



DPDK

DATA PLANE DEVELOPMENT KIT



TECHNISCHE UNIVERSITÄT
ILMENAU

6.5.2021

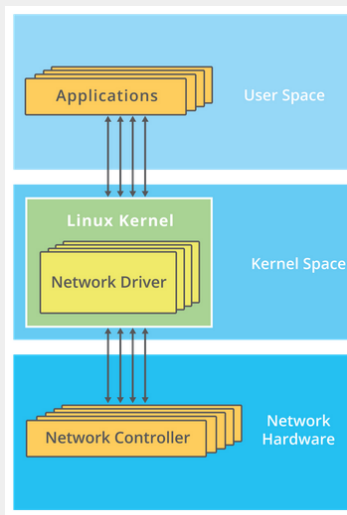
- Sammlung von Bibliotheken
- 2010 von Intel veröffentlicht
- freie OpenSource Lizenz
- unterstützt alle wichtigen CPU-Architekturen und NICs

- viele Engpässe verhindern Verarbeitung von Paketen
- Bottleneck im Linux Kernel selbst
- Kernel-Bypass
- Anwendungen kommunizieren direkt mit Netzwerkgeräten

- 1 DPDK vs. Kernel Processing
- 2 DPDK Aufbau
- 3 Paketrepräsentation
- 4 Empfang und Senden von Paketen

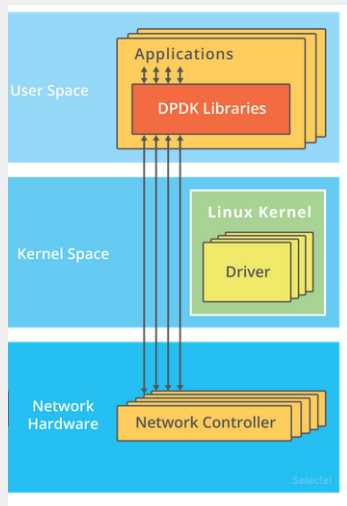
DPDK vs. KERNEL PROCESSING

LINUX KERNEL OHNE DPDK



- Netzwerkkarte sendet Paket zu Warteschlange
- per DMA in Hauptspeicher kopieren
- Kernel weißt Puffer zu **Interrupts!**
- Paket aus Puffer in User Space übertragen

LINUX KERNEL MIT DPDK



- Netzwerkports vom Kernel getrennt/ungebunden
- Eingehende Pakete gehen in Ringpuffer
- Wenn Puffer neue Paketdeskriptoren enthält, verweis auf die DPDK-Paketpuffer
- Wenn Ringpuffer keine Pakete enthält, stelle Netzwerkgeräte in Warteschlange
- erfordert Konfiguration von **hugepages**

DPDK AUFBAU



ENVIRONMENT ABSTRACTION LAYER (EAL)

- Satz von Programmierwerkzeugen
- bindet Anwendungen an DPDK
- alle Anwendungen müssen Header Dateien der EAL enthalten

ENVIRONMENT ABSTRACTION LAYER (EAL)

die häufigsten/wichtigsten Header¹:

rte_lcore.h verwaltet Prozessorkerne und Sockel

rte_memory.h verwaltet Speicher

rte_pci.h Schnittstelle für den Zugriff auf PCI-Adressraum

rte_debug.h Trace- und Debug-Funktionen

rte_interrupts.h verarbeitet Interrupts

¹http://dpdk.org/doc/guides/prog_guide/env_abstraction_layer.html

PAKETREPRÄSENTATION

- Nutzung von PMDs (Poll Mode Drivers)²
- Pakete werden in Speicherblöcken untergebracht
- diese Blöcke nennt man "Buffer" (konkret: Mbuf)³
 - ▶ nicht zu verwechseln mit Ring Buffer!

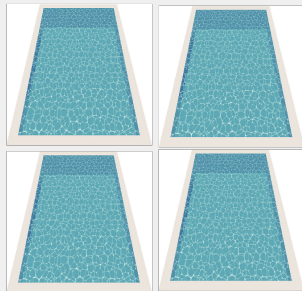
²http://doc.dpdk.org/guides/prog_guide/poll_mode_drv.html

³http://doc.dpdk.org/guides/prog_guide/mbuf_lib.html

MEMPOOL



memory pool allocation

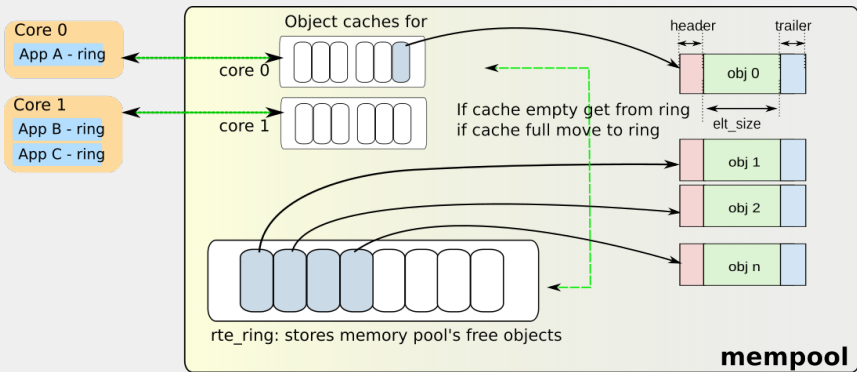


https://en.wikipedia.org/wiki/Memory_pool

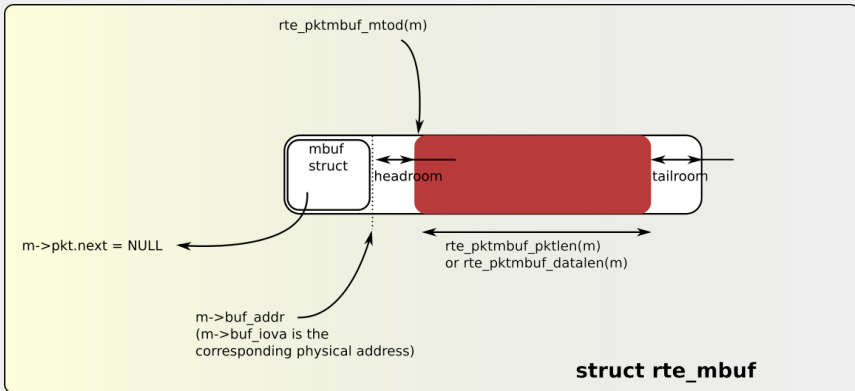
http://doc.dpdk.org/guides/prog_guide/mempool_lib.html

- Mempool **nutzt** intern **rte_ring** (Queue)
 - ▶ speichert dort Referenzen auf Speicherbereiche

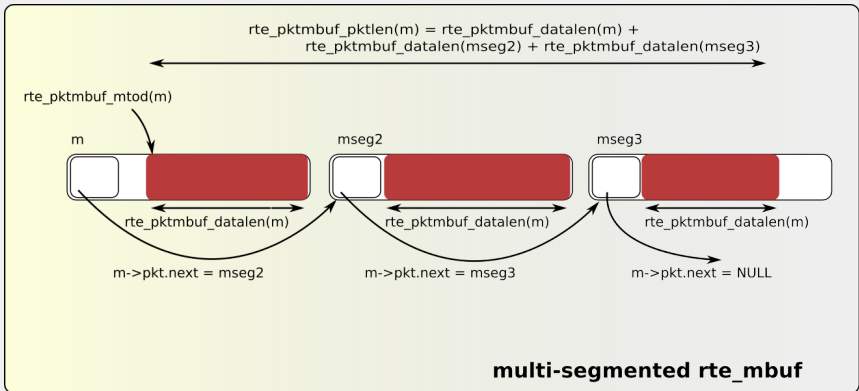
MEMPOOL: LOCAL CACHE



MBUF



MBUF: INDIRECT BUFFER



- Queue durch Bibliothek **rte_ring**⁴ implementiert
- aus FreeBSD Ringpuffer⁵ entwickelt
- sperrfreier Ringpuffer nach FIFO-Prinzip
- Zeiger mit vier Kategorien
 - ▶ prod_tail
 - ▶ prod_head
 - ▶ cons_tail
 - ▶ cons_head

⁴https://doc.dpdk.org/guides-16.07/prog_guide/ring_lib.html

⁵https://svnweb.freebsd.org/base/release/8.0.0/sys/sys/buf_ring.h

Vorteile

- Daten extrem schnell in Puffer schreiben
- viel seltener Cache-Misses

Nachteil

- feste unveränderliche Größe
- höherer Speicherverbrauch

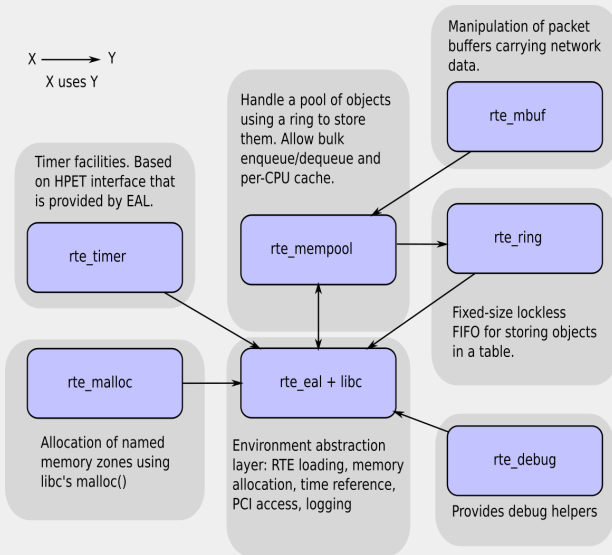
Programmierer sorgt dafür, dass:

1. alles richtig **konfiguriert** ist
2. **PMD** die Pakete holt
3. Speicher vom **Mempool** geholt wird
4. Mithilfe des Mempool ein **Mbuf** erzeugt wird
5. in den Pakete reingesteckt werden

Details stehen in der Dokumentation!

Es gibt viele verschiedene Wege, Beispielprogramme geben Orientierung.

ZUSAMMENHANG DER KOMponentEN



WAS SIND HUGEPAGES?

- Memory management unit (**MMU**):
 - ▶ Teil der CPU,
 - ▶ realisiert Zuordnung VA \mapsto PA
 - Translation lookaside buffer (**TLB**):
 - ▶ Speichert oft genutzte Einträge
 - ▶ schneller Zugriff auf Einträge
 - ▶ bei vielen Einträgen ineffizient
 - größere Pages \rightarrow weniger Einträge
- \rightarrow Inhalt ändert sich weniger oft
- \Rightarrow Zeiteffizienz erhöht!

EMPFANG UND SENDEN VON PAKETEN

1. runtime environment initialisieren
2. mempool anwenden
3. initialisiere NIC ports um tx/rx-Warteschlangen⁶ zu erhalten und diesen Speicher zuweisen
4. Definieren von mbuf und Anfordern von Speicher aus dem mempool

⁶DPDK - Dynamic Queue Test

1. holt Paket vom NIC-Port zu mbuf
2. analysiere mbuf und hole Paket
3. Paket in Anwendung verwenden z.B. Inhalt des Paket drucken

1. Paket in mbuf schreiben
2. bewege mbuf zu tx Warteschlange
3. Paket senden durch dpdk-api

- DPDK Linux Guide (Installation, Getting started)
- DPDK Programmer Guide (für ausführliche Erklärungen)
 - ▶ Ring Library
 - ▶ Mempool Library
 - ▶ Mbuf Library
 - ▶ Poll Mode Driver Library
 - ▶ ...
- DPDK API Documentation (zum Programmieren)

VIELEN DANK FÜRS ZUHÖREN

QUELLEN

DPDK.org

github.com/DPDK/dpdk

blog.selectel.com

net.in.tum.de

privateinternetaccess.com

slideshare.net

it-sobytie.ru

github.com/AltraMayor/gatekeeper

svnweb.freebsd.org

pcapplusplus.github.io

wikipedia.org/wiki/Memory_pool

ZEIT FÜR EURE FRAGEN