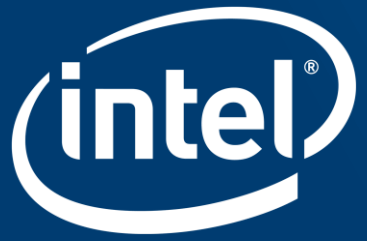


MIXING KOOL-AIDS!
ACCELERATE THE INTERNET WITH AF_XDP & DPDK

Kevin Laatz & Ciara Loftus

FOSDEM 2020

Introduction

https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/14-AF_XDP-dpdk-summit-china-2019.pdf

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikxb2Z7pnnAhVNY8AKHchXDfAQjhx6BAgBEAI&url=https%3A%2F%2Fwww.cdrecycler.com%2Finterstate-waste-apex-merger.aspx&psig=AOvVaw0QFTtUxvONAgVjKXIMC_-v&ust=1579873577016036

Introduction

- Userspace Libraries and Drivers.
- Accelerate packet processing workloads.
- Runs on wide variety of CPU architectures.
- Has its own memory management subsystem.
- Device specific PMDs (Poll Mode Drivers).

https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/14-AF_XDP-dpdk-summit-china-2019.pdf

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikxb2Z7pnnAhVNY8AKHchXDfAQjhx6BAgBEAI&url=https%3A%2F%2Fwww.cdrecycler.com%2Finterstate-waste-apex-merger.aspx&psig=AOvVaw0QFTtUxvONAgVjKXIMC_-v&ust=1579873577016036

Introduction

- Userspace Libraries and Drivers.
- Accelerate packet processing workloads.
- Runs on wide variety of CPU architectures.
- Has its own memory management subsystem.
- Device specific PMDs (Poll Mode Drivers).

AF_XDP

- Kernel based address family.
- Optimized for high performance packet processing.
- AF_XDP sockets redirect packets to userspace.
- By-passes kernel network stack
  - “In-kernel fast path”.

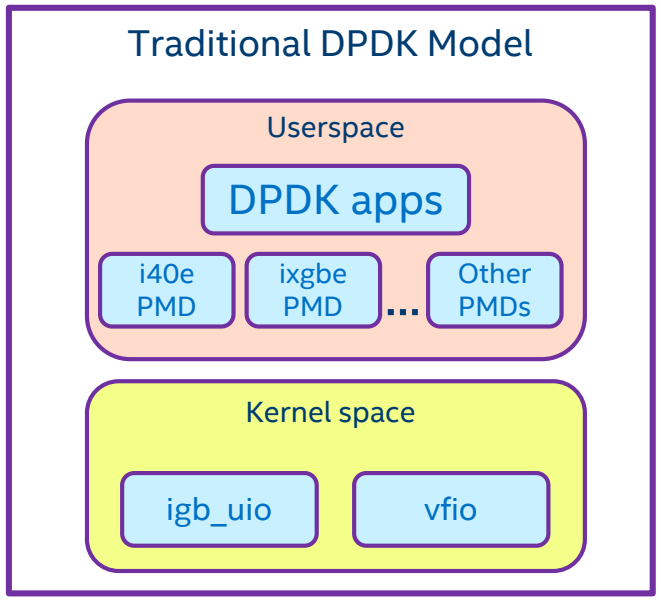
https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/14-AF_XDP-dpdk-summit-china-2019.pdf

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikxb2Z7pnnAhVNY8AKHchXDfAQjhx6BAgBEAI&url=https%3A%2F%2Fwww.cdrecycler.com%2Finterstate-waste-apex-merger.aspx&psig=AOvVaw0QFTtUxvONAgVjKXIMC_-v&ust=1579873577016036

Introduction

DPDK
DATA PLANE DEVELOPMENT KIT

- Userspace Libraries and Drivers.
- Accelerate packet processing workloads.
- Runs on wide variety of CPU architectures.
- Has its own memory management subsystem.
- Device specific PMDs (Poll Mode Drivers).



AF_XDP

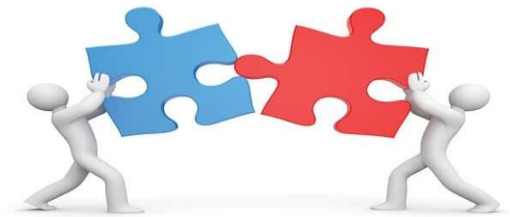
- Kernel based address family.
- Optimized for high performance packet processing.
- AF_XDP sockets redirect packets to userspace.
- By-passes kernel network stack
  - “In-kernel fast path”.

https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/14-AF_XDP-dpdk-summit-china-2019.pdf

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikxb2Z7pnnAhVNY8AKHchXDfAQjhx6BAgBEAI&url=https%3A%2F%2Fwww.cdrecycler.com%2Finterstate-waste-apex-merger.aspx&psig=AOvVaw0QFTtUxvONAgVjKXIMC_-v&ust=1579873577016036

Introduction

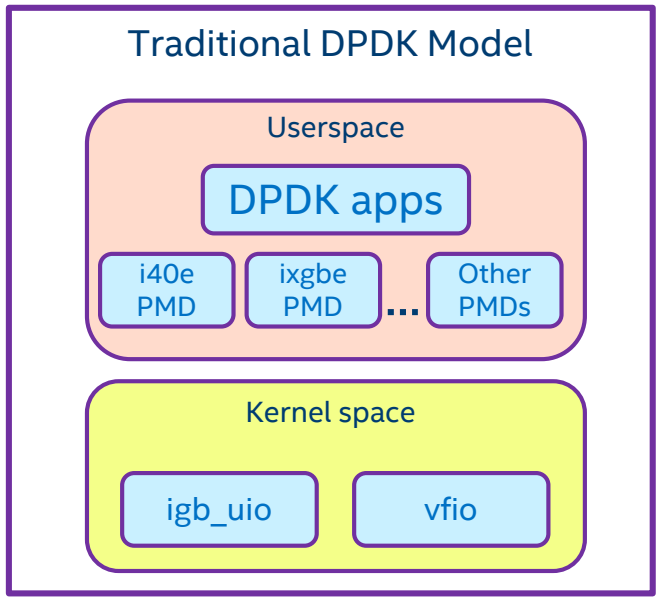
DPDK
DATA PLANE DEVELOPMENT KIT

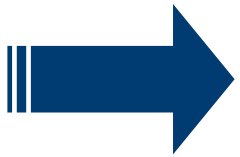


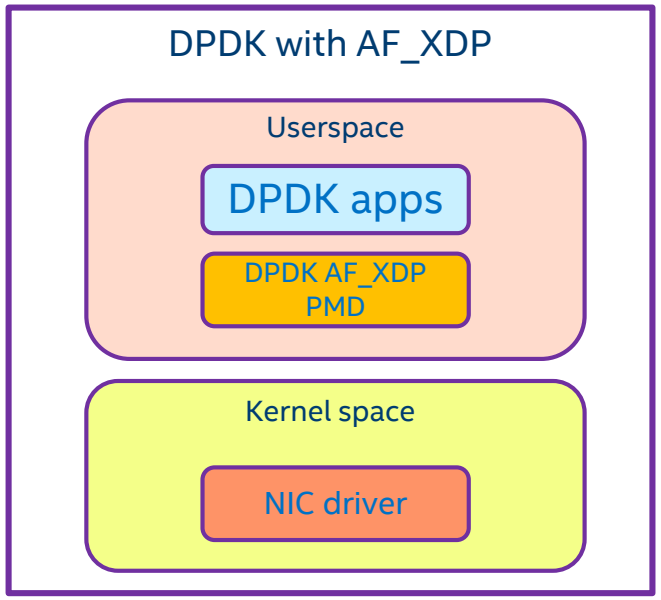
AF_XDP

- Userspace Libraries and Drivers.
- Accelerate packet processing workloads.
- Runs on wide variety of CPU architectures.
- Has its own memory management subsystem.
- Device specific PMDs (Poll Mode Drivers).

- Kernel based address family.
- Optimized for high performance packet processing.
- AF_XDP sockets redirect packets to userspace.
- By-passes kernel network stack
  - “In-kernel fast path”.







https://www.dpdk.org/wp-content/uploads/sites/35/2019/07/14-AF_XDP-dpdk-summit-china-2019.pdf

https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwikxb2Z7pnnAhVNY8AKHchXDfAQjhx6BAgBEAI&url=https%3A%2F%2Fwww.cdrecycler.com%2Finterstate-waste-apex-merger.aspx&psig=AOvVaw0QFTtUxvONAgVjKXIMC_-v&ust=1579873577016036

Problem Statement

Problem Statement

- The Goal:
  - All DPDK applications should be able to run out-of-the-box with the AF_XDP PMD.
  - DPDK app + AF_XDP PMD should run with good performance.

Problem Statement

- The Goal:
  - All DPDK applications should be able to run out-of-the-box with the AF_XDP PMD.
  - DPDK app + AF_XDP PMD should run with good performance.

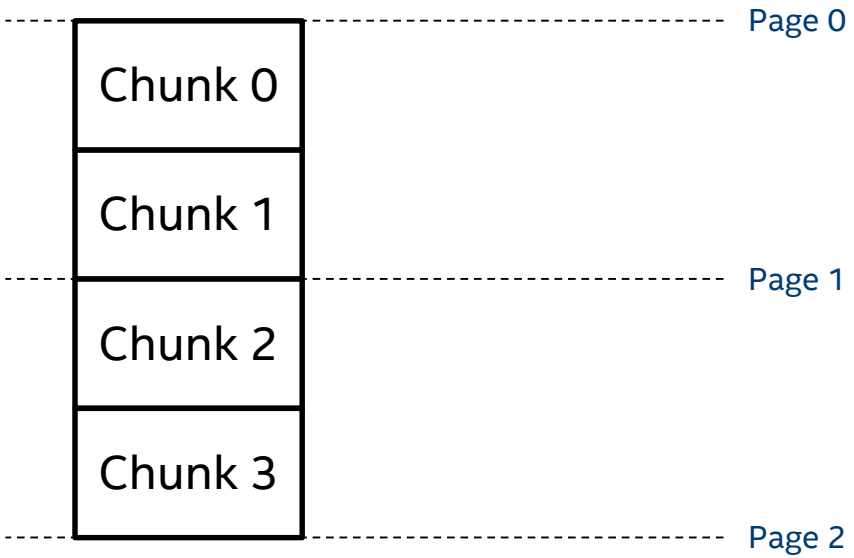
- The Challenge:
  - Frameworks like DPDK have their own memory management which come with their own assumptions and constraints.

Problem Statement

- The Goal:
  - All DPDK applications should be able to run out-of-the-box with the AF_XDP PMD.
  - DPDK app + AF_XDP PMD should run with good performance.

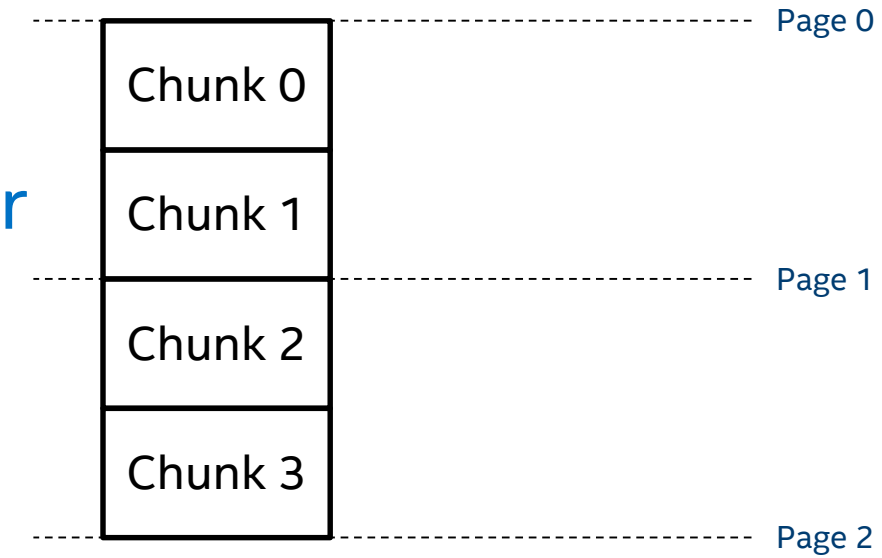
- The Challenge:
  - Frameworks like DPDK have their own memory management which come with their own assumptions and constraints.
  - Discrepancy between the DPDK and AF_XDP buffer alignment.
    - Prevents direct mapping of DPDK mempool to AF_XDP UMEM.
    - Extra complexity/work negatively impacting performance.

AF_XDP UMEM vs DPDK Mbuf Pool



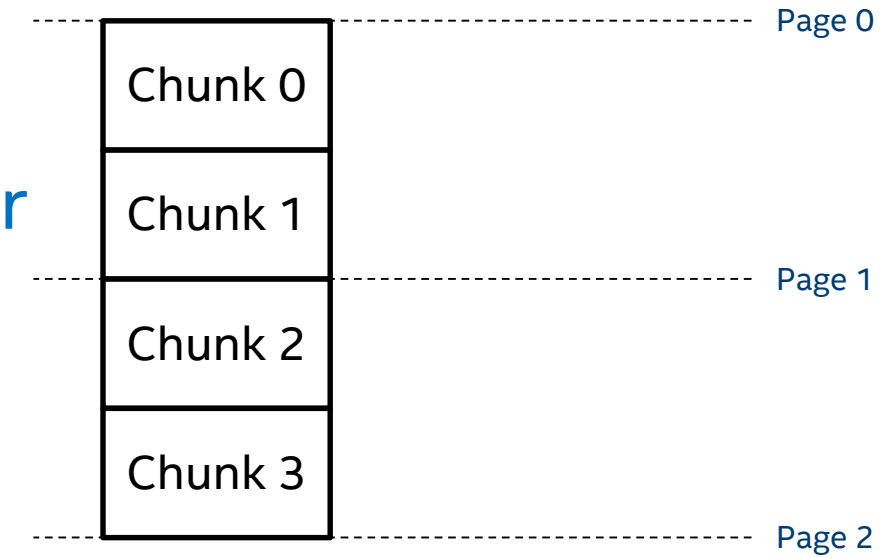
AF_XDP UMEM vs DPDK Mbuf Pool

- Area of memory allocated by the user for packet data



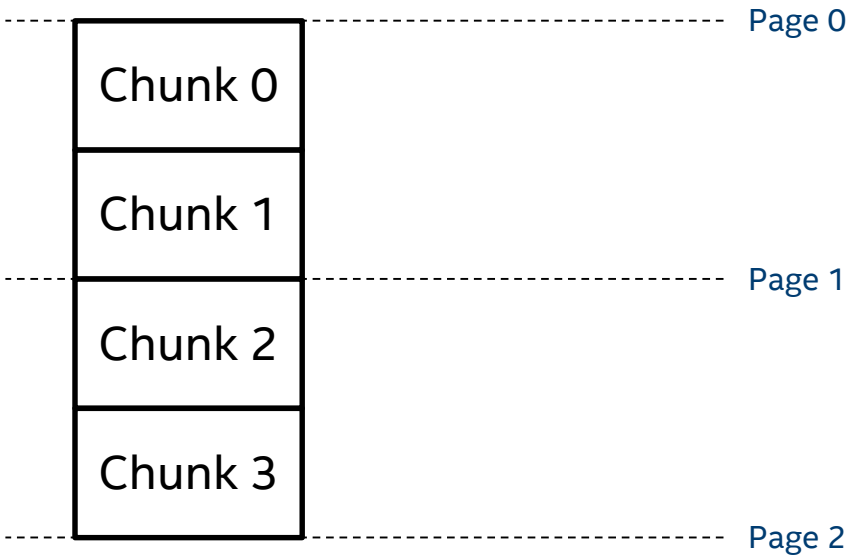
AF_XDP UMEM vs DPDK Mbuf Pool

- Area of memory allocated by the user for packet data
- Split up into equal sized-chunks



AF_XDP UMEM vs DPDK Mbuf Pool

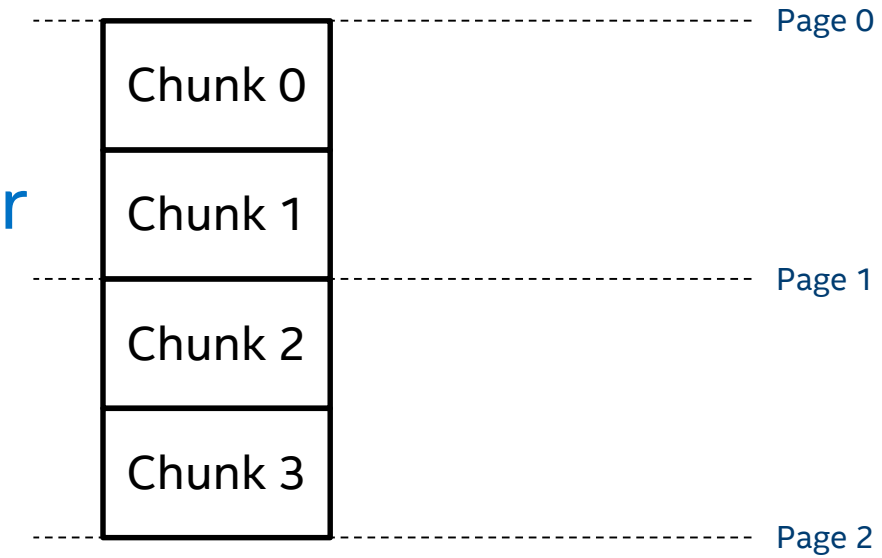
- Area of memory allocated by the user for packet data
- Split up into equal sized-chunks



- RX Path: Kernel places packet data in a chunk for userspace (DPDK) to retrieve

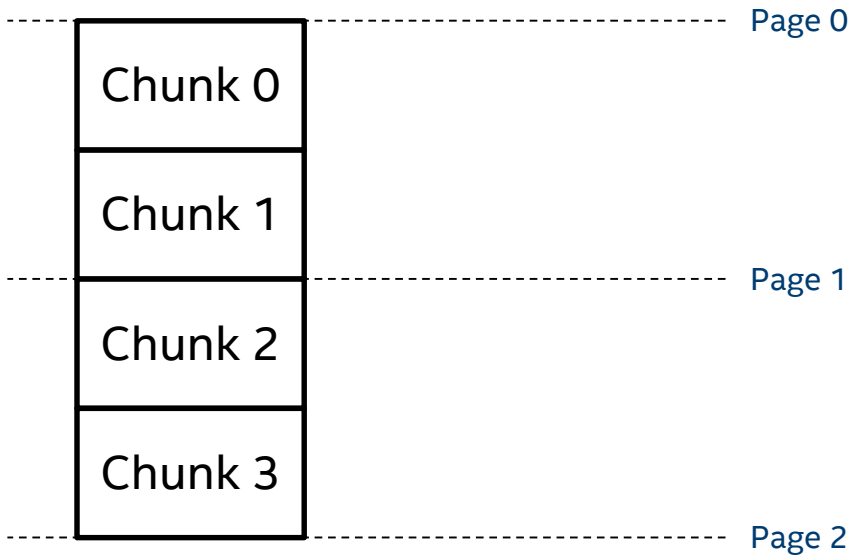
AF_XDP UMEM vs DPDK Mbuf Pool

- Area of memory allocated by the user for packet data
- Split up into equal sized-chunks



- RX Path: Kernel places packet data in a chunk for userspace (DPDK) to retrieve
- TX Path: Userspace places packet data in chunk for kernel NIC driver to transmit

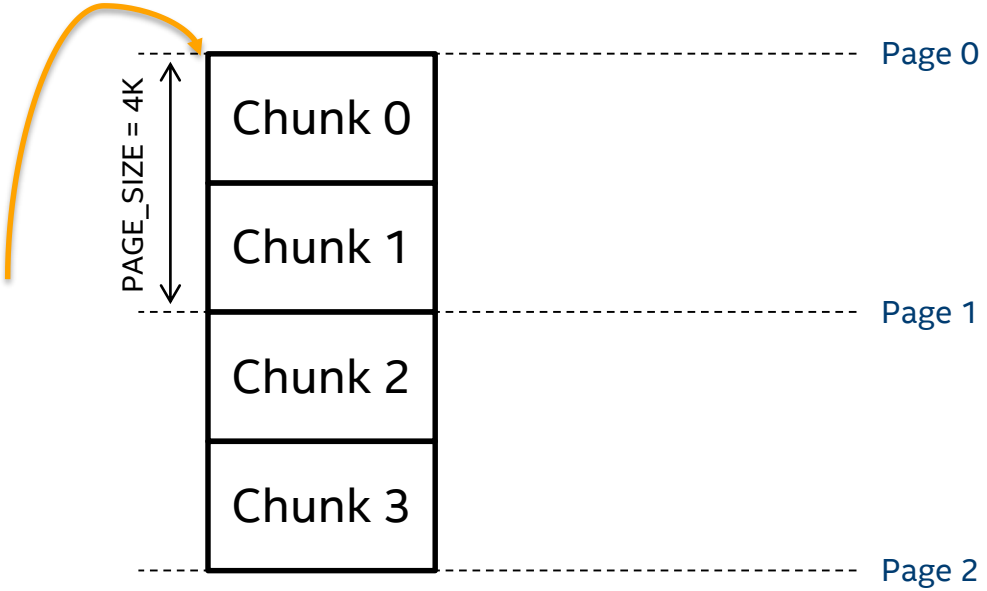
AF_XDP UMEM* vs DPDK Mbuf Pool



*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

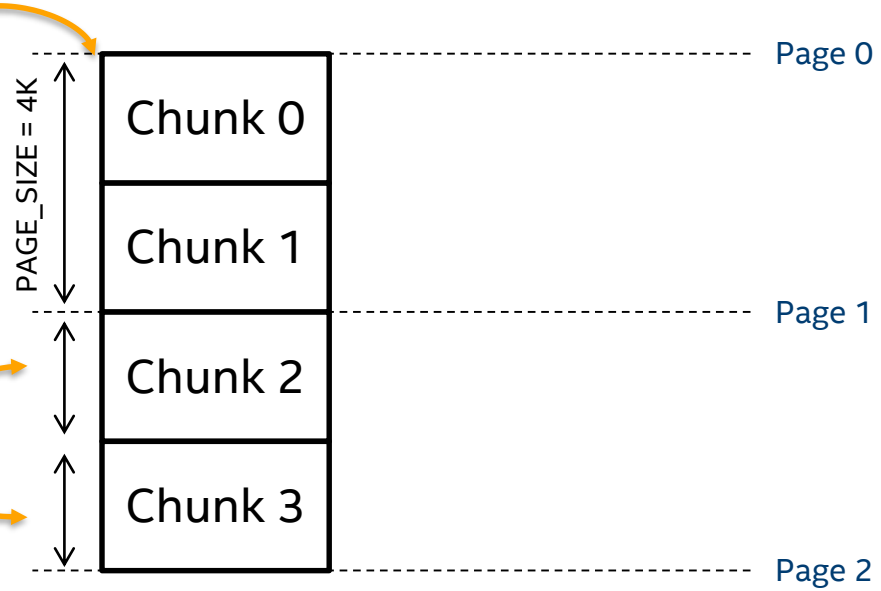


*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized



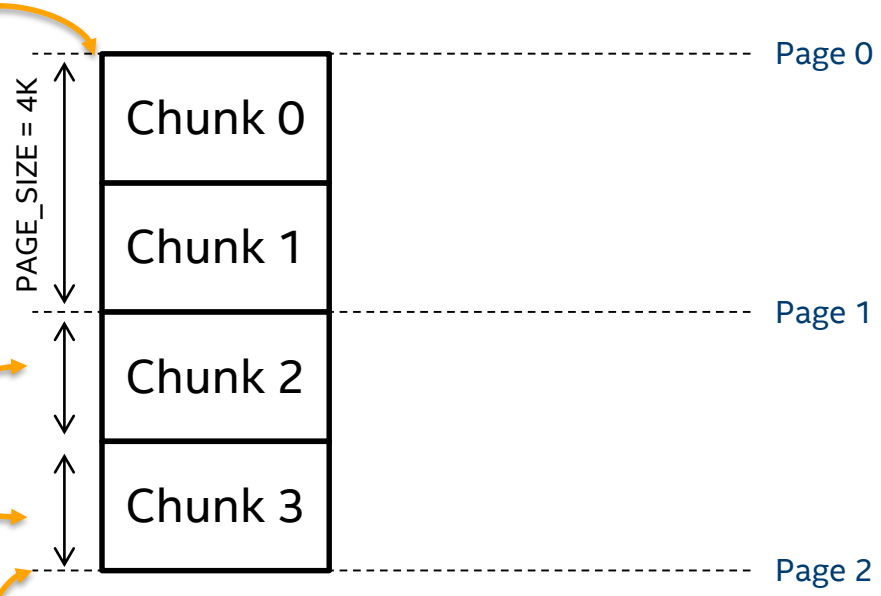
*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized

Chunks may not cross page boundaries



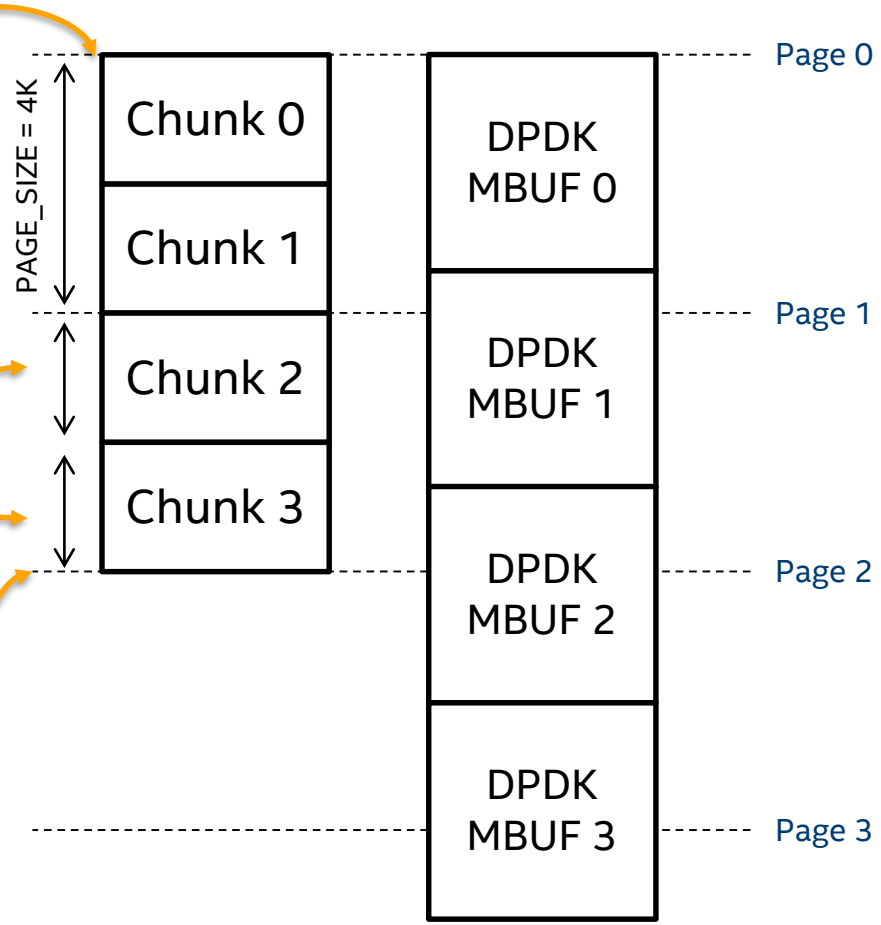
*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized

Chunks may not cross page boundaries



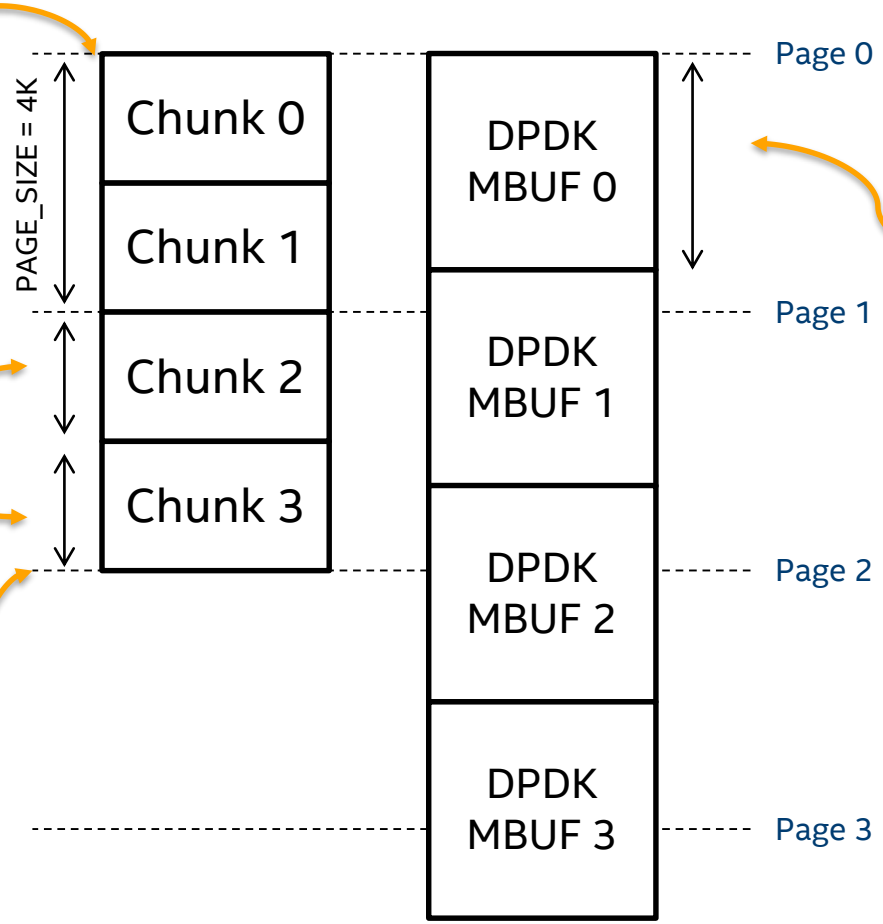
*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized

Chunks may not cross page boundaries



Mbufs can be any size

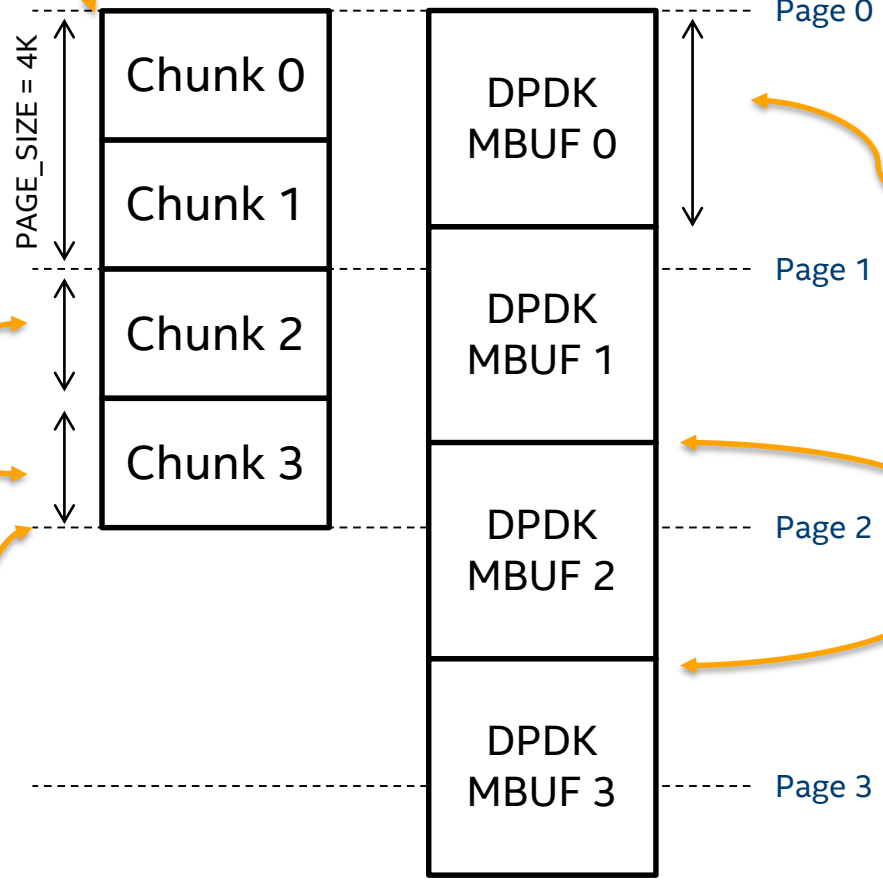
*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool

Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized

Chunks may not cross page boundaries

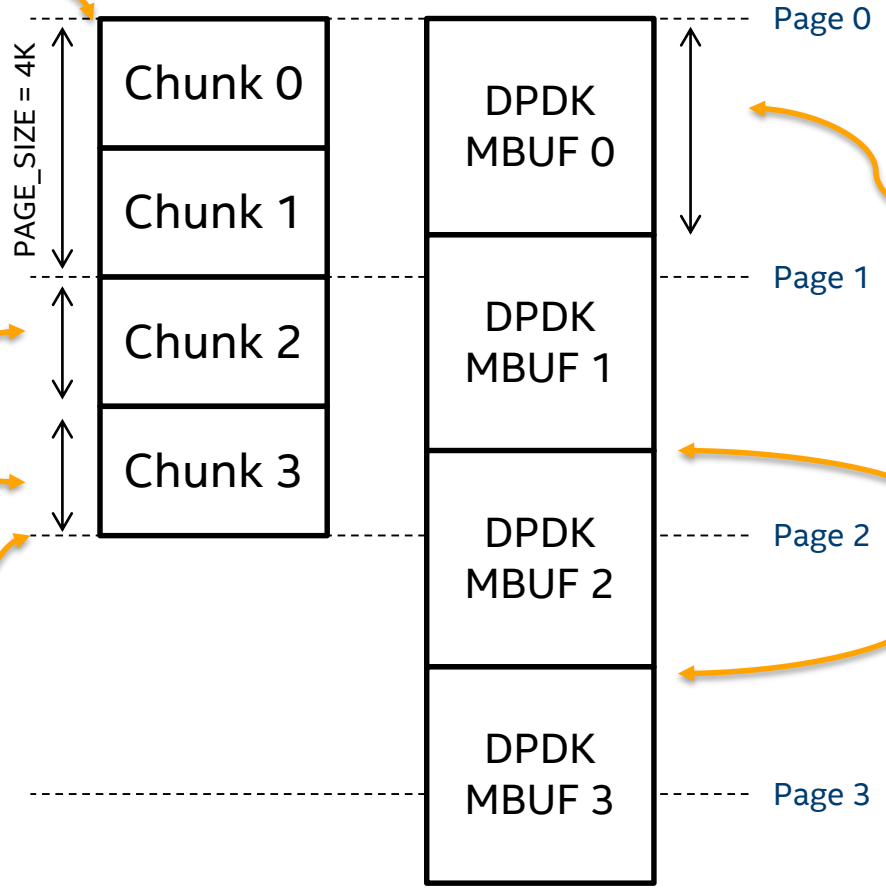


Mbufs can be any size

Mbufs can have arbitrary alignment

*Prior to Kernel 5.4

AF_XDP UMEM* vs DPDK Mbuf Pool



Start address must be PAGE_SIZE aligned

Chunks must be ^2 sized

Chunks may not cross page boundaries

Mbufs can be any size

Mbufs can have arbitrary alignment

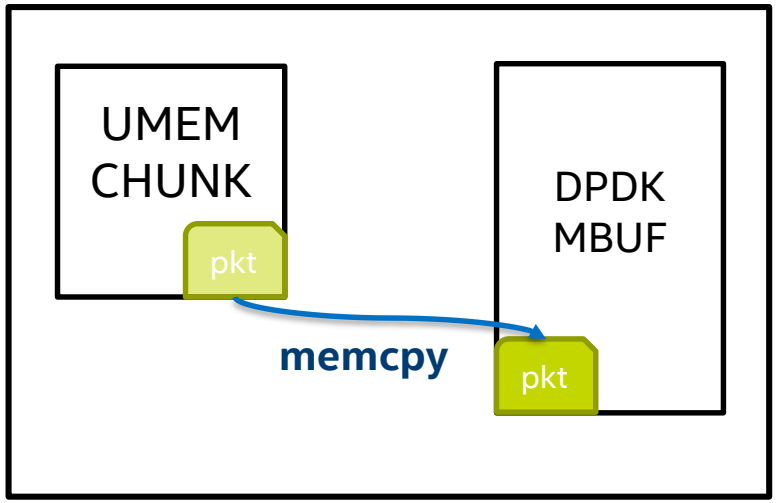
*Prior to Kernel 5.4

To get the highest performing integration of AF_XDP and DPDK, the DPDK mbuf pool must be mapped into the UMEM for a zero copy datapath.

DPDK Solutions

DPDK Solutions

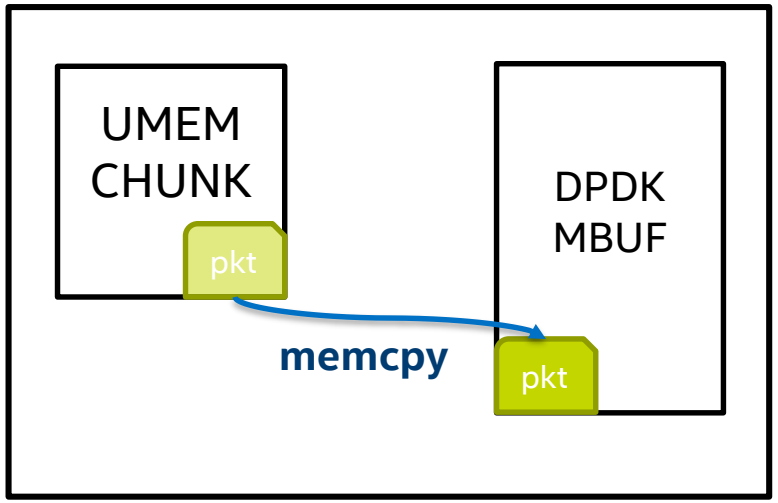
Copy Mode



- Memcpy packets between UMEM & mbufpool
- Cycle-heavy memcpy
- DPDK 19.05

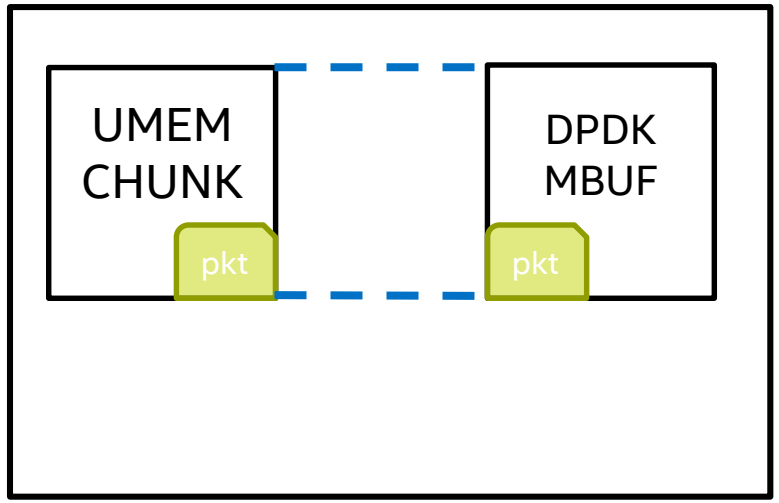
DPDK Solutions

Copy Mode



- Memcpy packets between UMEM & mbufpool
- Cycle-heavy memcpy
- DPDK 19.05

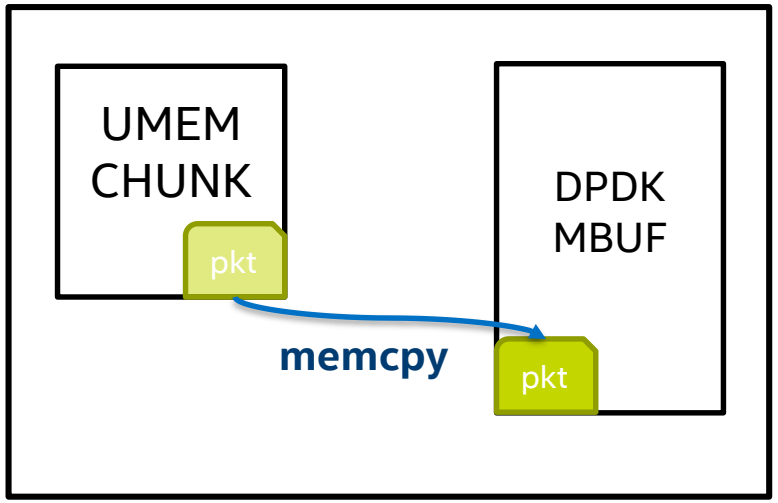
Alignment API



- API to change mbuf pool alignment, then 1:1 map
- Invasive change
- No DPDK release

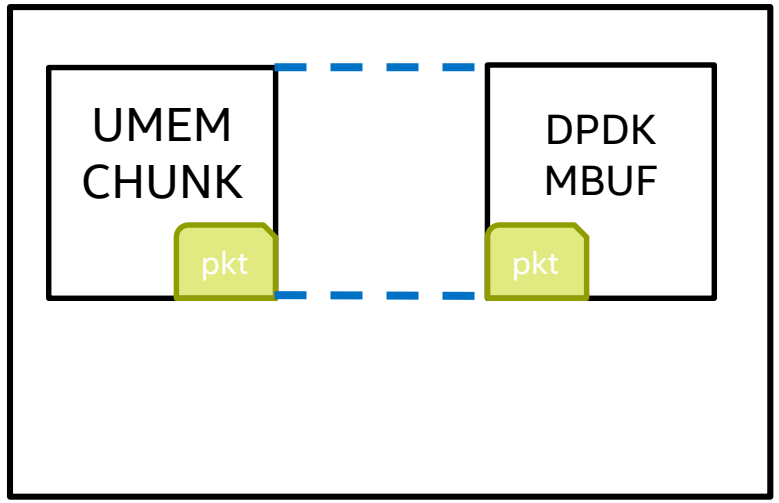
DPDK Solutions

Copy Mode



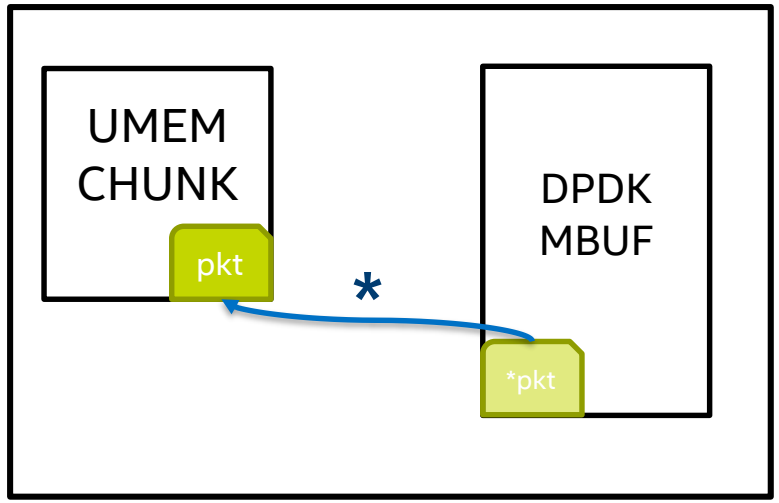
- Memcpy packets between UMEM & mbufpool
- Cycle-heavy memcpy
- DPDK 19.05

Alignment API



- API to change mbuf pool alignment, then 1:1 map
- Invasive change
- No DPDK release

External Mbuf



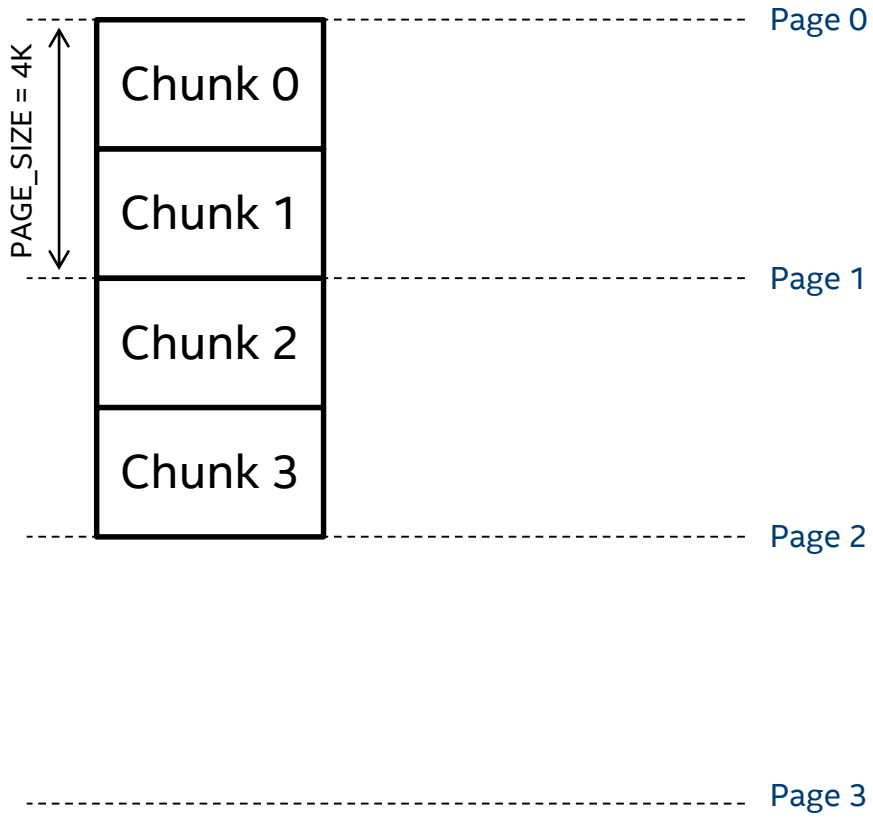
- Mbuf points to packet data in UMEM chunk
- Additional complexity
- DPDK 19.08

New Solution: Kernel Arbitrary Alignment

New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

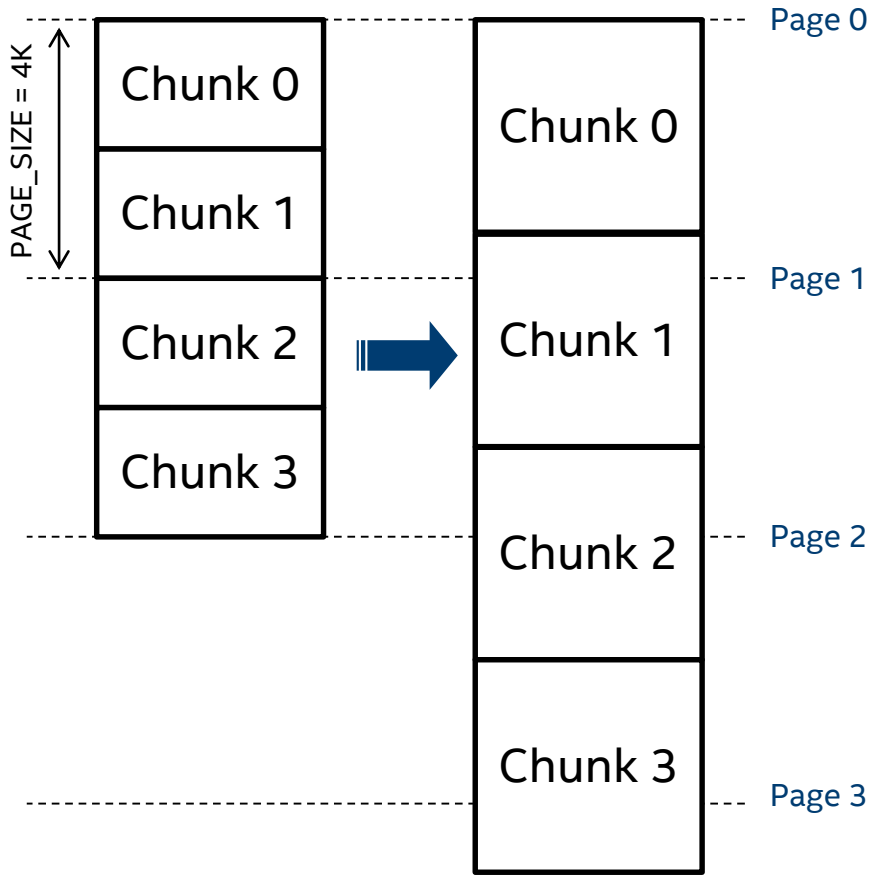
- Enables arbitrary chunk alignment.



New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

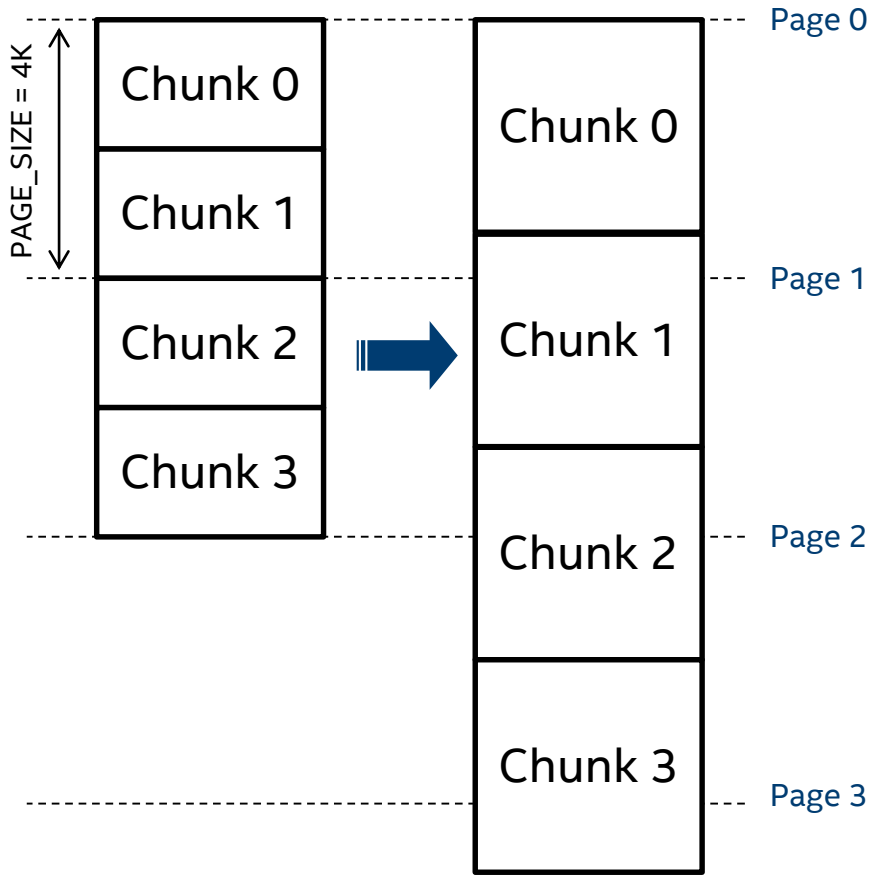
- Enables arbitrary chunk alignment.



New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

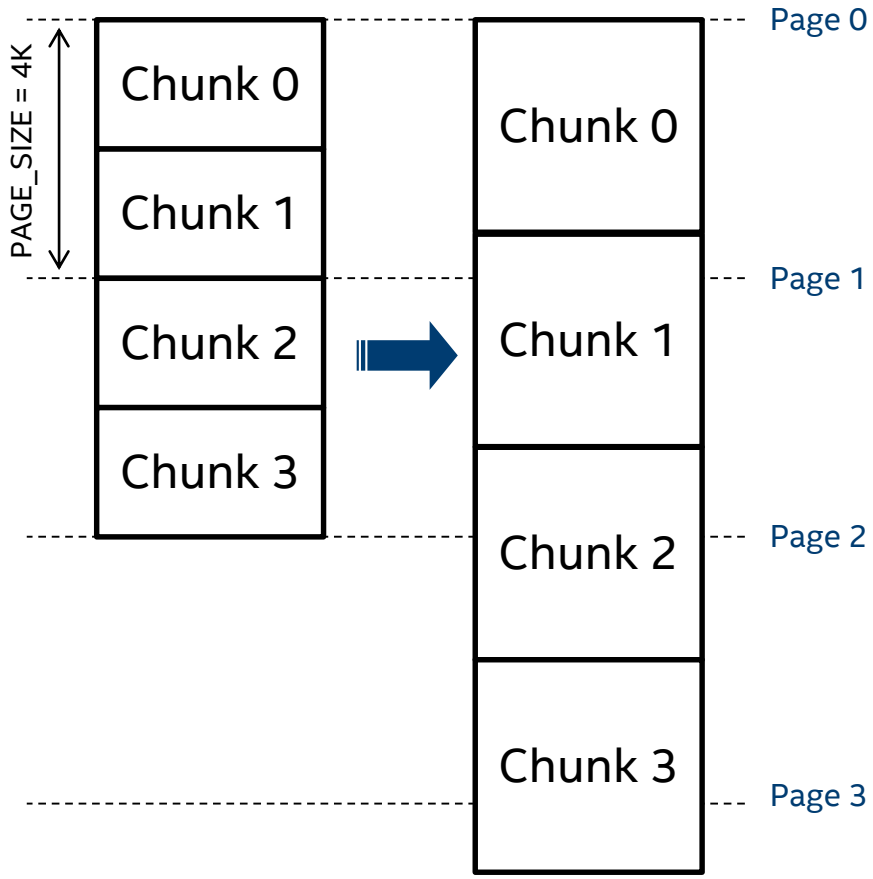
- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).



New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

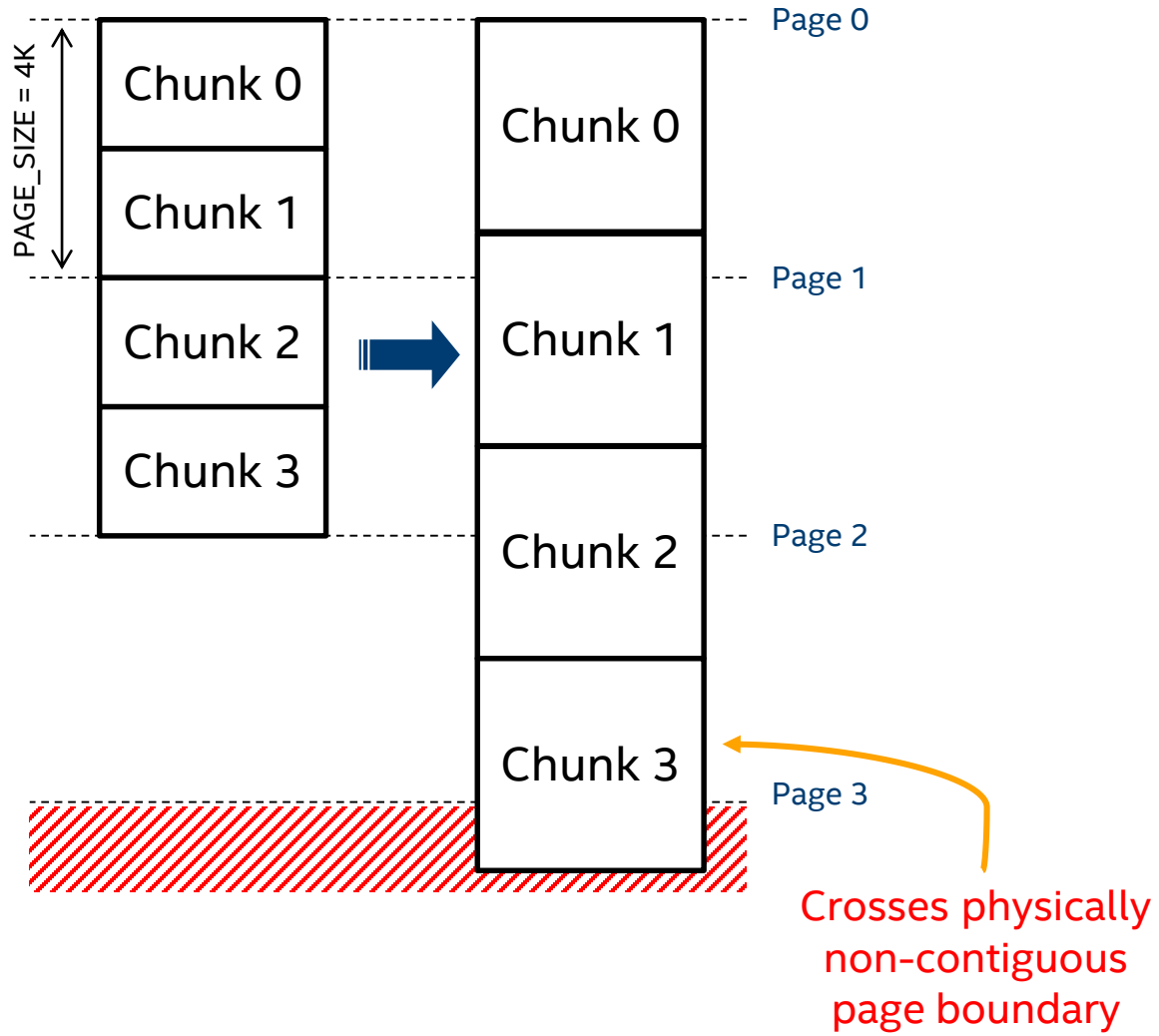
- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.



New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

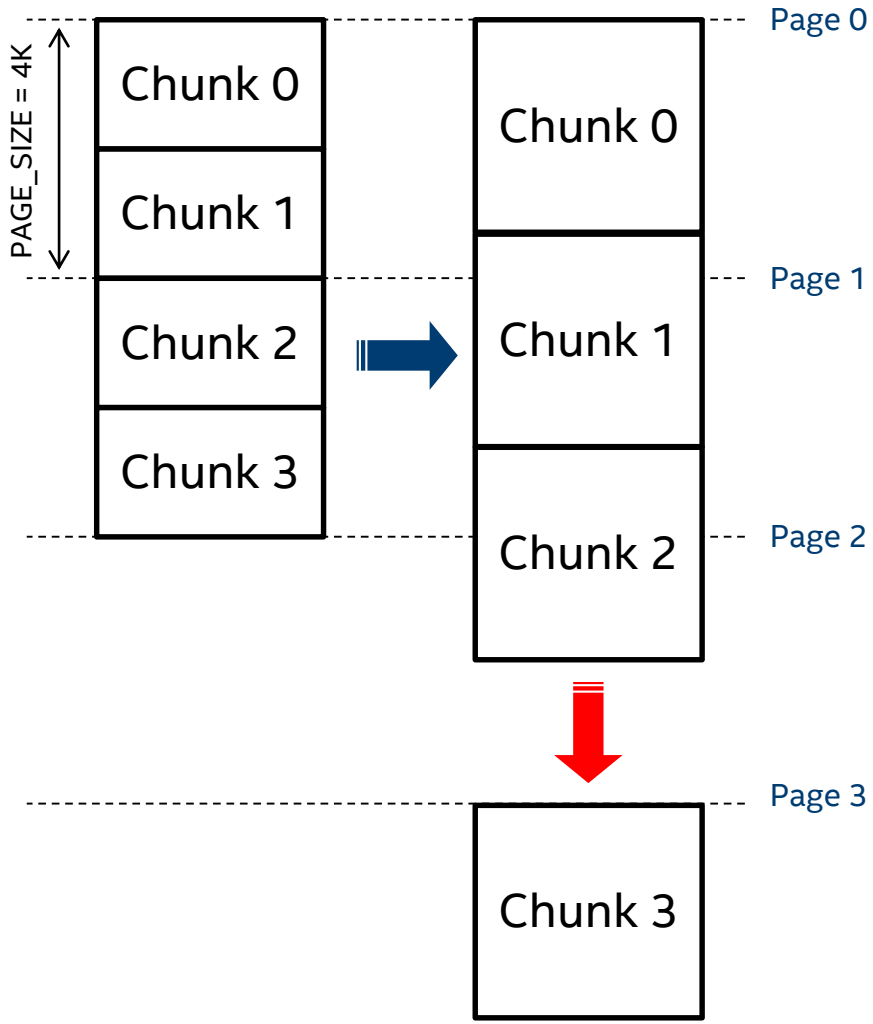
- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.
  - If physically contiguous!



New Solution: Kernel Arbitrary Alignment

Main benefits of the new solution:

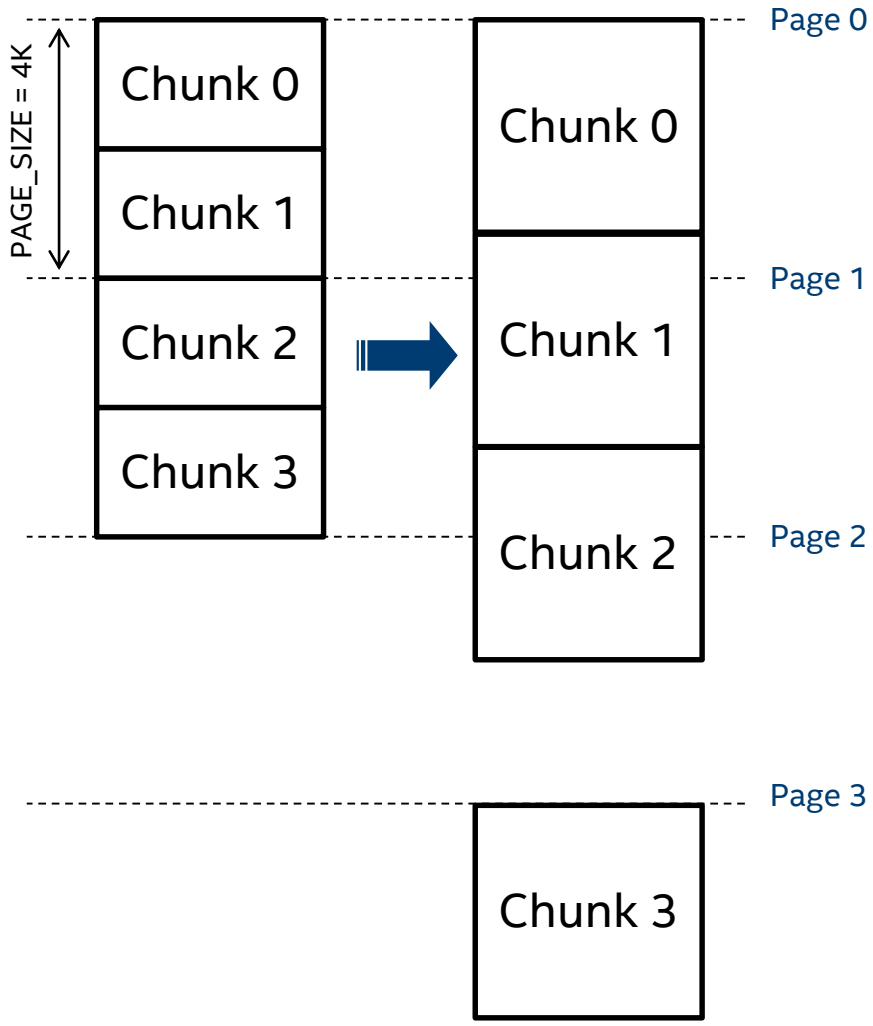
- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.
  - If physically contiguous!



New Solution: Kernel Arbitrary Alignment

AF_XDP Rx/Tx Descriptor:

64-bit “address” field

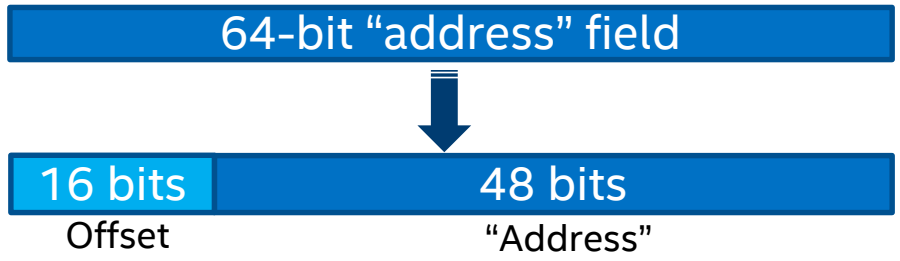


Main benefits of the new solution:

- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.
  - If physically contiguous!

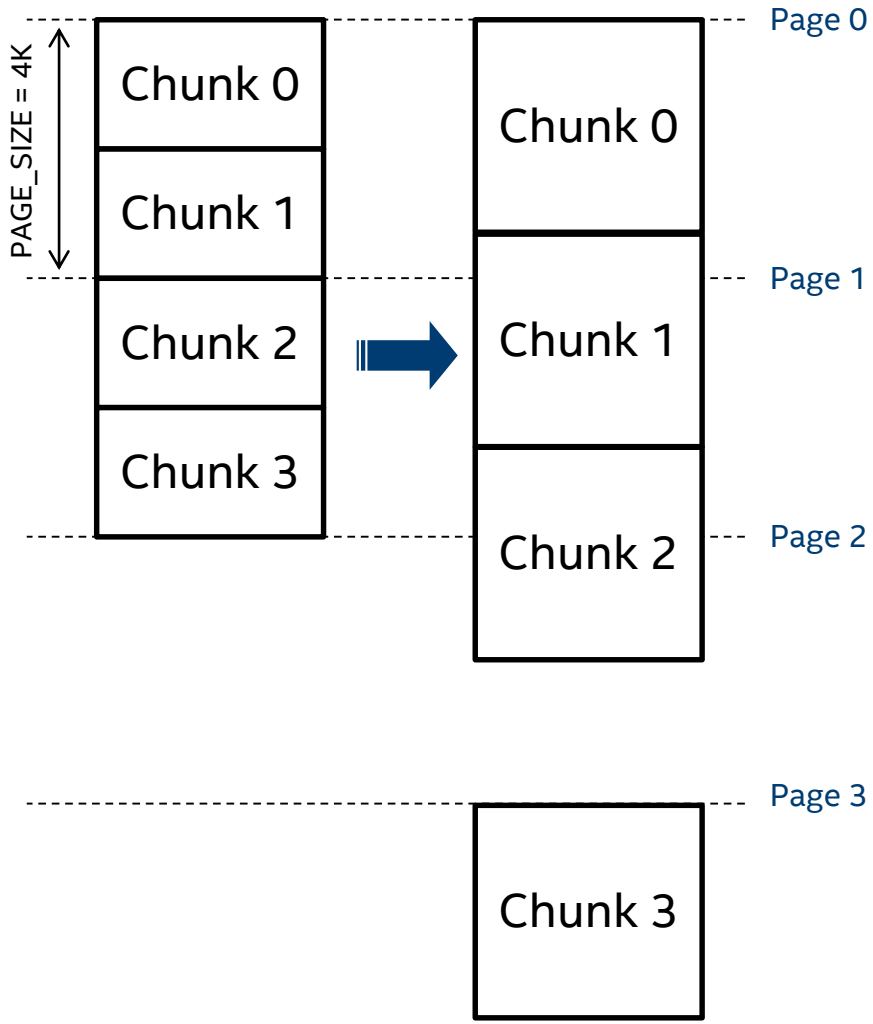
New Solution: Kernel Arbitrary Alignment

AF_XDP Rx/Tx Descriptor:



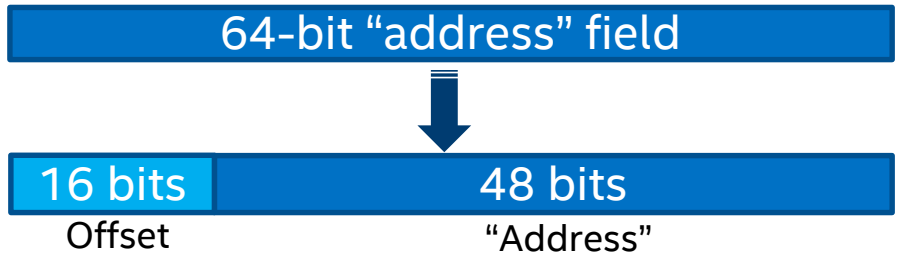
Main benefits of the new solution:

- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.
  - If physically contiguous!
- New descriptor format keeps original address which is useful for buffer recycling.



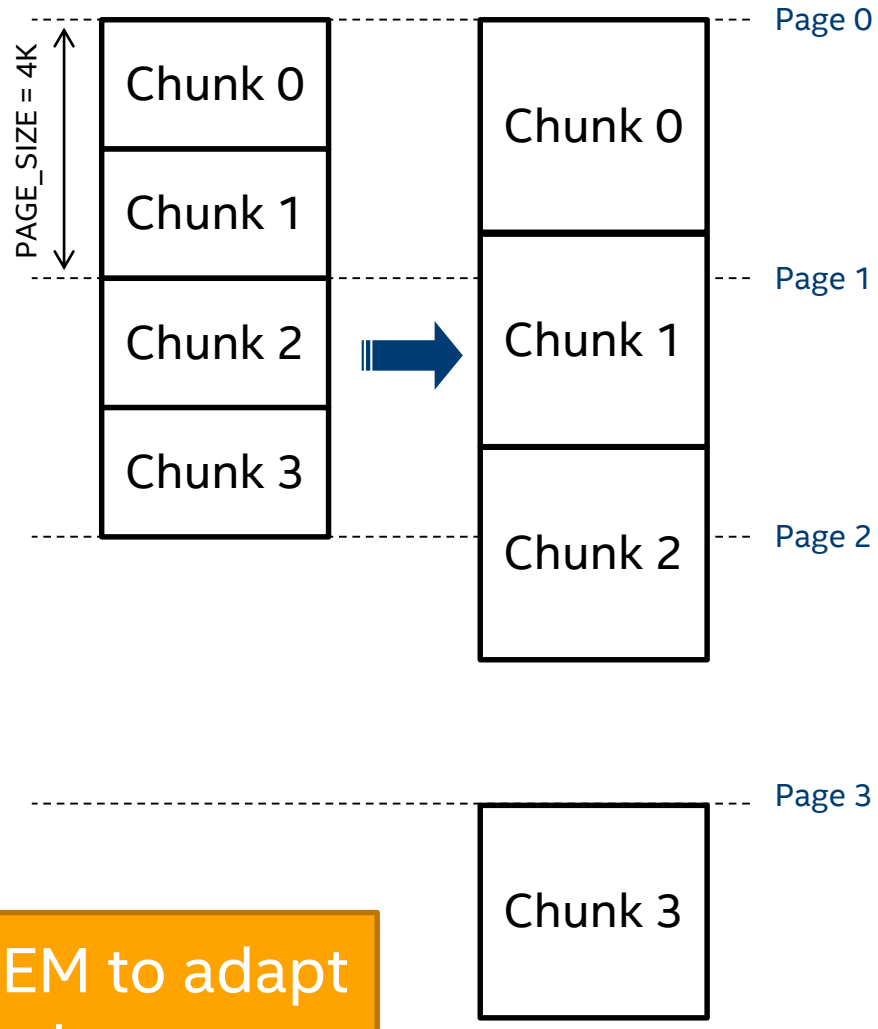
New Solution: Kernel Arbitrary Alignment

AF_XDP Rx/Tx Descriptor:



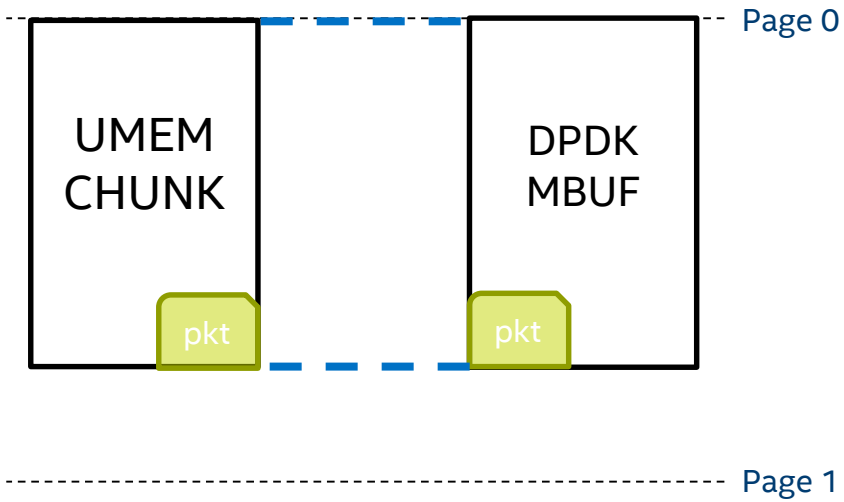
Main benefits of the new solution:

- Enables arbitrary chunk alignment.
- Enables use of arbitrary chunk size (up to 4k).
- Chunks can cross page boundaries.
  - If physically contiguous!
- New descriptor format keeps original address which is useful for buffer recycling.
- Makes integrating AF_XDP with existing frameworks more seamless.



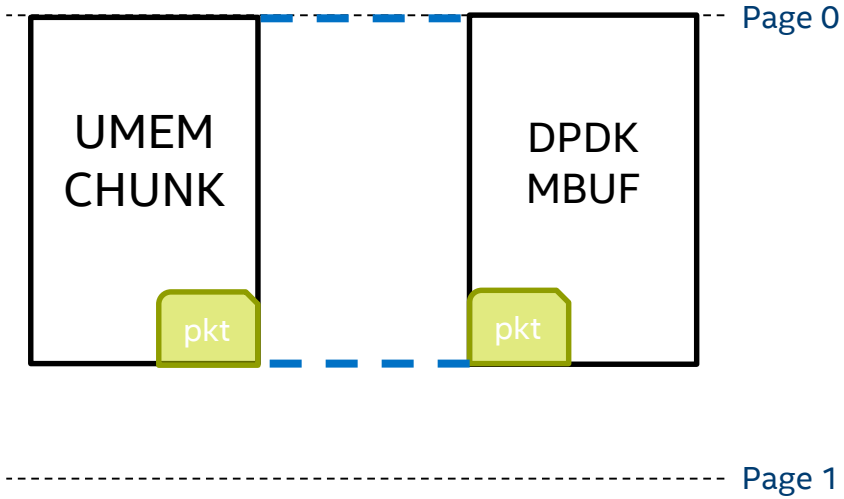
New arbitrary alignment enables the UMEM to adapt to the requirements of the application

DPDK Integration



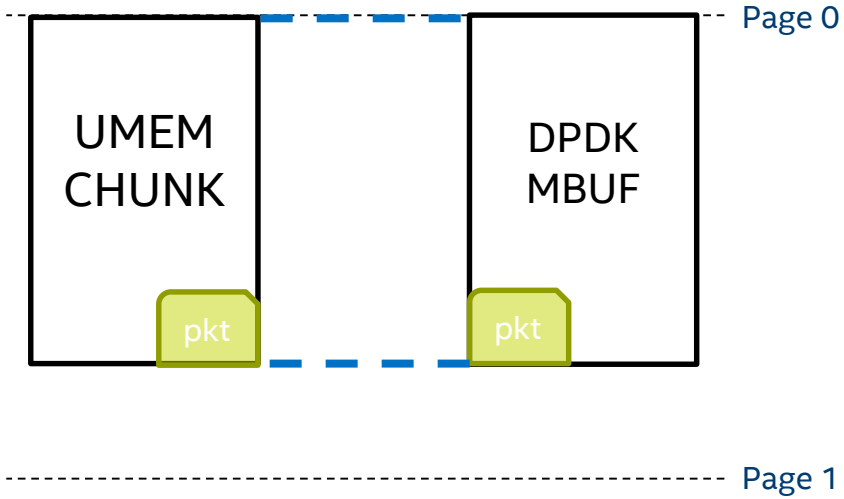
DPDK Integration

- Now that UMEM alignment constraints are relaxed, DPDK mbuf pools can be directly mapped into the UMEM



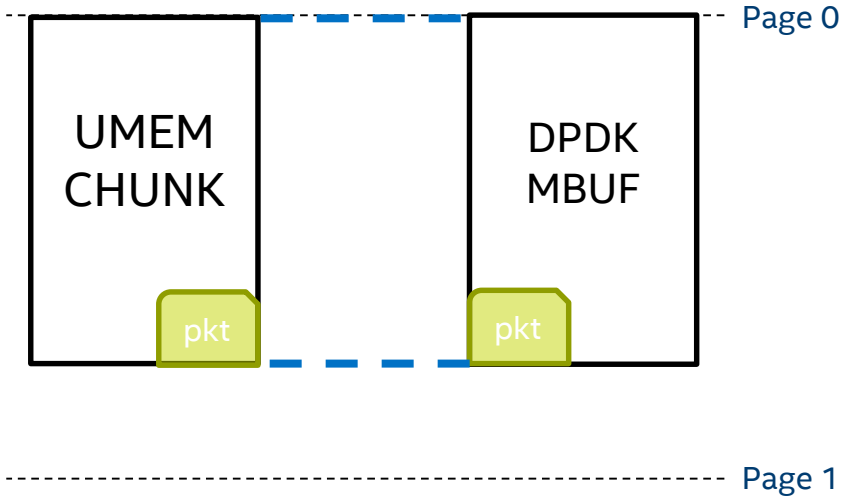
DPDK Integration

- Now that UMEM alignment constraints are relaxed, DPDK mbuf pools can be directly mapped into the UMEM
- Seamless zero-copy is now achievable



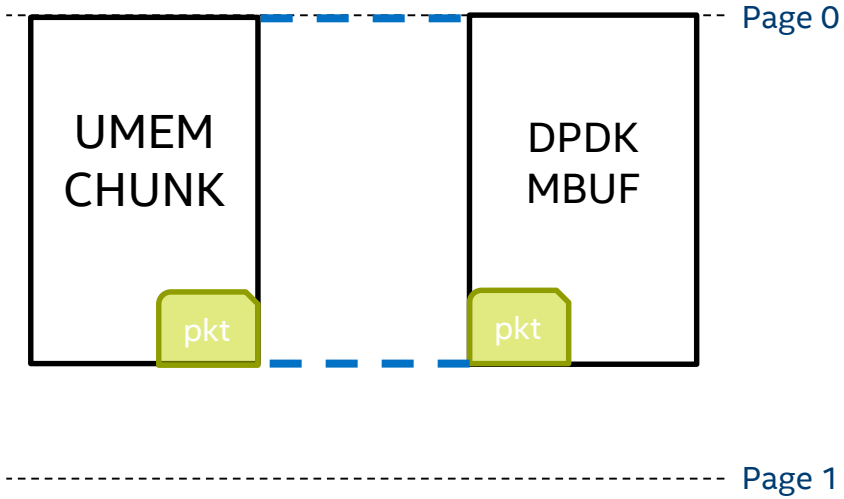
DPDK Integration

- Now that UMEM alignment constraints are relaxed, DPDK mbuf pools can be directly mapped into the UMEM
- Seamless zero-copy is now achievable
- No need to modify existing DPDK applications - they work OOTB



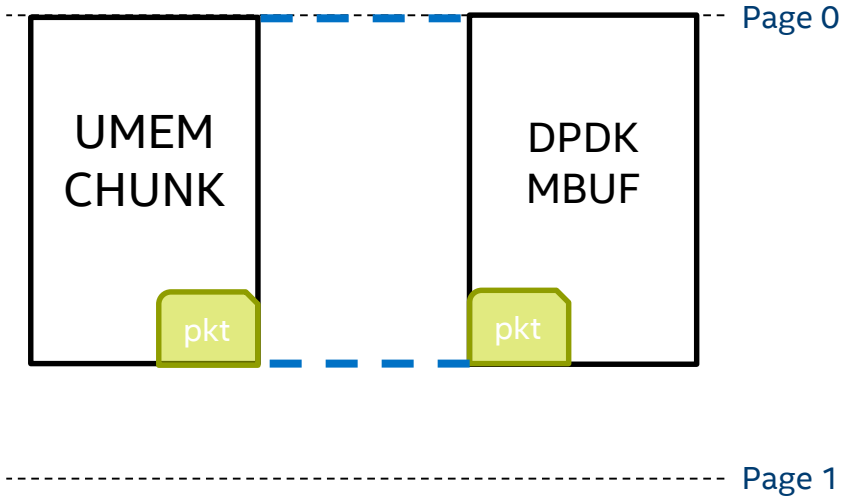
DPDK Integration

- Now that UMEM alignment constraints are relaxed, DPDK mbuf pools can be directly mapped into the UMEM
- Seamless zero-copy is now achievable
- No need to modify existing DPDK applications - they work OOTB
- Performant **and** portable solution

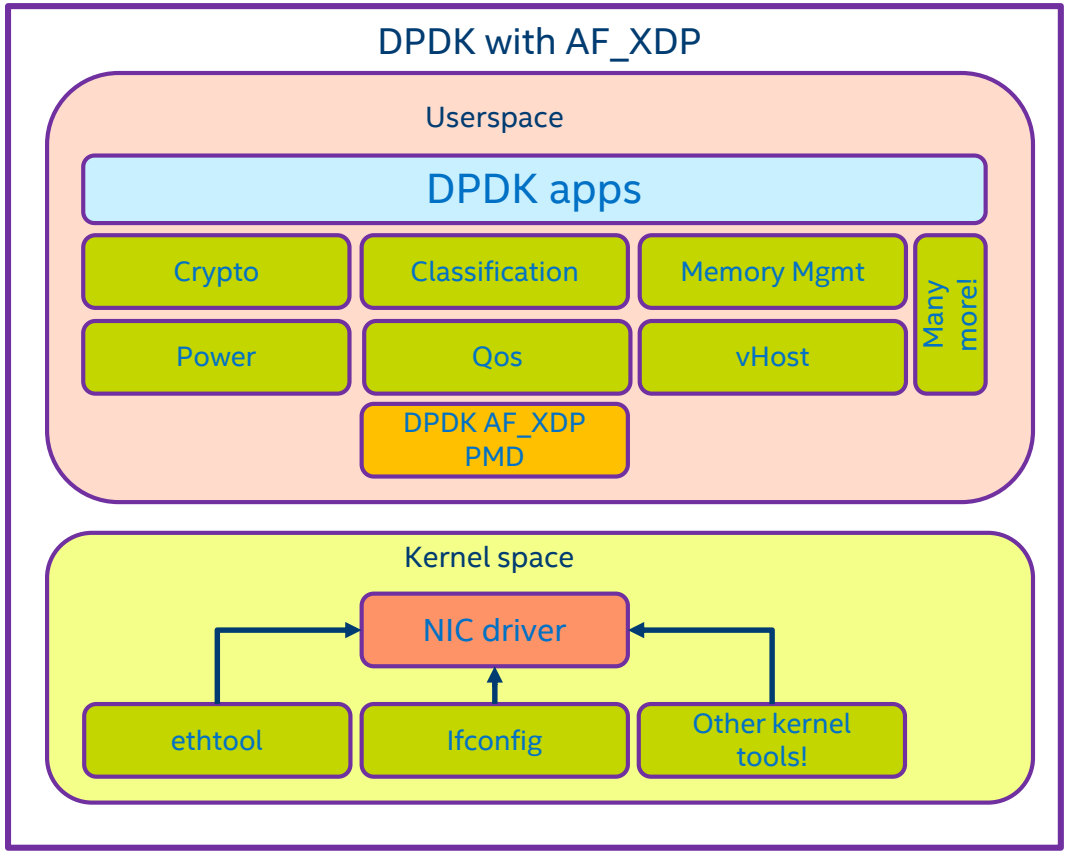


DPDK Integration

- Now that UMEM alignment constraints are relaxed, DPDK mbuf pools can be directly mapped into the UMEM
- Seamless zero-copy is now achievable
- No need to modify existing DPDK applications - they work OOTB
- Performant **and** portable solution
- DPDK 19.11 feature



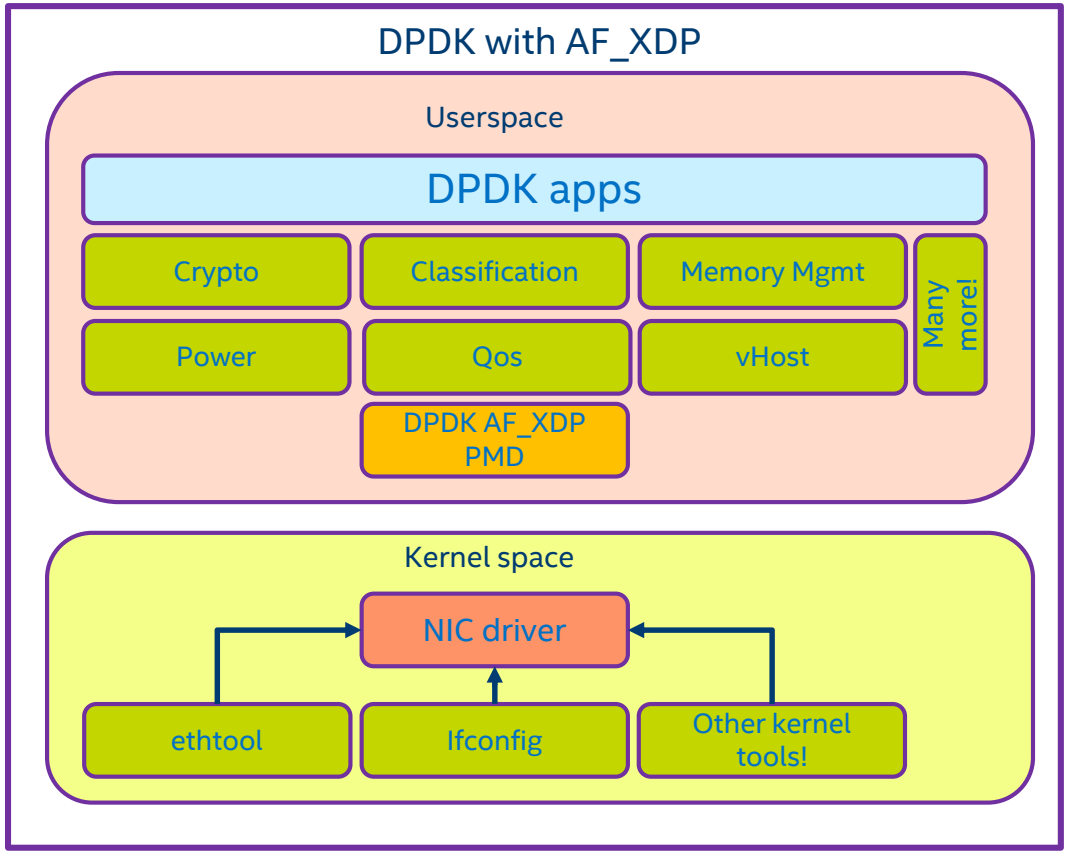
Summary


DPDK with AF_XDP
Userspace
DPDK apps
Crypto
Classification
Memory Mgmt
Many more!
Power
Qos
vHost
DPDK AF_XDP PMD
Kernel space
NIC driver
ethtool
Ifconfig
Other kernel tools!

Summary

- DPDK provides a wide range of functionality to an application, eg.:

  Memory & power management, crypto libraries, virtual networking & many more!



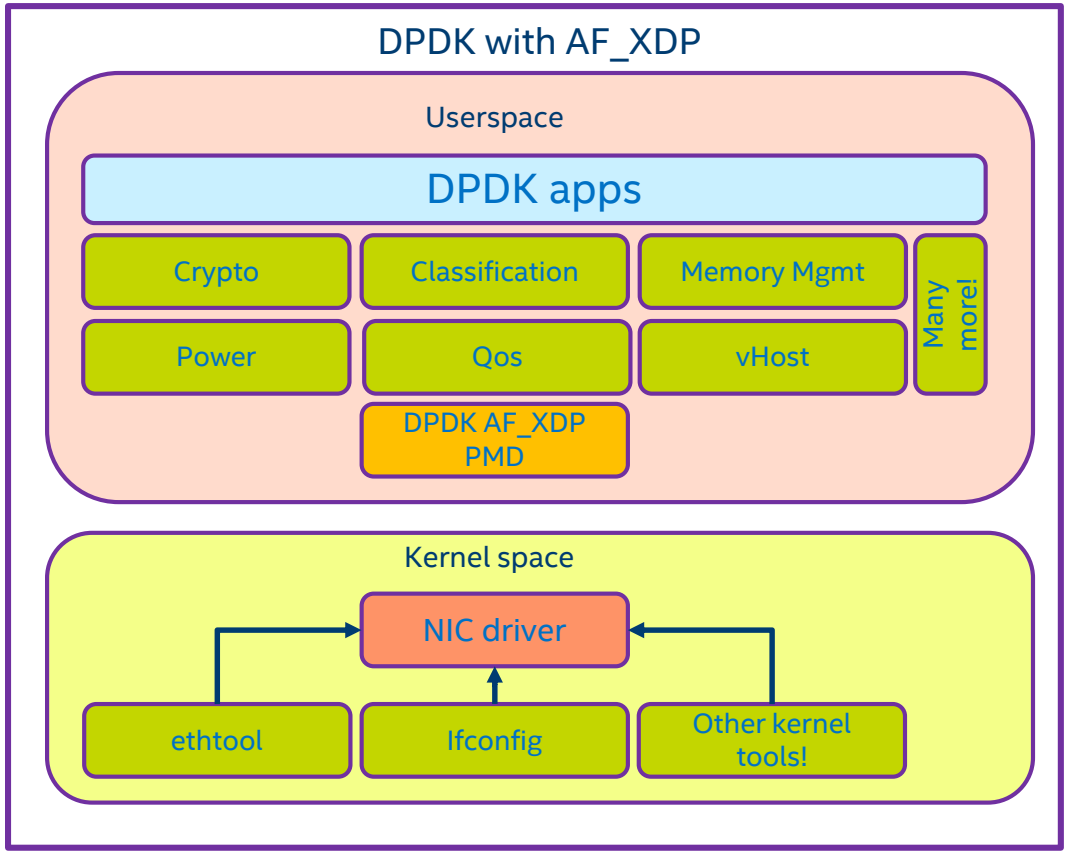
Summary

- DPDK provides a wide range of functionality to an application, eg.:

  Memory & power management, crypto libraries, virtual networking & many more!

- AF_XDP provides flexibility and usability through kernel control paths.

  Familiar tools eg. ifconfig, ethtool, etc.



Summary

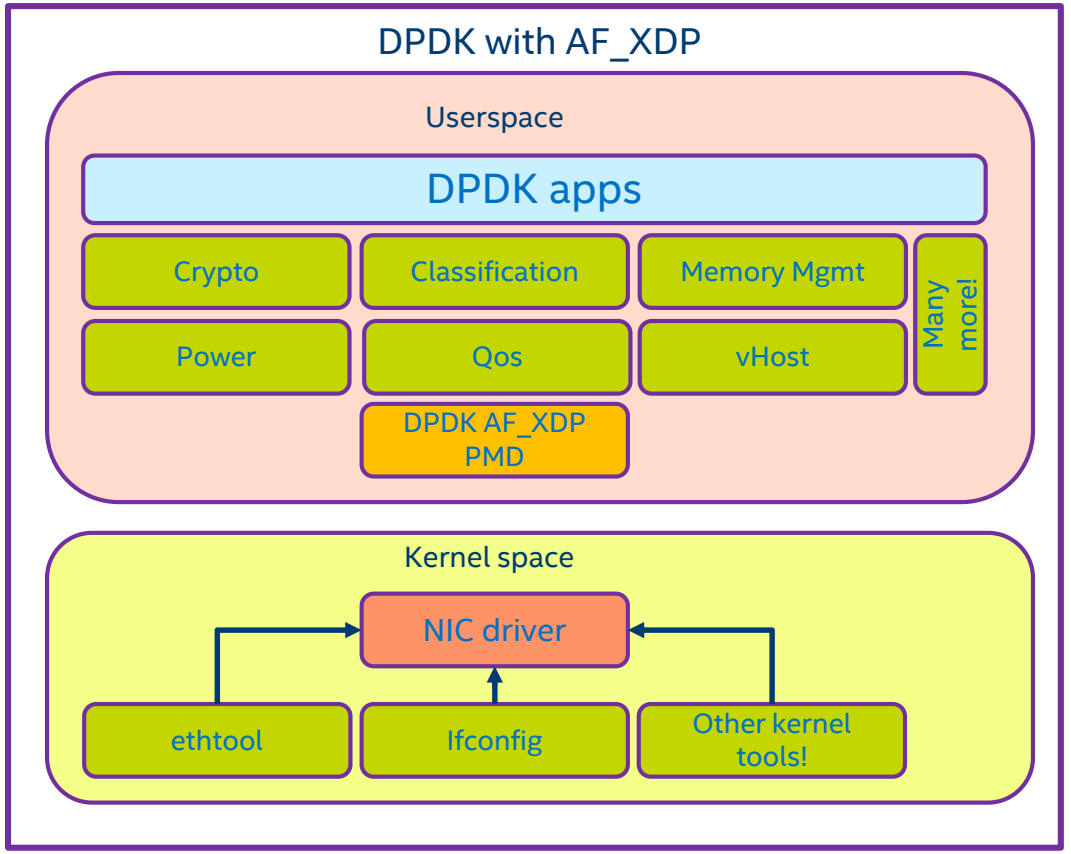
- DPDK provides a wide range of functionality to an application, eg.:

  Memory & power management, crypto libraries, virtual networking & many more!

- AF_XDP provides flexibility and usability through kernel control paths.

  Familiar tools eg. ifconfig, ethtool, etc.

- Together, the best of both worlds can be enjoyed.



Summary

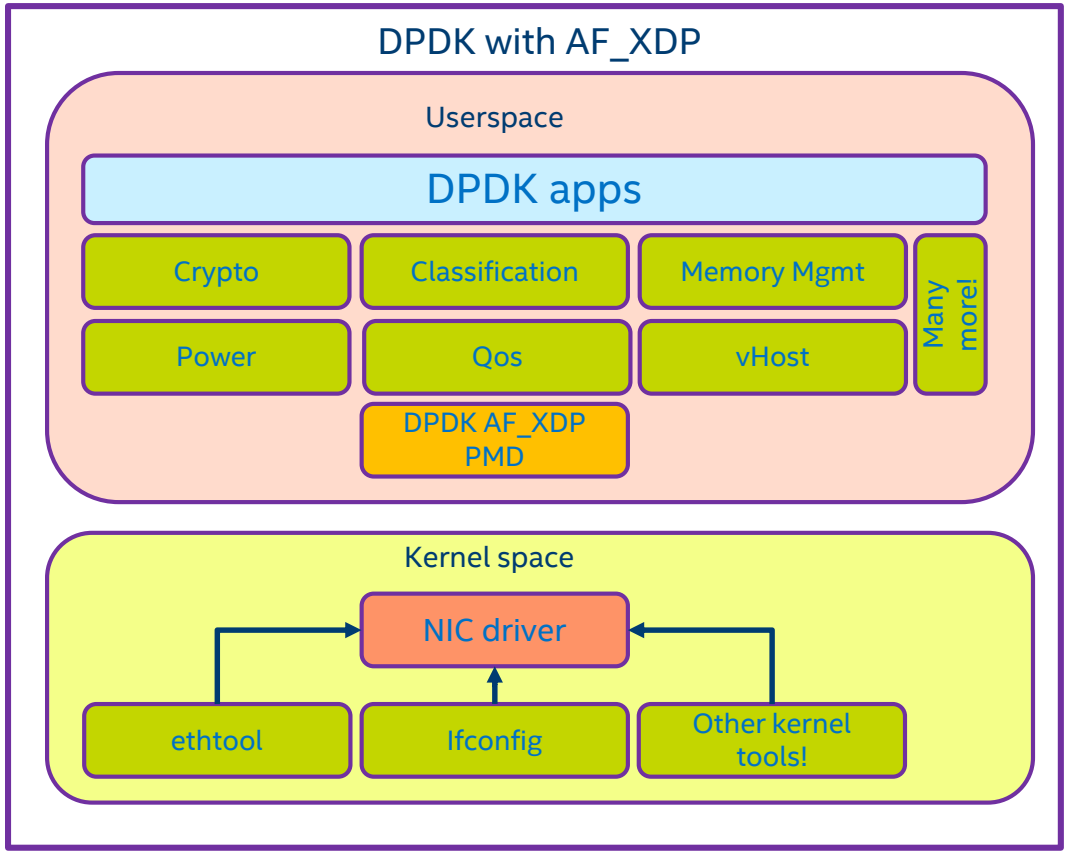
- DPDK provides a wide range of functionality to an application, eg.:

  Memory & power management, crypto libraries, virtual networking & many more!

- AF_XDP provides flexibility and usability through kernel control paths.

  Familiar tools eg. ifconfig, ethtool, etc.

- Together, the best of both worlds can be enjoyed.

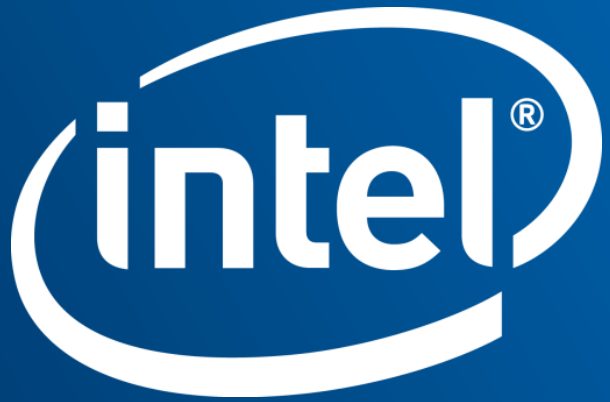


High performing, portable, fully-featured, accelerated, usable and flexible applications are possible with DPDK + AF_XDP

Thanks to..

Magnus Karlsson, Björn Töpel,
Bruce Richardson, Qi Zhang, Xiaolong Ye,
DPDK & Kernel Communities

Q&A


intel