I/O-Technologien im Umbruch

OSL Aktuell

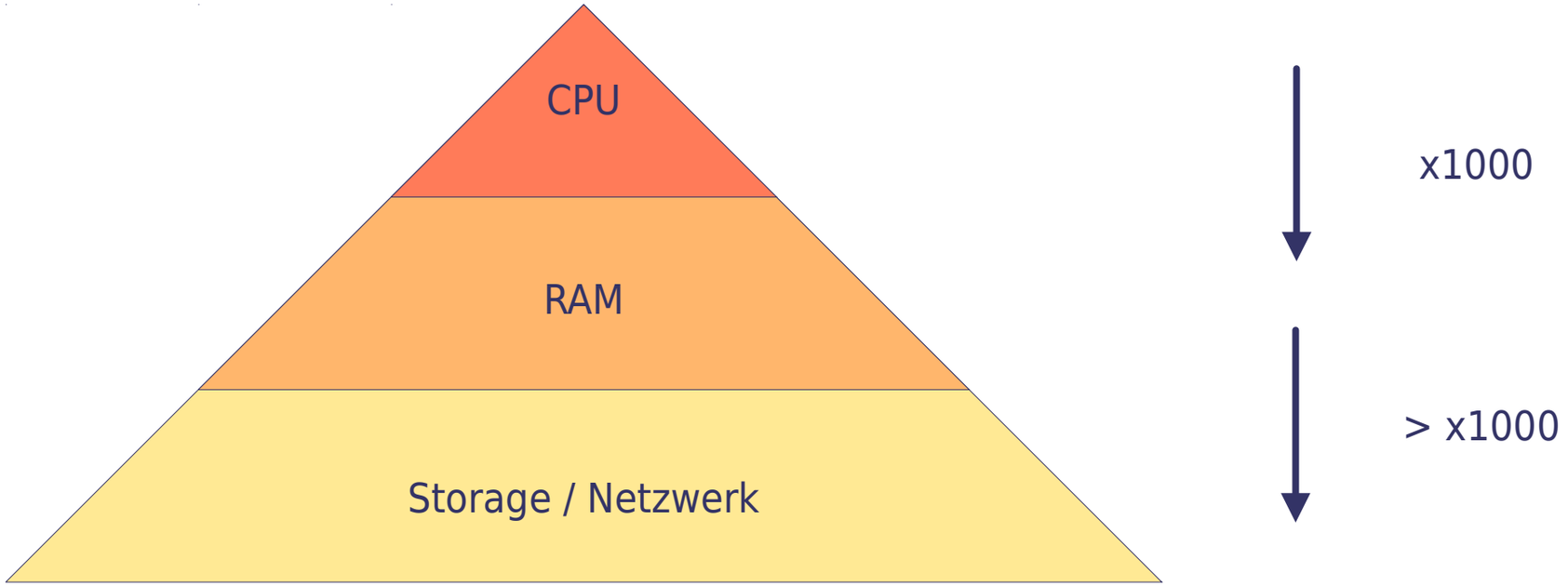
Schöneiche bei Berlin – 26. Mai 2021

till.miemietz@barkhauseninstitut.org

I/O-Geräte: Gestern noch zu langsam.
–
Heute schon zu schnell?

Geschwindigkeitsverhältnisse vor 2015

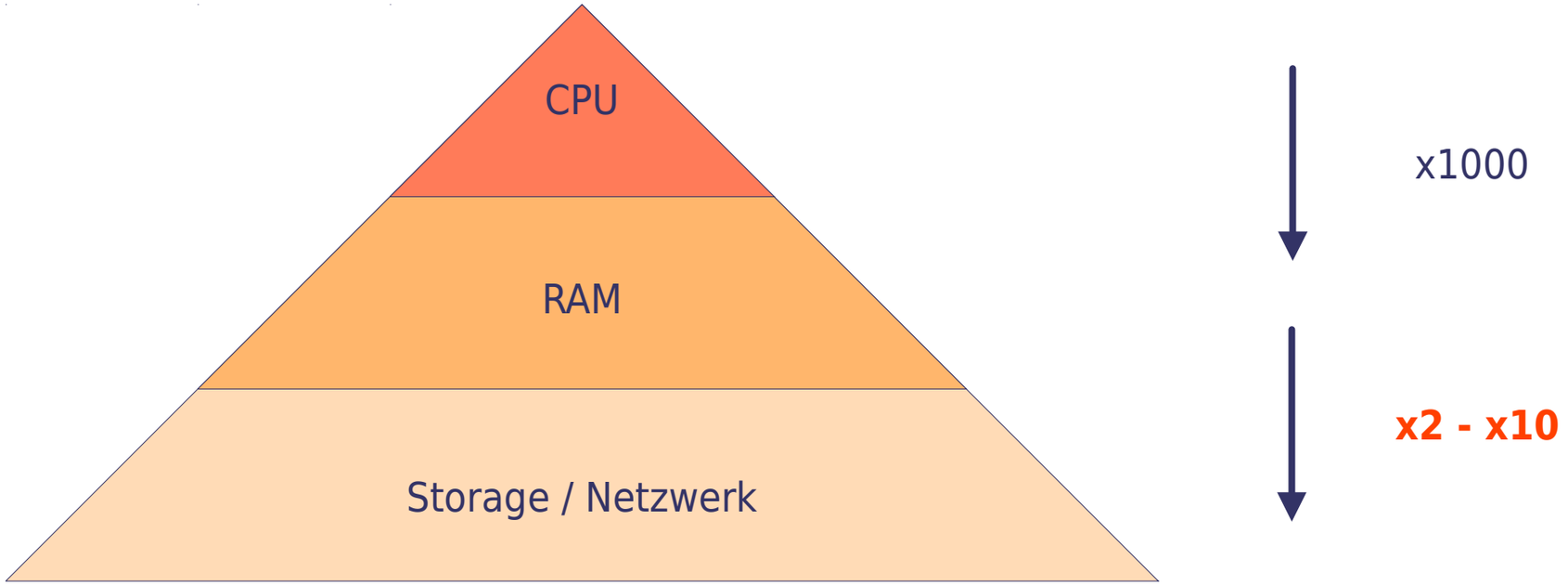
- Leistung von I/O-Geräten quasi unabhängig von CPU und Betriebssystem



- Wie verringert man I/O-induzierte Leerlaufzeiten?

Geschwindigkeitsverhältnisse im Jahr 2021

- I/O-Geräte sind nur nur noch eine Größenordnung langsamer als RAM!



- Wie können solche Geräte effizient ausgenutzt werden?

  → Und was bedeutet das für Betriebssysteme und Anwendungen?

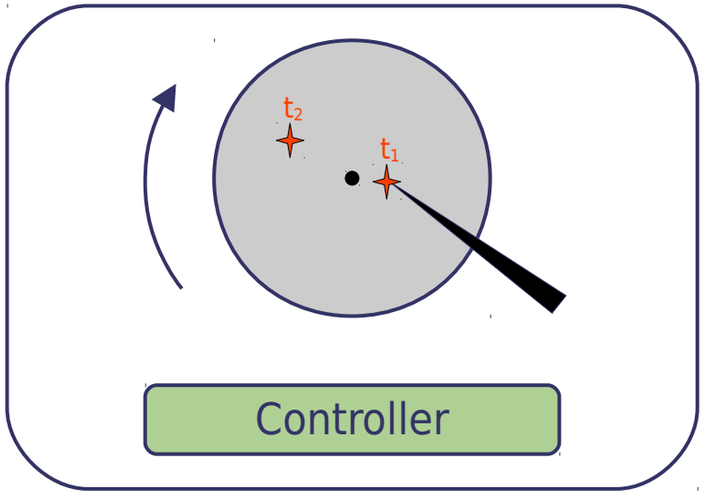
Über neuartige Speichergeräte

Flashbasierte Solid State Drives (SSDs)

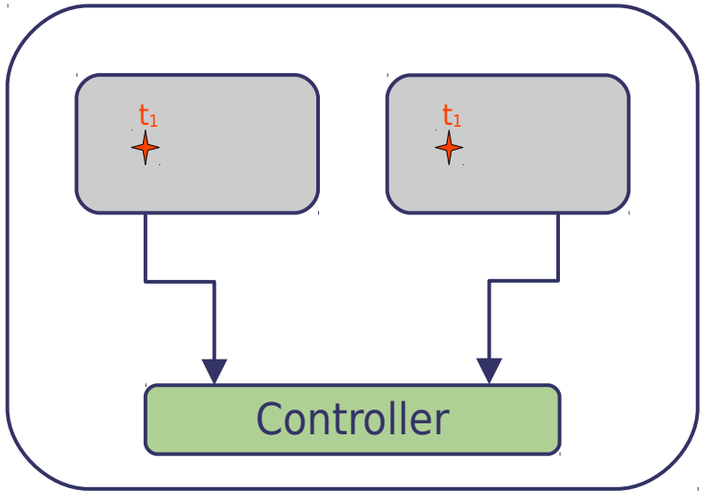
- Prinzipbedingte Überlegenheit gegenüber herkömmlichen Festplatten

  → Schnellerer Speichermechanismus (vollelektrisch vs. elektromechanisch)

  → Fähigkeit zur echt parallelen Requestbearbeitung



HDD



SSD

- NVMe: Weniger Overhead und Sichtbarmachen der Parallelität im Betriebssystem

Solid State Drives (SSDs) – Leistungsentwicklung

		Corsair Force LS[1] (S-ATA 3)	Samsung 970 evo[2] (NVMe & PCIe 3)	Samsung 980 pro[3] (NVMe & PCIe 4)
Jahr		2013	2018	2020
Kapazität		240 GB	250 GB (bis 2 TB)	500 GB (bis 2 TB)
Latenz [µs]	Lesen (wahlfrei)	170	66	45
	Schreiben (wahlfrei)	47	20	17
Durchsatz [MB/s]	Lesen (sequentiell)	510	3400	6900
	Schreiben (sequentiell)	382	1500 (2500)	5000

[1] https://www.storagereview.com/review/corsair-force-series-ls-ssd-review

[2] https://www.samsung.com/semiconductor/global.semi.static/Samsung_NVMe_SSD_970_EVO_Data_Sheet_Rev.1.0.pdf

[3] https://s3.ap-northeast-2.amazonaws.com/global.semi.static/Samsung_NVMe_SSD_980_PRO_Data_Sheet_Rev.1.2.pdf

Intel Optane (a.k.a. 3D XPoint)

- Nochmals verbesserter Speichermechanismus gegenüber Flash

  → Phase-Change Memory

  → Bessere Packungsdichte

- Beispiel: Intel DC P5800X[1]:

  → Latenz für Lesen und Schreiben bei 10 µs

  → Durchsatz lesend: 7200 MB/s

  → Durchsatz schreibend: 6100 MB/s

- Optane SSDs sind schon zu weit von CPU entfernt!

[1] https://ark.intel.com/content/www/de/de/ark/products/201860/intel-optane-ssd-dc-p5800x-series-800gb-2-5in-pcie-x4-3d-xpoint.html

Intel Optane als Non-volatile Memory (NVM)

- Ziel: Persistenter Speicher mit der Leistungsfähigkeit von DRAM

- Optane existiert auch im DIMM-Format

  → Relativ günstige Erweiterung des RAMs (Memory Mode)

  → Echter NVM (App-Direct Mode)

  → Interleaved Mode (ähnlich striped volumes)

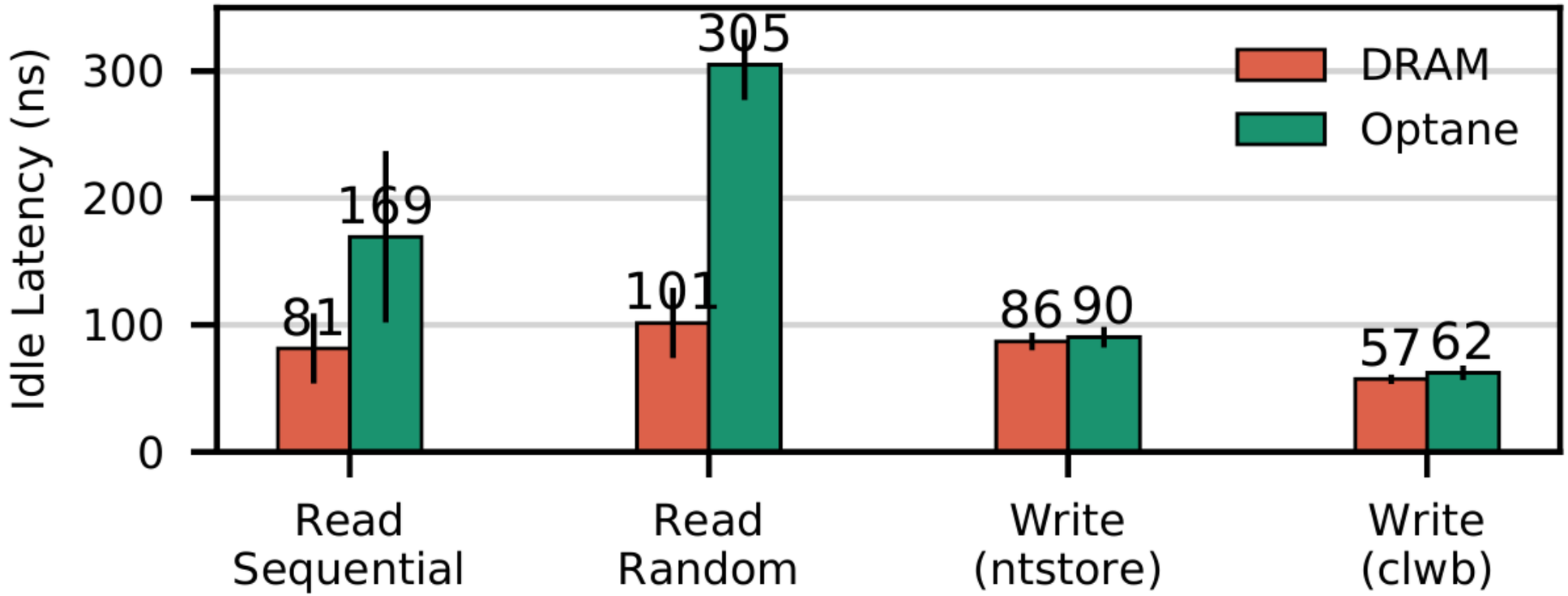
- Im Gegensatz zur SSD ist ein angepasster Chipsatz nötig

- Byte-Adressierbar, aber interne Blockgröße von 256B (Cache Line auf x86: 64B)

Intel Optane – DIMM Performance (Yang et al. FAST 2020)

- Latenz in gleicher Größenordnung wie DRAM

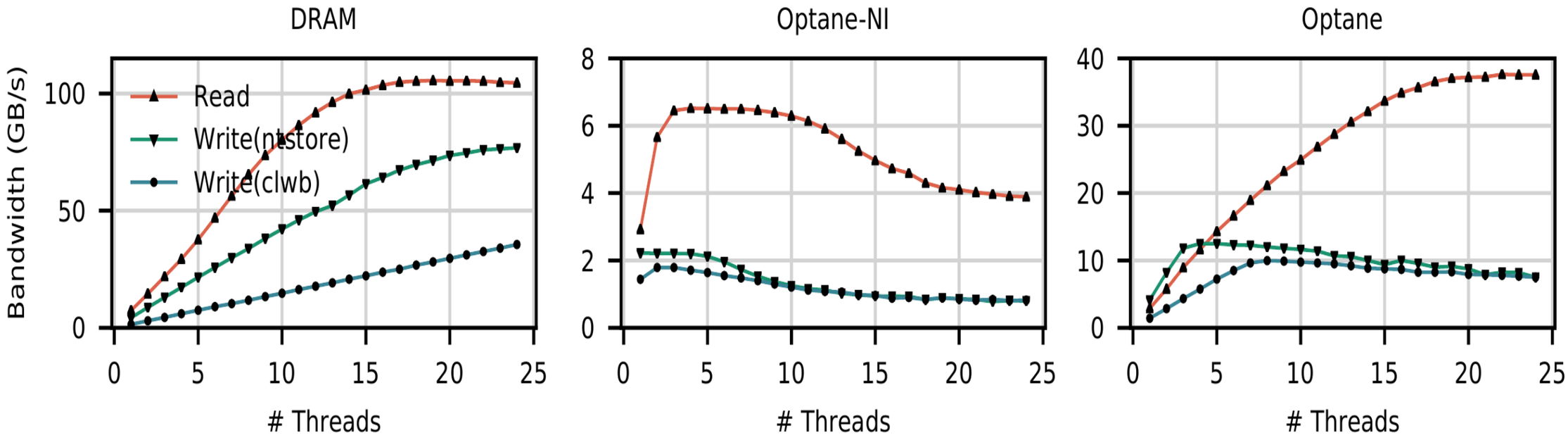
  → Ausreißer aber noch im Mikrosekundenbereich



Intel Optane – DIMM Performance (Yang et al. FAST 2020)

- Durchsatz fällt jedoch gegenüber DRAM deutlich ab

  → Ohne Interleaving ähnliche Bandbreite wie SSDs



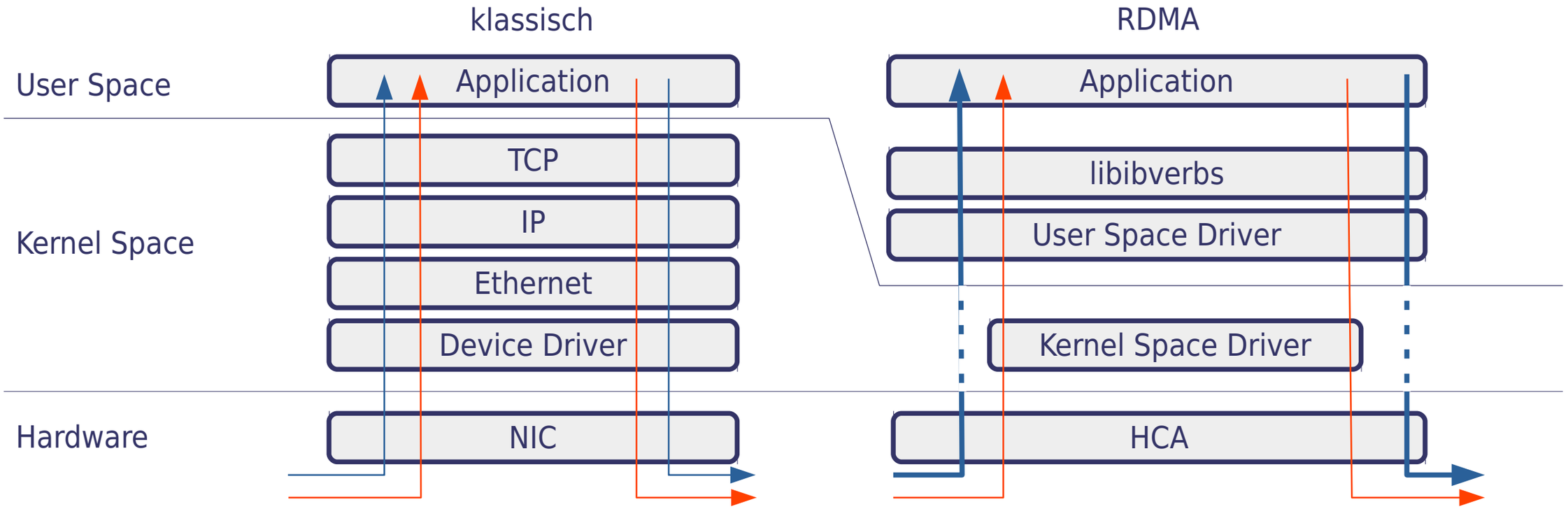
Was tut sich im Netzwerk?

Was ist der Stand der Technik?

- Datenraten von Netzwerkadaptern steigen kontinuierlich
  - → Heutiges Maximum: NVIDIA Connect X-7: bis zu 400 Gbit/s (pro Karte)
  - → Eher realistisch: EDR (100 GBit/s)
  - → Weniger als 800ns Adapterlatenz
- Problem: Traditionelle Protokolle zu ineffizient
  - → Betriebssystem muss jedes Paket behandeln (sehr hohe CPU-Nutzung!)
  - → Überholte Annahmen bzgl. Paketverlusten
  - → Komplexe Softwareschichten

Remote Direct Memory Access (RDMA)

- Idee: Netzwerkprotokolle in Hardware implementieren



- Vertreter: RDMA over Converged Ethernet (RoCE) und InfiniBand

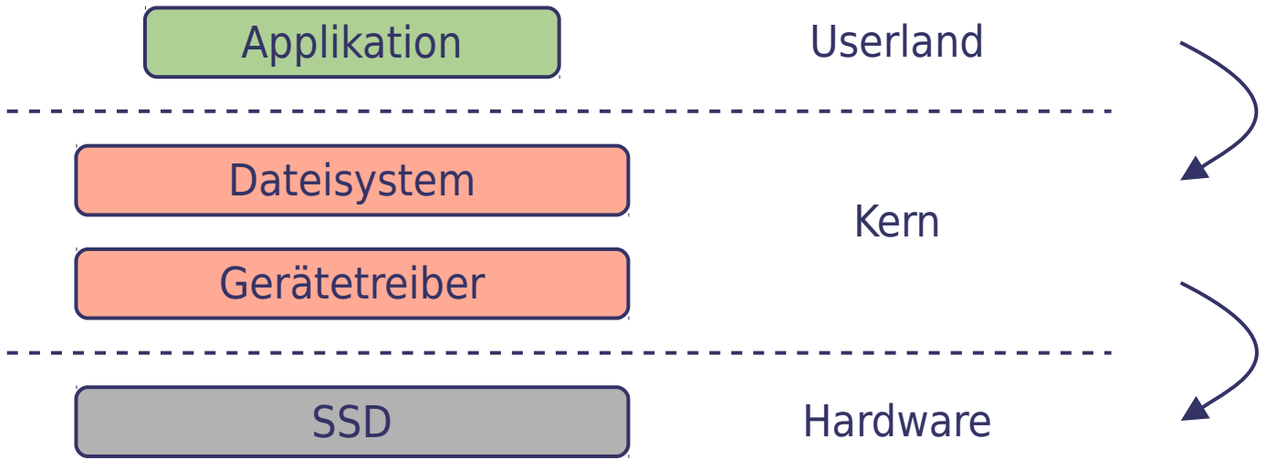
Warum wird das Betriebssystem zum Problem?

Klassisches Betriebssystemkonzept (Monolith)

- Teilung in Userland und Kern (privilegierter Modus)

  → Kern verwaltet Geräte

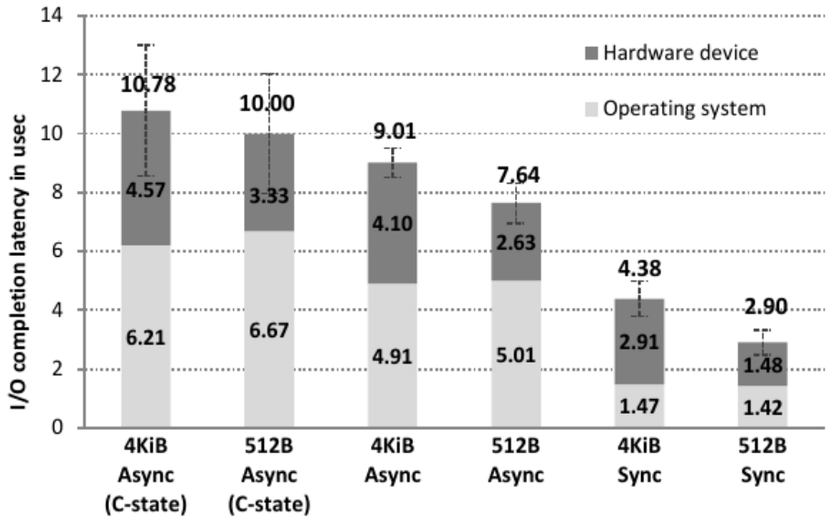
  → Kern setzt Isolation zwischen Nutzern durch



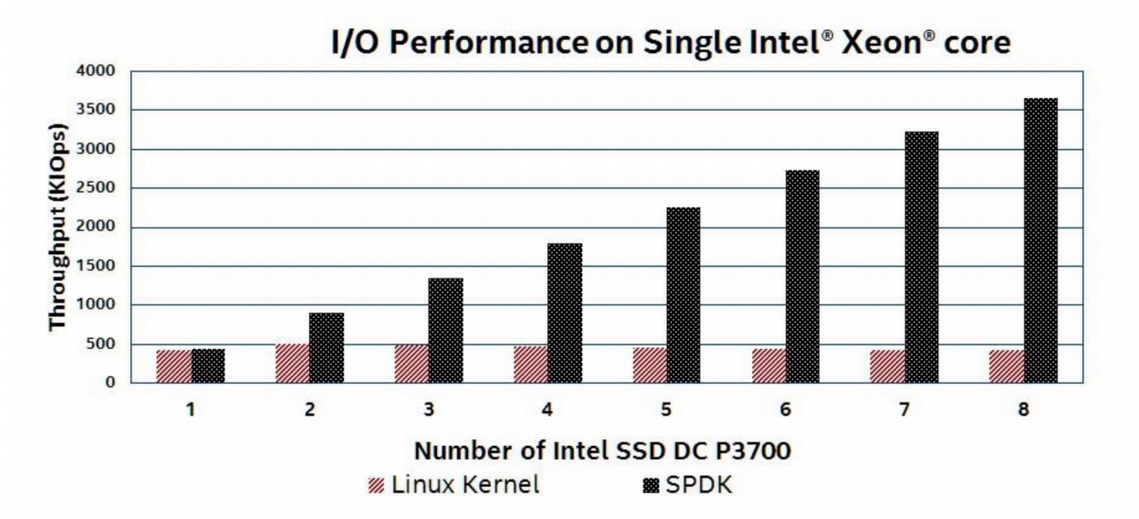
- Kontextwechsel ~ wenige µs, Kopieren von 1 MiB im RAM ~ 1 µs

Was bedeutet das heute?

- Kontextwechsel ~ wenige µs, Kopieren von 1 MiB im RAM ~ 1 µs



Yang et al.: When Poll is Better than Interrupt, FAST 2012



Yang et al.: SPDK: A development kit to build high performance storage applications, 2017

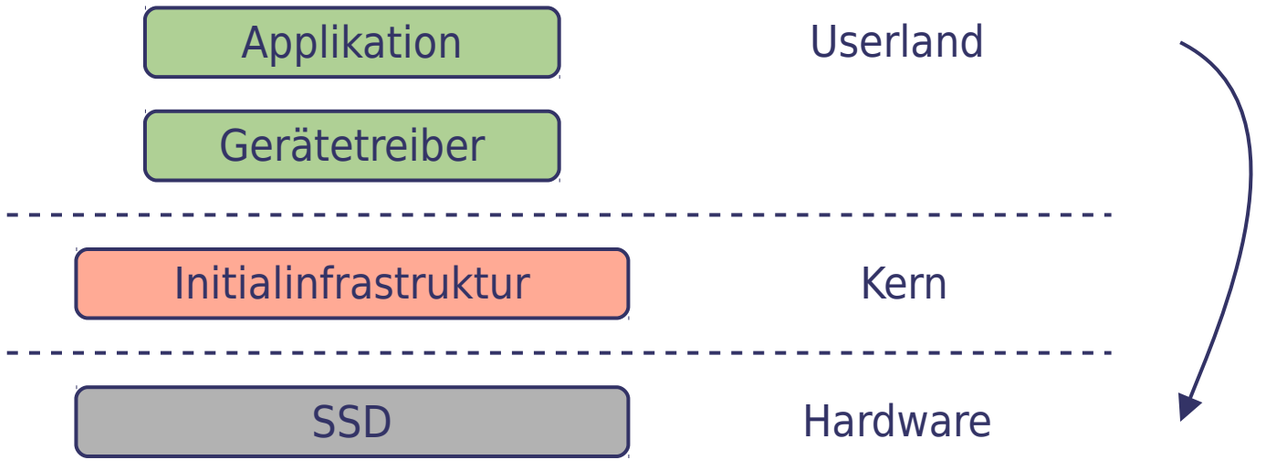
- Betriebssystem ist mit sich selbst beschäftigt und behindert die Geräte!

Userspacetreiber

- Verlagerung von Teilen des Kerns in die Applikation (als Bibliothek)

  → Sparen von Metaoperationen und dadurch Leistungssteigerung

  → Applikation wird Gerät direkt zugewiesen, dadurch kein OS-seitiges Caching



- Beispiele: RDMA, DPDK (Netzwerk); SPDK, NVMeDirect (Speichergeräte)

Was bedeuten High-Performance-Geräte für die Anwendungen?

Programmieren mit modernen Geräten

- Sämtlicher Komfort des Betriebssystemkerns geht mit Userspacetreibern verloren
  - → Geräte werden direkt durchgereicht, dadurch wird Sharing kompliziert
  - → Anwendungen müssen Dateisysteme etc. selbst implementieren (gut für Datenbanken)
- Neue CPU-Aufteilung: Ein Core pro Applikationsthread
  - → Vermeiden von Kontextwechseln
  - → Teils nötig durch Fehlen eines blockierenden I/O-Mechanismus
- CPU wird durch High-Performance-I/O-Stacks stark belastet!

Faustregeln für I/O-lastige Anwendungen

- Indirektionen vermeiden!
- Asynchrone Programmiermodelle nutzen!
- I/O-Operationen reduzieren!

Faustregeln für I/O-lastige Anwendungen

- Indirektionen vermeiden!
- Asynchrone Programmiermodelle nutzen!
- I/O-Operationen reduzieren!**?**

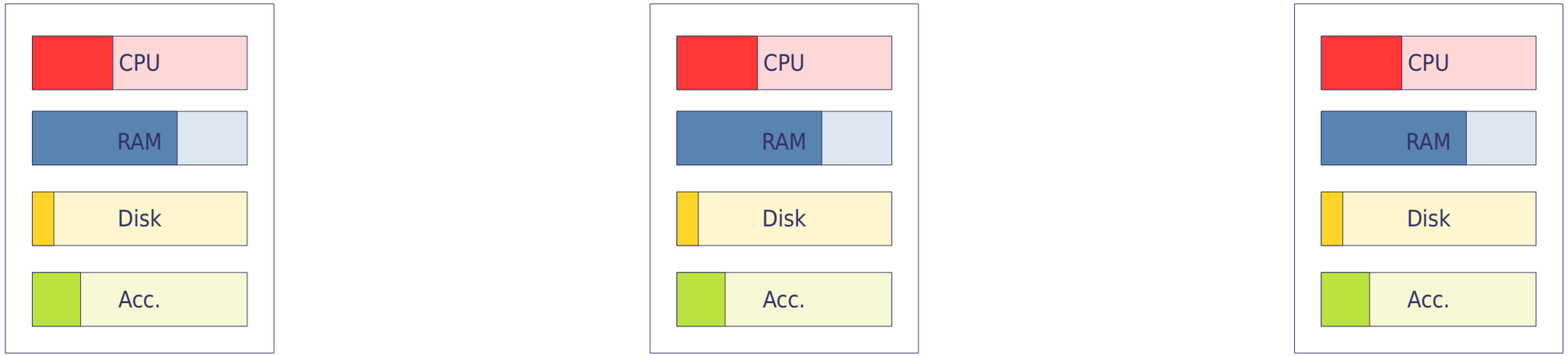
Neuartige RZ-Architekturen

Problemfall Auslastung

- Siloarchitekturen: Häufig Überprovisionierung von Ressourcen

  → Jeder Knoten muss für Lastspitzen ausgelegt sein

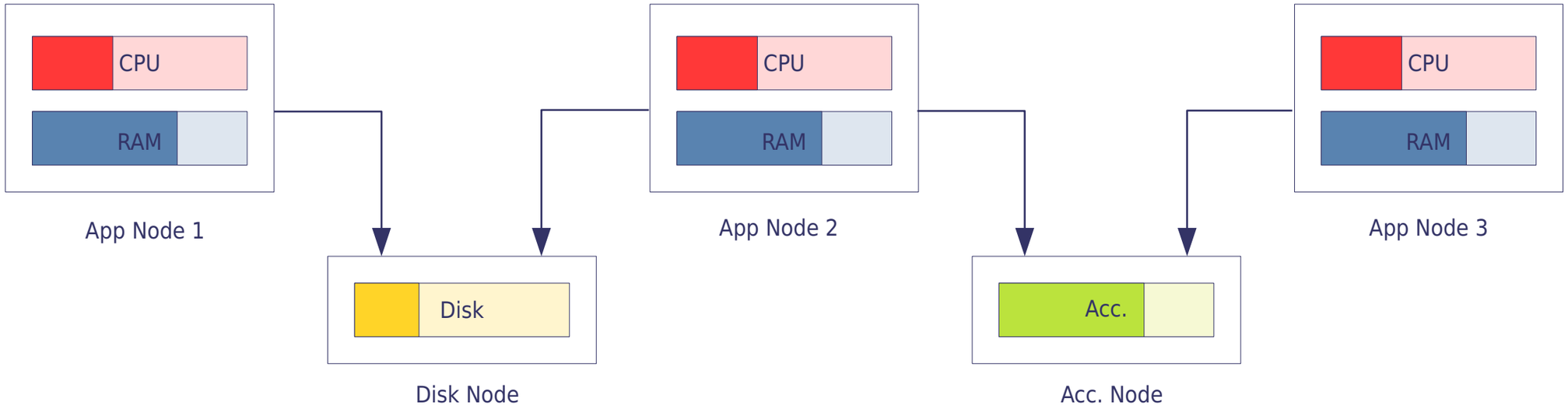
  → Aus ökonomischer Sicht nachteilig



Problemfall Auslastung

- Disaggregated Architectures: Reine Ressourcen-Nodes

  → Bessere Hardwarenutzung und unabhängige Skalierung einzelner Geräteklassen



- Mit heutiger Hardware nur minimaler Netzwerkoverhead

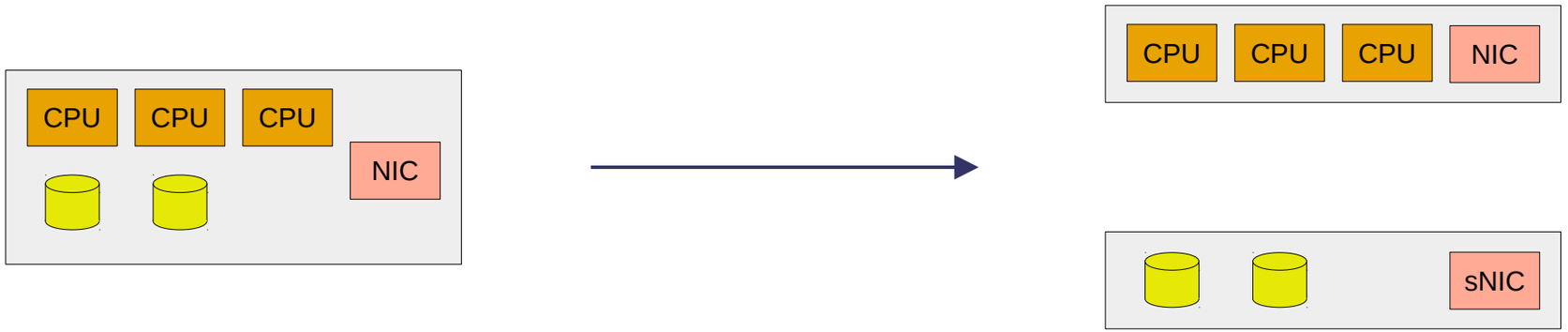
SmartNICs für das Disaggregated Data Center

- Muss jeder Ressourcenknoten ein vollwertiger Rechner sein?

- SmartNICs ist RDMA-NIC mit integriertem ARM-Subsystem

  → Kann sämtliche Kommunikation steuern und überwachen

  → Lässt sich vom restlichen Host abschotten

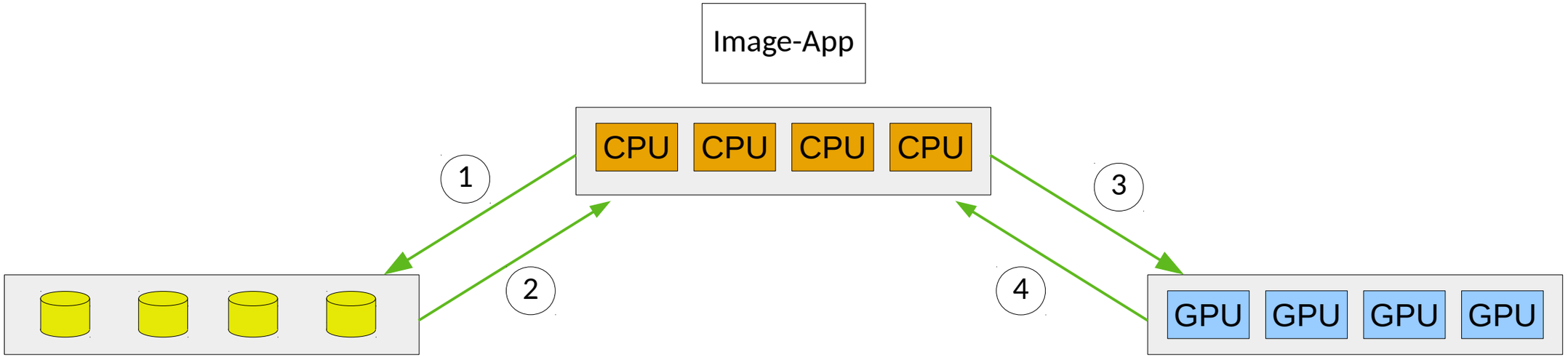


FractOS: Ein Framework für Disaggregated Data Center

- Beispiel Bildverarbeitung

  → Zergliederung eines Auftrages in Teilschritte

  → CPU-zentriertes Modell nicht optimal bzgl. Datenfluss

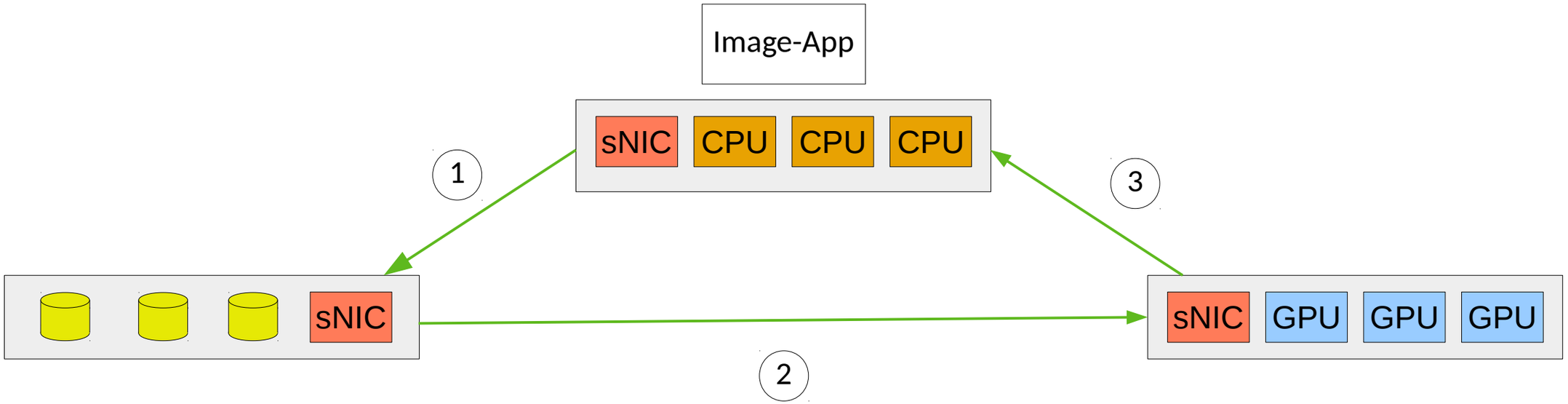


FractOS: Ein Framework für Disaggregated Data Center

- Auftrag wird von Anwendung gepackt und fließt „selbst“ durch das RZ

  → Einsparen von Nachrichten (Durchsatzsteigerung)

  → Bessere Ausnutzung der vorhandenen Ressourcen



Fazit

Was ändert sich mit der neuen Hardwaregeneration?

- Neue I/O-Technologien bringen nochmals deutliche Leistungssteigerung

  → Persistenter Speicher nähert sich RAM an

- Design von Betriebssystemen und Anwendungen verändert sich stark

  → Klassisches Konzept wird zum Flaschenhals

  → Wiederbelebung alter Programmiermodelle

- Fortschritt in Hardware ermöglicht neue RZ-Architekturen

  → Leistungseinbußen durch Netzwerk werden geringer

  → Anpassung des Rechenzentrums an die Applikation (anstatt andersherum)

virtualization and clustering – made simple