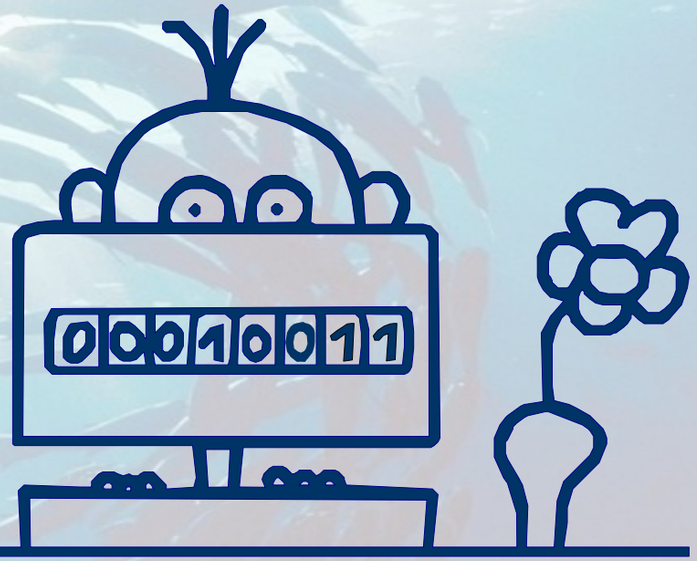
I/O-Technologien im Umbruch
Das kommerzielle RZ im Eigenbetrieb



OSL Aktuell

Schöneiche / Berlin • 26. Mai 2021

bert.miemietz@osl.eu

Herausforderungen und Möglichkeiten für das kommerzielle RZ im Eigenbetrieb

Nochmal ein paar Fakten zur Situationsbeschreibung

1. Die ZVE: Kaum schneller aber größer

- Taktung konvergiert gegen 5 GHz (theoretische Prognose vor 10 Jahren waren 10GHz)

  Intel: 1990: **20 MHz** 1995: **200 MHz** 2000: **1 GHz** 2005: **3,7 GHz** 2010 – **3,6 GHz** 2018 – **4,3 GHz***
  Mainframe: 2018: **5,2 GHz** (z14, 500 Watt je CPU)

- Multithreading, Pipelining, spekulative Ausführung → gigantische Zunahme der Komplexität

- Leistungsgewinn heute hauptsächlich über Parallelisierung und Software (Komplexität!)

- Wichtige Mechanismen für die Parallelisierung / Synchronisation kaum schneller geworden
  - Beispiel Threadwechsel per CV-Sync ca. 5µs
  - Programmiertechniken ändern sich (derzeit: Compiler-Optimierungen, Deferred Store, Lock-Free und Atomic Variables)

- Signalgeschwindigkeit Kupfer (universell) und Glasfaser ca. 0,67 * c → 200.000 km/s

Modulo-Operation	ca. 1ns	→	0,2 m
Mutex-Lock	ca. 10ns	→	2,0 m
CV-Sync	ca. 5µs	→	1.000,0 m

Nochmal ein paar Fakten zur Situationsbeschreibung

2. Arbeitsspeicher

- Speicherbandbreiten erfreulich hoch – allerdings nur im Multithread-Modus

  Linux ST: 4,2 GiB/s
  Linux MT: 20,0 GiB/s Xeon 6C/12T-x86-System

  Höhere Durchsätze nur mit größeren Systemen
  AMD hat mit speziellen ZVE-Sätzen die Nase vorn (bis 140GiB/s)

- Single-Thread-Performance genügt oft nicht mehr den Wünschen / Anforderungen
- Multithread-Konzepte implizieren auf größeren Systemen auch größere Latenzen
- Ausschöpfung Leistungspotential nur mit sehr speziellen Programmiertechniken
- Bedeutung Virtualisierungs-Layer für die RZ-Praxis nimmt zu

Nochmal ein paar Fakten zur Situationsbeschreibung

3. Massenspeicher und Netzwerke

- SSD erreichen heute Service-Zeiten um 10 µs, NVDIMM ca. 2 µs

  → Sie liegen damit in der Größenordnung teurer CPU-/RAM-Operationen

- HPN (InfiniBand, HPE) erreichen heute typische Latenzen um 1-2 µs, z. T. unter 1 µs

  → 1000GE ist heute vorstellbar
  → Latenzen im Netzwerkstack erscheinen nicht mehr so stark reduzierbar (100ns-Schallmauer => 20m Kupfer)

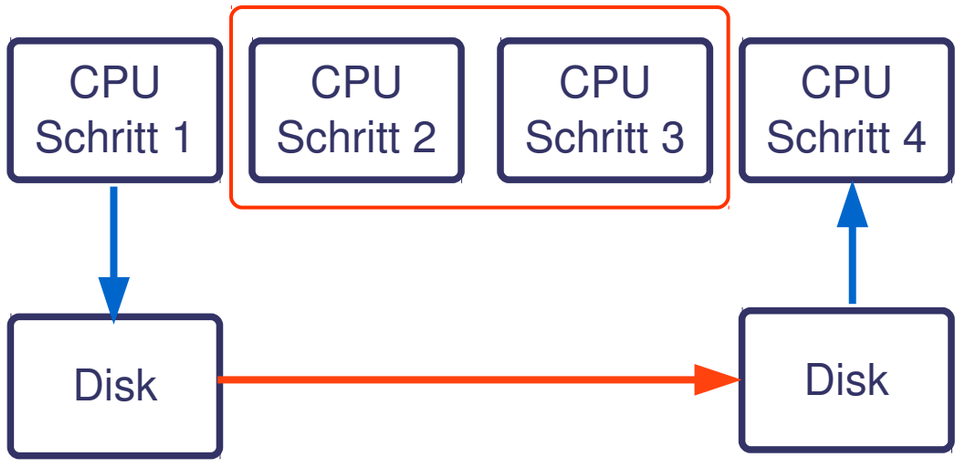
- Massenspeicher und Netzwerke werden gleichermaßen schneller

  → muss in der Programmierung heute ggf. berücksichtigt werden
  → zunehmende Bedeutung AIO
  → Anwendungsdesign wird wesentlich komplexer
  → Konventioneller Anwendungsstack + moderne Hardware = suboptimale Performance

Ein Punkt näher beleuchtet: AIO

Warum Anwendungen und z. T. auch Treiber von Grund auf neu zu entwerfen sind

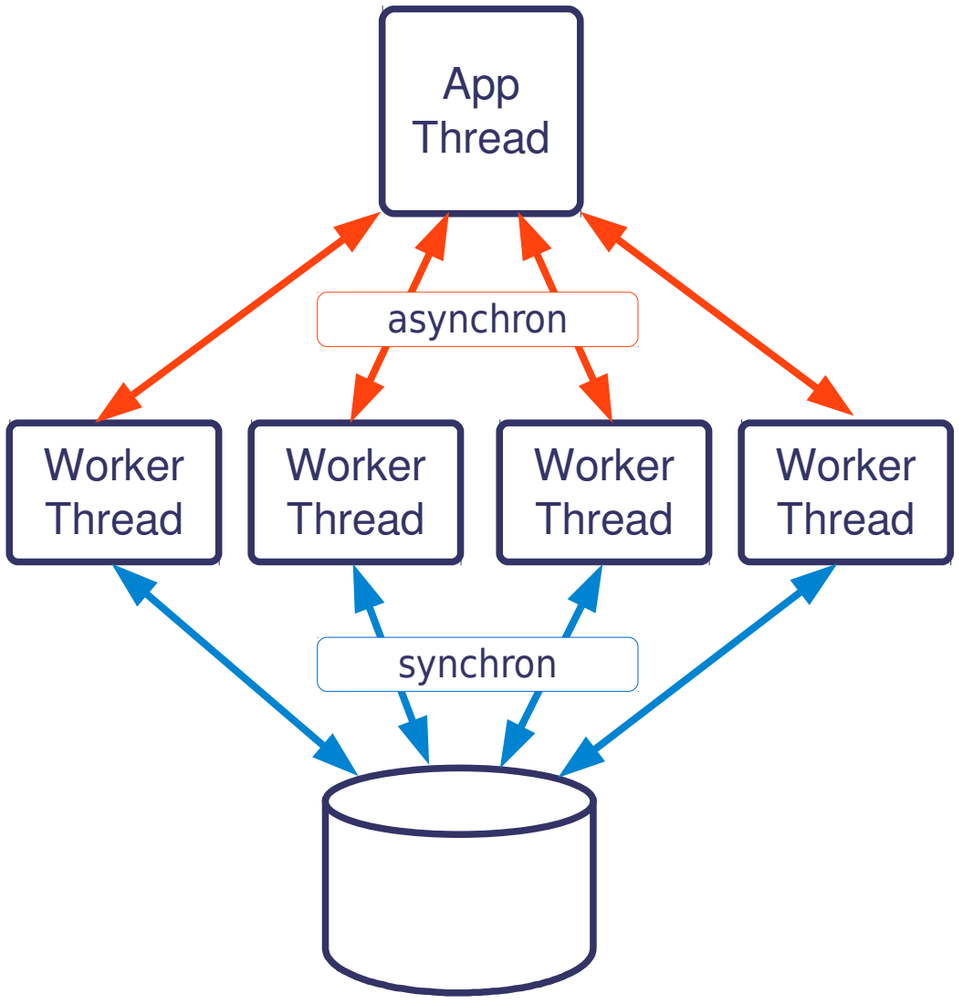
- Motivation früher: CPU Nutzung
  - → Statt nur auf I/O zu warten, könnte meine Applikation inzwischen weitere Aufgaben wahrnehmen
  - → Threadwechsel ermöglichen Multitasking und erlauben so verbesserte CPU-Auslastung
  - → Multicore-/Multithread-Design erschloß mit massiver Parallelisierung neue Möglichkeiten zur Steigerung des Gesamtdurchsatzes



- Dauer eines Threadwechsels ca. 5 µs
- Motivation heute: Threadwechsel vermeiden
  - → 100.000 – 200.000 Threadwechsel je Sekunde und Core markieren das Ende der Fahnenstange
- Netzprogrammiertechnologien erlauben mehr Wechsel des Ausführungskontextes innerhalb eines Threads

Ein Punkt näher beleuchtet: AIO

Früher das Allheilmittel: Synchrone Schnittstellen aber Asynchronität durch Multithreading

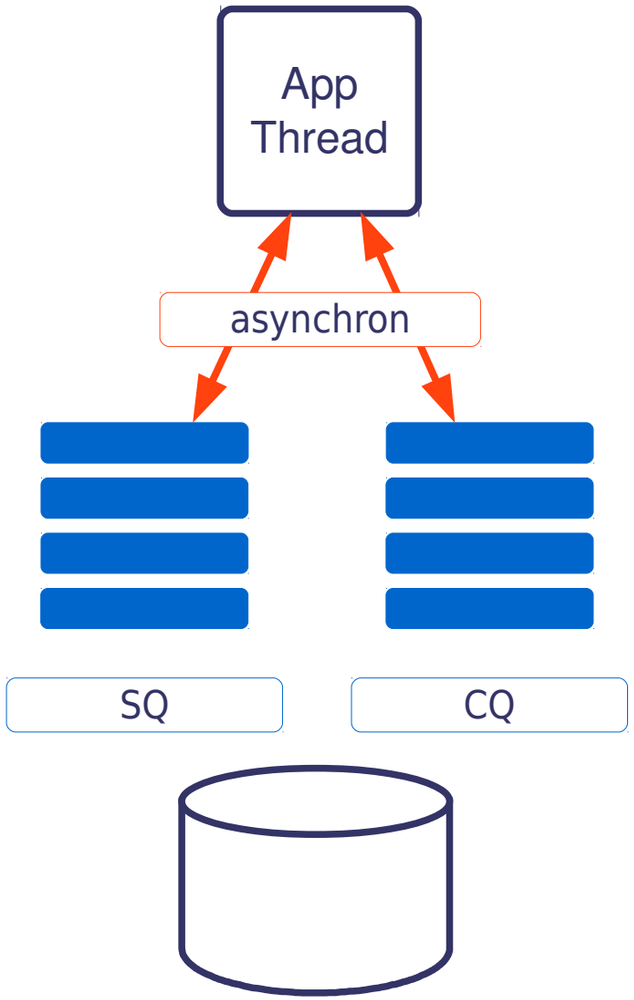


- Block-Layer arbeitet typisch synchron
- Lösung für Parallelität durch Threads
- Threadwechsel 5 µs
  Service-Zeit NVMe 10 µs
  Service-Zeit HDD 2000 µs
  Heute (2021) 1.000.000 IOPS je Gerät möglich
- Jeder laufende Thread braucht einen Core
- Geräte brauchen hohe Parallelität
- => bei 25.000 - 100.000 IOPS pro Thread “ist Schluß”
- Höhere Durchsätze bedeuten mehr Threadwechsel
  d.h. brauchen mehr CPUs (nicht lineare Skalierung)

Ein Punkt näher beleuchtet: AIO

Neu: AIO mit speziellen Schnittstellen

- Einlieferung der Request über spezielle Schnittstellen
- Abschlußverarbeitung über spezielle Schnittstellen
- Wenige Threads genügen
- Auch hier Problem der Signalisierung vs. “busy wait”
- Steigerung der Durchsatzmöglichkeit
  bis ca. 1.000.000 IOPS per Core
- Aber: I/O muß verarbeitet werden
- Komplexe Programmierung mit diversen
  impliziten Problemen



Schlußfolgerungen globaler Natur

Herausforderungen, den man sich stellen muß

- Minimale Latenzen nur erreichbar mit **direkter** Hardware und speziellen Programmiertechniken => Domäne der GAFAs mit speziellen Stacks

- Grund für Virtualisierung heute kaum mehr Performance, sondern überwiegend Flexibilität, einfache Anwendung und Hardwareabstraktion
  → Bedeutungsgewinn speziell der I/O-Virtualisierung

- (I/O)-Virtualisierung muß neu entworfen werden:
  - Portabilität ↔ Threadwechsel und Memory-Management
  - Vermeidung häufig frequentierter Sperren, Ablösung durch verteilte, granulare Konzepte und parallelisiertes Design
  - Isolation von „Zonen“ in großen Systemen aus Verfügbarkeits- und Performancegründen
  - spezialisiertes Memory-Management

- RZ-Architekturen können (und müssen!) neu entworfen werden, Unified Networking sinnvoll

- Neue Server-Generationen ermöglichen anderes Herangehen

Schlußfolgerungen für das eigene RZ

Ergebnisse umfangreicher Recherchen und Tests

- Massenspeicher nicht mehr entfernt
  SCSI ist 35 Jahre alt, FC 25 Jahre → zusammen ergibt das 60 Jahre! Klingt das modern?
  Implementierungen noch immer proprietär und bisweilen kräftezehrend
  Selbst moderne Netzwerke schwächen das Potential heutiger Speichertechnologien erheblich ab
- Abschied vom zentralen RAID-System
- SSD am besten via NVMe lokal nutzen
- Moderne SSD nicht in einen RAID-Verbund zwingen
- Entfernte Spiegel nur über Low-Latency-Netzwerke
- Keine Angst vor Consumer-Komponenten!

Sie haben nicht die Möglichkeiten der GAFAs → Ergeben deren Konzepte für Sie Sinn? Also:

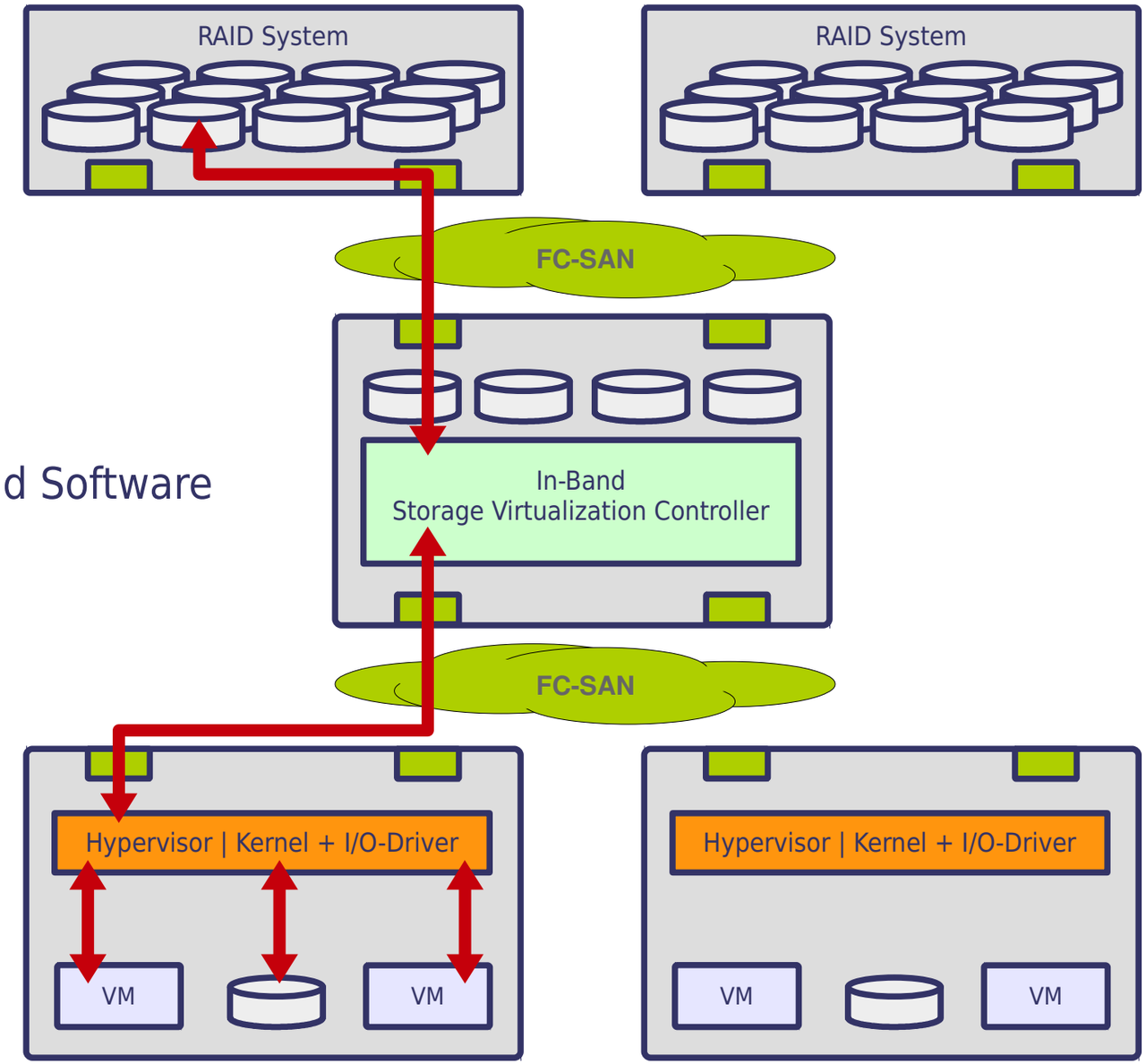
- Neue Technologien (auch Software) beobachten → den eigenen Weg gehen

Grundzüge der neuen Produktgeneration

Das klassische Modell

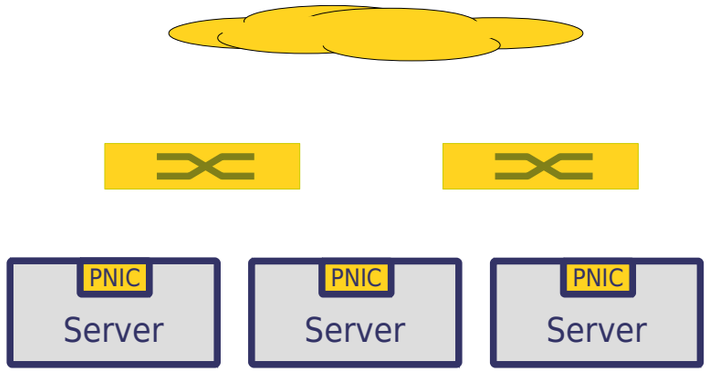
Immer noch brauchbar aber ...

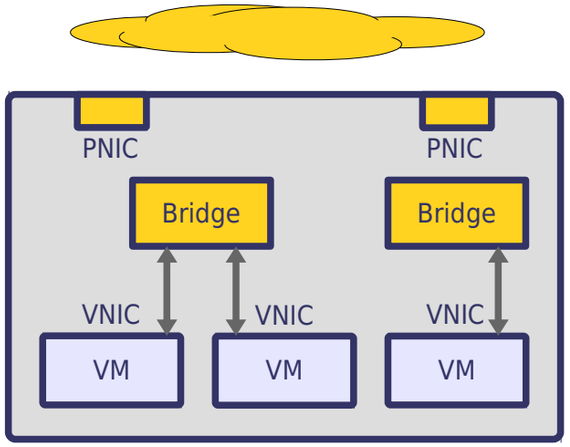
- lange Wege, viele Komponenten
- zentrale Ausfallpunkte
- Flaschenhälse erkennbar?
- vieles paßt nicht zu moderner Hard- und Software
- heterogenes Management
- ...

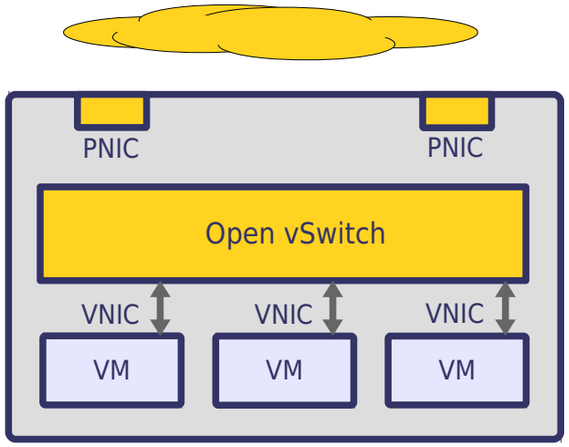


Analogieschluß

Netz und Block-I/O haben einiges gemein ...

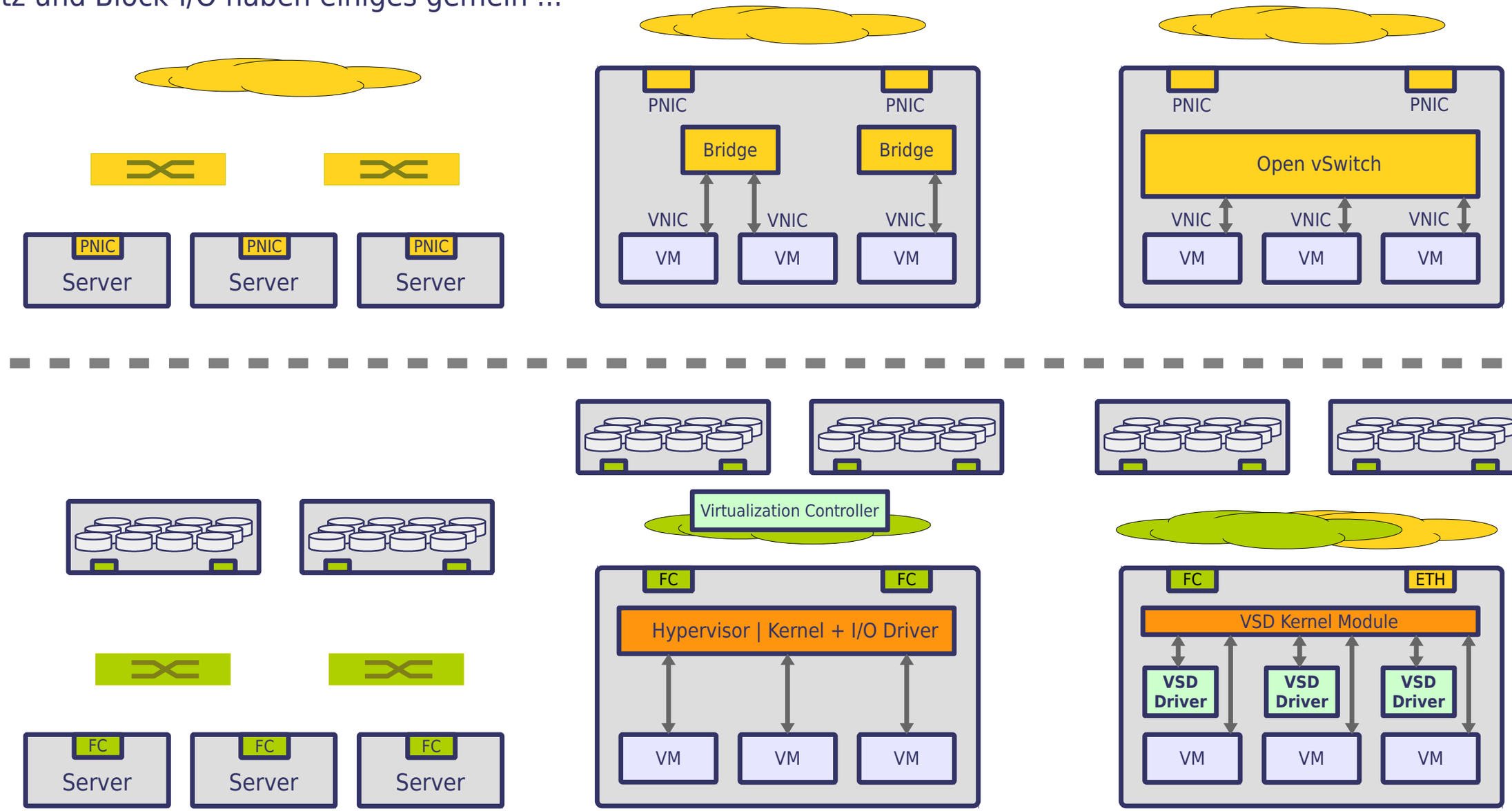






Analogieschluß

Netz und Block-I/O haben einiges gemein ...



Der Weg mit OSL Virtual Storage Domains

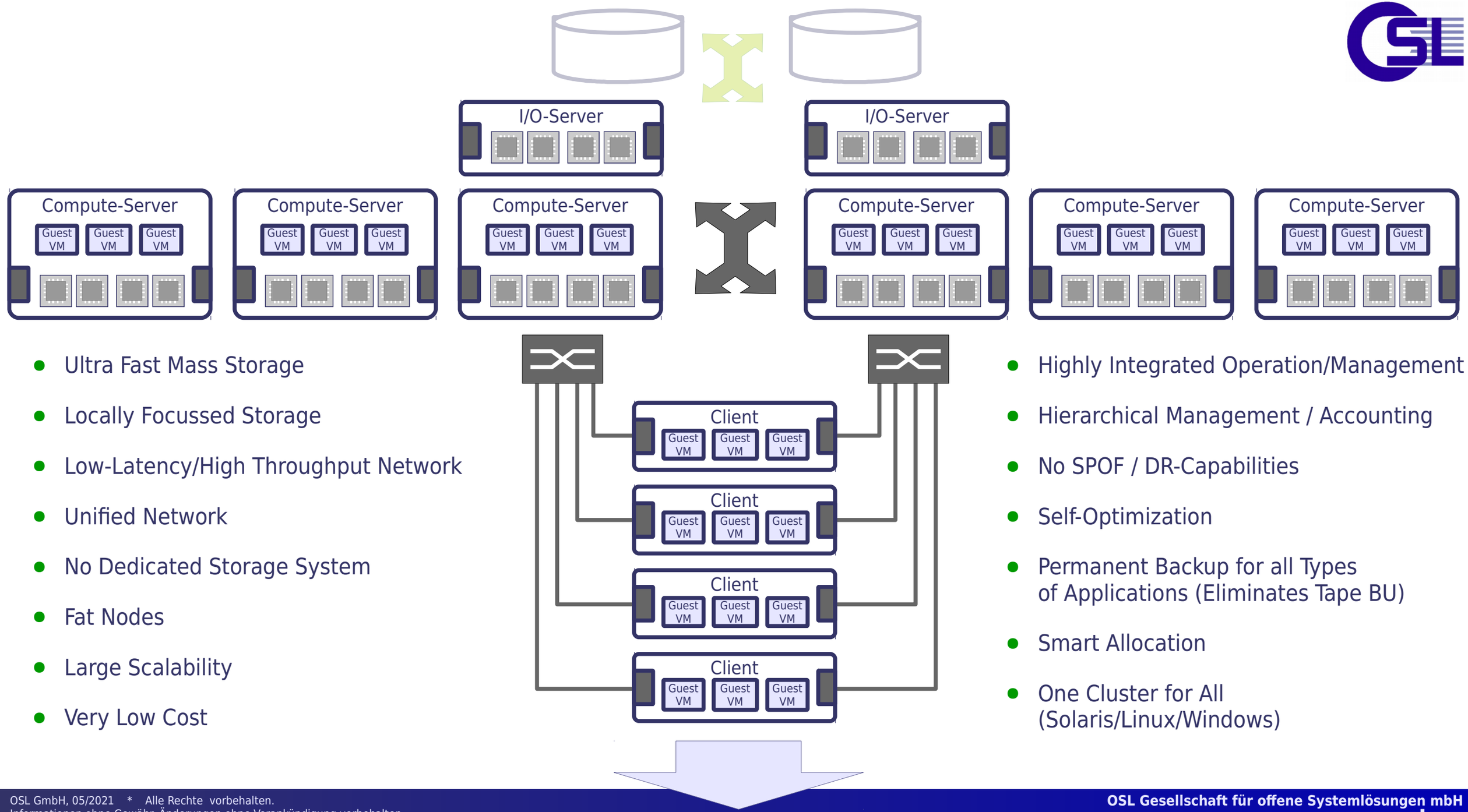
Speichervirtualisierung als Applikation

- Wir holen die Virtualisierung auf dasselbe System, auf dem auch die Applikation / VM läuft
  - es ist nur Software → geringer moralischer Verschleiß, geschütztes Investment
  - Einsparungen in der Infrastruktur / Verkabelung
  - sehr kurze Latenzen (lokale Kommunikation)
  - optimale Steuerungsmöglichkeiten und Homogenität
  - Nutzung leistungsfähiger Serverhardware (CPU, RAM, wio timeslice)

- Die Virtualisierung läuft im User-Space
  - es ist eine Software-Appliance
  - so gut wie möglich isoliert vom OS
  - portabel

- Ermöglicht clevere Verbindung von lokalem und entferntem Speicher
  - lokaler Speicher für kurze Latenzen unverzichtbar
  - modulare Einbeziehung von Netzwerktechnologien möglich
  - offen für HPC-Konstrukte und neue Schnittstellen und Programmiermodelle

- Großes Entwicklungspotential
  - extreme Flexibilität und Skalierbarkeit
  - langfristig angelegte, aus- und umbaufähige Architektur



- Ultra Fast Mass Storage
- Locally Focussed Storage
- Low-Latency/High Throughput Network
- Unified Network
- No Dedicated Storage System
- Fat Nodes
- Large Scalability
- Very Low Cost

- Highly Integrated Operation/Management
- Hierarchical Management / Accounting
- No SPOF / DR-Capabilities
- Self-Optimization
- Permanent Backup for all Types of Applications (Eliminates Tape BU)
- Smart Allocation
- One Cluster for All (Solaris/Linux/Windows)

virtualization and clustering – made simple