Dr. Stefan Fütterling – The Open Group Distinguished Architect – stefan.fuetterling@capgemini.com
Martin Bachmaier – IT Versatilist – mbachmaier@lenovo.com
Daniel Amor – The Open Group Master Architect – danny@dxc.com

Duale Hochschule Baden-Württemberg Stuttgart
IT Architekturen, 2025-11

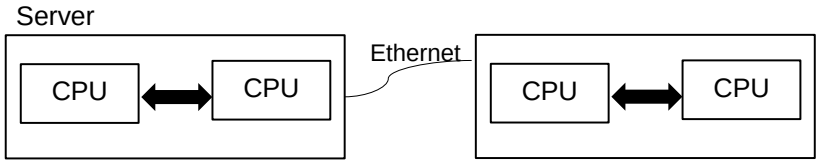
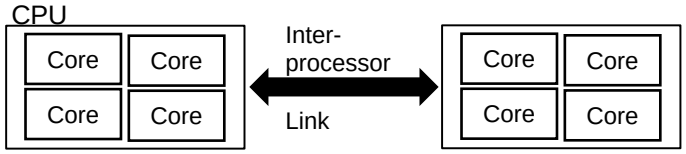
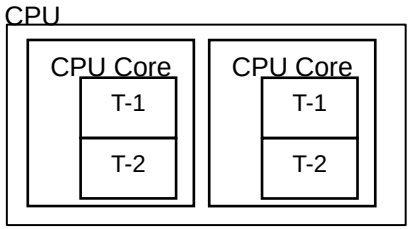
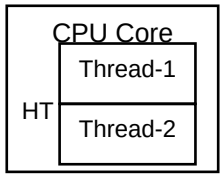
Clusterarchitekturen

Gesamtsicht der Vorlesung

- **Einführung**
  - 1.1 Einführung in IT Architektur
  - 1.2 Dynamische IT Infrastrukturen
  - 1.3 Cloud Computing
- **Server Virtualisierung**
  - 2.1 Einführung in die Server Konsolidierung und Virtualisierung
  - 2.2 Virtuelle Maschinen (VMs) am Beispiel VMware vSphere (ESXi)
  - 2.3 OS Containers am Beispiel Linux LXC und Docker
  - 2.4 Deep Dive x86 Virtualisierung
- **Zentralisierter Storage**
  - 3.1 Storage Area Networks (SAN) und Network Attached Storage (NAS)
  - 3.2 RAID Levels
  - 3.3 Disksysteme und Hyperconverged Infrastructure (HCI)
- **Clusterarchitekturen**
  - 4.1 Einführung in Clusterarchitekturen (LB-Cluster, HPC Cluster, HA Cluster)
  - 4.2 Scale Out Data Center
  - 4.3 Clustersoftware am Beispiel parallele Datenbanksysteme und Big Data Analysis Cluster (Hadoop)
- **IT Betrieb**
  - 5.1 Überblick DevOps, Application Management und Systems Management
  - 5.2 IT Service Management (ITIL)

4.1 Einführung in Clusterarchitekturen

Parallelisierung auf verschiedenen Ebenen im RZ

		Implementierung	Architekturelle Sichtbarkeit
Viele Server		Endkunde	Applikation
Server mit zwei multi-core CPU sockets		System-hersteller	BIOS und aufwärts
Multicore CPU		Silicon	BIOS und aufwärts
Singlecore CPU mit Hyperthreading		Silicon	BIOS und aufwärts

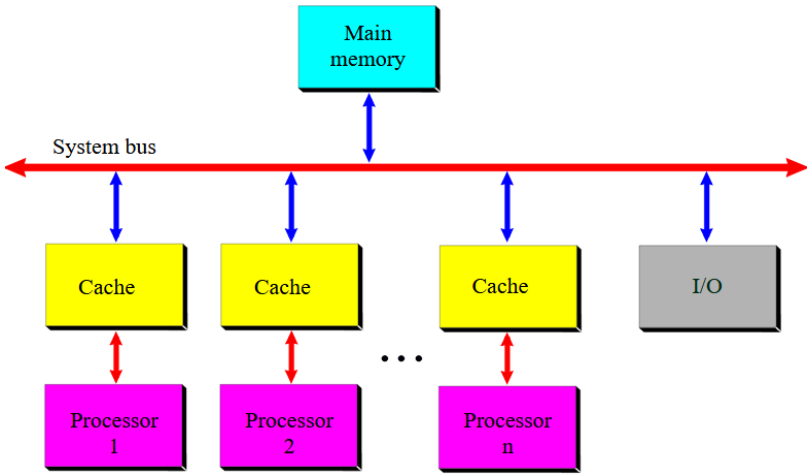
Wichtig auf jeder Ebene:
- How to parallelize task? Programmer vs compiler?
- How to synchronize between tasks?
- How to implement coherency between tasks?

Multiprozessor, SMP und Clustersysteme

- Ein **Multiprozessor System (MP)** ist ein paralleles System, welches aus mehreren Prozessoren besteht.

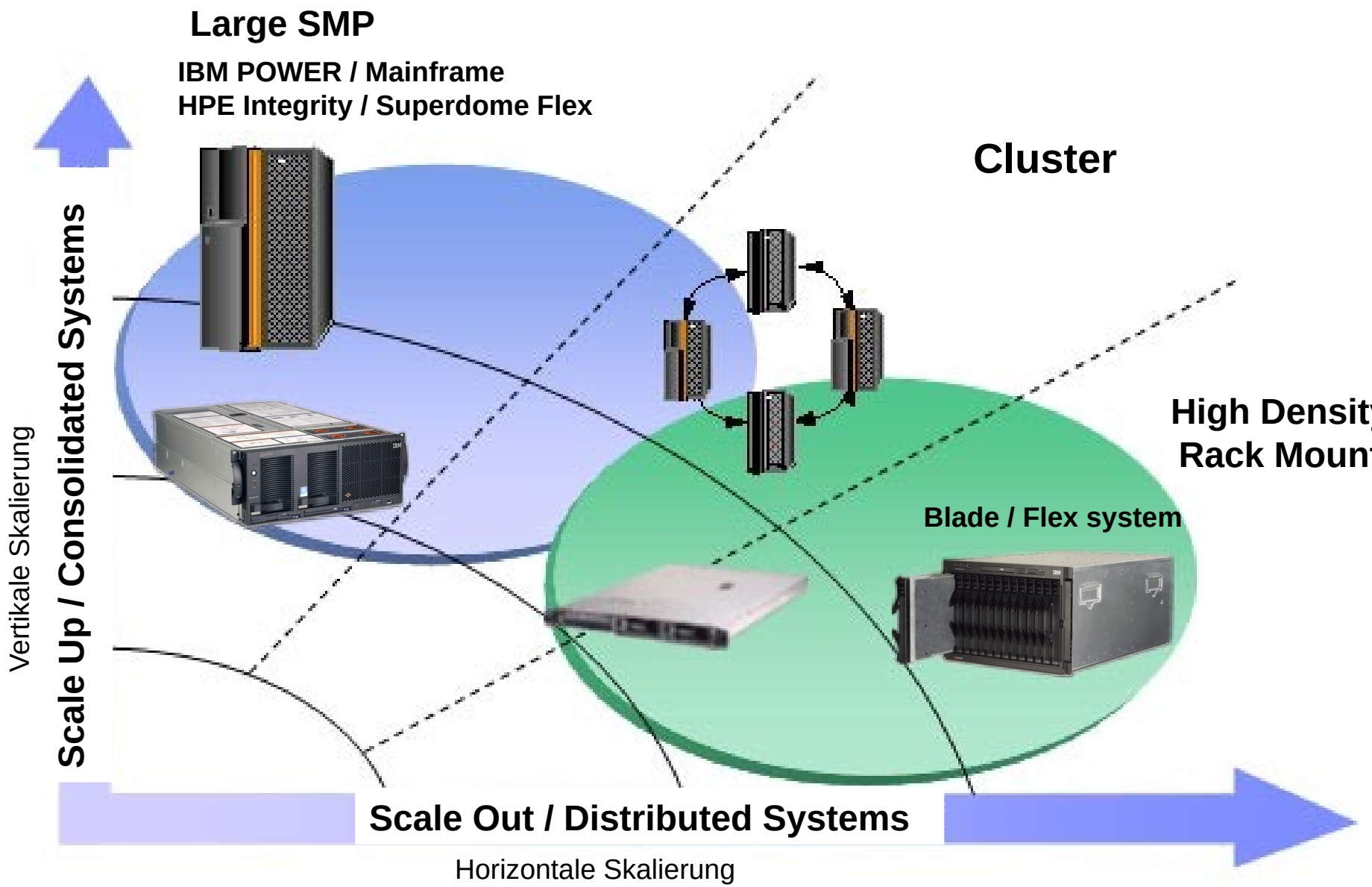
- Ein **Symmetric Multiprozessor System (SMP)** ist ein paralleles System
  - aus **mehreren gleichberechtigten Prozessoren**
  - mit **gemeinsamen Hauptspeicher**

  Das System ist aus Prozessorensicht symmetrisch.
  - *Weitergehendes Thema: die Grafik zeigt eine UMA Architektur*
    Uniform memory access
    Alle modernen x86 Architekturen heutzutage haben NUMA
    non-uniform memory access
    Hauptunterschied: Hauptspeicher hängt an jeder CPU; CPUs sind untereinander verbunden



- Ein **Cluster** besteht aus mehreren Systemen (Nodes), die
  - über geeignete **Verbindungen** (Links) miteinander verbunden sind
  - über eine **Clustersoftware** miteinander zu einer logischen Prozessiereinheit gekoppelt werden, so dass Programme oder Programmteile parallel auf diesen Systemen ausgeführt werden können.

SMP und Cluster – oder: Scale Up und Scale Out

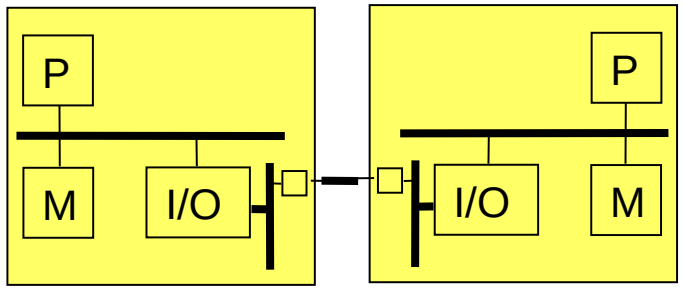


Clusterarchitekturen - Kategorisierung

Man unterscheidet folgende Clusterkategorien:

- **Message-Based** oder **Message-Passing Cluster**:

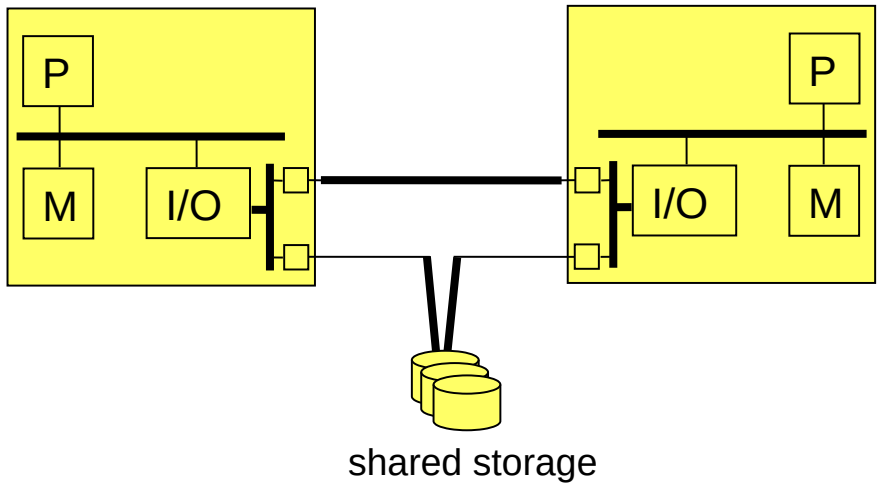
  In einem solchen System erfolgt die Kommunikation zwischen den Clusterknoten über Messages, die die Kommunikationsdaten zwischen den Clusterknoten transportieren.



- **Shared Storage Cluster**:

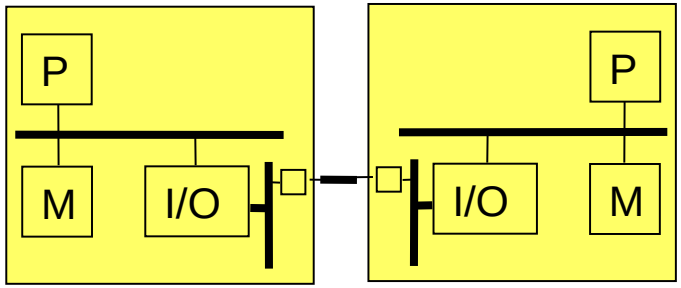
  Ein Shared Storage Cluster verfügt neben der Kommunikation über Messages noch über einen zusätzlichen Shared Storage, auf dem Daten abgelegt werden, auf die alle Clusterknoten Zugriff haben.

  Der Shared Storage ist das primäre Architekturmerkmal des Clusters.



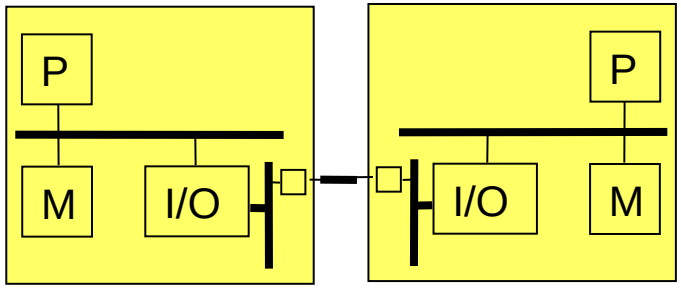
Beispiele für die Clusterkategorie: Message based Cluster

1.) Load Balancing (LB) Cluster
2.) High Performance Computing (HPC) Cluster



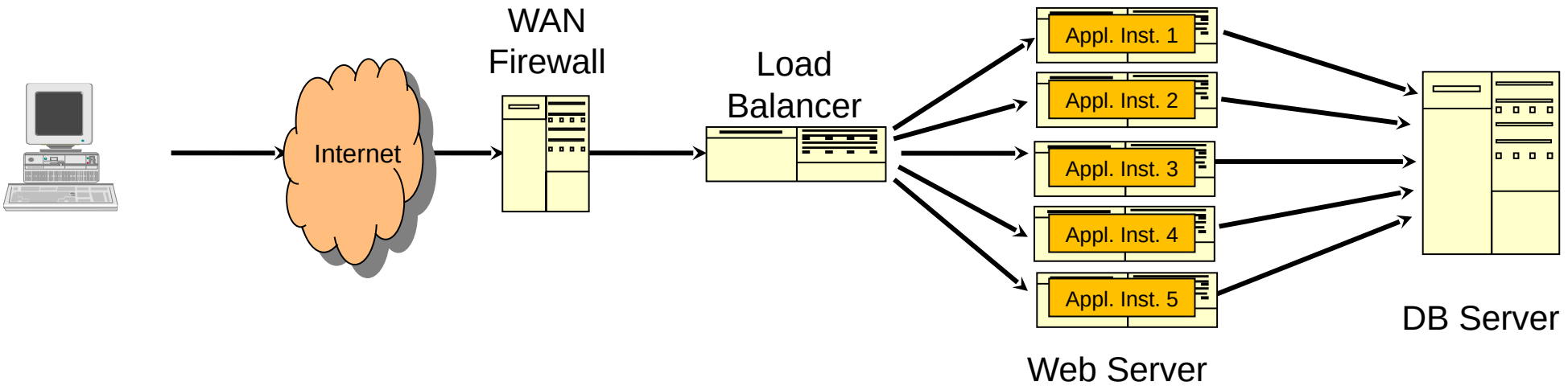
Message based Cluster

- Einfachste Form der Clusterbildung
- Keine spezielle Hardware nötig
- Alle Clusterknoten werden über LAN oder einen anderen Interconnect verbunden
- Beispiele:
  - Parallelisierte Web-Anwendungen
  - Anwendungscluster, z.B. WebSphere Cluster
  - Compute Cluster, z.B. x86 scale-out HPC
  - Parallele Datenbanksysteme auf Basis der Shared-Nothing-Architektur



1.) Load Balancing (LB) Cluster - Beispiel Web Server

- Clusterkategorie: **Message based**
- Der Load Balancer verteilt Anfragen auf mehrere Web Server (Load Balancing)
  - gleichmäßig auf alle Web Server
  - nach Auslastung der Web Server
- Web Server
  - Erhalten die Anfragen vom Load Balancer
  - Bearbeiten die Anfragen in ihrer Anwendungsinstanz
  - Senden die Antwort zurück



2.) High Performance Computing (HPC) Cluster

- Clusterkategorie: **Message based**
- Verbindung von einer großen Anzahl von Systemen (Clusterknoten) über schnelle Verbindungen (Interconnects)
- Ziel ist die schnelle parallelisierte Bearbeitung rechenintensiver Aufgaben
- Aufgaben werden auf Software Ebene parallelisiert und auf den Clusterknoten parallel bearbeitet.
- Beispiel Leibniz Rechenzentrum (LRZ) SuperMUC-NG (x86 basiert, 2019):
  - 6480 server mit 2 bzw 4 CPUs 24 cores per Intel CPU, insgesamt 311.040 cores, 719 TB RAM, 3,4 MW
  - Peakleistung 26,9 Petaflop
  - Materialwissenschaften
    Klimamodellierung
    Teilchenphysik
    Biomolekulare Simulationen
    Astrophysik

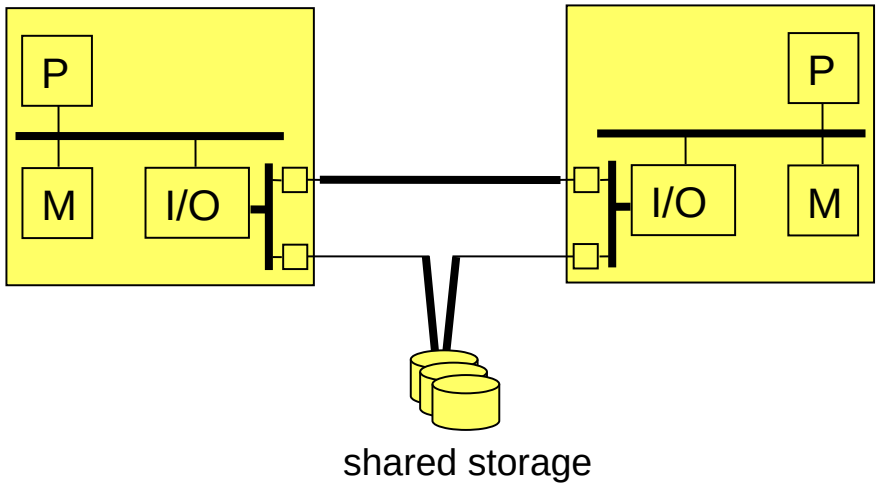
Vertiefung dieser Clusterarchitektur im Kapitel Scaleout Datacenter

http://www.lrz.de
https://www.youtube.com/watch?v=GxGrLm4ufYE

Beispiel für die Clusterkategorie: Shared Storage Cluster

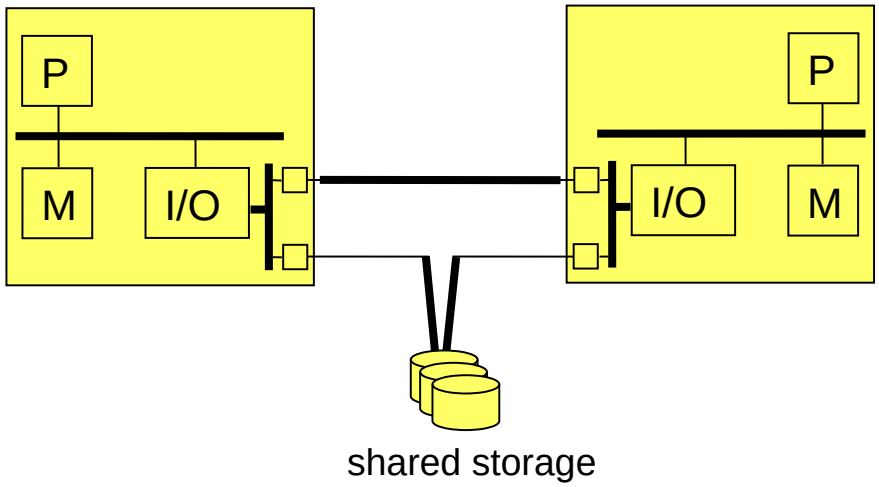
3.) HA Cluster **mit Failover einer Anwendung**

4.) HA Cluster **mit Failover einer ganzen VM**

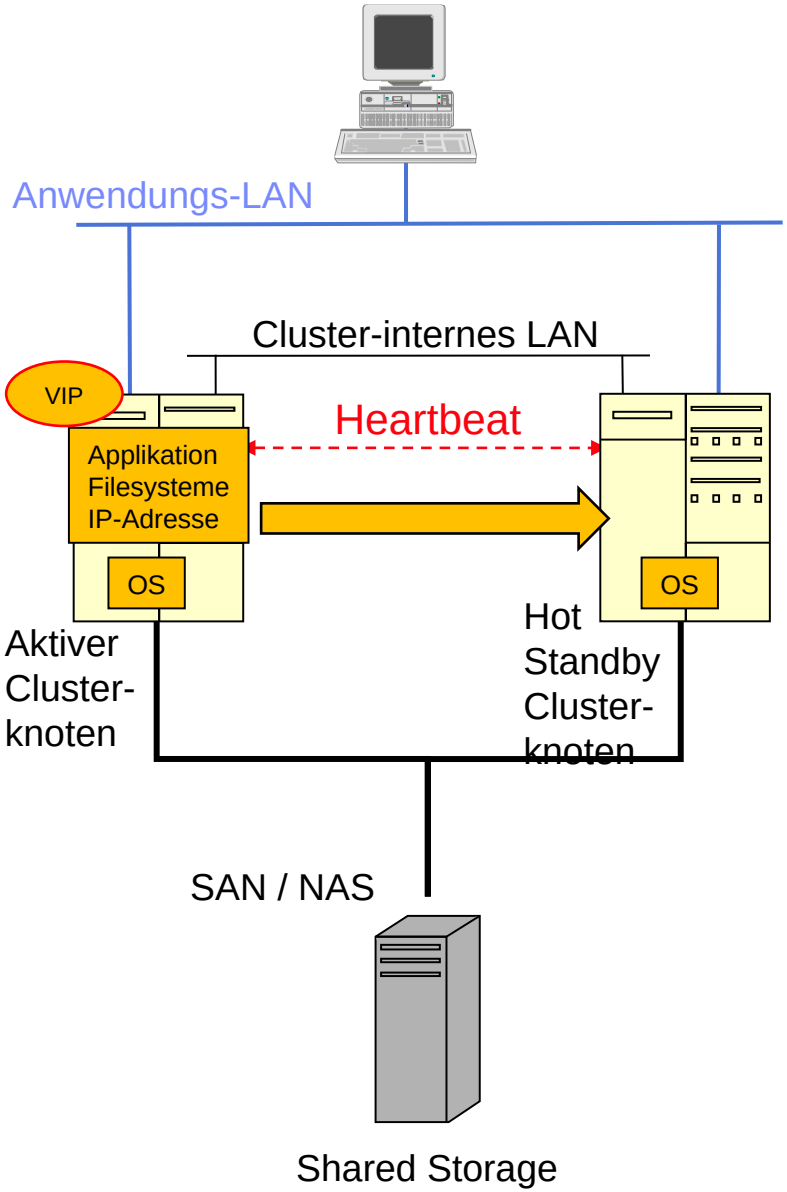


I/O Attached – Shared Storage („Shared Disk“)

- Daten befinden sich auf Shared Storage
- Alle Cluster Knoten können gleichberechtigt auf die Daten zugreifen
- Workload Management zwischen den Clusterknoten ist möglich
- High Availability (HA) Failover Cluster verwenden meist diese Form des Interconnects
- Beispiele für **Failover einer Anwendung** auf einen zweiten Knoten:
  - IBM PowerHA (for AIX and i)
  - SUSE SLES HA Extension
  - HPE Serviceguard
  - Windows Failover Clustering (WSFC)
- Beispiele für **Failover einer kompletten virtuellen Maschine**:
  - VMware vSphere HA
- Beispiele für **Failover eines Containers**:
  - Durch Docker Swarm, Kubernetes, ..

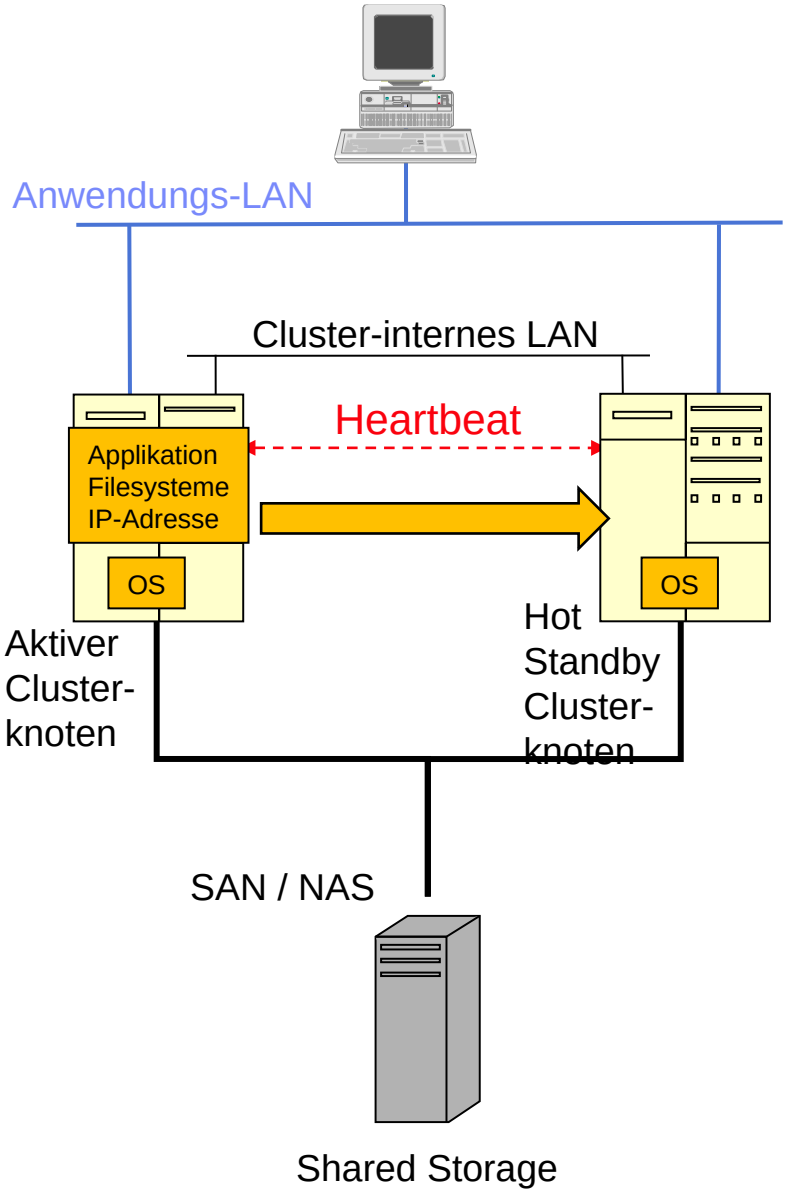


3.) High-Availability (HA) Cluster mit Shared Storage und Heartbeat: Failover einer Anwendung auf einen zweiten Knoten



- Clusterkategorie: **Shared Storage Cluster**
- Zwei Server (Clusterknoten):
  - Anwendung läuft nur in einer Instanz auf dem aktiven Knoten
  - Passiver Knoten als Hot Standby
  - gegenseitige **Heartbeat** Überwachung
- Hochverfügbarkeit auf Betriebssystem-Ebene
  - über eine HA Zusatzkomponente, z.B. SLES HA Extension
  - Vorteil: Die Anwendung selbst muss nicht HA-fähig sein
- Automatisierte Übernahme (**Failover**) der Anwendung bei Ausfall des aktiven Clusterknotens:
  - Shared Filesysteme mit Appl. und Daten
  - IP Adresse
  - Start- und Stop-Skripte
- Nachteil: Im Übernahmefall Downtime wenige Minuten und Reconnect der Anwender

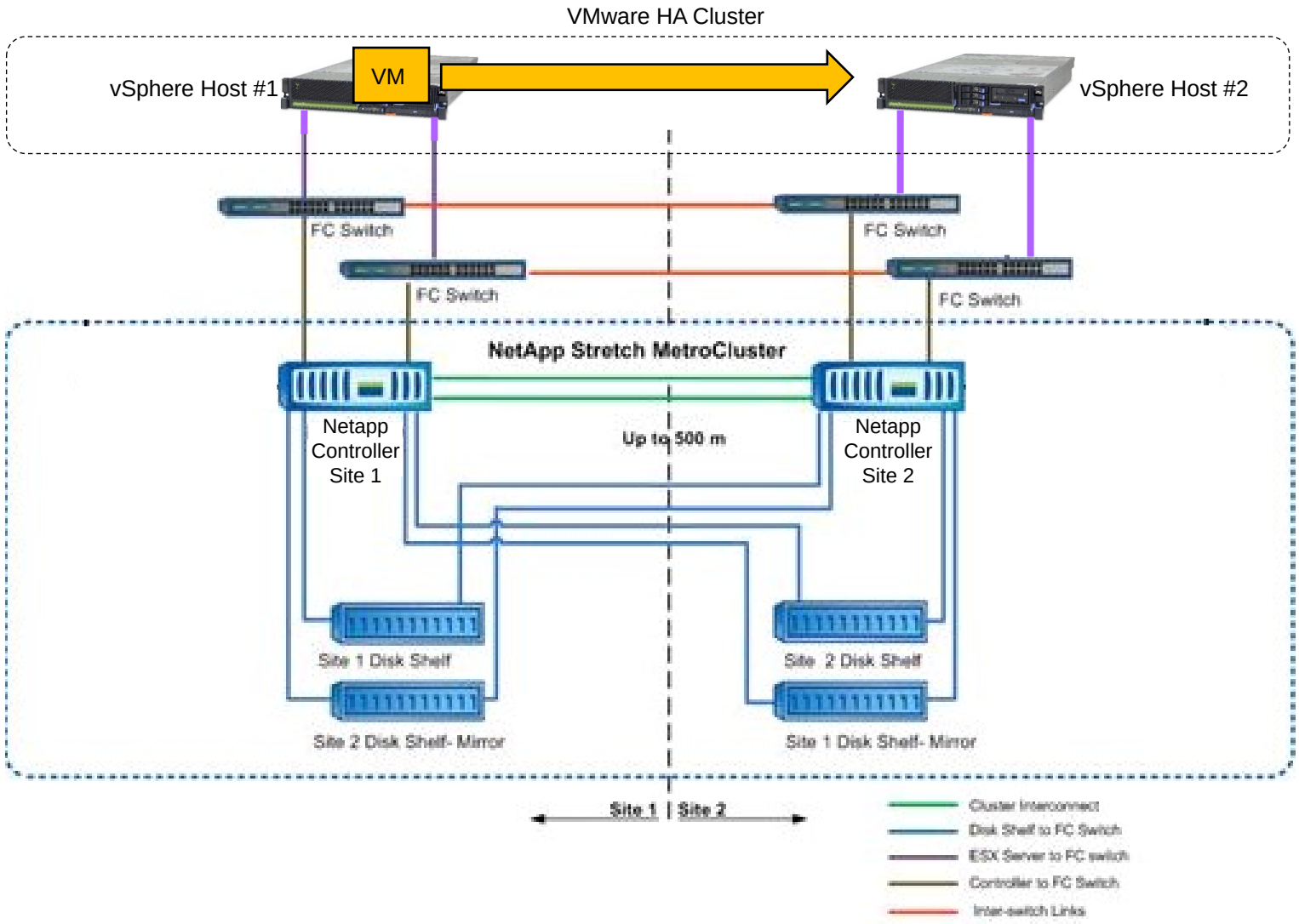
3.) High-Availability (HA) Cluster mit Shared Storage und Heartbeat: Failover einer Anwendung auf einen zweiten Knoten



Failover process

- Problem: wie stellt man sicher, dass der bisher-aktive Knoten wirklich nicht mehr im Cluster aktiv ist?
- STONITH
  Shoot the other node in the head
- Standby ist jetzt einzig verbliebener Knoten
- Erst danach beginnt der failover (Storage, IP, etc)

4.) High-Availability (HA) Cluster mit Shared Storage: Failover einer kompletten VM auf einen zweiten Knoten



- Clusterkategorie: **Shared Storage**
- MetroCluster Storage spiegelt die Daten synchron in 2 Sites
- VMware HA startet bei einem Ausfall die VMs auf einem anderen vSphere Host (**Failover mit Downtime**)
  - Kompensiert den Ausfall eines vSphere Hosts
  - Keine HA Zusatz-komponente in den Gast-Betriebs-systemen nötig