Fachhochschule Nordwestschweiz
Hochschule für Technik

Application Performance Management

FS 2021

Clustering

Michael Faes

Inhalt

Rückblick: Load Balancing

Clustering

- Eigenschaften & Komponenten
- Anwendungs-Anforderungen

Virtualisierung

- Prinzip
- Arten von Hypervisoren
- Bedeutung für Clustering/Cloud

Speichersysteme in einem Cluster

- DAS, NAS, SAN, ...
- Split Brain & Quorum

Übung

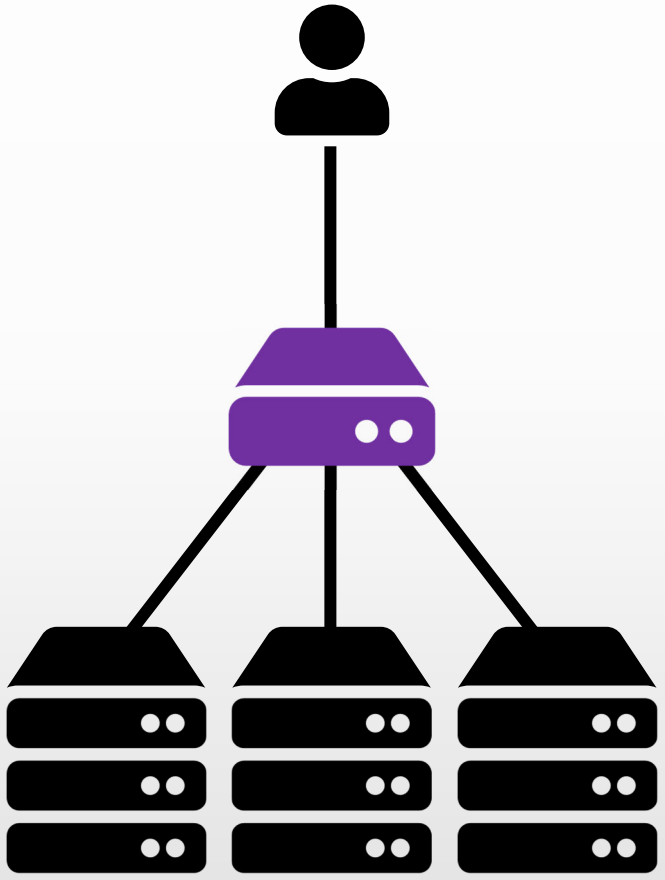
Rückblick: Load Balancing

Server-Requests werden auf mehrere Server verteilt

Neue Komponente: **Load Balancer**!

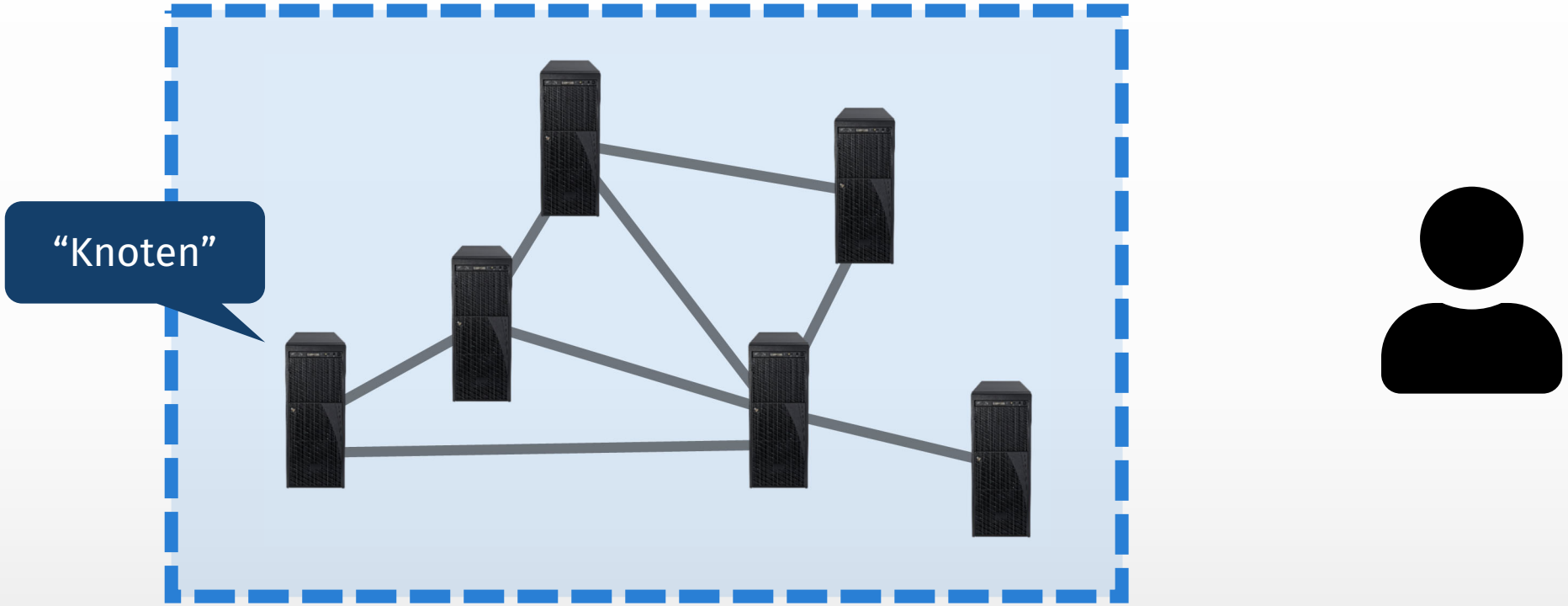
Herausforderungen:

- **Verfügbarkeit**
- Monitoring
- Persistenz



Clustering

Was ist ein “Cluster”?



Mehrere vernetzte Rechner, die zusammen arbeiten, und die man im Prinzip als ***ein*** System ansehen kann.

Arten von Clusters

Compute Clusters

- Rechenpower für High-Performance Computing
- Applikationen werden explizit parallelisiert

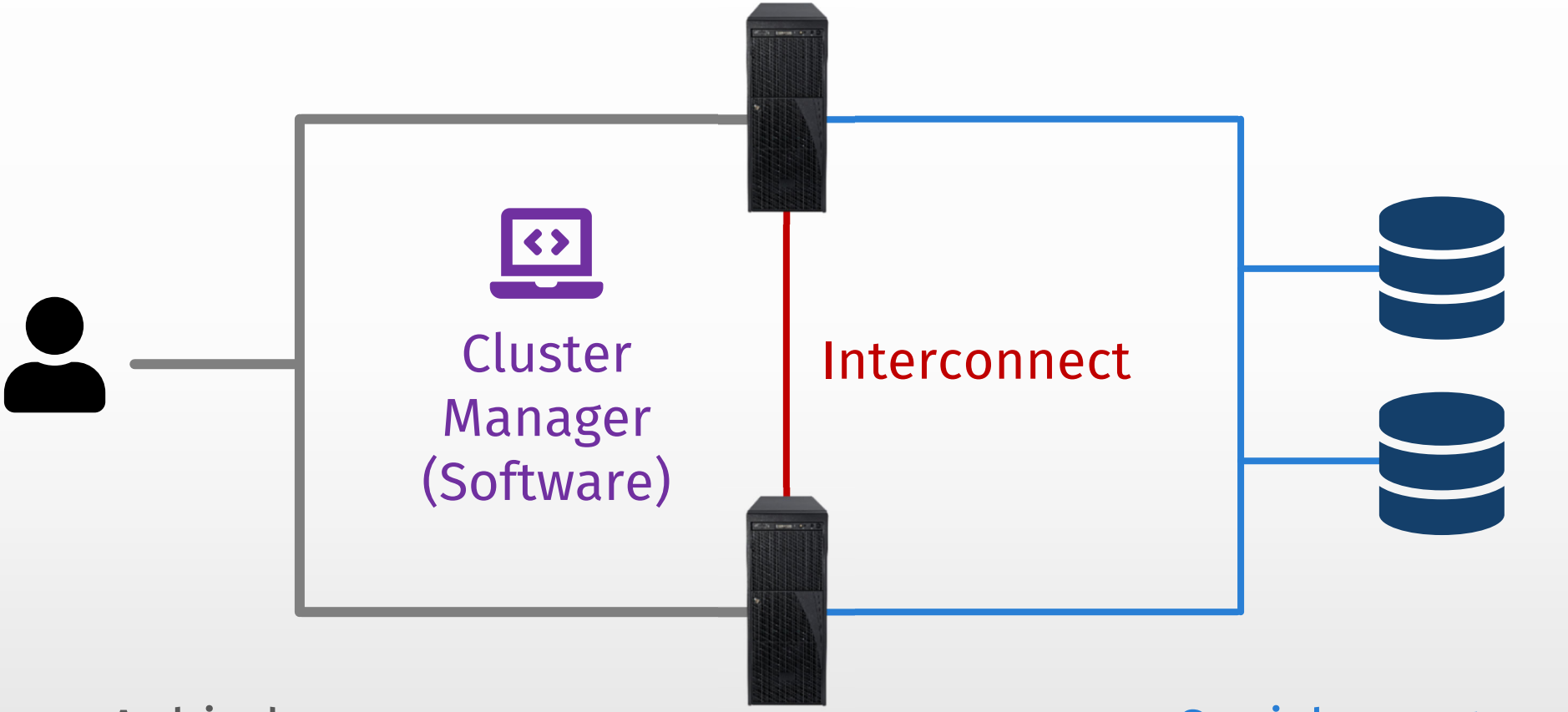
Load-balancing Clusters

- Verteilen von “gleichartigen” Requests
- Wie unsere Tomcats

High-availability (HA) Clusters

- Wenn Knoten ausfällt, übernimmt ein anderer (“failover”)
- Wie unsere Load Balancers

Komponenten eines HA-Clusters



Anwendungs-Anforderung

1. Einfacher Weg, App zu starten, stoppen und Status abzufragen (automatisierbar!)
2. App muss geteilten Speicher verwenden können
3. Zustand der App muss möglichst oft und vollständig auf persistentem Speicher gesichert werden
4. Keine Datenkorruption bei Crash oder Neustart

Viele der Anforderungen können durch Virtualisierung (Containerisierung) minimiert werden!

Virtualisierung

Virtualisierung: Prinzip

Applikation

Betriebssystem

Maschine

Applikation | ... | ...

Betriebssystem | ... | ... “Guest OS”

Virtuelle Maschine | ... | ... “Guest”

Hypervisor

Reale Maschine “Host”

Funktionsweise

Einfachste Technik: Emulation

- Kann beliebige Hardware virtualisieren (z.B. ARM auf x86-64 CPU)

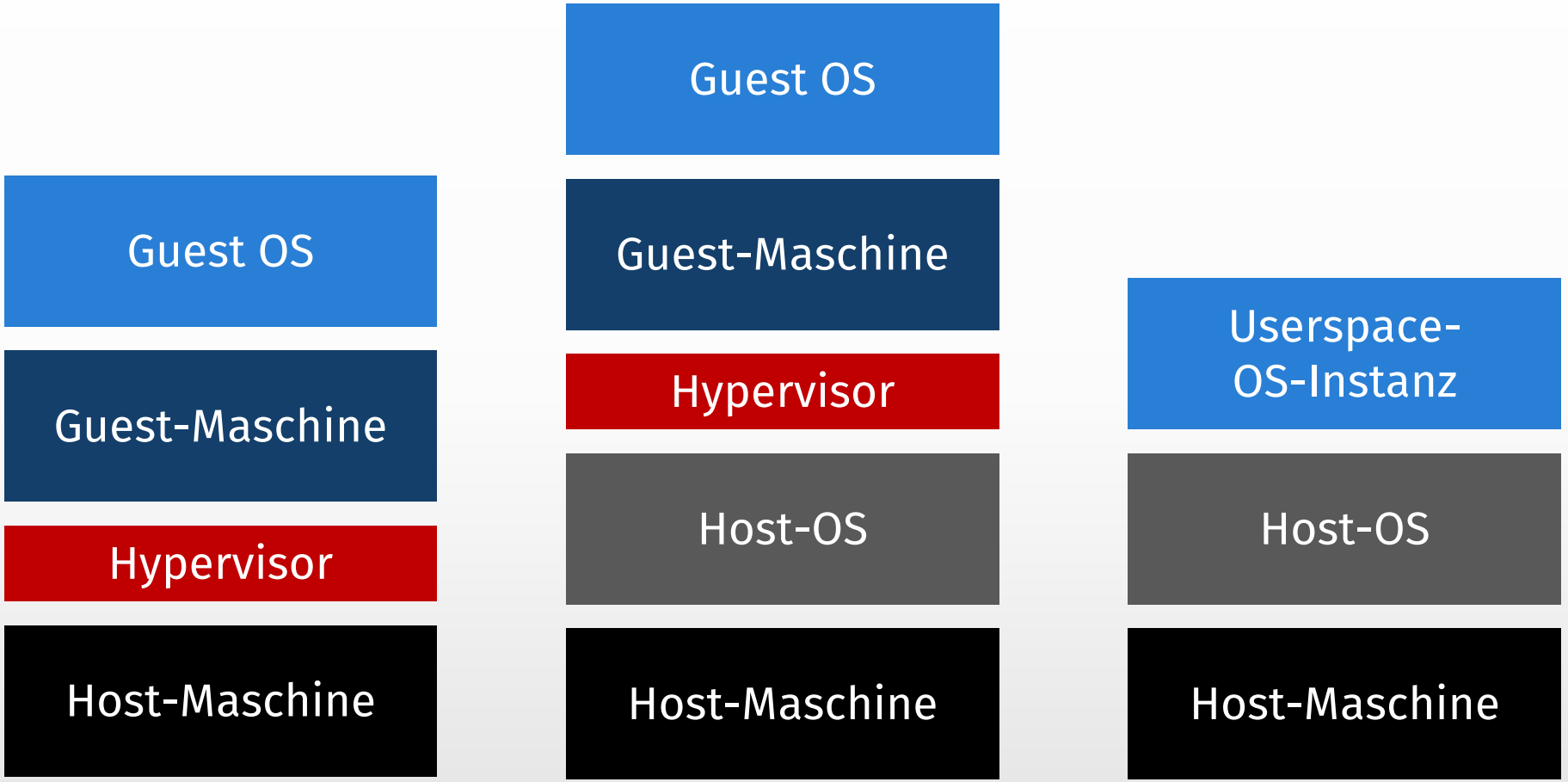
Bei gleichem Instruction Set (z.B. x86-64), können Instruktionen direkt auf Host-CPU ausgeführt werden

- Nicht alle! Gewisse sind OS-Kernel vorbehalten (“Ring 0”)

Hardware-assisted Virtualisation erlaubt noch effizientere Ausführung

- CPU gaukelt dem Gast vor, er würde in Ring 0 laufen, aber schützt Host-OS von unerwünschten Änderungen
- Beispiele: Intel VT-x, AMD-V

Arten von Virtualisierung



“Typ-1”
(Bare-Metal) Hypervisor

“Typ-2”
Hypervisor

Containerisierung

Bedeutung für Clustering/Cloud

Ressourcen-Pooling

NIST: "*Computing-Ressourcen werden zusammengelegt ('gepoolt'), um mehrere Kunden mit denselben physischen Ressourcen zu bedienen.*"

- CPU & RAM: Hypervisor kann Limiten für VMs festlegen
- Speicherplatz: Zentralisierter Speicher für alle Gäste, wird nach Bedarf aufgeteilt
- "Thin Provisioning": Speicherplatz wird Guest zugeschrieben, aber erst alloziert, wenn er wirklich verwendet wird
- Deduplication: Identische Blöcke von VMs werden nur 1x gespeichert

Live Migration

RAM-Inhalt und Netzwerk-Verbindungen werden beibehalten, wenn VM von Host zu Host migriert wird

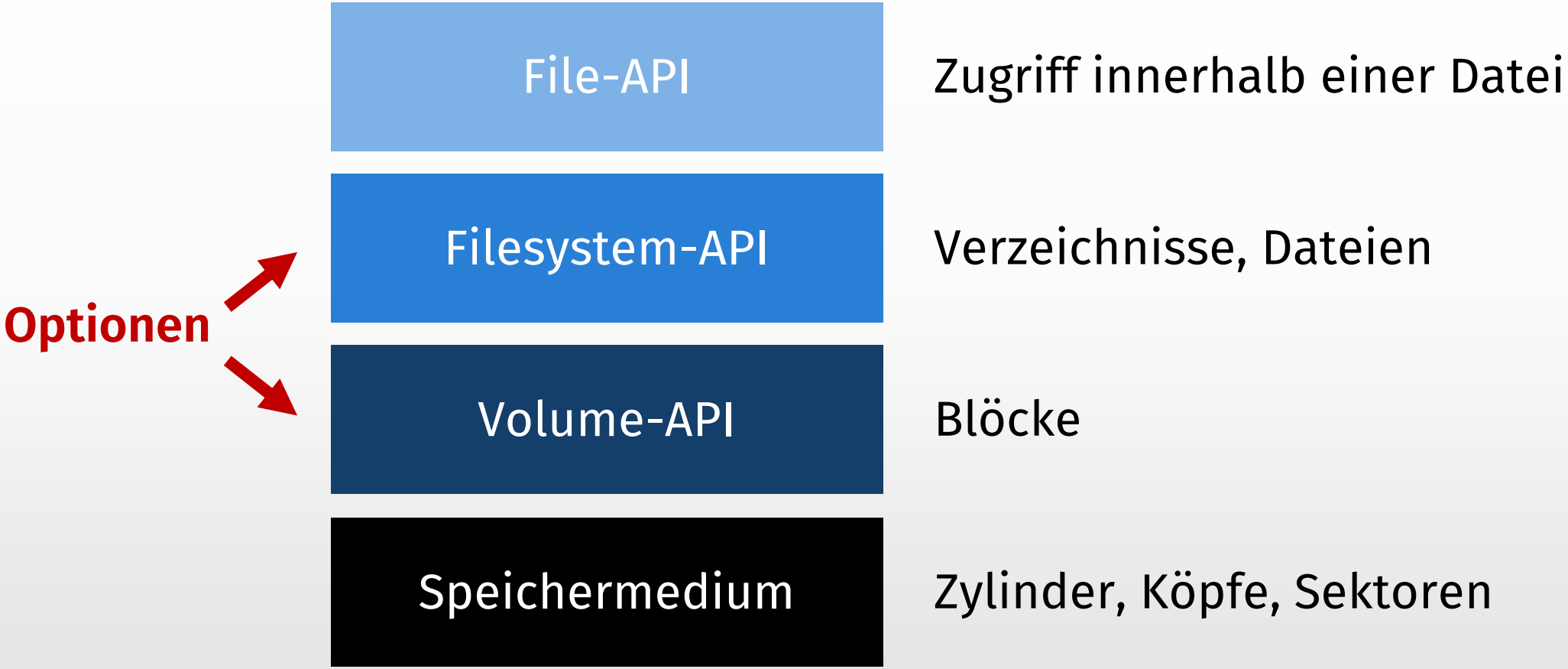
Speichersysteme in einem Cluster

Direct-Attached Storage (DAS)

Fancy Name für Festplatten, die direkt an Host angeschlossen sind

- Schnittstellen: SCSI, SATA, eSATA, USB, …
- *Block-basierte* Schnittstelle

Speicher-APIs



DAS: Vor-/Nachteile

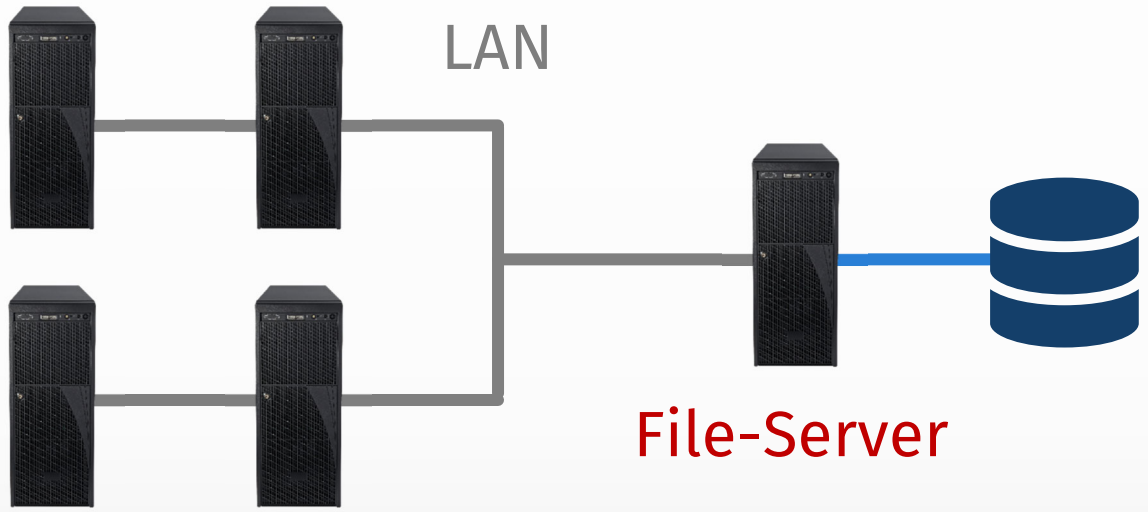
Super-einfach

Direkter Zugriff nur von einem Host möglich

Zugriff von anderen Hosts:

- Protokolle wie FTP
- Netzwerk-Dateisysteme wie NFS oder SMB
- Oder...

Network-Attached Storage (NAS)



Dedizierter File-Server in LAN, Zugriff von allen anderen Hosts möglich

- Zugriff normalerweise über NFS oder SMB (d.h. über TCP/IP)
- Einfache Lösung, aber: Limitierte Bandbreite, belastet LAN

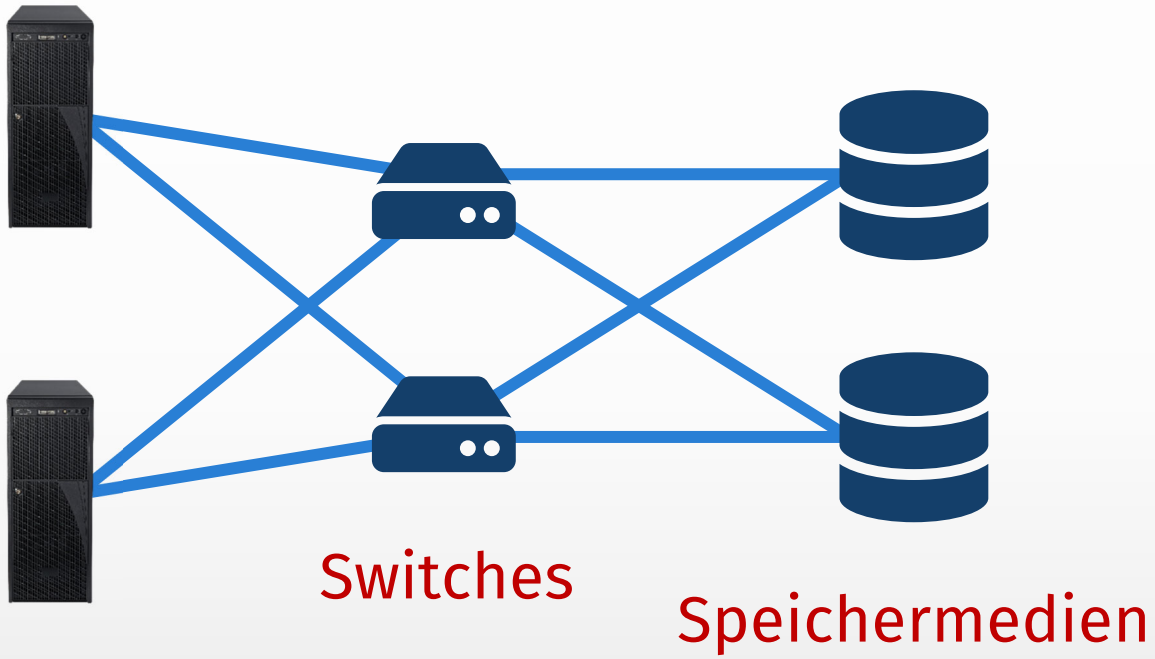
DAS & NAS: High Availability?

Wenn Speichermedium an *einen* Host angeschlossen ist, **wie verhindern wir Single Point of Failure**?

Ansätze:

- Regelmässiges Spiegeln auf anderen Host?
- Verteiltes Dateisystem?!
- **SAN!**

Storage Area Network (SAN)



Dediziertes Netzwerk für Speicher

Extra-Features: Automatisches Backup, Monitoring

SAN: Eigenschaften

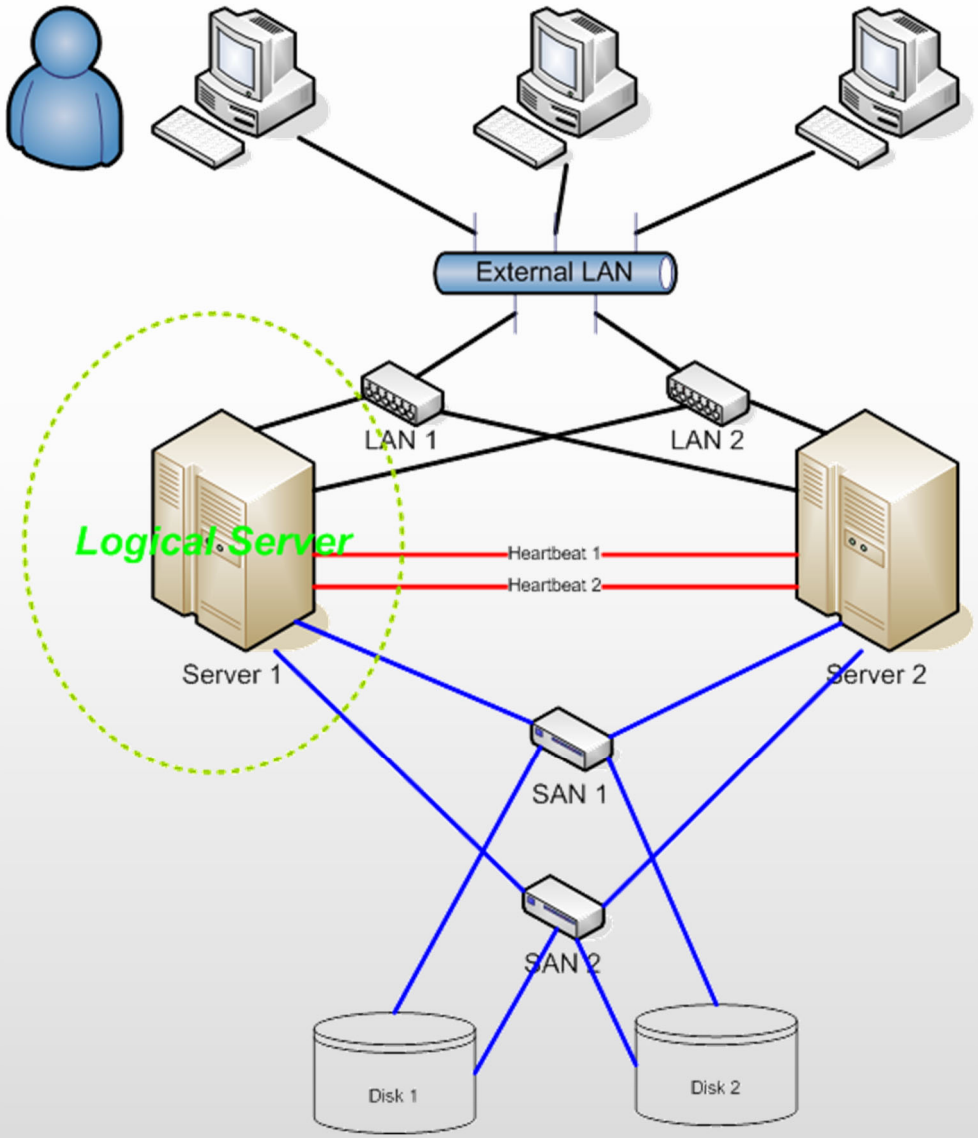
Spezielle Protokolle: Fibre Channel, iSCSI, Infiniband

- Darunter: Kupfer oder Glasfaser, für bis 10 km-Leitungen
- Höhere Bandbreite: z.Z. 50 Gbit/s pro Link, mit 4x, 8x, 12x…

Zugriff ist block-basiert! (Wie bei direct-attached storage)

- D.h. Dateisystem wird von Host implementiert
- Braucht spezielles “shared disk file system”
- Zugreifende Hosts müssen **Zugriff koordinieren**!

Echte Redundanz



SAN: HA-Alternativen

Direct-attached storage mit (echtem) **verteiltem FS**

- Beispiel: Hadoop Distributed File System (Open Source)

Für weniger Daten: **Verteilter In-Memory Store**

- Beispiel: Hazelcast (Open Source)
- Werden wir später einsetzen!

Probleme mit Shared Storage

Nodes müssen genau wissen, welche anderen Nodes “alive” sind und ebenfalls auf Speicher zugreifen.

- SAN: Block-basierter Zugriff!
- Auch ohne SAN ein Problem: Datenkonsistenz

“Split Brain”

Links zwischen zwei Knoten sind down, aber Anbindung an öffentliches Netz und an Speicher steht noch

Jeder Knoten meint, er sei der einzige und koordiniert nicht mehr!

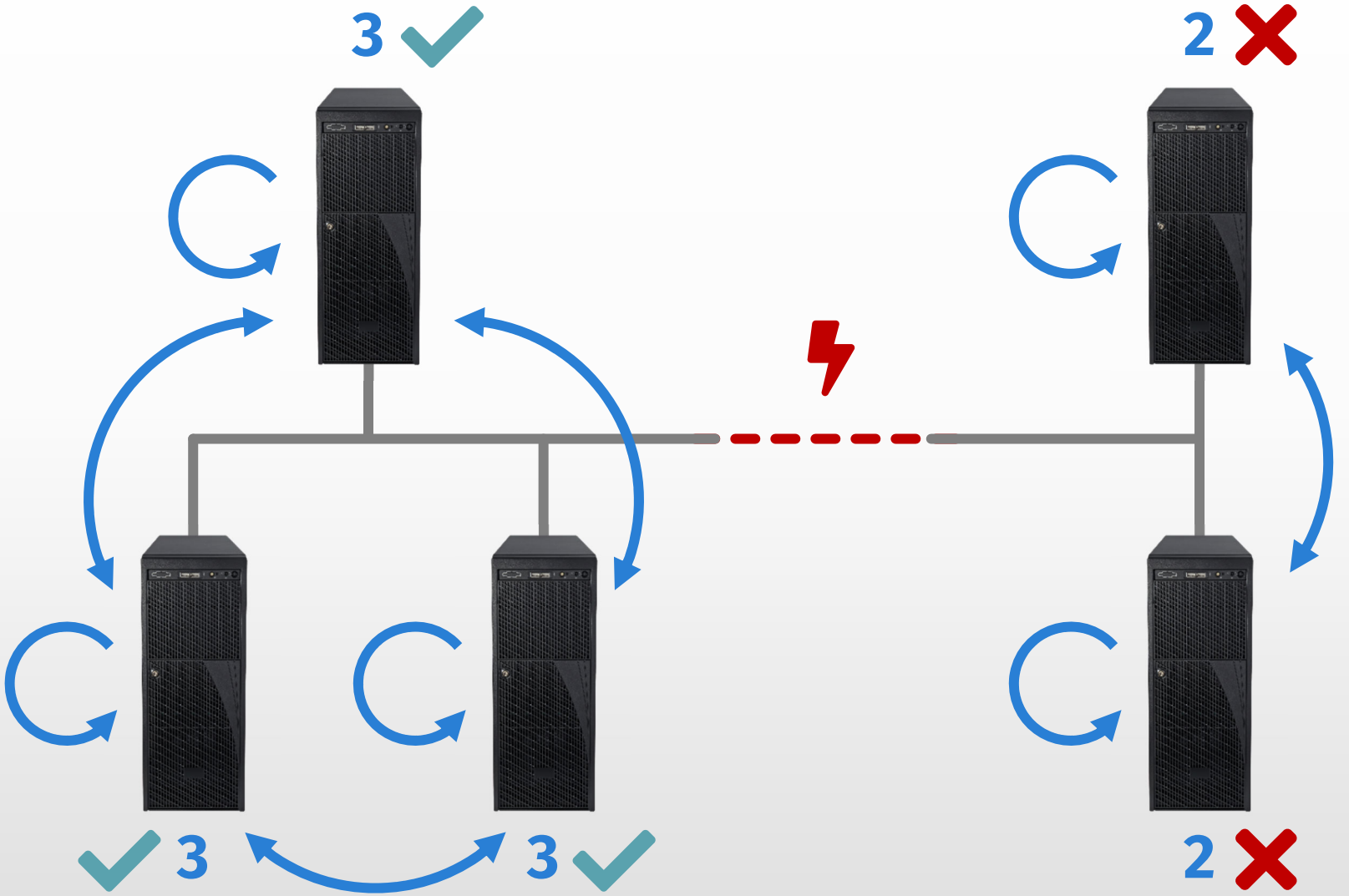
Quorum

Methode, um zu bestimmen, welcher Teil eines Clusters weiterläuft, wenn Links ausfallen

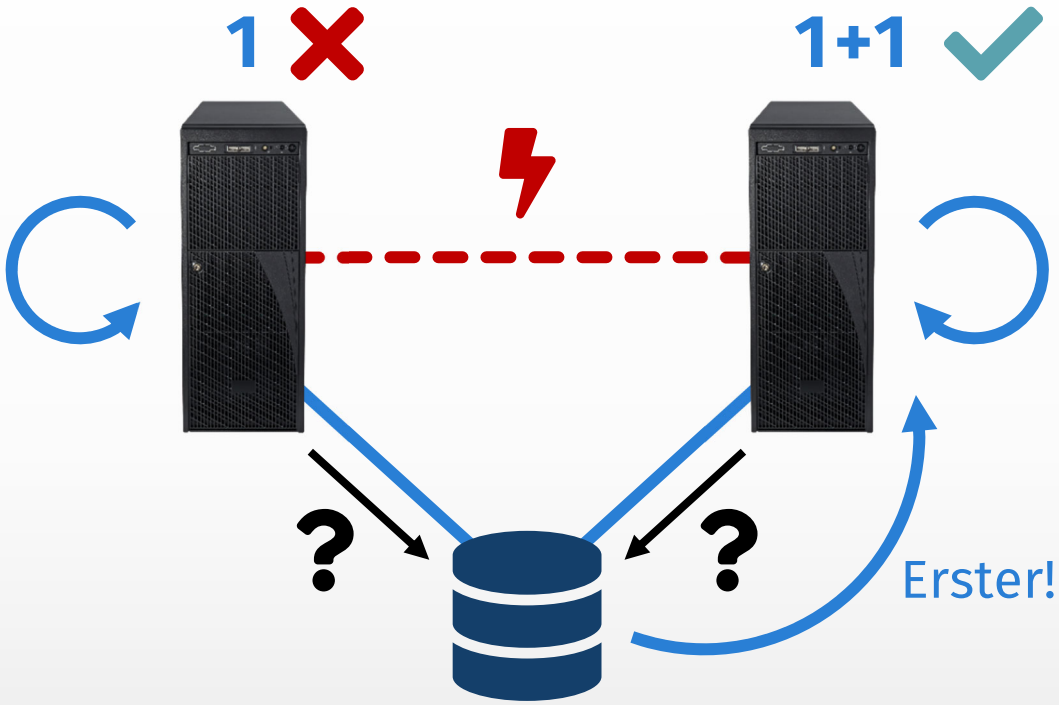
Prinzip: **Abstimmung**

- Jeder Knoten hat eine *“Stimme”*
- Knoten erhalten für jeden “sichtbaren” anderen Knoten dessen Stimme
- Jeder Knoten, der >50% der Stimmen hat, läuft weiter
- Alle anderen “schalten sich aus”

Quorum: Beispiel



Quorum: Gerade Anzahl?



Lösung: Zusätzlicher “Zeuge” (*witness*) mit Extra-Stimme:
Wer die Stimme (als erstes) holen kann, gewinnt!

- Beispiele: Voting Disk (Quorum Disk) in SAN, oder File Share

Übung: High-Availability

mit keepalived