-end.**

Asynchronous CPU-CBDMA Enqueue Pipeline



- The 1st and 3rd steps of CPU and the 2nd and 4th steps of CBDMA execute in parallel.
- Thus, we can save precious CPU cycles to do meaningful jobs and hide CBDMA copy overhead in executing CPU logics.

vHost-ioat PMD

- vHost-ioat is a polling mode driver for CBDMA-accelerated VirtIO backend.
  - It implements CBDMA-accelerated data path.
  - For control path, it directly leverages the vhost library.

- vHost-ioat PMD provides basic functionality of packet reception and transmission.
  - In the TX direction, it offloads memory copy operations the CBDMA asynchronous.
  - It just supports CPU-based RX operation currently. CBDMA-accelerated RX operation is a work in progress.

vHost-ioat PMD

- Users can specify the following arguments in '--vdev' option:
  - *iface*: specify a path to connect front end
  - *queues*: the number of queues
  - *client*: client mode or server mode
  - *ioats*: specify the CBDMA address used by a queue
- An example of creating a vhost-ioat port:

--vdev 'ioat_vhost_0,iface=/tmp/sock0,queues=2,ioats=(txq0@00:04.0;txq1@00:04.1),client=0'

- Limitation
  - A CBDMA device can only be used by one queue.

Key Takeaways and Future Work

- Key takeaways
  - Offload memory copy inside vhost-user to the CBDMA asynchronously to improve performance.
  - Asynchronous CPU-CBDMA enqueue pipeline is designed for CBDMA-accelerated enqueue operation.
  - Provide vhost-ioat PMD for CBDMA-accelerated backend.
- Future work
  - Support CBDMA-accelerated dequeue operation.
  - Support sharing CBDMA among vhost-ioat queues and ports.

Thanks

jiayu.hu@intel.com