

Hochverfügbarkeit und Virtualisierung

- Die optimale Kombination beider Konzepte -

MEGWARE HPC & Cluster Systeme

Supercomputing Technology – „Made in Germany“

Peter Großöhme – HPC-Ingenieur

Inhalt (1)

- Kurzüberblick High-Performance-Computing
- Grundlagen
 - Virtualisierung
 - Clustertheorie / -arten
- Aufbau und Konzeption eines hochverfügbaren Systems
 - Anforderungen sowie grundlegende Überlegungen
 - Linux-HA / Pacemaker

Inhalt (2)

- HA-Konfigurationen
 - FTP-HA-Cluster mit DRBD
 - Webserver-HA-Cluster mit Load Balancer
 - MySQL-Cluster
 - Komplexbeispiel
 - HA-Virtualisierungscluster mit iSCSI SAN
- Zusammenfassung / Ausblick
 - neue Technologien
 - Entwicklungen und Trends

Unternehmensprofil

- Gründung 1990
- in Chemnitz – Sachsen
- rund 60 hoch ausgebildete, motivierte Mitarbeiter
- rund 500 installierte HPC Systeme in Europa



Chemnitzer Linux Cluster (CLiC)

- Prof. Hans-Werner Meuer im Jahr 2000:
„das weltweit beste Preis-Leistungs-Verhältnis“
- Position 126 der TOP 500
- 528 Rechenknoten
 - Intel Pentium III 800 MHz
 - 512 MB RAM
- Leistung von 143,3 Gflops
(heute: ein Server)



Anwendungsgebiete weltweit

- **im Forschungs-Universitäts-Bereich:**
Astronomie, Biologie, Physik, Chemie, Earth Science, Gesundheitsforschung, Klimaforschung
- **im Engineering:**
Automobilbau, Aerospace, Verteidigung-Rüstung, Öl- und Gas
- **weitere Bereiche - Datenverwaltung:**
Finanzwesen, Internet, Media

High-Performance-Computing (HPC)

- für Probleme, die eine sehr hohe Rechenleistung und / oder viel Speicher benötigen sowie nicht in vertretbarer Zeit auf einzelnen Systemen berechenbar sind
- parallelisierte Anwendungen bzw. Programmcodes
 - **aber:** nicht jedes Problem ist einfach parallelisierbar
- spezielle Hochleistungssysteme / -rechner

3 Arten von Hochleistungssystemen

- x86 HPC Cluster und Server mit Linux oder Windows HPC
- Hybrid HPC Cluster mit x86 und Tesla GPU
- Parallele Storage Cluster

x86 HPC Cluster Linux / Windows

- Einsatz modernster Technologien
- große Auswahl aus verschiedener Hardware
- Verwendung neuester Intel® und AMD® CPUs
 - Intel: Sandy-Bridge-Architektur
 - AMD: Bulldozer-Architektur (Interlagos)

x86 HPC Cluster Linux / Windows

- aktuellste Netzwerktechnologien
- verschiedene Arten von Netzwerken seit 2001 (Myrinet 2000, 10G Ethernet, Dolphin SCI, InfiniBand)
- unabhängig und herstellerneutral



x86 HPC Cluster Linux

- Einsatz verschiedener Linux Distributionen
 - Open Suse / SUSE Linux Enterprise Server
 - Red Hat, CentOS, Scientific Linux



x86 HPC mit Nvidia Tesla GPU

- Performance Erhöhung
- GPUs nicht mehr nur im Grafikbereich
 - **General Purpose Computation on Graphics Processing Unit (GPGPU)**
- Rechenapplikationen vielfach schneller rechnen

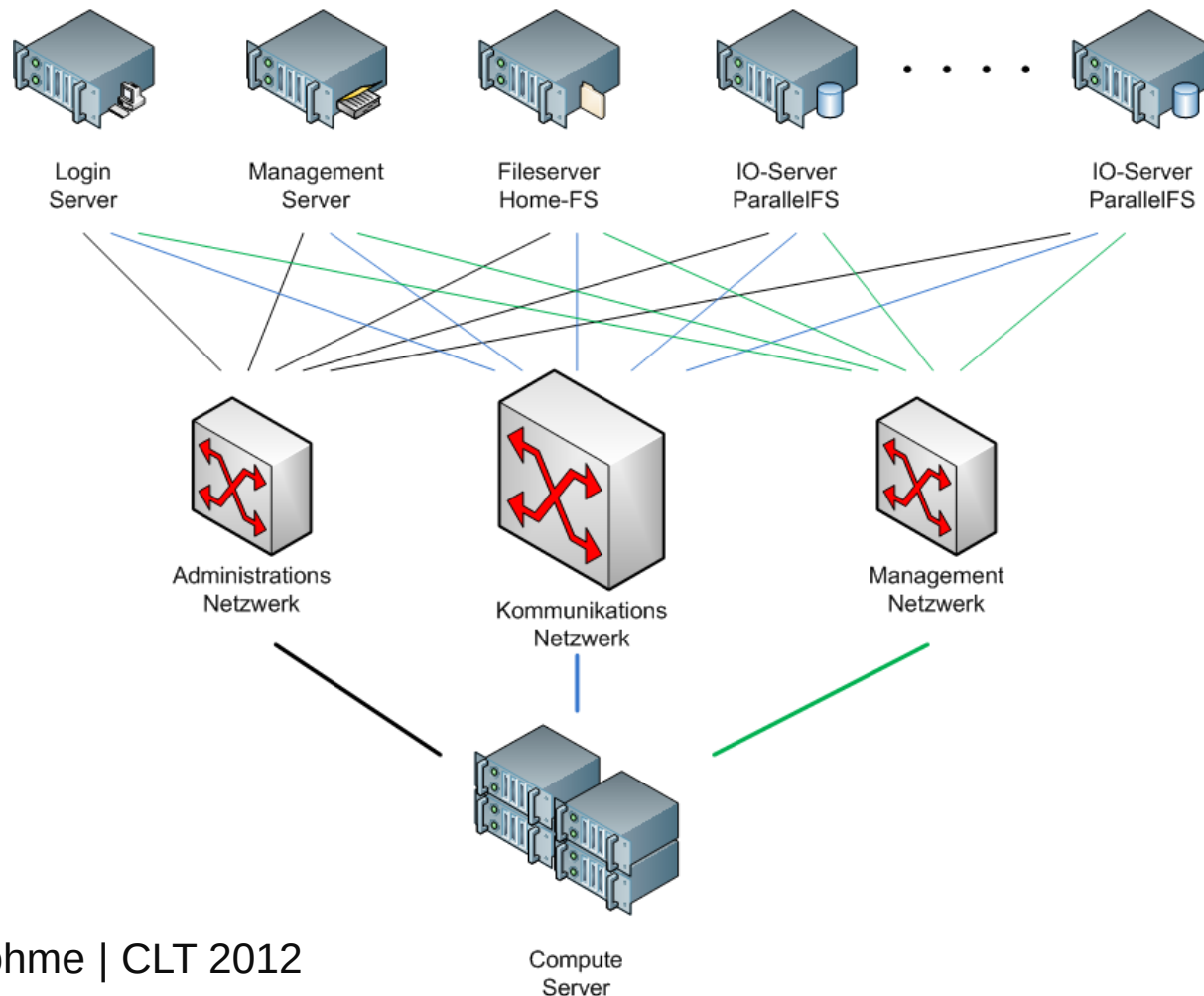


Parallel Storage Cluster

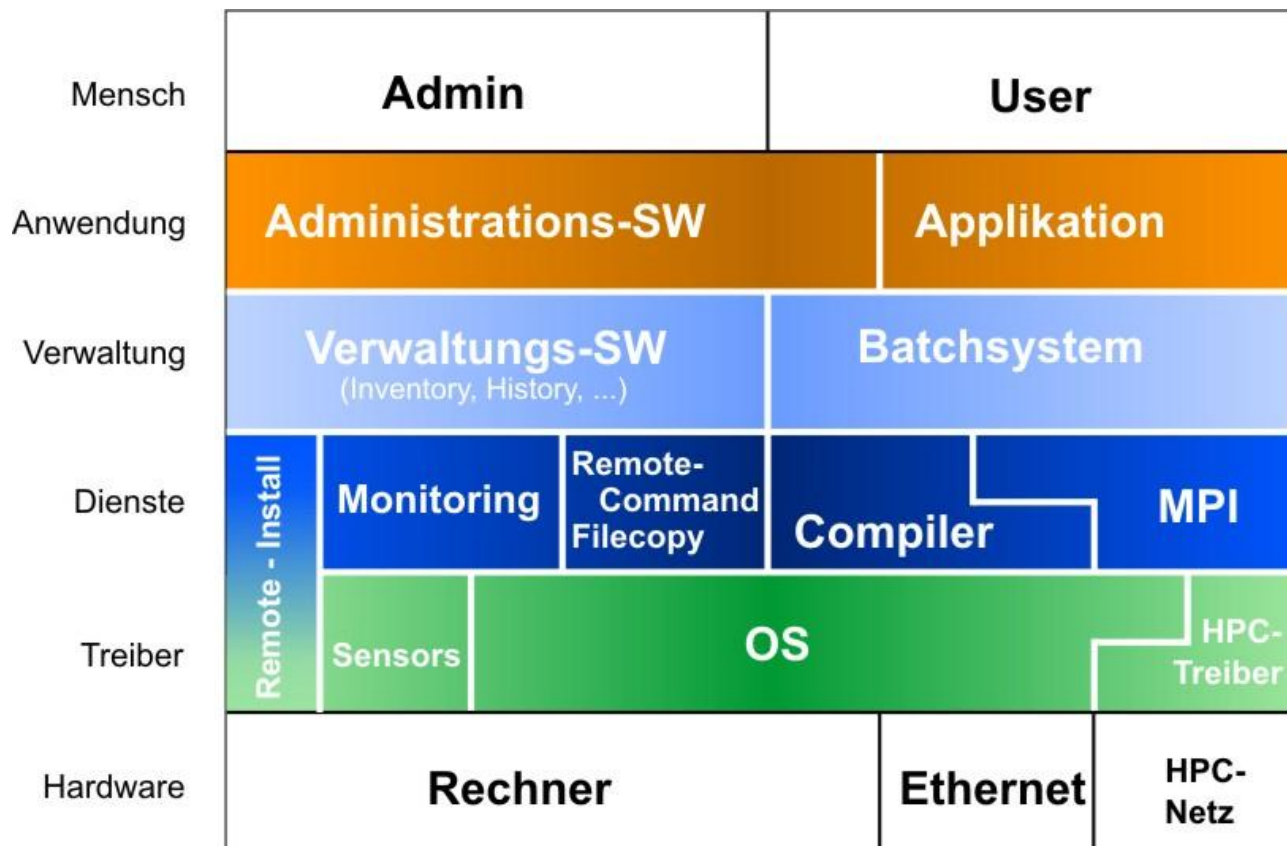
- differenzierte Filesysteme (Lustre, FhGFS, PanFS)
- High Performance Scratch Data Filesystem mit Lustre
- hohe Verfügbarkeit mit Panasas
- Performance, Verfügbarkeit, Kapazität



Aufbau eines HPC Clusters



Hardware- / Software-Stack eines HPC Clusters



Bewertung von HPC Systemen (1)

- verschiedene Standardbenchmarks
- SPEC für Parallelrechner kaum von Bedeutung
- „Quasi“-Standard: LINPACK Benchmark
 - Lösung linearer Gleichungssysteme
 - Performance wird in Gleitkommaoperationen pro Sekunde angegeben (**F**loating **P**oint **O**perations **P**er **S**econd – FLOPS)
 - www.top500.org / www.green500.org
- STREAM Benchmark
- Intel MPI Benchmark
- HPCC Benchmark bestehend aus 7 Einzeltests

Bewertung von HPC Systemen (2)

- Berechnung der Effizienz eines Systems

$$Effz = \frac{R_{\max}}{R_{peak}}$$

$R_{\max}$...Linpackleistung
 R_{peak} ...theoretische Maximalleistung

$$R_{peak} = Anzahl_{FPU} * Taktrate_{CPU} * Anzahl_{CPU}$$

Bewertung von HPC Systemen (3)

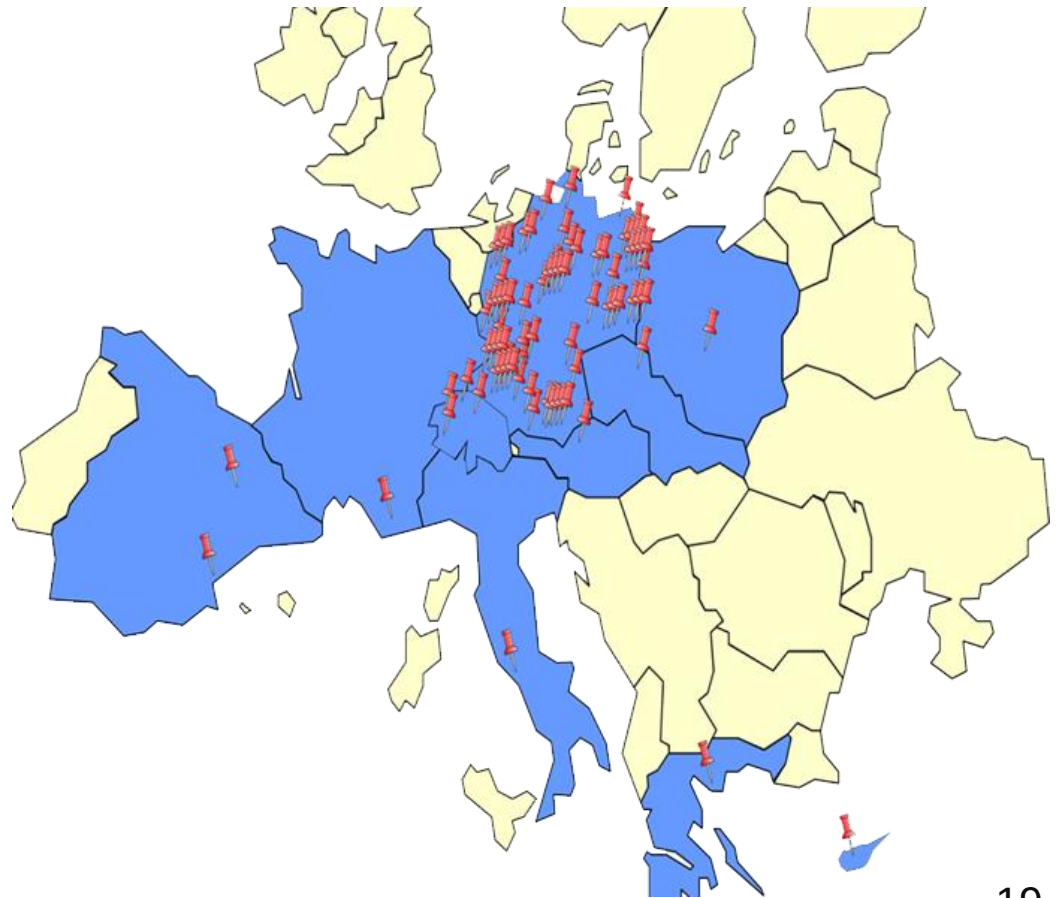
- Beispielrechnung theoretische Leistung
 - AMD Opteron 6274 (16 Cores @ 2,20 GHz)

$$R_{peak} = 2,2_{GHz} * 4_{FLOP/cycle} * 16_{CPU}$$

$$R_{peak} = 140,8 GFLOPS$$

500 installierte Systeme in Europa

- Frankreich
- Griechenland
- Polen
- Tschechien
- Österreich
- Zypern
- Schweiz ...



Vienna Scientific Cluster VSC-2

- Österreichs schnellster HPC
- 1314 SlashFive[®] -Knoten
- 30 Racks – Knürr[®] CoolDoor[®] mit passiv gekühlten Rücktüren
- Prozessorkerne: 21.024
AMD Opteron 6132 HE – 2,2 GHz
- Mellanox QDR InfiniBand
- 12 Storage Server FhGFS
(Kapazität: ca. 216 TB)



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology



Universität für Bodenkultur Wien



universität
wien

500 installierte Systeme in Europa



Friedrich-Schiller-Universität Jena



Inhalt (1)

- Kurzüberblick High-Performance-Computing
- Grundlagen
 - **Virtualisierung**
 - Clustertheorie / -arten
- Aufbau und Konzeption eines hochverfügbaren Systems
 - Anforderungen sowie grundlegende Überlegungen
 - Linux-HA / Pacemaker

Hochverfügbarkeit (1)

- Verfügbarkeit
 - Ein Dienst gilt als verfügbar, solange dieser die Aufgaben erfüllt, für die er vorgesehen ist.
- Berechnung der Verfügbarkeit
 - $\text{Verfügbarkeit} = \frac{\text{Uptime}}{\text{Uptime} + \text{Downtime}}$
- Hochverfügbarkeit
 - Anwendung auch im Fehlerfall weiterhin verfügbar
 - keine oder nur kurze Unterbrechung

Hochverfügbarkeit (2)

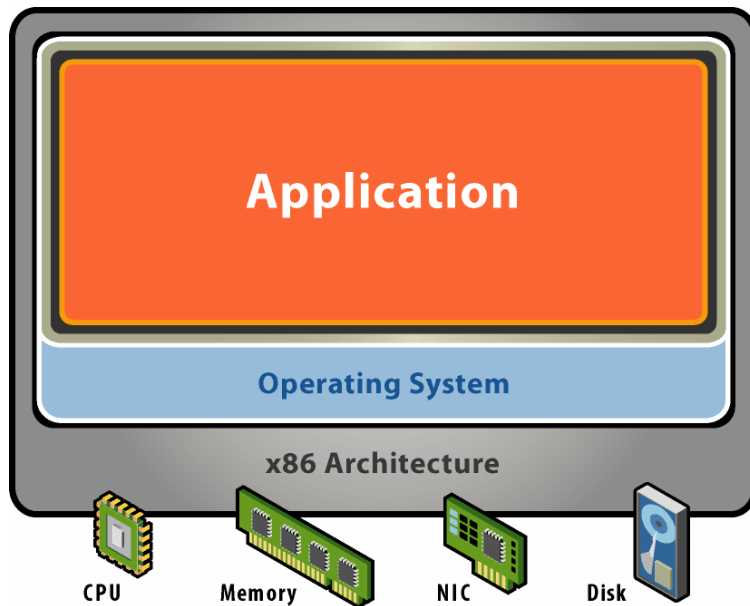
VBK-Klasse	Bezeichnung	Verfügbarkeit	Downtime pro Jahr
		90 %	36,50 Tage
		95 %	18,25 Tage
		98 %	7,30 Tage
2 (AEC-0)	stabil	99,0 %	3,65 Tage
		99,5 %	1,83 Tage
		99,8 %	17,52 Stunden
3 (AEC-1)	verfügbar	99,9 %	8,76 Stunden
		99,95 %	4,38 Stunden
4 (AEC-2)	hochverfügbar	99,99 %	52,60 Minuten
5 (AEC-3)	fehlerunempfindlich	99,999 %	5,25 Minuten
6 (AEC-4)	fehlertolerant	99,9999 %	31,50 Sekunden
7 (AEC-5)	fehlerresistent	99,99999 %	3 Sekunden

Hochverfügbarkeit (3)

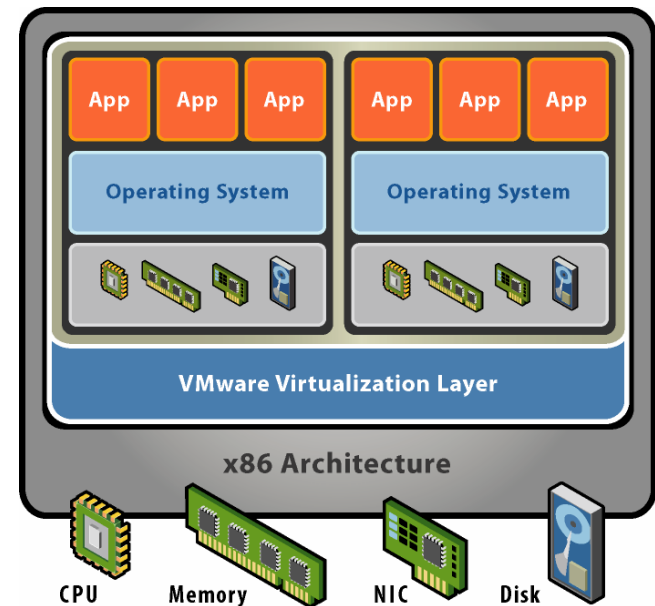
- Voraussetzungen
 - Minimierung der geplanten und ungeplanten Ausfälle
 - ungeplant:
 - Hardware- / Software-Fehler
 - fehlerhafte Bedienung / böswillige Nutzer
 - geplant:
 - Wartungsfenster durch Administrator
 - Einsatz fehlertoleranter Hard- und Software
 - Vermeidung von Single Point of Failures (SPOF)

Virtualisierung – Was ist das? (1)

einzelner Server



virtualisierte Infrastruktur



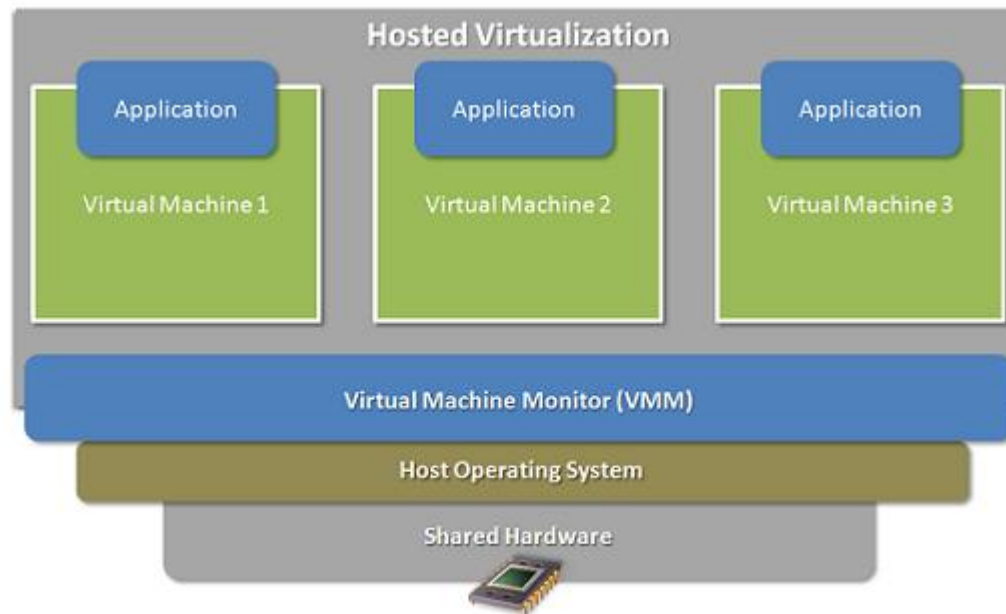
Virtualisierung – Was ist das? (2)

- **Ziele**

- Einsatz mehrerer Betriebssysteme in einer wohldefinierten Hardwareumgebung auf einem physischen Server
- effizientere Nutzung der vorhandenen Ressourcen (Konsolidierung)
- Automatisierung

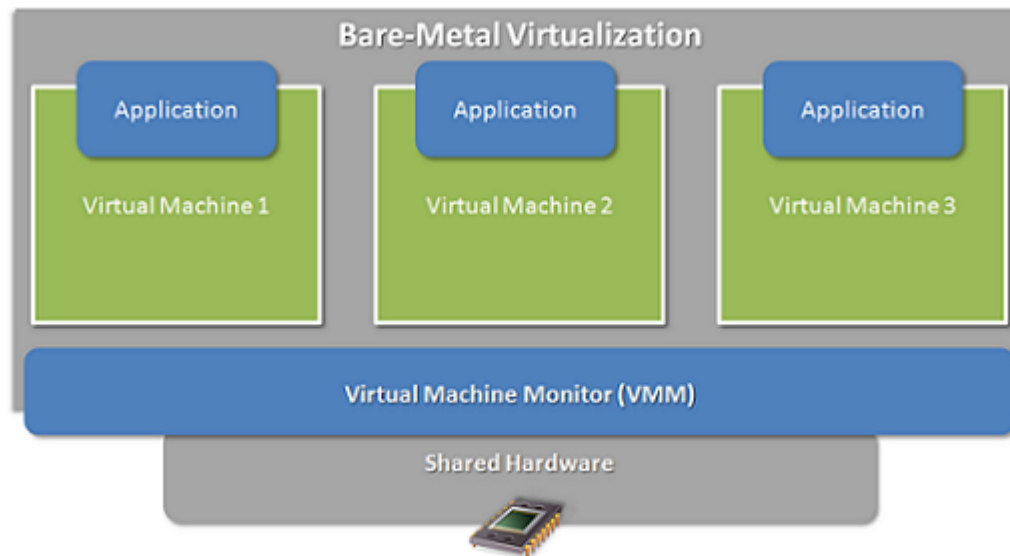
Virtualisierungsarten (1)

- Hosted-Vollvirtualisierung



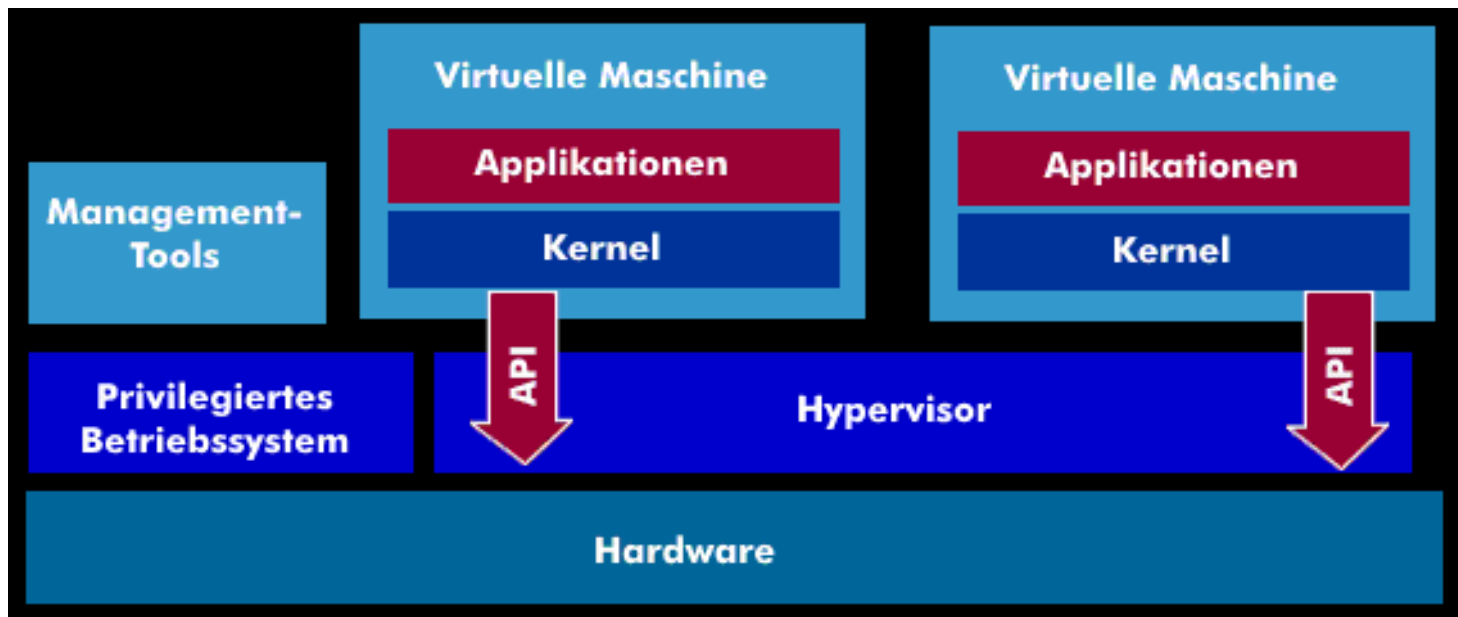
Virtualisierungsarten (2)

- Bare Metal-Vollvirtualisierung



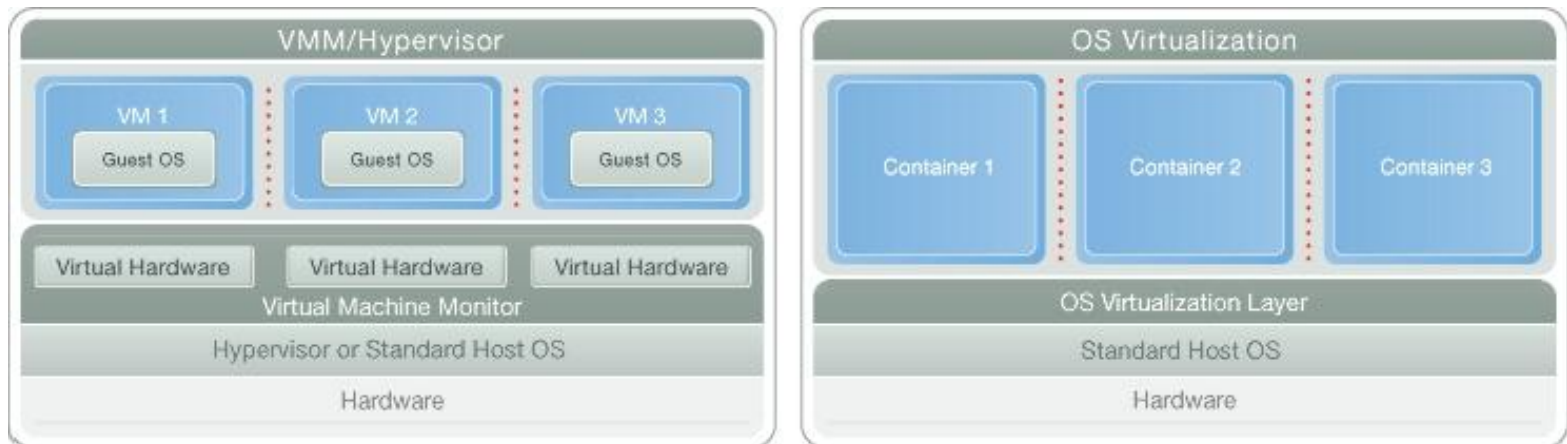
Virtualisierungsarten (3)

- Paravirtualisierung



Virtualisierungsarten (4)

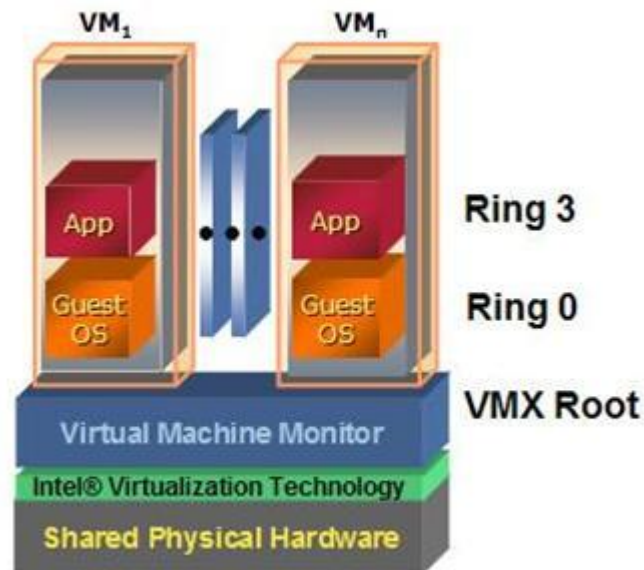
- „Virtualisierung“ auf Betriebssystemebene



Vergleich HV-Virtualisierung / OS-„Virtualisierung“

Virtualisierungsarten (5)

- Hardware-basierte Virtualisierung



Intel Virtualization Technology VT-x

Zusammenfassung Virtualisierungsarten (1)

- **Vollvirtualisierung**
 - Gastbetriebssystem ohne vorherige Anpassung lauffähig
 - Hardware nahezu vollständig virtualisiert
- **Paravirtualisierung**
 - Anpassung des Host- und Gastbetriebssystems
 - keine Emulation der Hardware

Zusammenfassung

Virtualisierungsarten (2)

- „Virtualisierung“ auf Betriebssystemebene
 - Partitionierung des vorhandenen Betriebssystems
 - modifizierter Host-Kernel zur Bereitstellung von virtuellen Einheiten
 - weniger flexibel, jedoch weitaus weniger Overhead

Zusammenfassung

Virtualisierungsarten (3)

- Hardware-basierte Vollvirtualisierung
 - Sonderform der Vollvirtualisierung
 - Intel VT-x und AMD-V
nicht kompatibel zueinander
 - **Trend:**
 - Erweiterung / Optimierung der Virtualisierungsfunktionen im Prozessor

Clustertheorie

- Definition
 - Der Zusammenschluss von Computersystemen über ein Netzwerk zu einem bestimmten Zweck wird als Cluster bezeichnet.
 - Die Mitglieder eines Clusters bezeichnet man als Knoten.

Clusterarten (1)

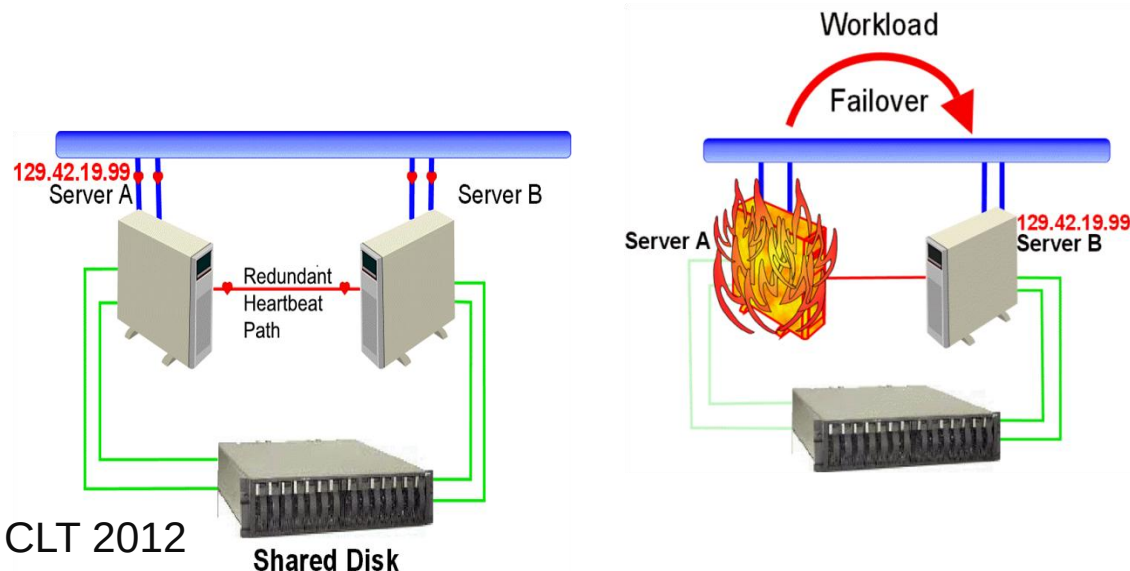
- Load-Balancing Cluster
 - Lastverteilung auf verschiedene Systeme
 - je nach Leistungsbedarf flexibel erweiterbar
- HA-Cluster
 - Verbesserung der Verfügbarkeit
 - Fokus: **Ausfall eines oder mehrerer Knoten**

Clusterarten (2)

- HPC-Cluster
 - Berechnung komplexer Probleme
 - hohe Rechenleistung
 - parallele / verteilte Abarbeitung einzelner Jobs
 - Anwendung in Forschung und Industrie
 - Kommunikation häufig über ein Hochgeschwindigkeits-Interconnect-Netzwerk (z. B. InfiniBand oder SCI)

HA-Clusterarten (1)

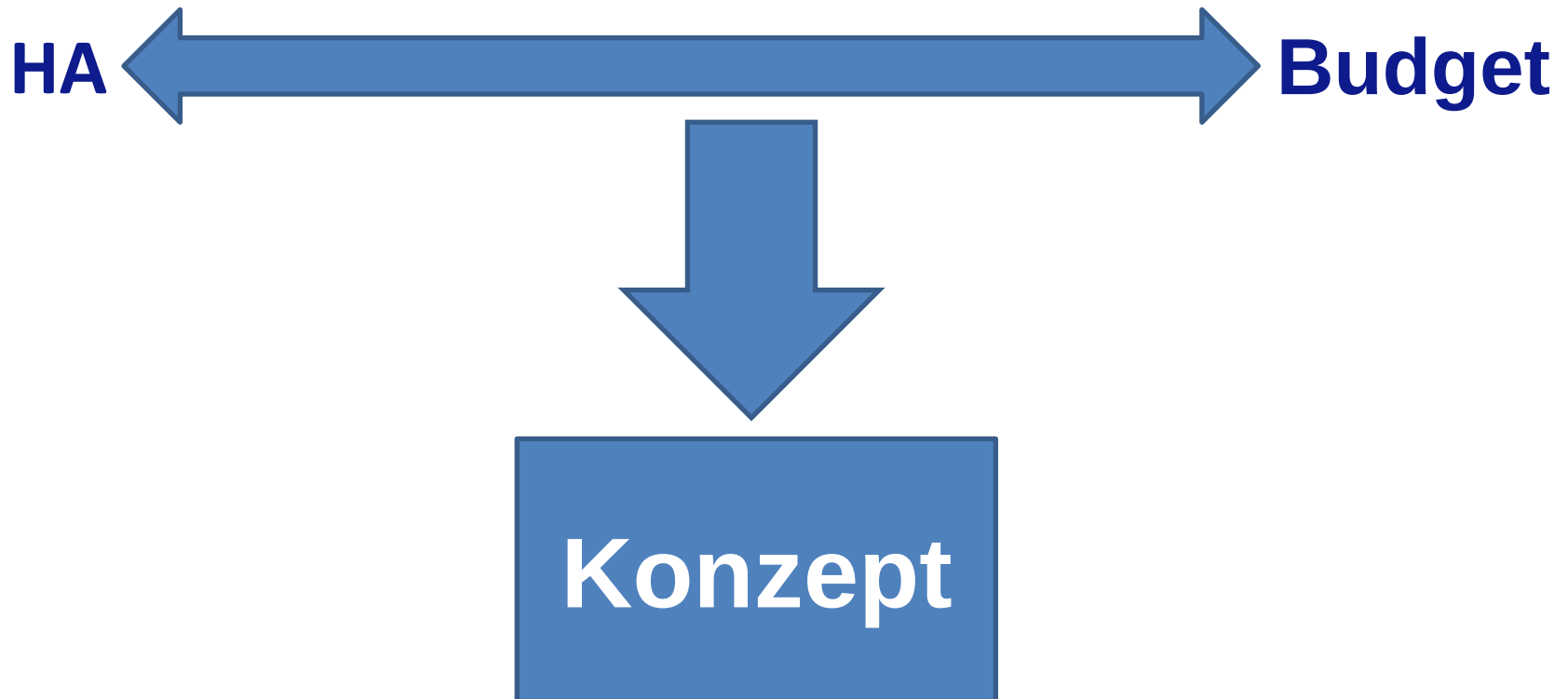
- Active- / Passive-Cluster
 - einfachste Clusterform bestehend aus zwei Knoten
 - Überwachung des Masters durch Slave per Heartbeat
 - bei Ausfall des Masters => Übernahme der Ressourcen



HA-Clusterarten (2)

- Active- / Active-Cluster
 - Bereitstellung verschiedener Dienste
 - gegenseitige Überwachung dieser
- M-to-N-Cluster
 - Verteilung einer Anzahl von m Ressourcen auf n Knoten
 - zum Teil (hoch)-komplexe Bedingungen zur Ressourcenverteilung

Anforderungen an Hochverfügbare Systeme (1)



Anforderungen an Hochverfügbare Systeme (2)

- Welche Verfügbarkeit soll erreicht werden?
- Auf welcher Ebene?
 - Hardware (fehlertolerant)
 - Software (Applikationscluster)
- Welche Kosten entstehen bei Ausfall des Dienstes?
- Wie wird der Zugriff auf dynamische Daten realisiert?
 - Replikation
 - RAID-Controller für Cluster
 - Shared Storage (SAN, NAS)
- Wie wird die Funktionalität des Dienstes gewährleistet?

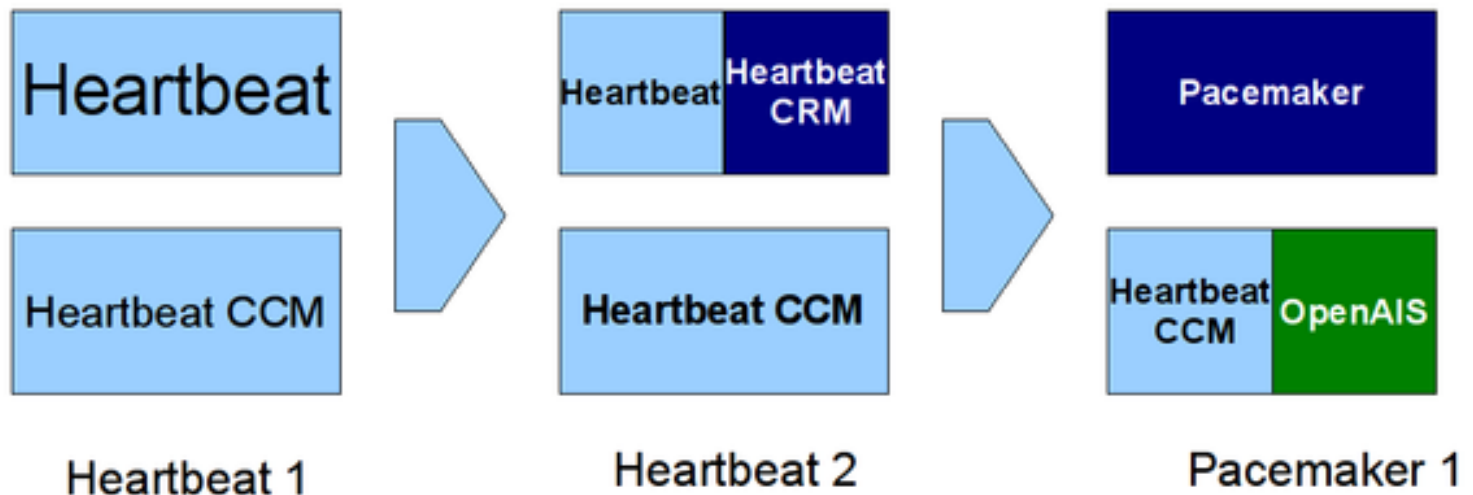
Linux-HA / Pacemaker (1)

- plattformunabhängige OpenSource-Lösung
- Verwaltung von Ressourcen durch Cluster Manager
- nahezu alle Dienste hochverfügbar realisierbar
- mittlerweile neuer Cluster Manager Pacemaker mit eigenem Kommunikationsstack OpenAIS / Corosync



Linux-HA / Pacemaker (2)

- Projektentwicklung



Linux-HA / Pacemaker (3)

Vergleich Funktionsumfang

Linux-HA Version 1

- maximal 2 Knoten
- nur Active- / Passive-Cluster
- Text-basierte Konfiguration
- monolithischer Aufbau
- Überwachung des Zustands der Knoten

Linux-HA Version 2

- bis zu 16 Knoten
- jede HA-Clusterart
- XML-basierte Konfiguration
- modularer Aufbau
- Überwachung des Zustands der Dienste

HA-Konfigurationen Ideen & Konzepte (1)

- Ursprünglicher Ansatz
 - Kombination der Virtualisierung mit Linux-HA Version 2
 - Problem: **dynamische Ressourcenverwaltung**
 - exponentiell ansteigende Komplexität führt zur Unbeherrschbarkeit des Gesamtsystems
 - bis vor kurzem unklare Projektstruktur bei Linux-HA Version 2

HA-Konfigurationen

Ideen & Konzepte (2)

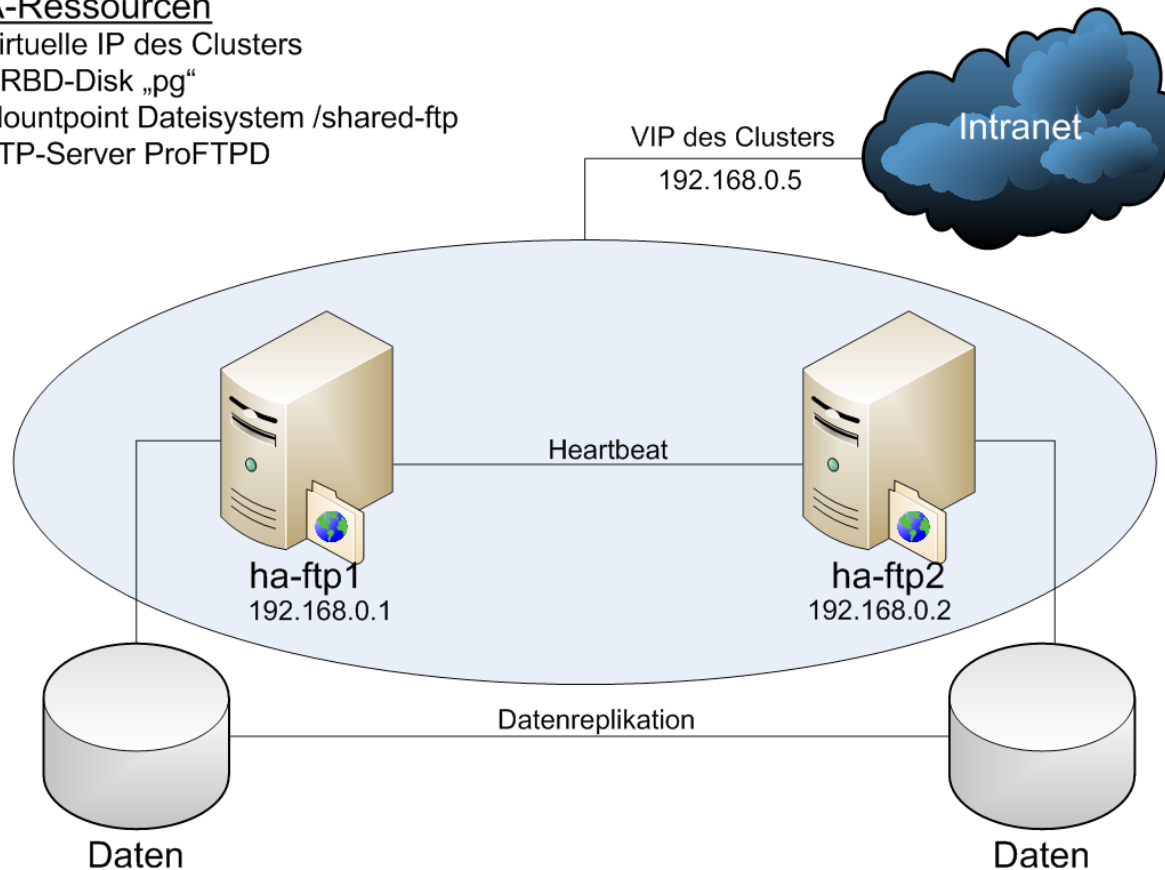
- Praktische Umsetzung
 - **KISS-Prinzip** („Keep It Simple, Stupid!“)
 - Active- / Passive-Cluster mit Linux-HA Version 1
=> überschaubare Konfiguration
 - Einsatz der Virtualisierungsfunktionen
 - dynamische Skalierbarkeit mittels Linux Virtual Server

HA-Konfigurationen (1)

FTP-HA-Cluster mit DRBD

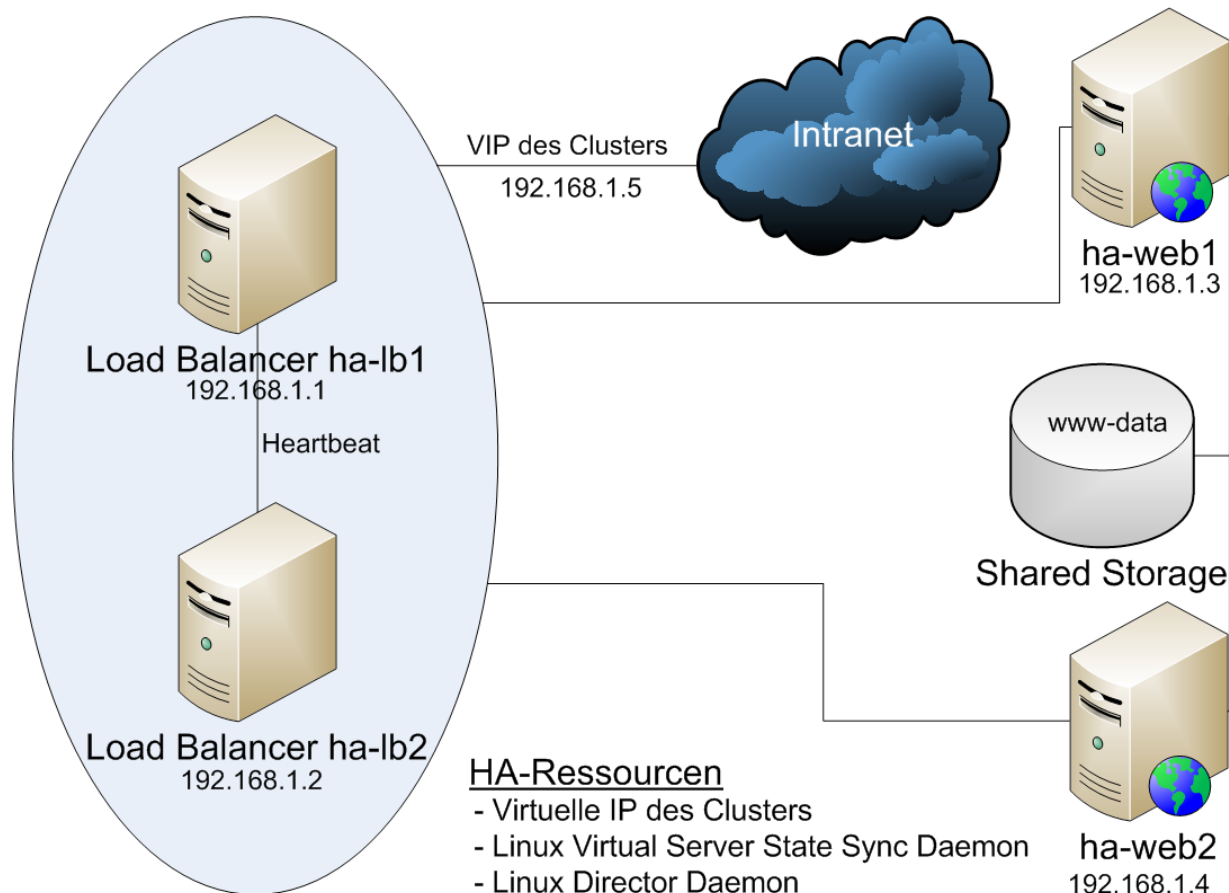
HA-Ressourcen

- Virtuelle IP des Clusters
- DRBD-Disk „pg“
- Mountpoint Dateisystem /shared-ftp
- FTP-Server ProFTPD



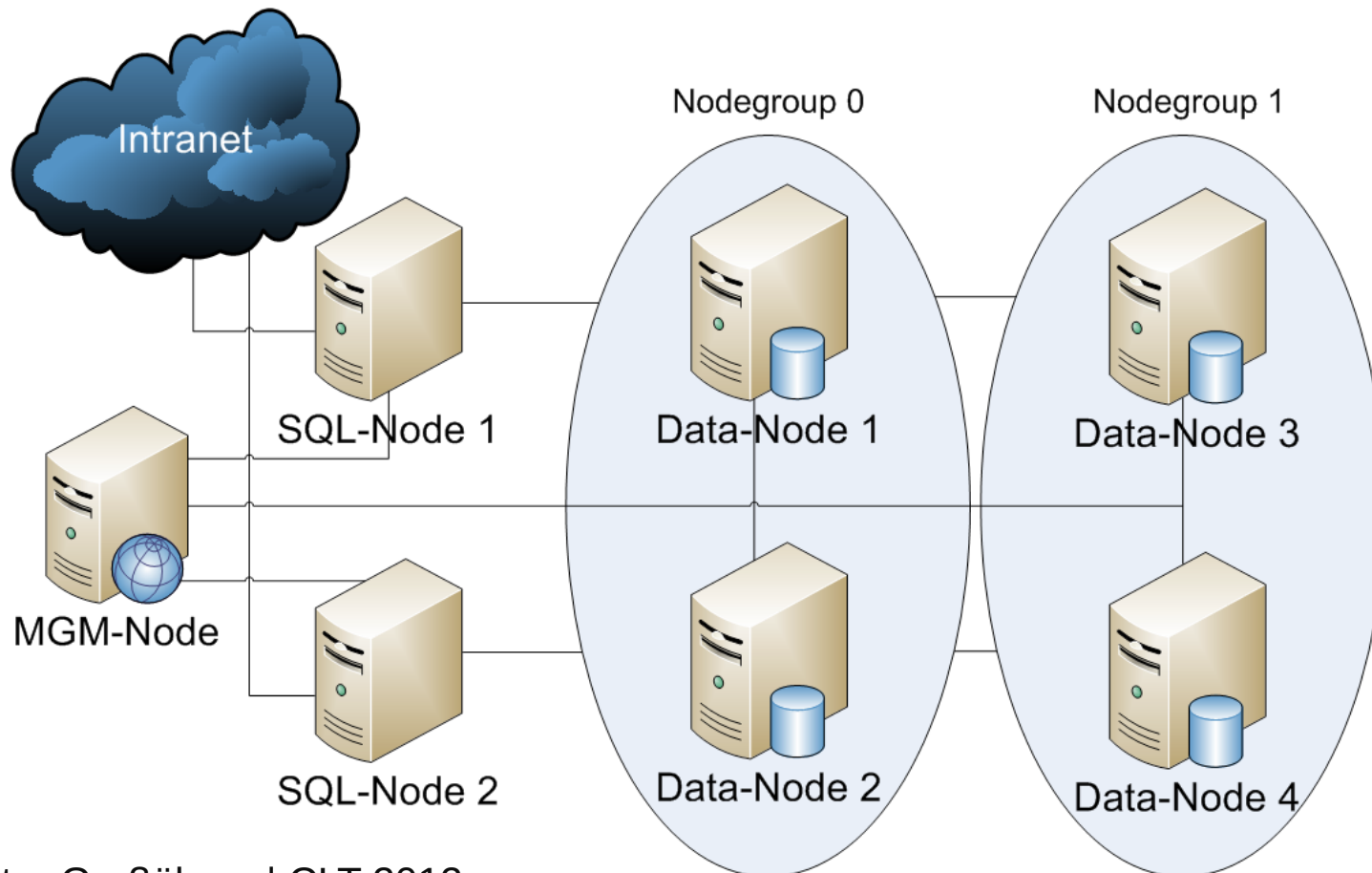
HA-Konfigurationen (2)

Webserver-HA-Cluster mit Load Balancer



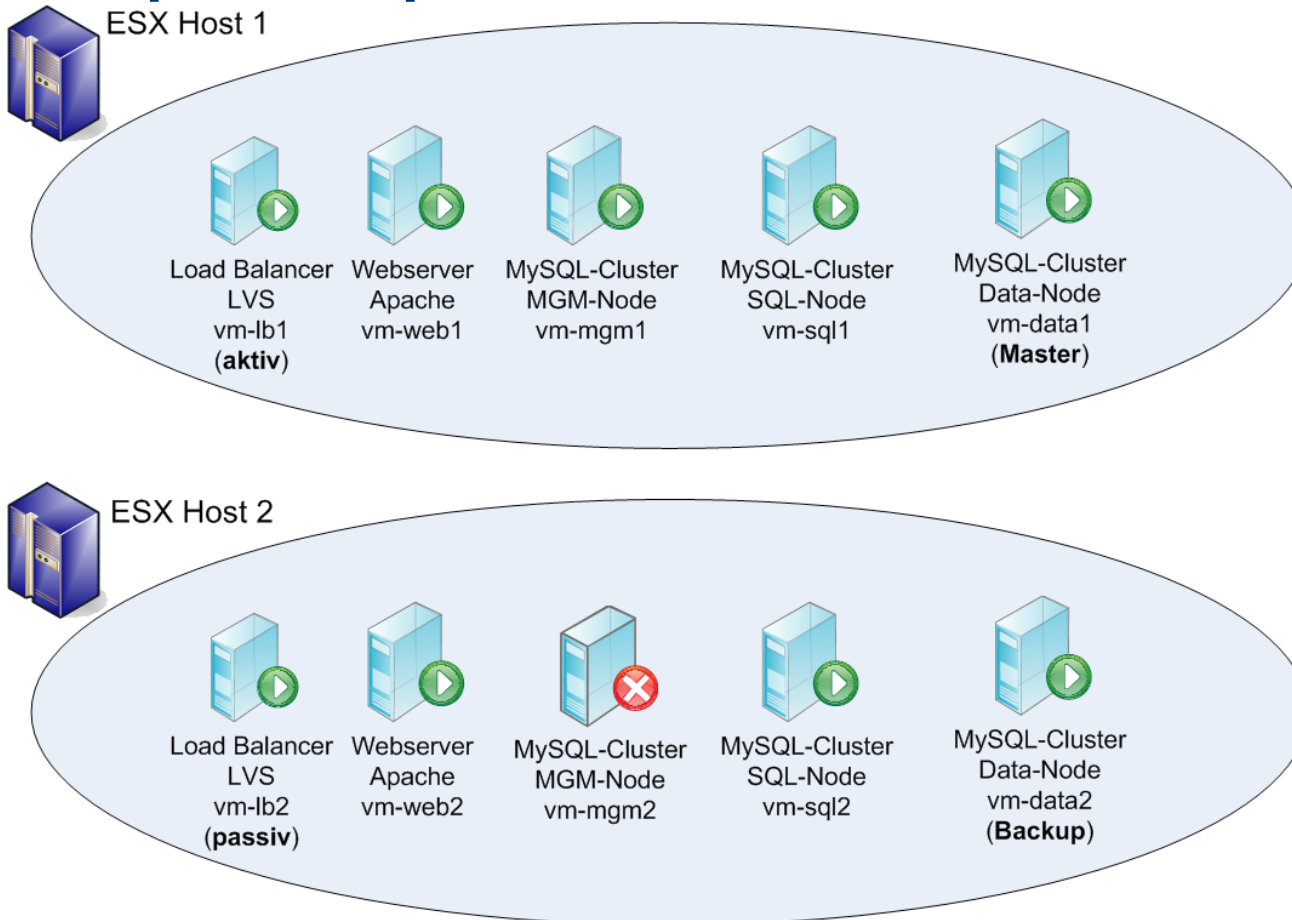
HA-Konfigurationen (3)

MySQL-Cluster



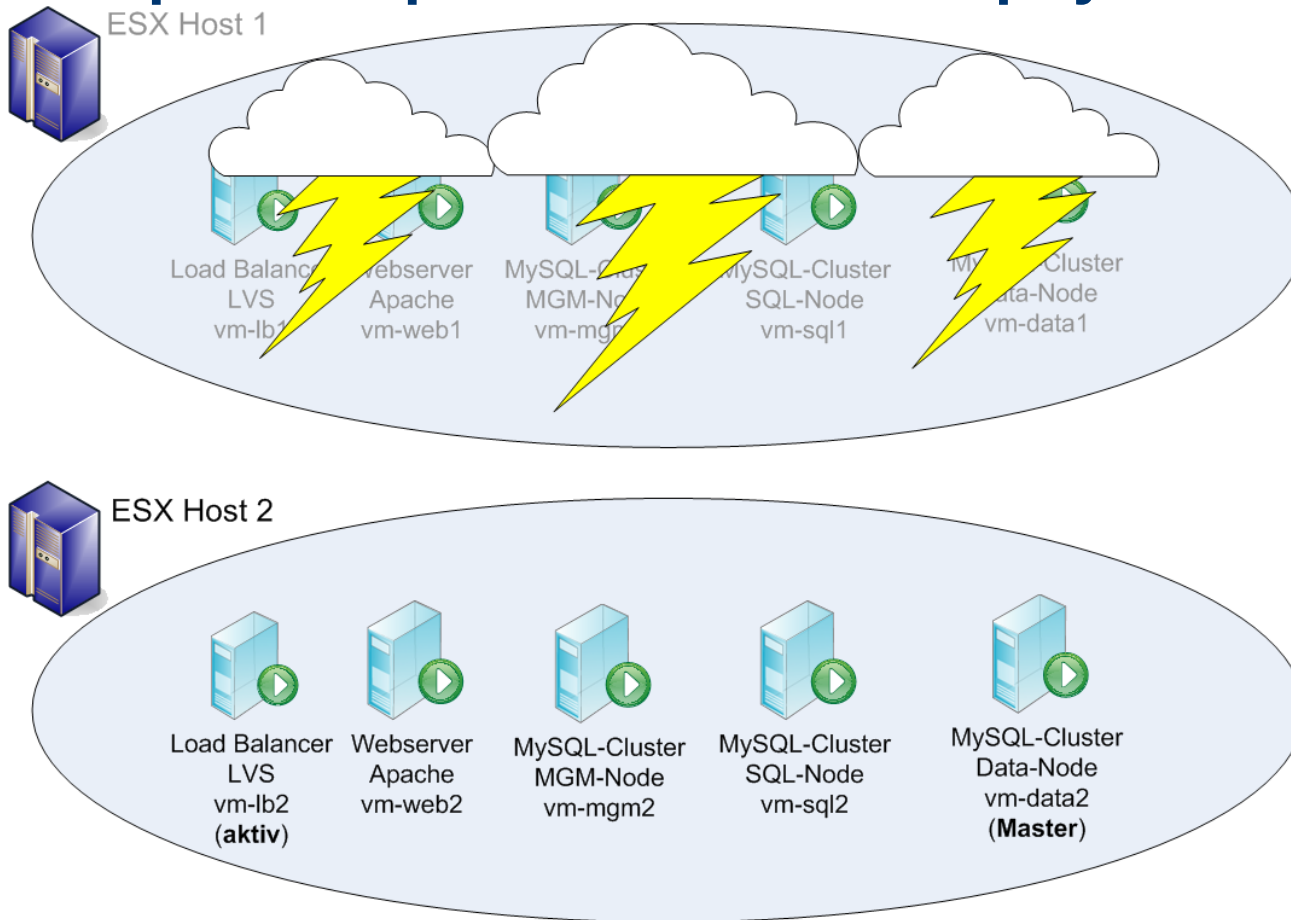
HA-Konfigurationen (4)

Komplexbeispiel



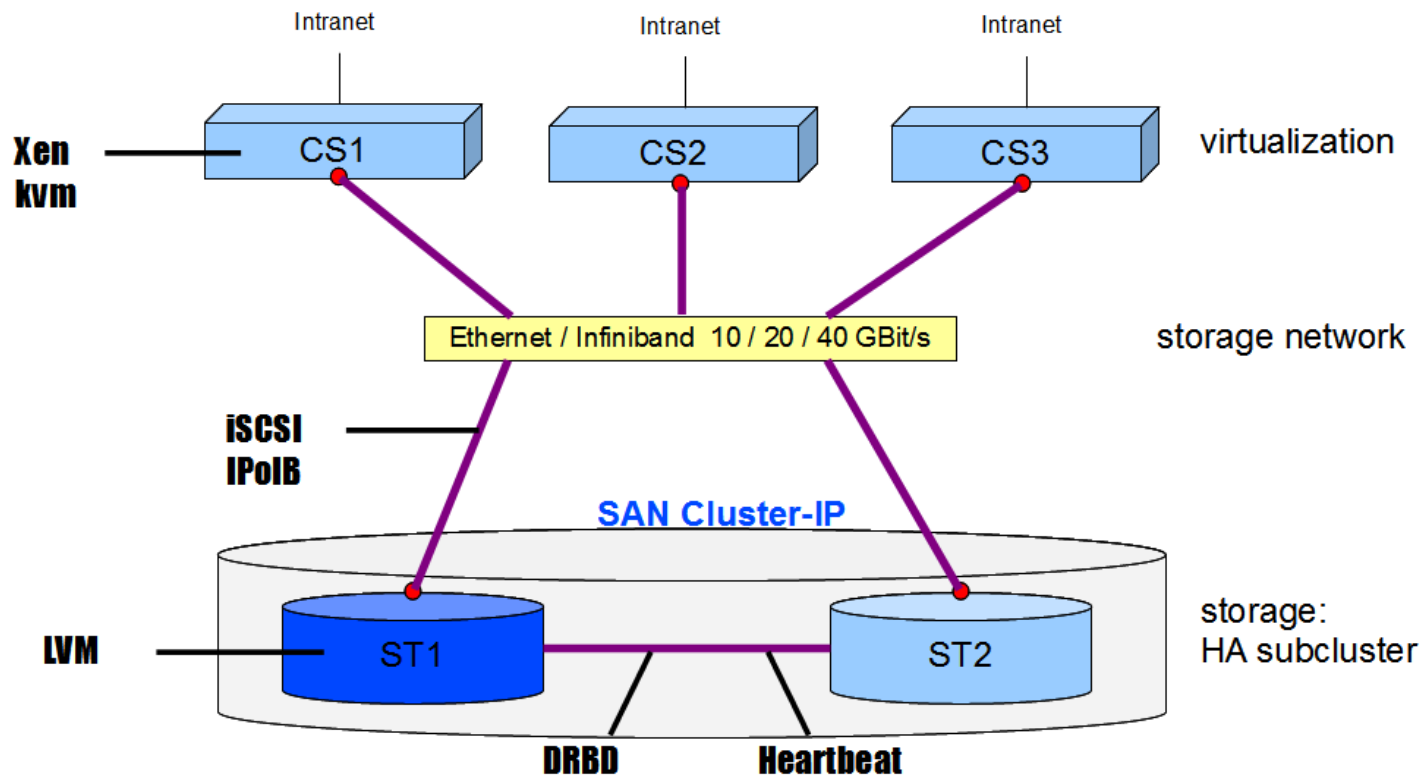
HA-Konfigurationen (5)

Komplexbeispiel – Ausfall eines physischen Hosts



HA-Konfigurationen (6)

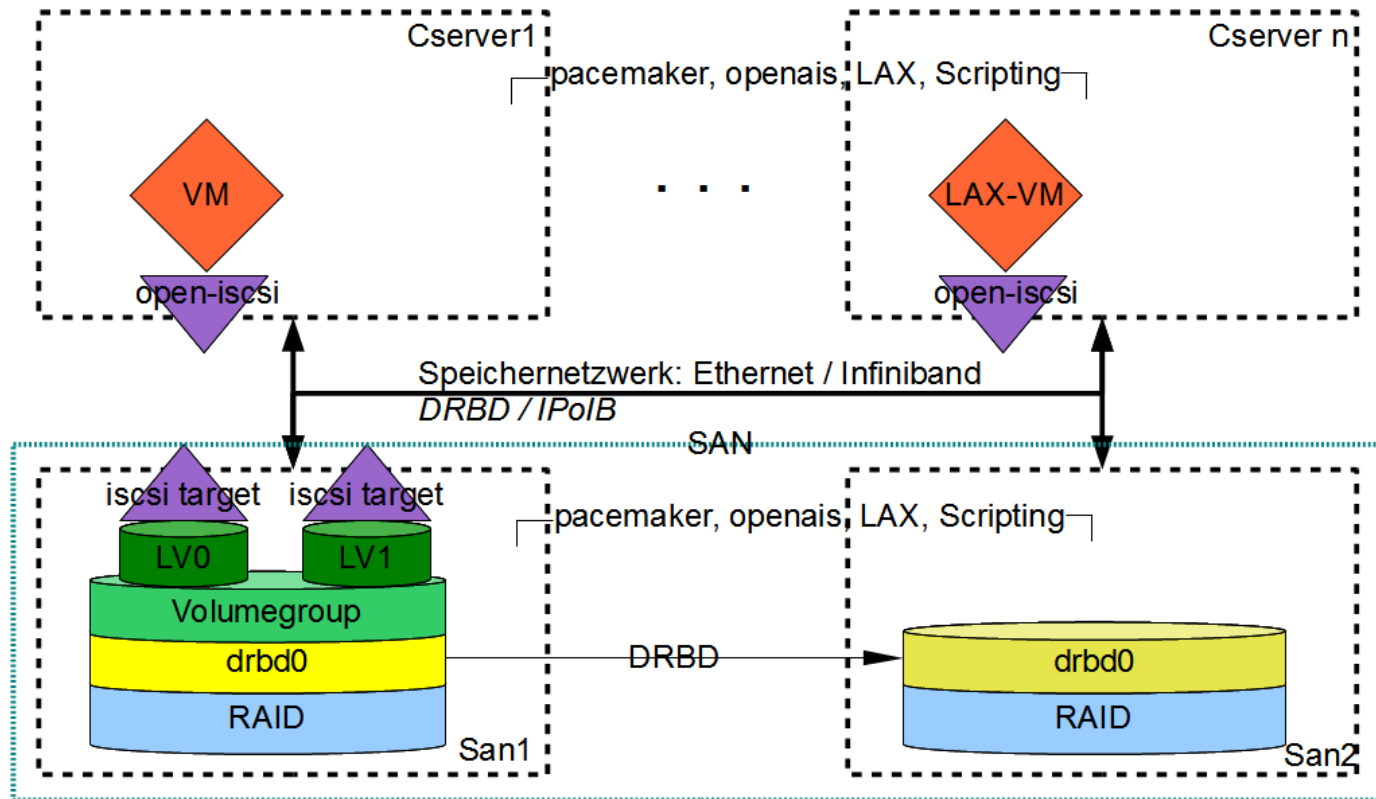
HA-Virtualisierungscluster mit iSCSI SAN (Hardware)



Quelle: teegee – Thomas Groß

HA-Konfigurationen (7)

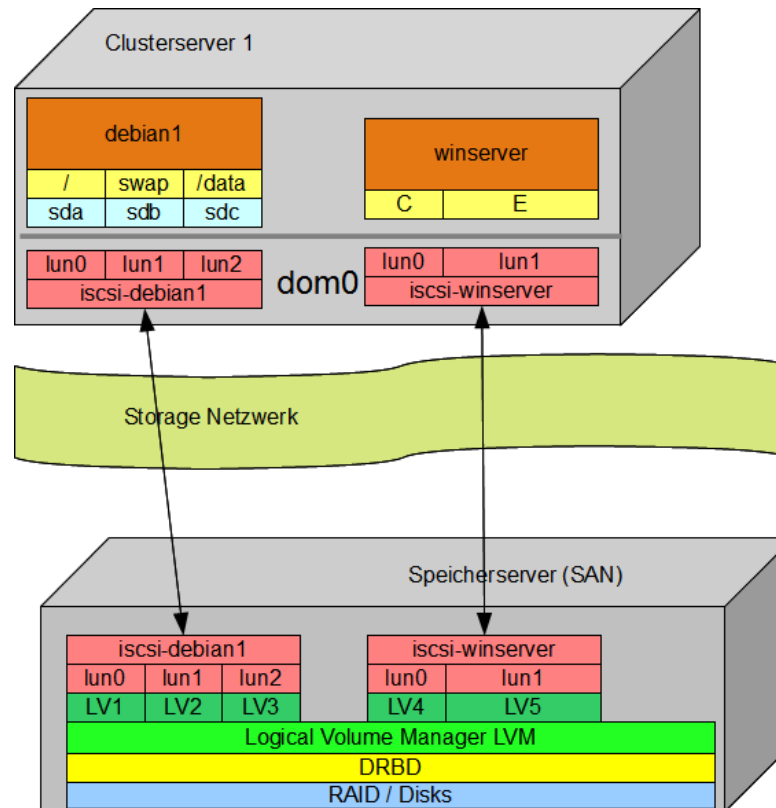
HA-Virtualisierungscluster mit iSCSI SAN (Software)



Quelle: teegee – Thomas Groß

HA-Konfigurationen (8)

HA-Virtualisierungscluster mit iSCSI SAN (Software)



Quelle: teegee – Thomas Groß

HA-Konfigurationen (9)

HA-Virtualisierungscluster mit iSCSI SAN (Mgmt)

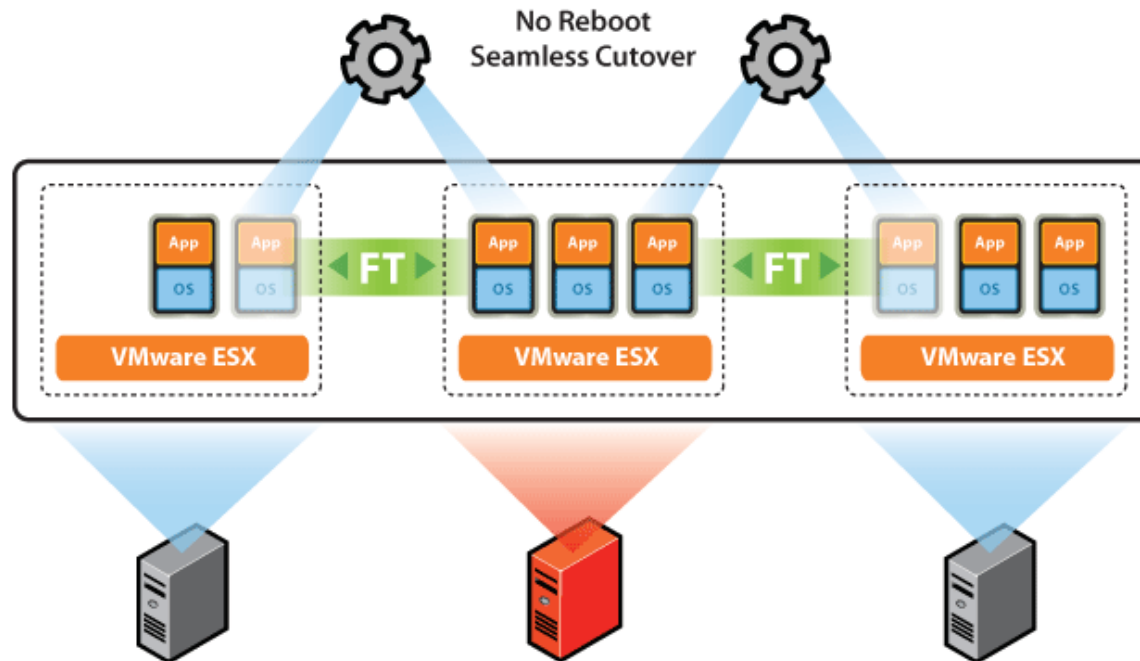
- Clustermanagement mittels LAX
 - Funktionsumfang
 - HA-Virtualisierungscluster Overview
 - VM-Verwaltung
 - VM-Backup
 - Monitoring / Alarming
 - Fehlerbehandlung
 - <http://www.teegee.de/lax>

Zusammenfassung

- Kombination der Cluster- und Virtualisierungsfunktionen
Ziel: hoher Automatisierungsgrad
- Einsatz von OpenSource-Lösungen, wie z.B. Linux-HA;
effiziente Ressourcennutzung durch Virtualisierung
Ziel: Beachtung ökonomischer Randbedingungen
- Beschränkung auf Active- / Passive-Cluster (mit LVS)
Ziel: Beherrschbarkeit des Gesamtsystems

Trends und Entwicklungen

Fault Tolerance (1)



Fault Tolerance (FT) im Einsatz

Trends und Entwicklungen

Fault Tolerance (2)

- Voraussetzungen / Einschränkungen
 - Hardware-Virtualisierung (Intel VT-x, AMD-V)
 - nur spezielle Intel- bzw. AMD-Prozessorreihen
 - Gigabit Ethernet (besser: 10 Gigabit Ethernet bei mehreren FT VMs)
 - gleiche ESXi Version und Patch Level der physischen Hosts im Cluster
 - VMs müssen sich auf einem Shared Storage befinden
 - Fibre Channel
 - NAS
 - iSCSI
 - thick provisioned VMDK („Festplatte“ der VM)
 - Verwendung eines FT unterstützten Betriebssystems
 - nur eine virtuelle CPU im jeweiligen Gastsystem

Trends und Entwicklungen

Fault Tolerance (3)

- Funktionsweise
 - **Ansatz:** „Active- / Passive-Cluster“
 - Live-Schatteninstanz der primären VM auf anderen physischen Host
 - Synchronisation aller Operationen (CPU-Instruktionen / Festplatten- und Netzwerkzugriffe) der primären VM über das Netzwerk an die Sekundäre
 - Zugriff auf gemeinsame „virtuelle“ Festplatte
 - Transparenz für Applikationen mittels einer IP- und MAC-Adresse
 - Heartbeat zwischen primärer und sekundärer FT VM
 - bei Ausbleiben des Heartbeats (im Millisekunden-Bereich) übernimmt die jeweils andere VM sämtliche Aufgaben ohne Verlust von Daten

Trends und Entwicklungen

Fault Tolerance (4)

- Implementierungen
 - kostenpflichtig
 - VMware Fault Tolerance
 - Marathon everRun VM für Citrix XenServer
 - „frei“
 - Remus für Xen
 - <http://wiki.xen.org/xenwiki/Remus>
 - Not tested yet! 😊

Trends und Entwicklungen

weitere VMware Technologien

- VMotion
 - Live-Migration virtueller Maschinen auf andere physische Hosts
- Storage VMotion
 - Live-Migration der Festplattendateien virtueller Maschinen über heterogene Storage-Arrays hinweg
- Storage I/O Control
 - Priorität des Zugriffs auf Storage-Ressourcen
- Hot Add
 - dynamische Zuweisung von Ressourcen (CPU, RAM) ohne Unterbrechung und Neustart des Betriebssystems

Trends und Entwicklungen

Ausblick / Ideen & Visionen (1)

- Desktop-Virtualisierung (z. B. VMware View)
- OCF-Agenten in Linux-HA zum Betrieb hochverfügbarer virtueller Maschinen
=> Verschmelzung beider Konzepte
- Cloud-Computing (z. B. OpenStack)
- Kernel Based Virtual Machine (KVM) im Trend
 - Red Hat Enterprise Virtualization (RHEV) als Basis einer Virtualization Management-Lösung für Server- und Desktop-Virtualisierung

Trends und Entwicklungen

Ausblick / Ideen & Visionen (2)

- Einsatz von Virtualisierung in HPC-Umgebungen
 - Betrieb des gesamten HPC-Clusters in Form von virtuellen Maschinen über das Queueing System
 - komplette Verlagerung des High-Performance-Computings in die Cloud als Software as a Service
 - Basis der Managementsoftware: OpenStack
 - Kombination von High-Performance- und Cloud-Computing

Vielen Dank! Fragen?

**MEGWARE Computer
Vertrieb und Service GmbH**

Nordstraße 19
09247 Chemnitz-Röhrsdorf

Telefon: +49 (3722) 528-0
Telefax: +49 (3722) 528-15

www.megware.com

Peter Großöhme

peter.grossoehme@megware.com

Telefon: +49 (3722) 528-45