TSBE Bern
Höhere Fachschule

Virtuelle Systeme – Verfügbarkeit & Clustering

FS-2018

Christoph Bühlmann

Agenda

1. Einleitung
2. Verfügbarkeit
3. Cluster
4. Gruppenarbeit Clientvirtualisierung
5. Hands on

Einleitung - Definition

Was

- Verschiedene Systeme stellen eine Funktion bereit
- Transparent von aussen (Black Box)

Warum

- Verschiedene Cluster – Verschiedene Gründe
  - Hochverfügbarkeit (HA)
  - Load Balancing
  - High Performance
- Vorteile
  - Geschwindigkeit
  - Mehr Last kann bewältigt werden
  - Skalierbarkeit (je nach Model)

Einleitung - Verfügbarkeit

Prozentsatz der geforderten Betriebszeit während denen das System normal funktioniert

- Beispiele: 99%, 99.99%

Betriebszeit

- Zeitplan während dem das System normal verfügbar sein muss
- Beispiele: Montag bis Freitag von 0800 bis 1800 oder 24h x 365d

Ausfallzeit

- Zeit während das System nicht verfügbar ist
- Ausfall kann beabsichtigt sein (Wartung) oder nicht (Incident)

Verfügbarkeit

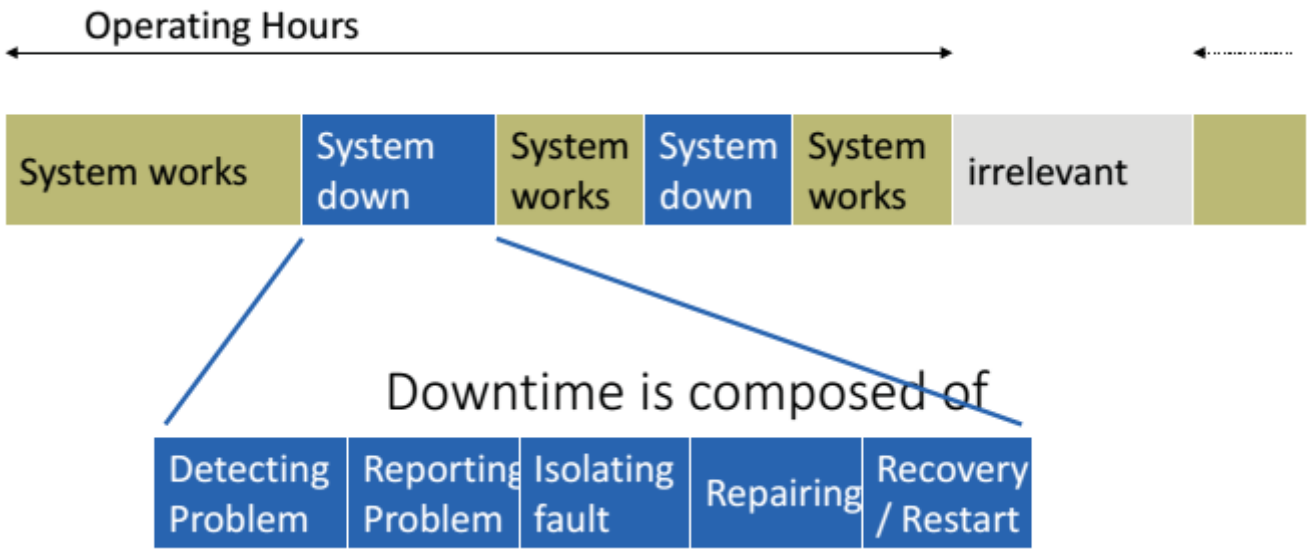
Beispiel für 24h x 365d

Verfügbarkeit %	Minimal Uptime [h:m]	Maximal Downtime [h:m:s]
99	8648:38	87:21:36
99.9	8727:15	8:44:10
99.99	8735:07	0:52:25
99.999	8735:54	0:05:14

Beispiel für 24h x 6d x 12m

Verfügbarkeit %	Minimal Uptime [h:m]	Maximal Downtime [h:m:s]	Rest [h]
99	3706:33	37:26:24	4992
99.9	3740:15	3:44:38	4992
99.99	3743:37	0:22:28	4992
99.999	3743:57	0:02:15	4992

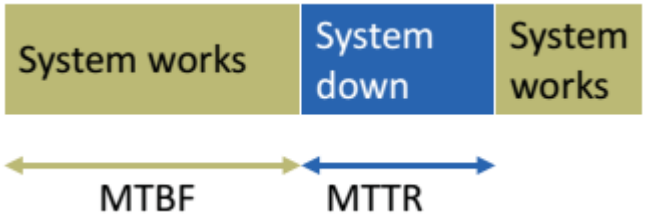
Verfügbarkeit



Verfügbarkeit

Statistische Angaben

- MTBF Mean Time Between Failure
- MTTR Mean Time To Repair



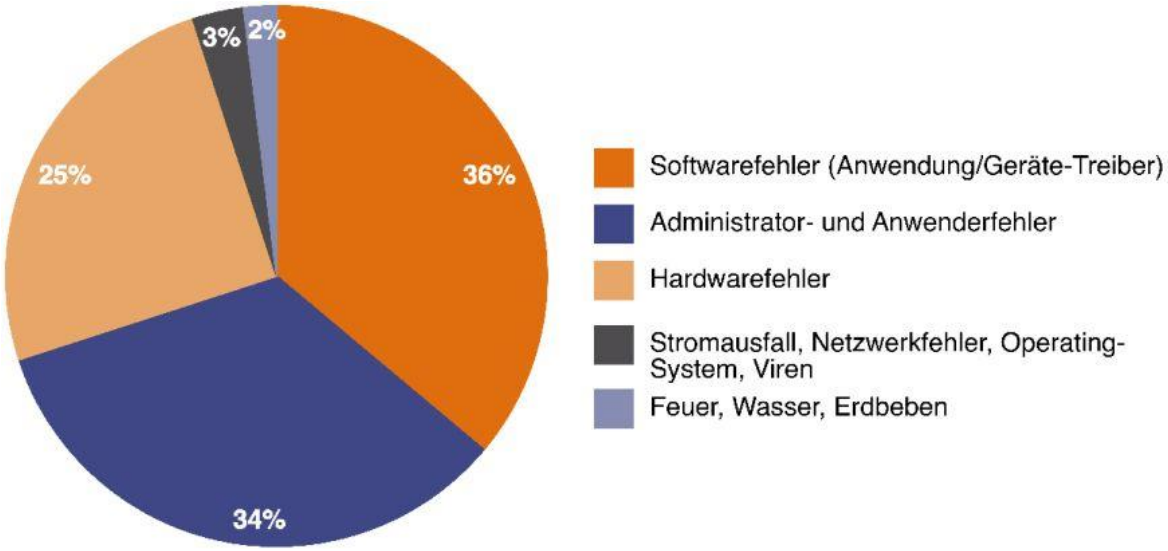
Berechnung

- $\text{Verfügbarkeit} = \frac{MTBF}{MTBF+MTTR} = \frac{Betriebsbereitschaft - Downtime}{Betriebsbereitschaft}$

Verfügbarkeit - Strategien

Redundanz

- Was ist das billigste System um die geforderte Verfügbarkeit zu erreichen
- Der Schlüssel zu höherer HW-Verfügbarkeit ist Redundanz

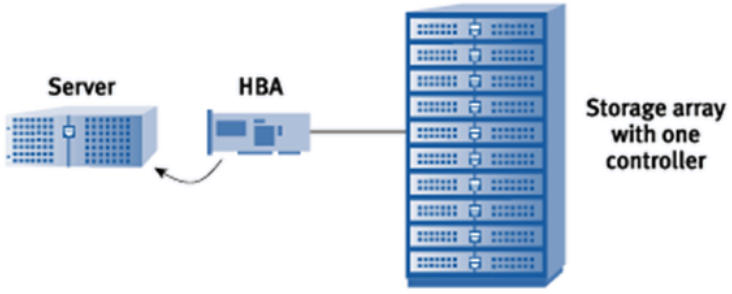


Achtung

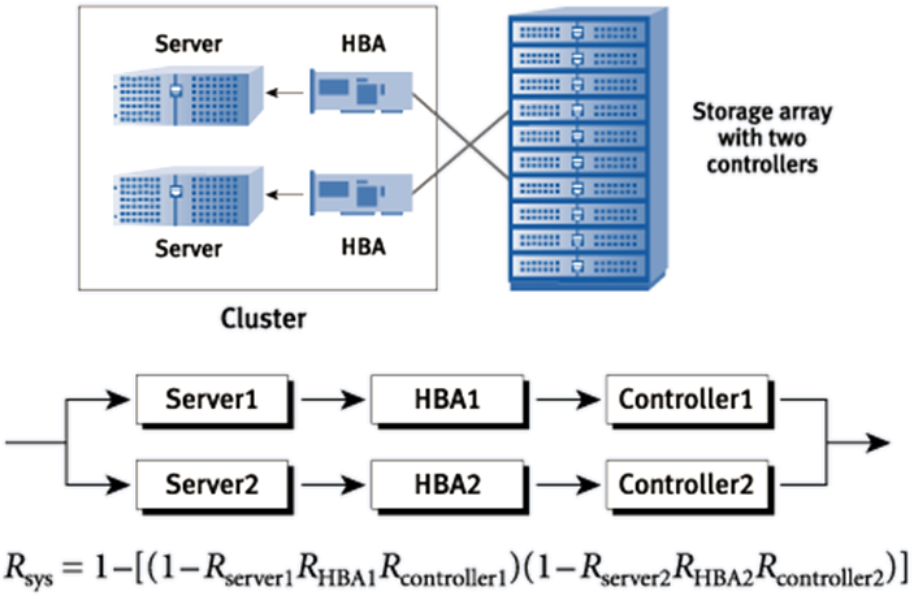
- Nur zwischen 15% und 25% aller Ausfälle sind Hardwareausfälle
- Die Verfügbarkeit kann in der Realität (fast) nicht durch Hardwaremassnahmen alleine gesteigert werden!
- Redundanz macht das System komplexer (~34% der Ausfälle durch Admins)

Verfügbarkeit - Strategien

Berechnung Normal- / Parallelbetrieb



$$R_{sys} = R_{server} R_{HBA} R_{controller}$$

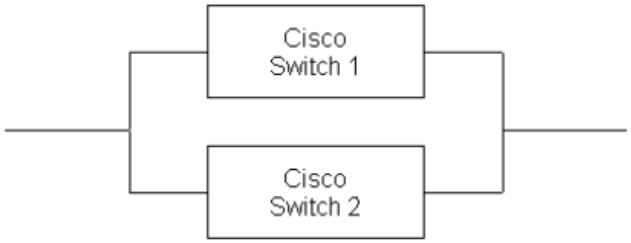


$$R_{sys} = 1 - [(1 - R_{server1} R_{HBA1} R_{controller1})(1 - R_{server2} R_{HBA2} R_{controller2})]$$

Verfügbarkeit - Strategien

Beispiel

- Redundante Ethernet-Switche
- Cisco weisst eine MTBF von 200’000h aus
- Unsere Prozesse lassen eine MTTR von 5h zu



Berechnung

- $\text{Verfügbarkeit} = \frac{400'000h}{400'000+5h} = 0.9999875 \rightarrow 99.99875\%$

Verfügbarkeit – Single Points of Failures

Single Points of Failure

- Welche (Teil)-komponenten eines Systemes kommen als Single-Point-of-Failure in Frage

Failure Point	Possible solutions
Disk	Disk mirroring, RAID-5
Network Service (DNS etc.)	Multiple DNS services
Power Outage	UPS
NIC	Multiple NICs in a Host
Hub	Multiple interconnected network paths
OS / SW crash	Clustering, switching to healthy node
Firewall	Firewall Cluster or high-availability Firewall

Verfügbarkeit – Applikation

Was kann / muss die Applikation beitragen

- Lose Kopplung bzw. Entkopplung einzelner Komponenten (Fehlertoleranz)
- Clusterfähigkeit
  - Replikation / Load Balancing
  - Failover

Was kann / muss der Mensch (Administrator) beitragen

- Hohe HW-Verfügbarkeit
- Infrastructure as Code, manuelle Eingriffe vermeiden
- Sauberes (und reaktives) Staging
- Aktives Monitoring

Clusterdefinition

Shared Nothing (Verbund)

Replizierte Systeme

Shared Disk

Failover

Failover

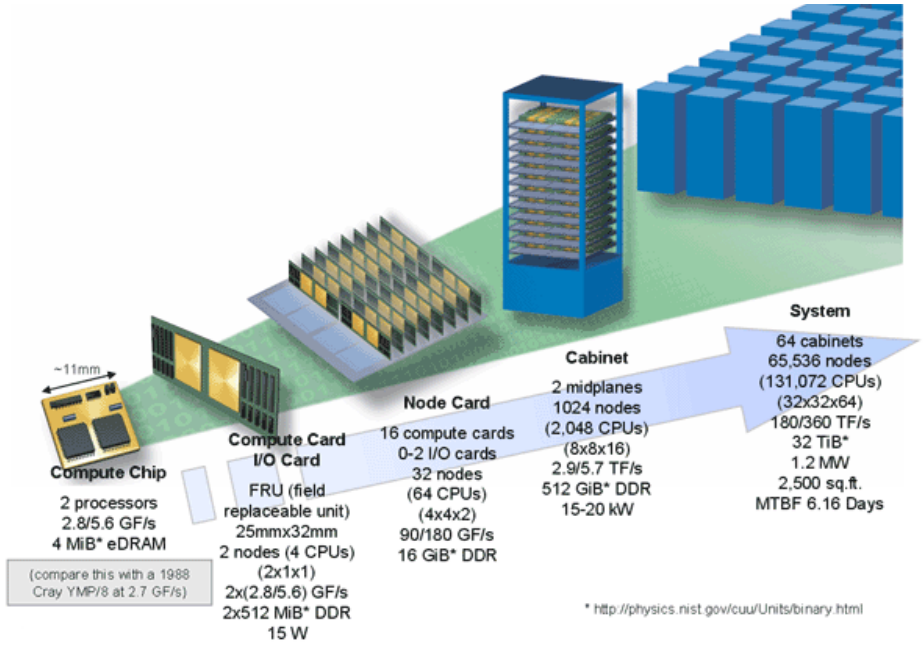
- Aktiv / Aktiv
- Aktiv / Passiv

Shared Everything

Cluster – Shared Nothing (Verbund)

Merkmahle

- Eine Storage-Instanz pro Rechner / CPU
- Daten sind über n Nodes verteilt und allenfalls auf Applikatinslevel repliziert



Cluster – Shared Nothing (Verbund)

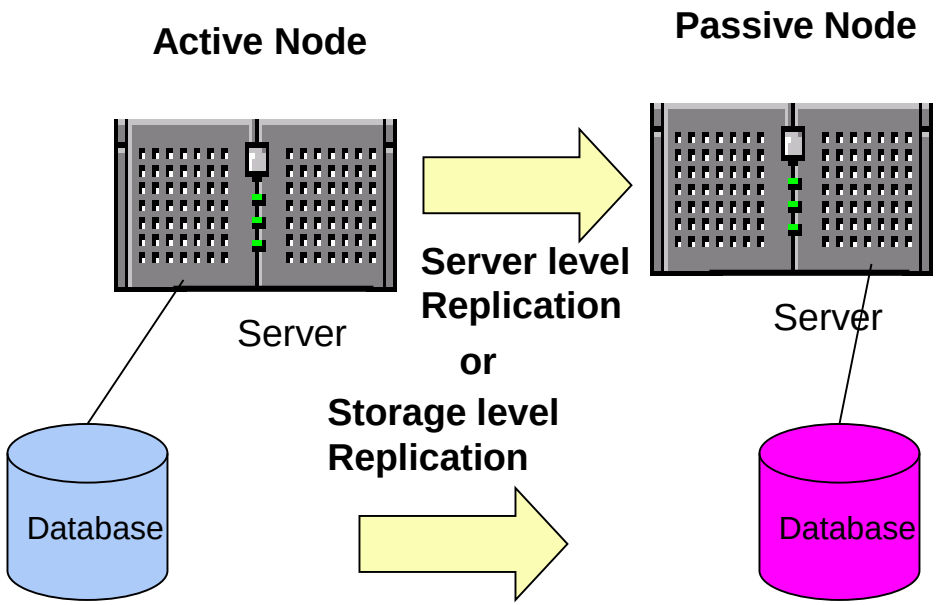
Ausprägung

- Massive Parallel Computing (MPC) – IBM Grossrechner
- Gluster Storage (applikatorisch repliziertes FS)
- Datawarehouses / BigData – Hadoop

Apache Hadoop ist ein freies, in Java geschriebenes Framework für skalierbare, verteilt arbeitende Software. Es basiert auf dem MapReduce-Algorithmus von Google Inc. sowie auf Vorschlägen des Google-Dateisystemsund ermöglicht es, intensive Rechenprozesse mit großen Datenmengen (Big Data, Petabyte-Bereich) auf Computerclustern durchzuführen. (wikipedia)

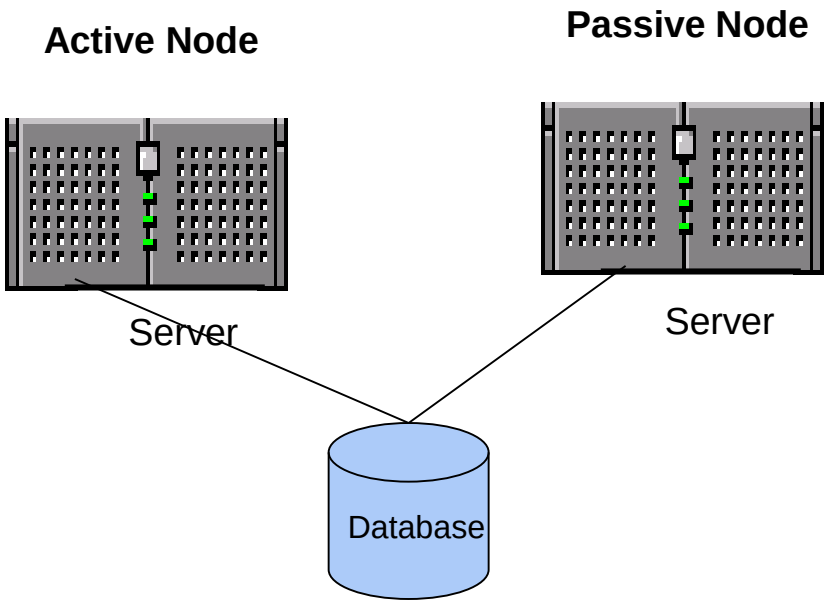
Cluster – Replizierte Systeme

- Daten werden repliziert
  - Auf Server-Level (Netzwerk, NAS)
  - Auf Storage (SAN) Level
- Daher sind mehrere Kopien des gleichen Datensatzes vorhanden
- Meistens ist die Implementation Aktiv/Passiv (mehr später)
- Zwischen den Nodes wird ein Failover realisiert



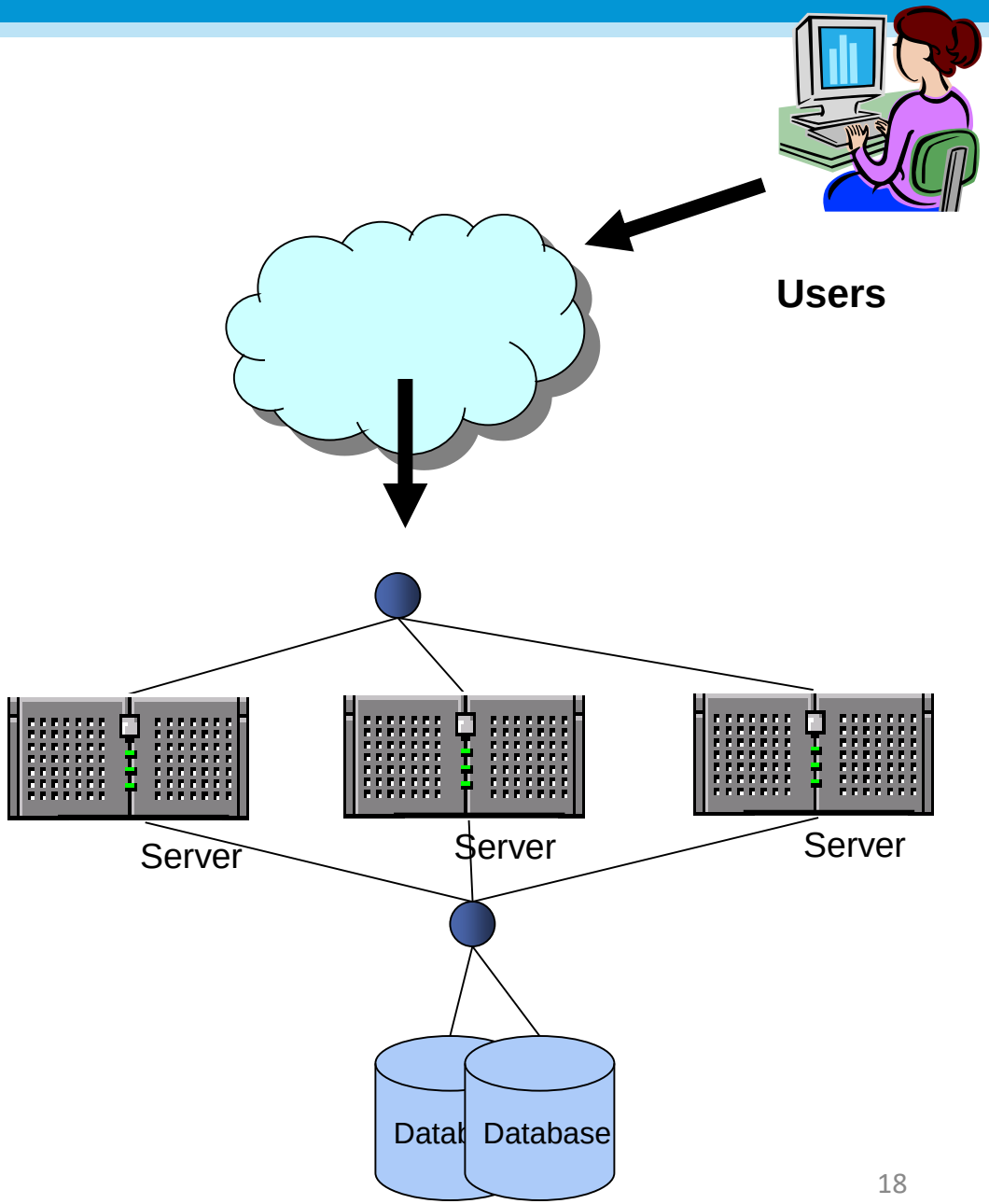
Cluster – Shared Disk

- File-System geteilt (Es muss Cluster-Aware sein) -> Mehrere Systeme haben Zugriff auf ein Filesystem
- Somit haben alle Nodes die gleichen Daten
- Eine Node hat «ownership» des Datensatzes
- Fällt ein System aus kann das zweite übernehmen
- Kein aufwändiges Synchronisieren des Filesystem



Cluster – Failover

- Einfachste Form für Redundanz
- Ermöglicht High Availability (HA)
- Keine Applikatorische Unterstützung nötig
- Normalerweise wird ein Failover nur über 2 Nodes realisiert.
- Bei einem Serverausfall werden die betroffenen VM's auf einem andern Server neu gestartet



Cluster – Failover

Aktiv/Passiv

- Eine Node ist Aktiv
- Die andere ist solange Passiv, bis ein Failover auftritt

Aktiv/Aktiv

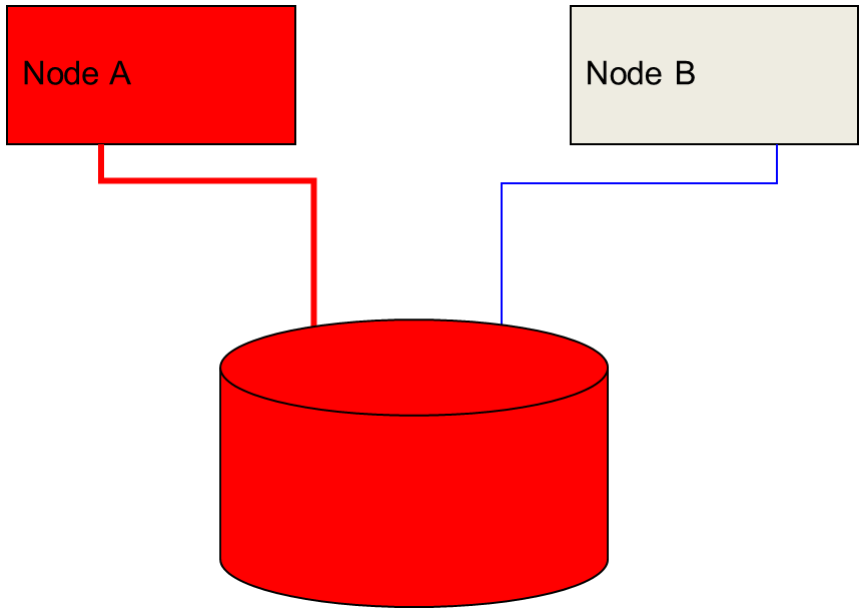
- Technologie gleich wie bei Aktiv/Passiv
- Jedoch sind beide Systeme produktiv
- 2 verschiedene Systeme mit gegenseitiger Failoverbereitschaft

Beide Varianten sind nur Failover!

Cluster – Failover

Aktiv/Passiv

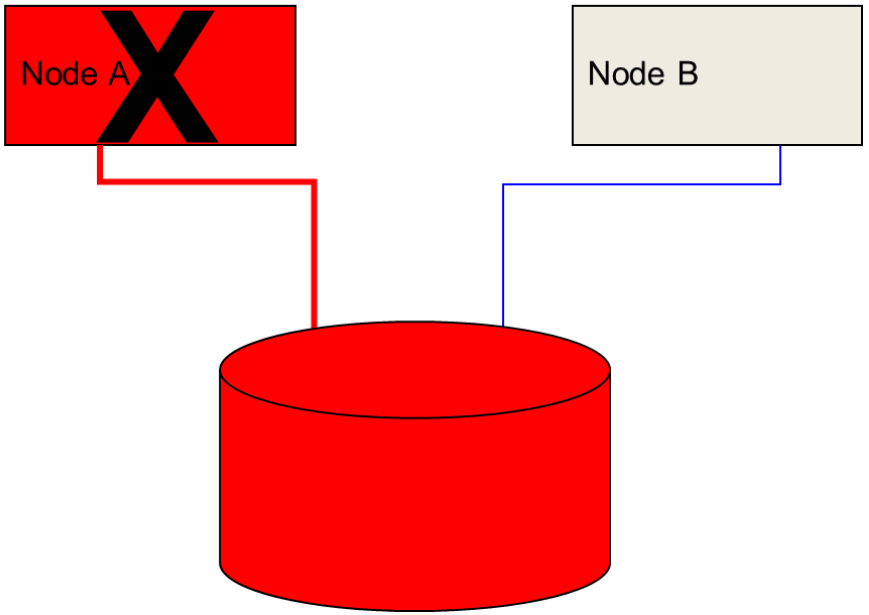
- Ausgangslage
  - A ist aktiv
  - B ist passiv
- Failover
  - A fällt aus
  - B wird aktiv



Cluster – Failover

Aktiv/Passiv

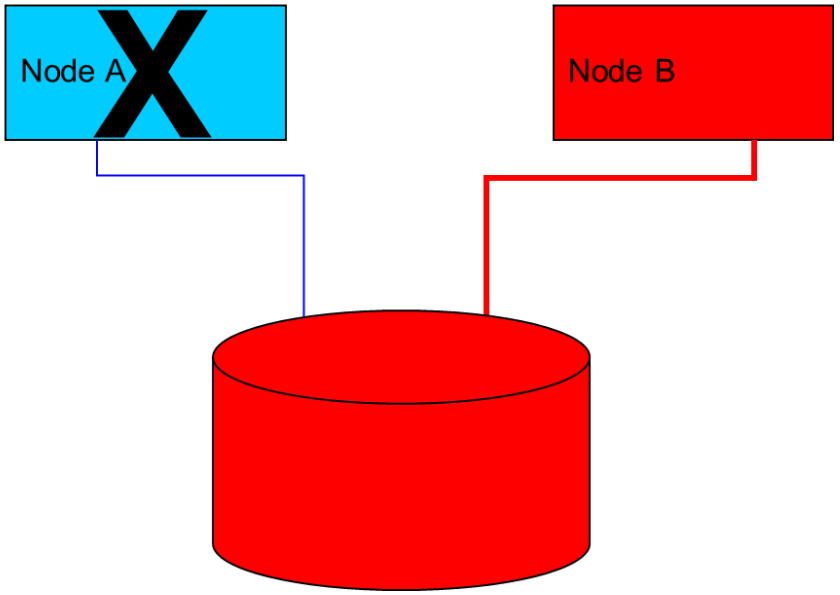
- Failover
  - A fällt aus



Cluster – Failover

Aktiv/Passiv

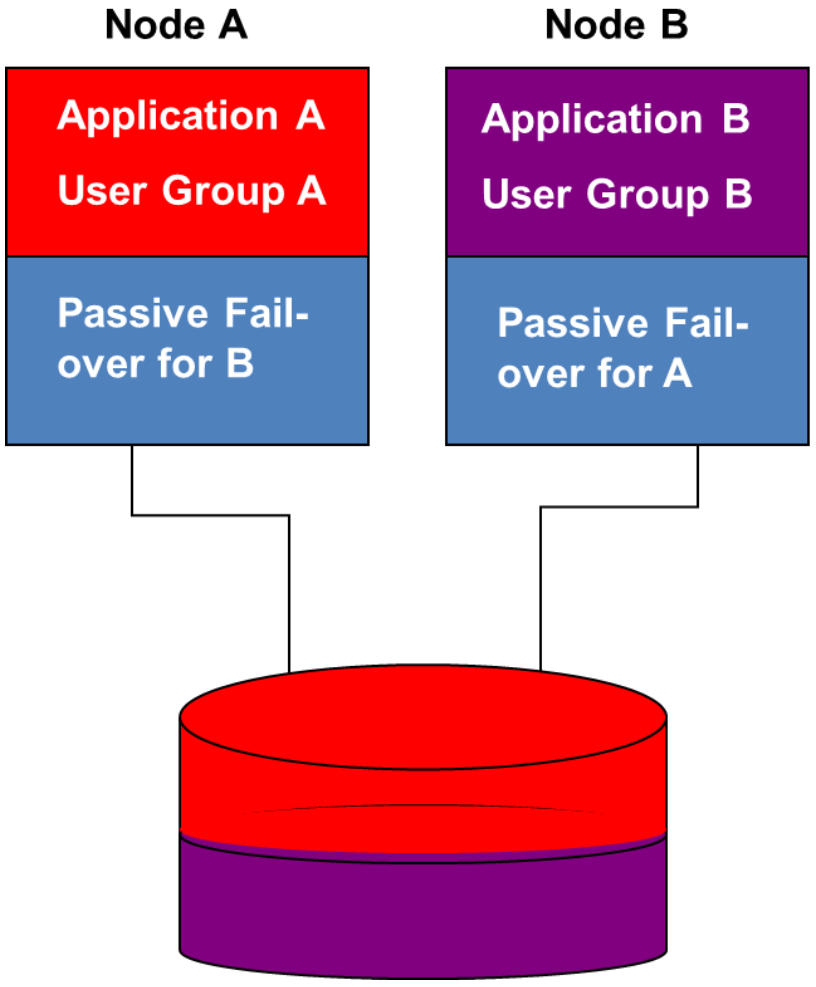
- Failover
  - B wird aktiv
  - B bleibt bis zur manuellen Recovery aktiv.
  - Nach einem Recovery kann B Aktiv bleiben oder die Ausgangslage wiederhergestellt werden.



Cluster – Failover

Aktiv/Aktiv

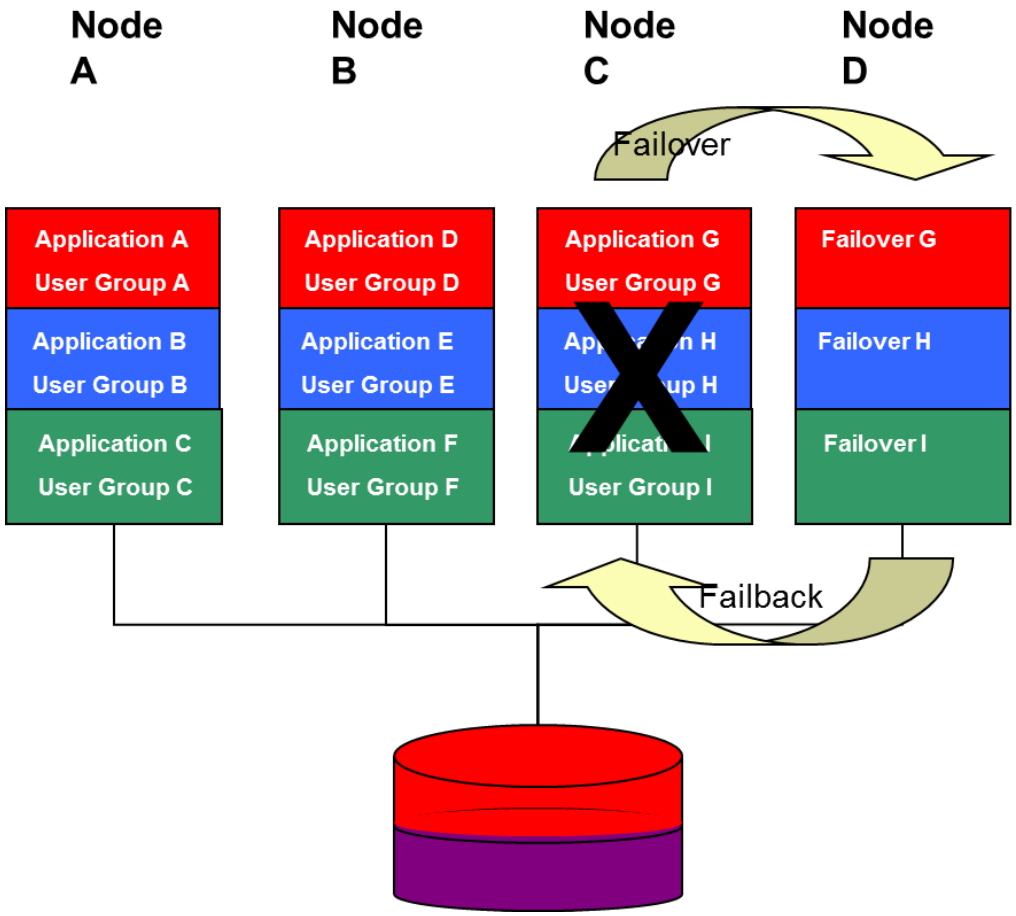
- Applikations- und Usergruppe A sind **aktiv** auf Node A
- Applikations- und Usergruppe B sind **aktiv** auf Node B
- Beide Nodes fungieren als Failover für den anderen Node.



Cluster – Failover

Spezialfall N-to-1 Failover

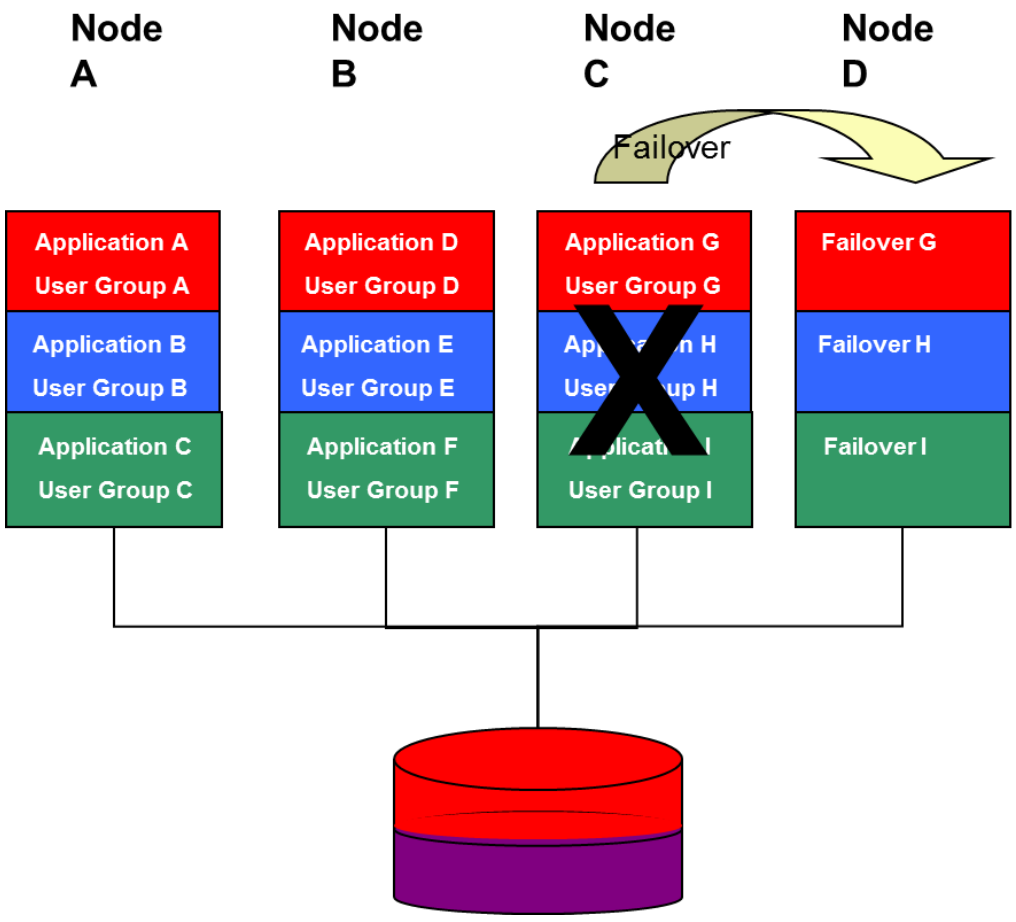
- Node D ist eine dedizierte Failovernode für A, B und C
- So kann die Anzahl aktiver Nodes erhöht werden
- Hat D einmal den Betrieb einer aktiven Node übernommen, müssen die Services von Node D wieder zurück auf die Ausgangs-Node zurückfallen um die Hochverfügbarkeit wieder herzustellen



Cluster – Failover

Spezialfall N + 1 (N+M) Failover

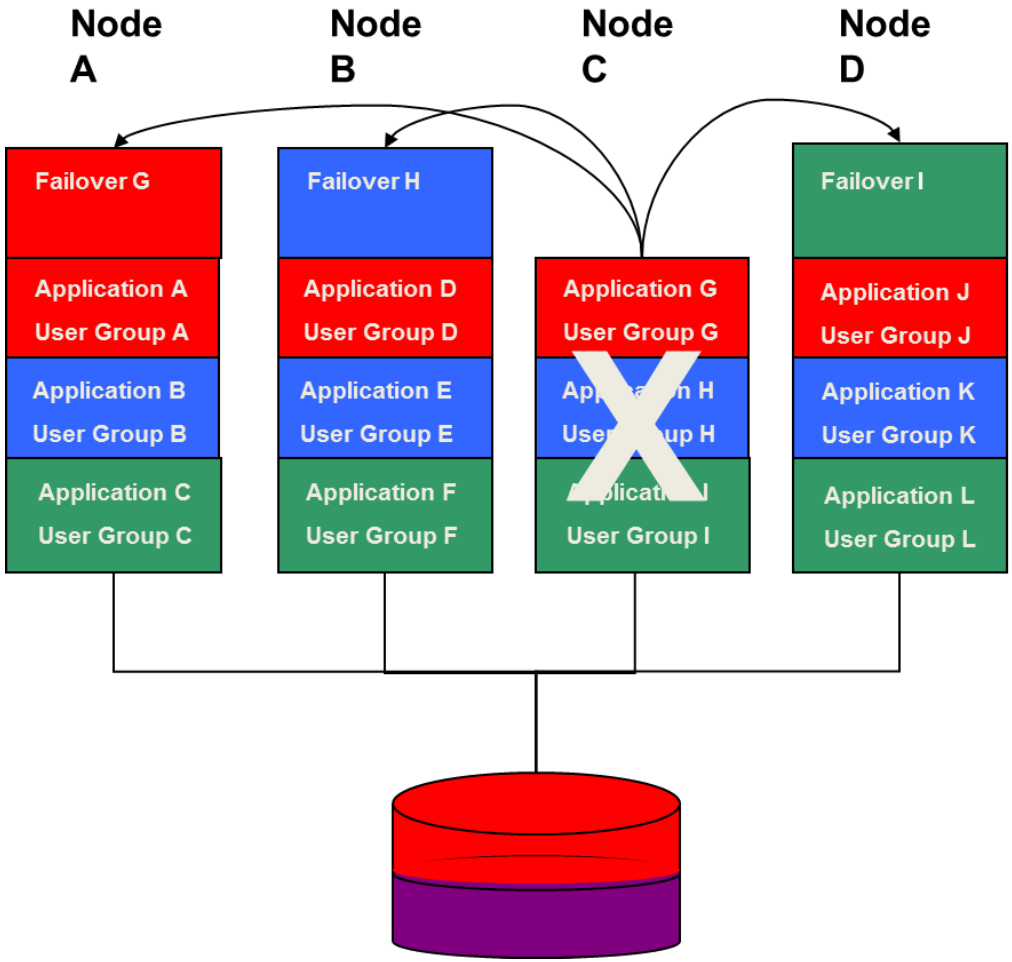
- Node D ist eine dedizierte Failovernode für A, B und C
- So kann die Anzahl aktiver Nodes erhöht werden
- Hat D einmal den Betrieb einer aktiven Node übernommen, wird die wiederhergestellte Node (im Beispiel C) zur neuen Failovernode



Cluster – Failover

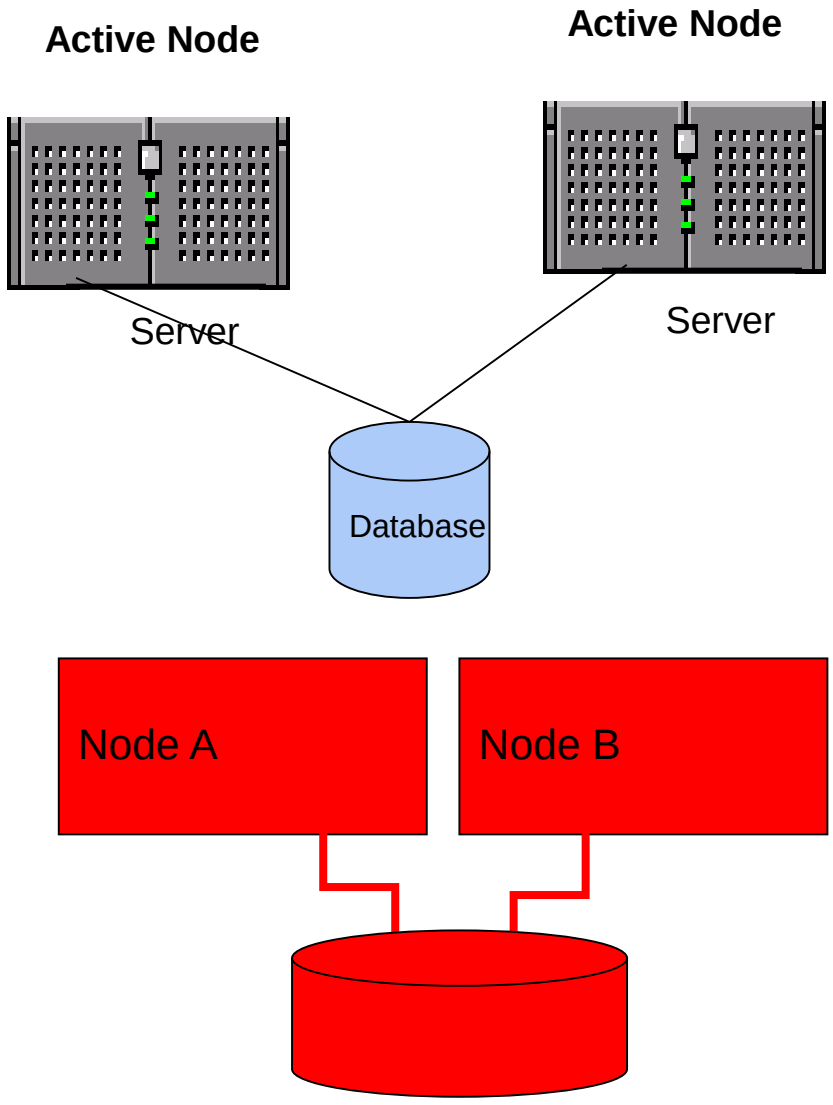
Spezialfall N-to-N Failover

- Node D ist eine dedizierte Failovernode für A, B und C
- So kann die Anzahl aktiver Nodes erhöht werden
- Hat D einmal den Betrieb einer aktiven Node übernommen, müssen die Services von Node D wieder zurück auf die Ausgangs-Node zurückfallen um die Hochverfügbarkeit wieder herzustellen



Cluster – Shared Anything

- Eine einzige Datenbasis
- Beide Nodes arbeiten gleichzeitig auf dieser Datenbasis
- Somit ist ein transparentes Failover möglich
- Die Applikation muss Cluster Aware sein
- Höchstes Level der Ausfallsicherheit
- So wird auch eine Lastverteilung realisiert



Cluster - Interkommunikation

Wo wird was gespeichert

- VM-Config ist entweder verteilt (repliziert) oder auf einem dedizierten Drive (SAN)
- VM-Disk ist auf einem geteilten Drive, so haben alle Clustermembers zugriff
- Vor allem spezialisierte DB-Cluster brauchen ein gemeinsames Caching

Wie können mehrere Systeme Synchron gehalten werden

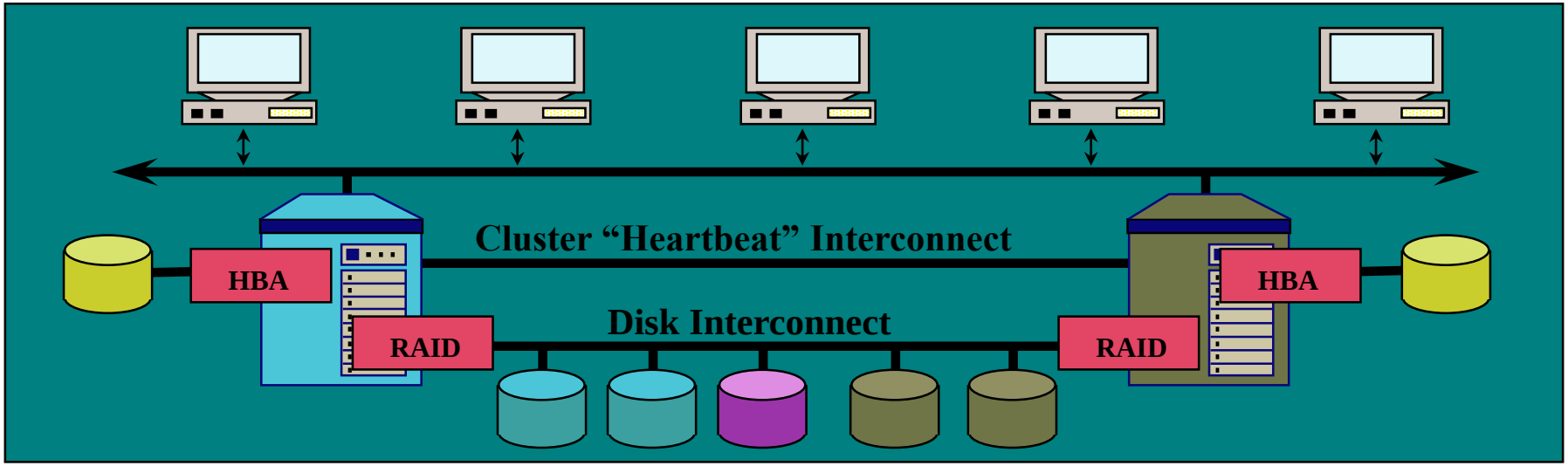
- Ein Shared-Disk Cluster benötig neben den normalen Kanälen eine Clustersynchronisation. So werden auch die VM-Configs verteilt
  - Über Ethernet
  - Oder spezialisierte Technologien
- Zum Monitoring des Clusterzustandes ist üblicherweise ein Heartbeat zwischen den einzelnen Nodes implementiert
  - Kann auch über eine quorum-disk via SAN gelöst oder ergänzt werden (later)

Der Cluster benötigt in jedem Fall ein separates «Management-Network». Viele dieser Mechanismen nutzen UDP und Multicast / Unicast.

Cluster - Interkommunikation

Transaktionskoordination, oder wer ist der Böse

- Quorum-Disk (oder Voting-Disk, violett)
- Sichert die Datenintegrität über den Cluster
- Entscheidet beim auftrennen des Clusters welcher Teil des Clusters aktiv bleibt (wer ist also der Böse Teil)
- Typischerweise bleibt bei asymmetrischer Aufteilung der grösserer Clusterteil aktiv
- Beim Aufbau des Clusters oder dem neu initialisieren wird der erste Node die Quorum-Disk übernehmen. Die weiteren Nodes treten dann nur noch dem Cluster bei.



Cluster - Interkommunikation

Split Brain

- Werden alle Zwischenverbindungen eines Clusters geteilt spricht man von einer Split-Brain Problematik. Varianten
  - (Ab-)Trennung eines Einzelknotens, ein Extrembeispiel dafür ist die Teilung eines 2-Knoten-Clusters
  - Auftrennung eines Mehr-Knoten-Clusters (>2) in ungleiche Teile
  - Auftrennung eines Mehr-Knoten-Clusters (>2) in gleiche Teile
- Parallele Schreibzugriffe im getrennten Cluster können zu massiven Konflikten führen

Gegenmassnahmen

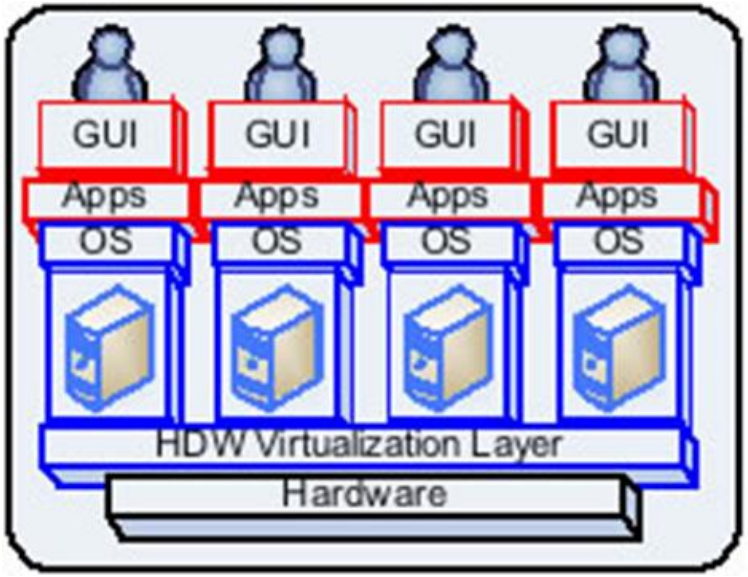
- Einsatz von Quorum und Cluster Interconnect gleichzeitig
- Rulesets: es überlebt nach dem Verlust des Interconnect
  - der Teil/Knoten mit der Sicht auf die meisten der Quoren
  - der Teil/Knoten mit der höchsten Arbeitslast.
- Gewisse Hersteller setzen zusätzlich mehrere Quoren ein, um einen Ausfall des Quorum zu vermeiden. Möglich ist auch eine Storage-seitige Replikation (im SAN)

Problematisch sind Cluster mit einer geraden Zahl Nodes, insbesondere 2!

Clientvirtualisierung

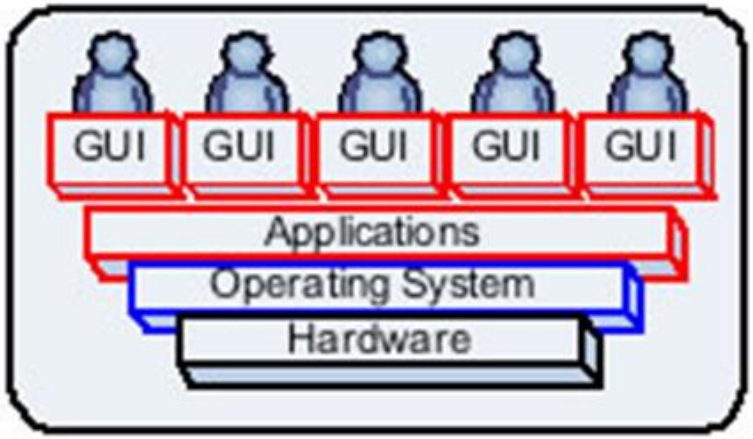
Clientvirtualisierung

- Pro User eine VM
- Volle Einstellmöglichkeit
- Abgrenzung kritischer Anwendungen
- Ressourcenintensiv



Terminal Services

- Mehrere User teilen sich eine Instanz
- Kleiner Administrationsaufwand
- Schlechte Isolation / Abgrenzung



Fragen

Gruppenarbeit – Clientvirtualisierung

Aufgabe

Untersuchen Sie in 2er Gruppen ein marktrelevantes System auf Grund der folgenden Kriterien. Stellen Sie die Ergebnisse der Klasse vor. Zeigen Sie vor- und Nachteile!

Systeme

- Microsoft: Integriert in Windows ist Remote Desktop Protocol (RDP) und in neueren Versionen RemoteFX
- RedHat: Hat mit dem Kauf von Qumranet (Kernel-based Virtual Machine (KVM)) auch Simple Protocol for Independent Computing Environments (SPICE) übernommen
- Citrix: zu XEN Independent Computing Architecture (ICA) mit HDX
- VMware: PCoIP
- RealVNC: VNC (seit 1998 opensource)

Mögliche Kriterien

- Übertragung / Bandbreite / WAN-Fähigkeit
- Technologie / Kompression
- Grafikbeschleunigung
- unterstützte Plattformen Server
- unterstützte Plattformen Client
- Zeroclients
- Sound / USB / Zwischenablage
- Lizenzierung
- Maintenance, Zukunftssicherheit

Kompetenznachweis

Prüfung

- Samstag 24. März 2018
- Beginn: 08:10 Uhr
- Dauer ca. 1h
- Gesamter Stoff bis zu diesem Datum
- ohne Unterlagen, ohne Computer

Praktische Arbeit, Demonstration Samstag 28. April 2018

- Funktionierender Proxmox Cluster im 3er Gruppen (mind. 3 Nodes)
- 2 VM's mit Paravirtualisierten Treibern für Ethernet und Storage (1x Windows, 1x Linux), Zugriff auch via SPICE
- 0 Downtime Migration von Node A zu Node B mit konfigurierter HA
- Funktionierende Docker-Instanz mit Apache Guacamole
- Eigenen Docker geschrieben
- Erfüllung aller obligatorischer Punkte = Note 5
  - Zusatzpunkte möglich, der Kreativität sind keine Grenzen gesetzt

Hands On – Cluster / Clientvirtualisierung

Cluster

- Untersuchen Sie die Möglichkeiten des HA-Managers. Erreichen Sie ein automatisches Failover?

VM's

- Installieren Sie pro Cluster mindestens (sollte schon gemacht sein)
  - 1 Windows-Guest
  - 1 Linux Guest mit GUI
  - Verwenden Sie bei beiden Installationen paravirtalisierte Ethernet und Storage-Treiber (Stichwort virtio)
  - Lassen Sie Memory dynamisch zuweisen. Wie viel Memory sieht ihr Guest-system? Wie wird dieser Effekt erzielt?

Thin-Client (optional)

- Realisieren Sie mit ihren Raspberry-PI Thinclients, welche über SPICE und RDP automatisch beim starten auf ihre VM's verbinden.
  - Raspberry Pi Thin Client project: http://rpitc.blogspot.ch/