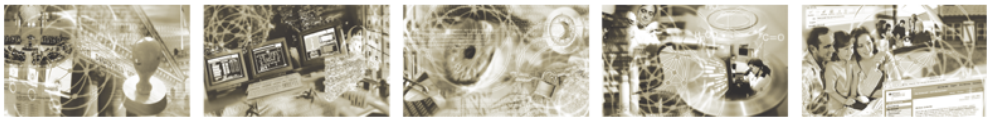
Bundesamt
für Sicherheit in der
Informationstechnik



Band B, Kapitel 4: Cluster

Im Umfeld der Hochverfügbarkeit

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582-0
E-Mail: hochverfügbarkeit@bsi.bund.de
Internet: https://www.bsi.bund.de
© Bundesamt für Sicherheit in der Informationstechnik 2013

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Ein wichtiger Ansatz zur Einführung von Redundanz in IT-Systemen ist der Einsatz von mehreren funktional ähnlichen Einzelelementen, um Redundanzen im Betrieb einzuführen. Solche Systeme werden Cluster genannt. Diese Architektur wird in diesem Abschnitt dargestellt.

Ein Cluster stellt eine Kombination aus einzelnen Systemen zu einem Systemverbund dar. Dazu können sowohl logische als auch physikalische Kombinationen gewählt werden. Die Einzelsysteme sind bezogen auf ihre Wirkfunktion gleichwertig. Neben der eigentlichen Wirkfunktion verfügt ein Cluster in seiner Gesamtheit über zusätzliche Arbitrierungsfunktionen, die die Synchronisation der Einzelsysteme sicherstellen. Dies wird durch eine spezielle Clustersoftware realisiert. Bezogen auf die Zielsetzung, mit der ein Cluster aufgebaut wird, unterscheidet man Hochverfügbarkeits-Cluster, Lastverteilungs-Cluster und Hochgeschwindigkeits-Cluster.

In diesem Kompendium ist die Betrachtung von Cluster-Strukturen, welche besonders für HV-Konzepte geeignet sind, von besonderer Bedeutung. Kommt es darauf an, eine hohe Verfügbarkeit für einen Dienst zu erreichen, sind neben dem HV-Cluster Lastverteilungs-Cluster eine mögliche Lösung. Hochgeschwindigkeits-Cluster sind auf spezielle Leistungsparameter optimiert und spielen bei der Betrachtung von Hochverfügbarkeitslösungen keine wesentliche Rolle.

Cluster können nach verschiedenen Kriterien kategorisiert werden:

- Art der Gruppierung,
- Topologie der Gruppierung und
- Arbitrierung des Zugriffs auf den Cluster.

Je nach Betrachtungsart werden im Folgenden Vor- und Nachteile von Cluster-Systemen dargelegt.

1.1 Gruppierungsarten

Die Gruppierungsarten sind gekennzeichnet durch die Zuordnung logischer und physikalischer Einheiten zueinander. Wenn wir mit N die Anzahl der logischen Einheiten und M die Anzahl der physikalischen Einheiten bezeichnen, so ergeben sich folgende Grundformen:

- N-in-1-Cluster
- 1-in-M-Cluster

Durch Kombination der obigen Cluster-Gruppierungsarten ergibt sich der:

- N-in-M-Cluster

Ein N-in-1-Cluster besteht aus N logischen Einheiten, die auf einer physikalischen Einheit kombiniert werden. Die Art der logischen Einheit differiert je nach Anwendungsfall. Es kann sich dabei um eine vollständige Betriebssystem-Instanz handeln, wie sie etwa bei Hardwarevirtualisierungen anzutreffen ist. Es kann sich aber auch um die mehrfache Ausführung einzelner Dienste auf einer Instanz (wie bei Webserver-Farmen) handeln.

Die Vorteile dieser Clusterbildung liegen in der einfachen Wartbarkeit von logischen Einheiten. Typischerweise stellt der Cluster seinen logischen Elementen eine einheitliche Betriebsumgebung bereit. Dadurch ergeben sich Vereinheitlichungen, die sich beim Betrieb der entsprechenden

Lösungen positiv bemerkbar machen. Ein weiterer Vorteil besteht darin, dass die logischen Einheiten meistens problemlos auch zwischen physikalischen Einheiten verschoben werden können. Wenn sich also mittel- bis langfristig im Rahmen des Monitorings abzeichnet, dass ein Engpass auf einer Hardware-Einheit zu Problemen oder Ausfällen führen kann, besteht die Möglichkeit zur manuell initiierten Lastverteilung.

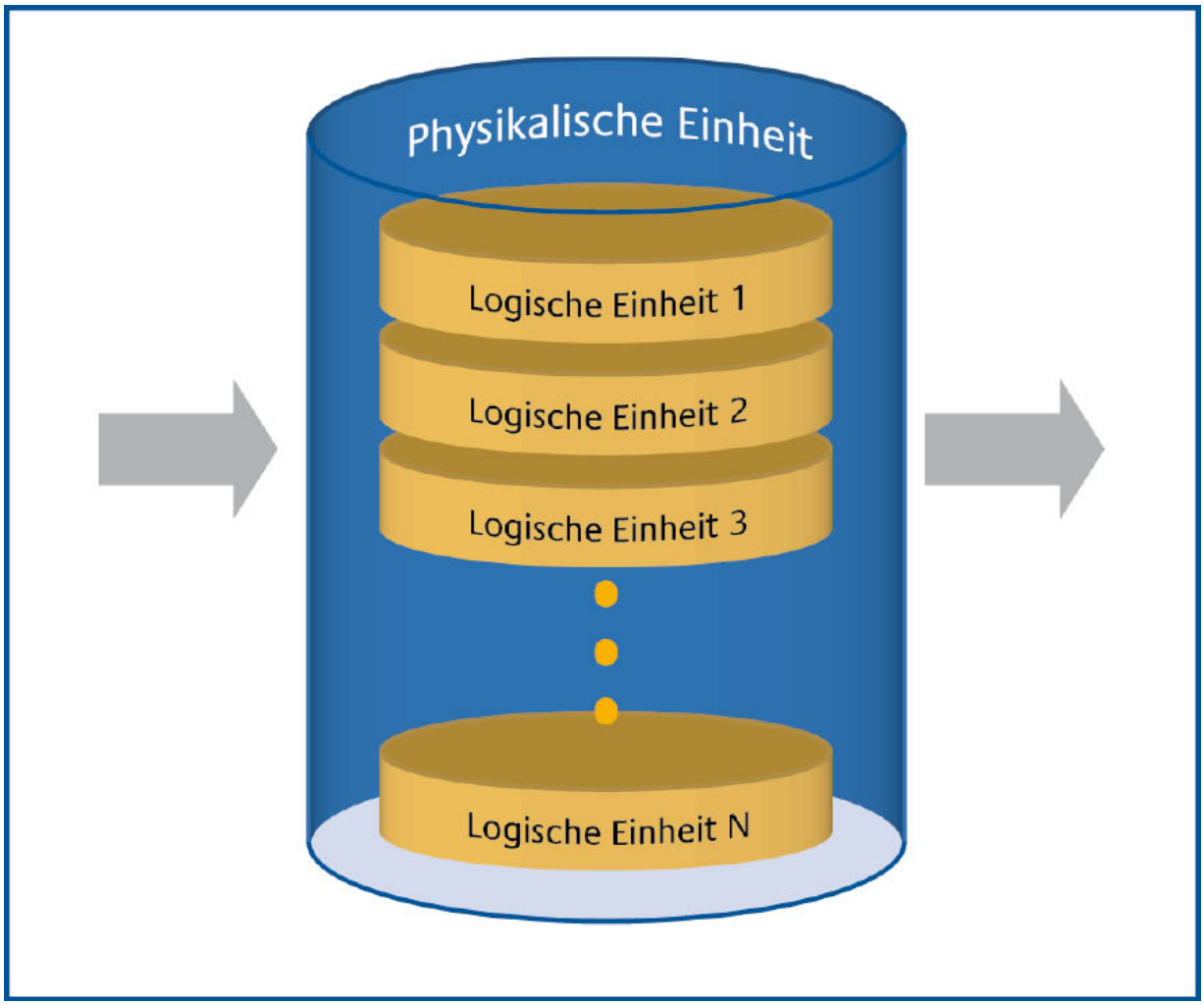


Abbildung 1: N-in-1-Cluster

Nachteile dieser Cluster bestehen darin, dass sie per se keine Redundanz auf der Ebene des Host-Systems bieten. Es besteht lediglich die Möglichkeit, mehrere N-in-1-Cluster parallel zu betreiben bzw. vorzuhalten. Ein solcher Verbund bietet jedoch von sich aus keine Möglichkeit, automatische Gegenmaßnahmen einzuleiten, die bei einem Ausfall aktiviert werden. Die Redundanz ist dann typischerweise manuell zu überwachen und zu aktivieren.

Ein 1-in-M-Cluster besteht aus einer logischen Einheit, die auf verschiedene Hardwareeinheiten verteilt wird. Typisches Beispiel dieser Gruppierung ist ein Mail-Cluster, der über eine Adresse angesprochen wird, deren Datenbestände aber über verschiedene Hardware-Systeme verteilt sind. Derartige Cluster verfügen typischerweise über eine zusätzliche Instanz, die dafür sorgt, dass Informationen zwischen verschiedenen Cluster-Knoten verteilt werden. Es ist wichtig zu berücksichtigen, dass 1-in-M-Cluster intern häufig nur über strangredundante Subkomponenten verfügen.

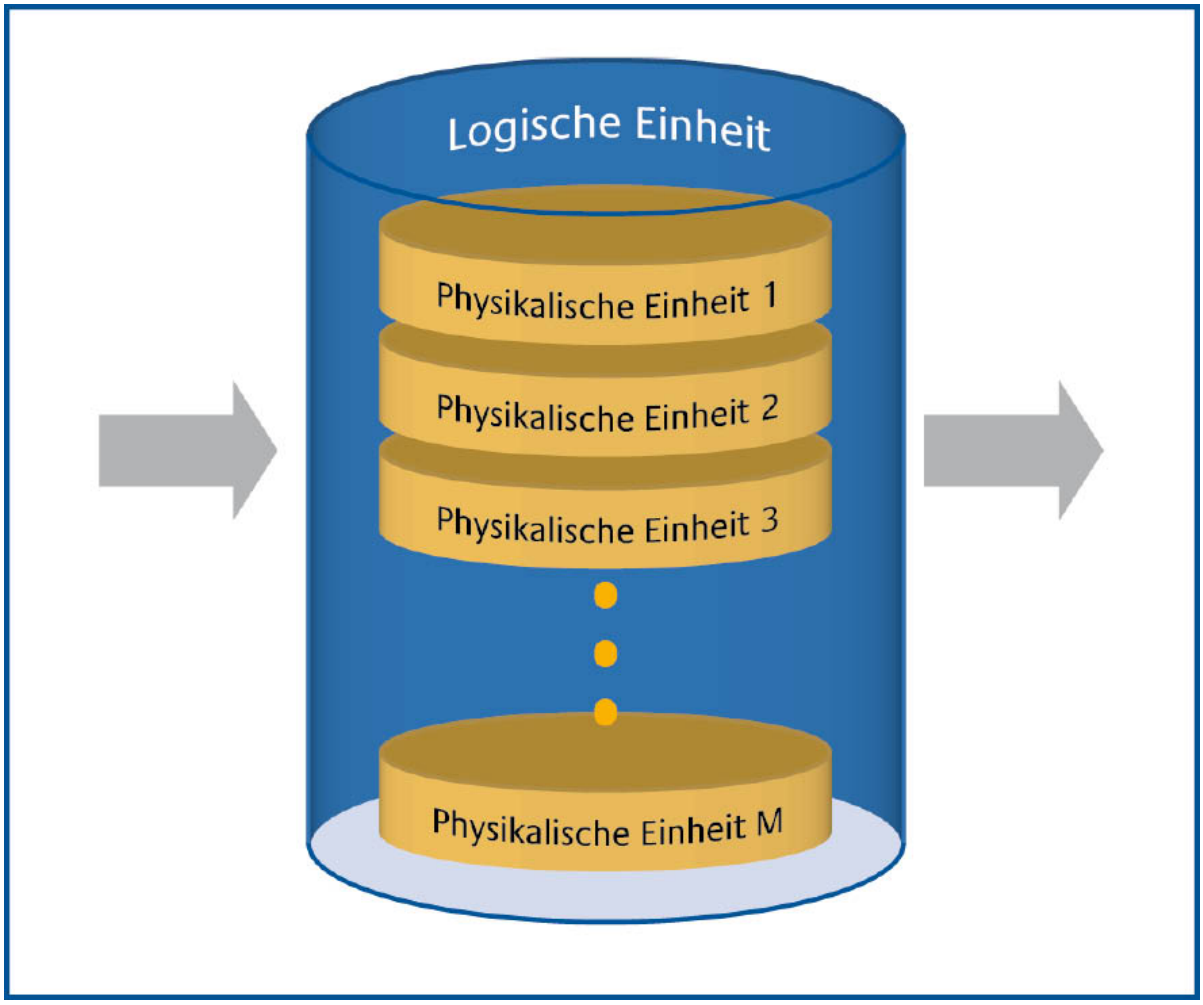


Abbildung 2: 1-in-M-Cluster

Vorteile eines 1-in-M-Clusters liegen in der Unempfindlichkeit gegenüber Einzelsystemausfällen. Häufig bieten 1-in-M-Cluster eine gute Skalierbarkeit. Der Umfang der Skalierbarkeit ergibt sich aus der Topologie des Clusters. Trotz der einfachen Skalierbarkeit darf nicht übersehen werden, dass die Kosten des Gesamtsystems nicht zwangsläufig in gleicher Weise skalieren, sondern durch den Koordinierungsaufwand und Verwaltungs-Overhead zwischen den Elementen deutlich stärker anwachsen.

Die Nachteile von 1-in-M-Clustern liegen in der Spezialisierung auf einzelne Dienste. Ein 1-in-M-Cluster bietet keine Flexibilität wie ein N-in-1-Cluster, da sich die Funktionalität auf eine spezielle Anwendung beschränkt. Grund dafür ist die Art des Datenabgleichs bzw. die Optimierung der Datenabgleichalgorithmen auf die beabsichtigte Funktionalität des Clusters.

Kombiniert man N-in-1 und 1-in-M-Cluster ergibt sich der N-in-M-Cluster. Diese Clustervariante ist in der Lage, unterschiedliche logische Einheiten auf mehrere physikalische Einheiten zu verteilen.

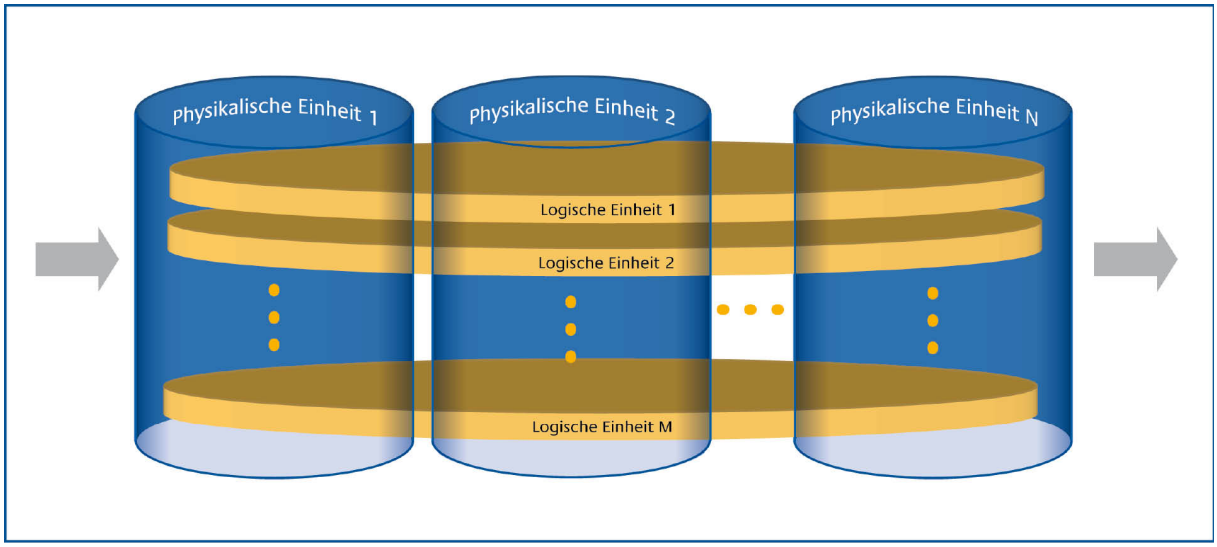


Abbildung 3: N-in-M-Cluster

1.2 Topologien

Sowohl 1-in-M als auch N-in-M-Clustern unterscheiden sich in der Art, wie die verschiedenen Hardware-Komponenten zu einem Verbund zusammengeführt werden. Die Art, wie die Kommunikation der einzelnen Elemente untereinander realisiert ist, entscheidet maßgeblich, welche Qualitätsmerkmale dauerhaft verfügbar sind. Diese Kommunikation kann entweder über die Nutzdatenverbindungen oder aber alternativ über ein dediziertes Kommunikationssegment zum Zwecke des Monitorings und der Synchronisation geschehen. Die garantierte Datenrate, die Grundlage der Verfügbarkeit des Gesamtsystems ist, wird von dieser Entscheidung besonders beeinflusst. Man unterscheidet bei Cluster-Systemen folgende Topologien:

- Shared-Memory-Cluster,
- Paar-Cluster,
- Baum-Cluster,
- Bus-Cluster und
- vermaschte Cluster.

In diesem Abschnitt werden die spezifischen Charakteristika der verschiedenen Cluster-Topologien dargestellt und deren Vor- und Nachteile thematisiert.

1.3 Shared-Memory-Cluster

Ein Shared-Memory-Cluster basiert auf der gleichnamigen Shared-Memory-Architektur von Computersystemen. Bei einem Shared-Memory-Cluster wird der Gesamtverbund aus einem Multiprozessor-System gebildet. Je nach Variation bieten derartige Systeme unterschiedliche Leistungsmerkmale, deren wesentliche Kenngrößen die Anzahl der parallelen CPUs und die einfache Austauschbarkeit von Teilkomponenten durch Ersatzkomponenten im laufenden Betrieb sind.

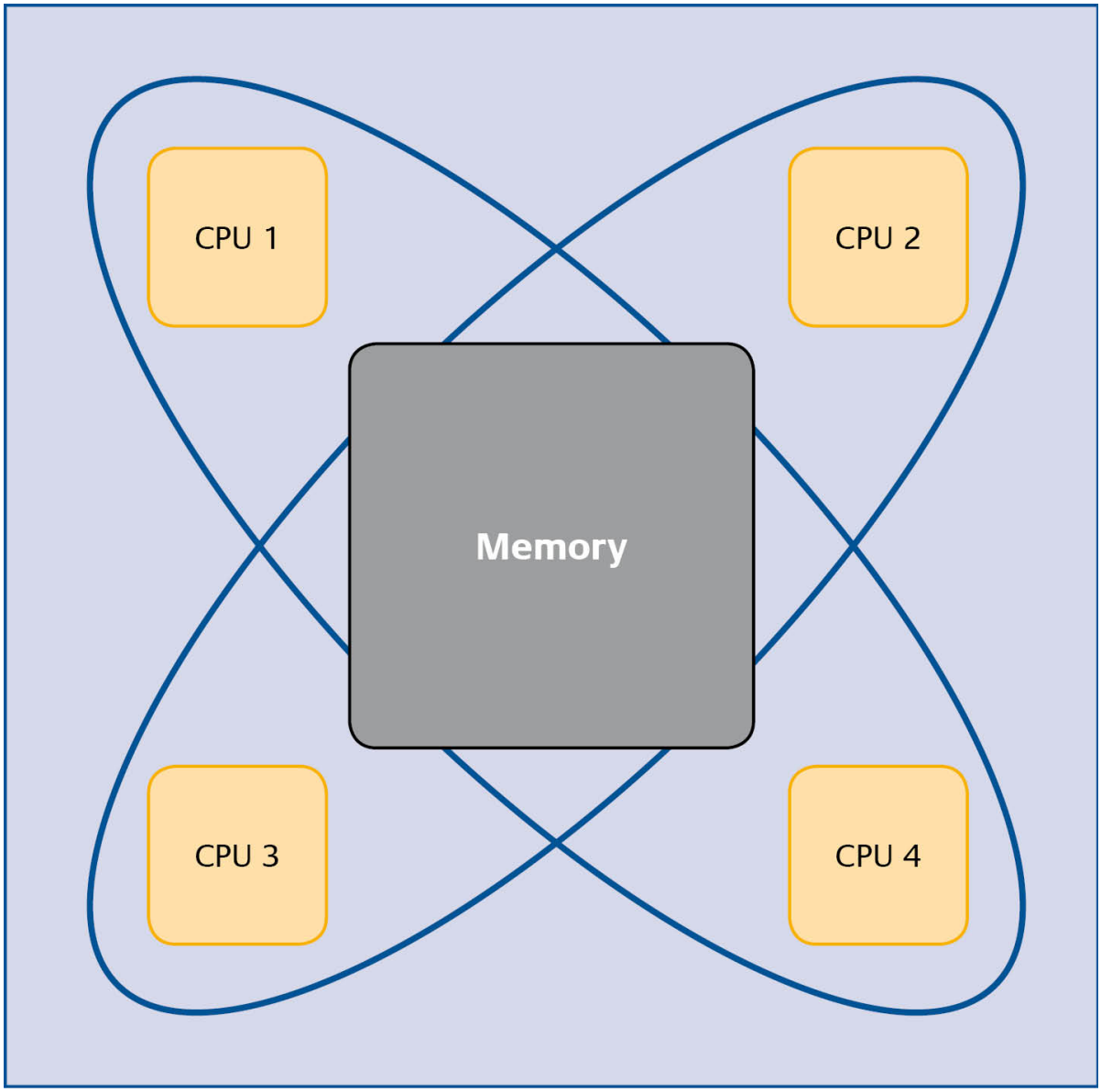


Abbildung 4: Shared-Memory

Die Synchronisation der Clusterelemente erfolgt über den gemeinsam verwendeten Speicher. Parallele Zugriffe auf geteilte Speicherbereiche werden durch Semaphore serialisiert oder durch Speicherarchitekturen ermöglicht, die ein gleichzeitiges Lesen von mehreren CPUs ermöglichen.

Die Überwachung von Ausfällen obliegt speziellen Überwachungsbausteinen, die Ausfälle von CPUs erkennen können und ausgefallene CPUs abschalten.

1.3.1 Paar-Cluster

Ein Paar-Cluster besteht aus zwei (oder prinzipiell auch mehreren) Einzel-Systemen, die quasi gleichwertig sind. Man unterscheidet zwischen aktiv-aktiv und aktiv-passiv arbeitenden Clustern.

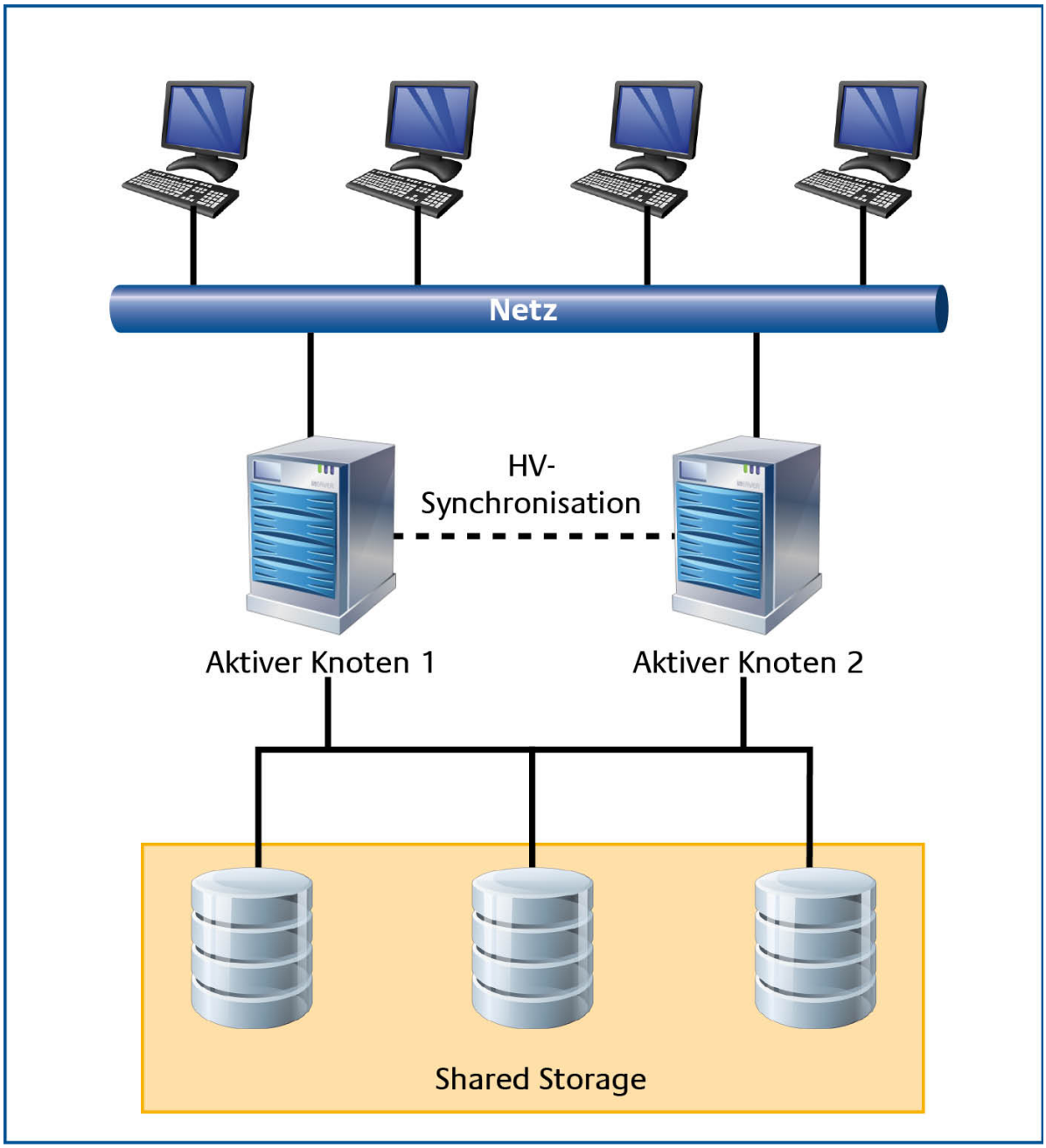


Abbildung 5: Aktiv-aktiv-Paar-Cluster

Ein aktiv-aktiv betriebenes System zeichnet sich dadurch aus, dass beide Elemente gleichzeitig produktiv arbeiten. Dies impliziert nicht automatisch ein Loadbalancing. Häufig ist eine separate

Loadbalancing-Einheit notwendig. Es gibt aber auch Produkte, die die Loadbalancing-Funktionalität bereits integriert haben.

Die Aktiv-passiv-Betriebsart (auch asymmetrische Knoten genannt) zeichnet sich dadurch aus, dass immer nur ein System zu einem Zeitpunkt produktiv arbeitet. Es handelt sich dabei jedoch normalerweise nicht um einen Zeitscheibenbetrieb. Der Wechsel der aktiven Komponente erfolgt typischerweise sehr selten und nur in Folge einer Fehlfunktion oder einer Wartungstätigkeit.

Die aktiv-aktiv-Variante (auch symmetrische Knoten genannt) bietet gegenüber der Aktiv-passiv-Variante eine höhere Burstleistung im Normalbetrieb. Die garantierte Minimalleistung der beiden Varianten ist aber gleich. Bei der Dimensionierung eines Clusters ist es wichtig, den Vorteil der Burstleistung nicht überzubewerten, da er sich im Fehlerfall schlagartig reduziert. Die Überlastung des verbleibenden Systems kann unter Umständen zum Ausfall der verbleibenden Komponente führen. Tritt dieser Fall ein, bedeutet dies automatisch den Totalausfall des Clusters. Durch Fehldimensionierung von Aktiv-aktiv-Paar-Clustern können somit Totalausfälle provoziert werden. Die Aktiv-passiv-Variante der Paar-Cluster hat diesen Nachteil nicht.

Die Synchronisation von Paar-Clustern erfolgt entweder über eine dedizierte Verbindung oder über die Nutzdatenverbindung (z. B. Cluster Heartbeat bei Aktiv-aktiv-Betriebsart). Die Clusterelemente müssen über die Funktionalität verfügen, sich gegenseitig zu überwachen und gegebenenfalls zu synchronisieren. Realisiert wird dies über ein individuelles Protokoll, das den Synchronisationsdatenaustausch ermöglicht. Die Überwachungsfunktionen sind je nach Anwendung applikationsindividuell.

	Aktiv-Passiv	***Aktiv-Aktiv***
Burstleistung	niedrig	hoch
Loadbalancing	nicht möglich	extern, intern, ohne
Totalausfallgefahr	niedrig	hoch bei Burstleistungsbeanspruchung

Tabelle 1: Paar-Cluster-Eigenschaften

1.3.2 Baum-Cluster

Der Baum-Cluster besteht typischerweise aus einer zweistufigen Baumstruktur. Im Gegensatz zum Paar-Cluster ist beim Baum-Cluster immer eine zusätzliche Komponente, nämlich der Dispatcher notwendig. Diese erste Stufe sorgt für die Verteilung der Daten oder Aufträge an die einzelnen operativen Cluster-Elemente. Ein active-active-Paar-Cluster mit vorgeschaltetem Loadbalancer ist beispielsweise eine mögliche Ausprägung des Baum-Clusters. Die Funktionalität des Dispatchers ist in manchen Anwendungsfällen im Client integriert. Dies gilt z. B. für den Domain Name Service (DNS).

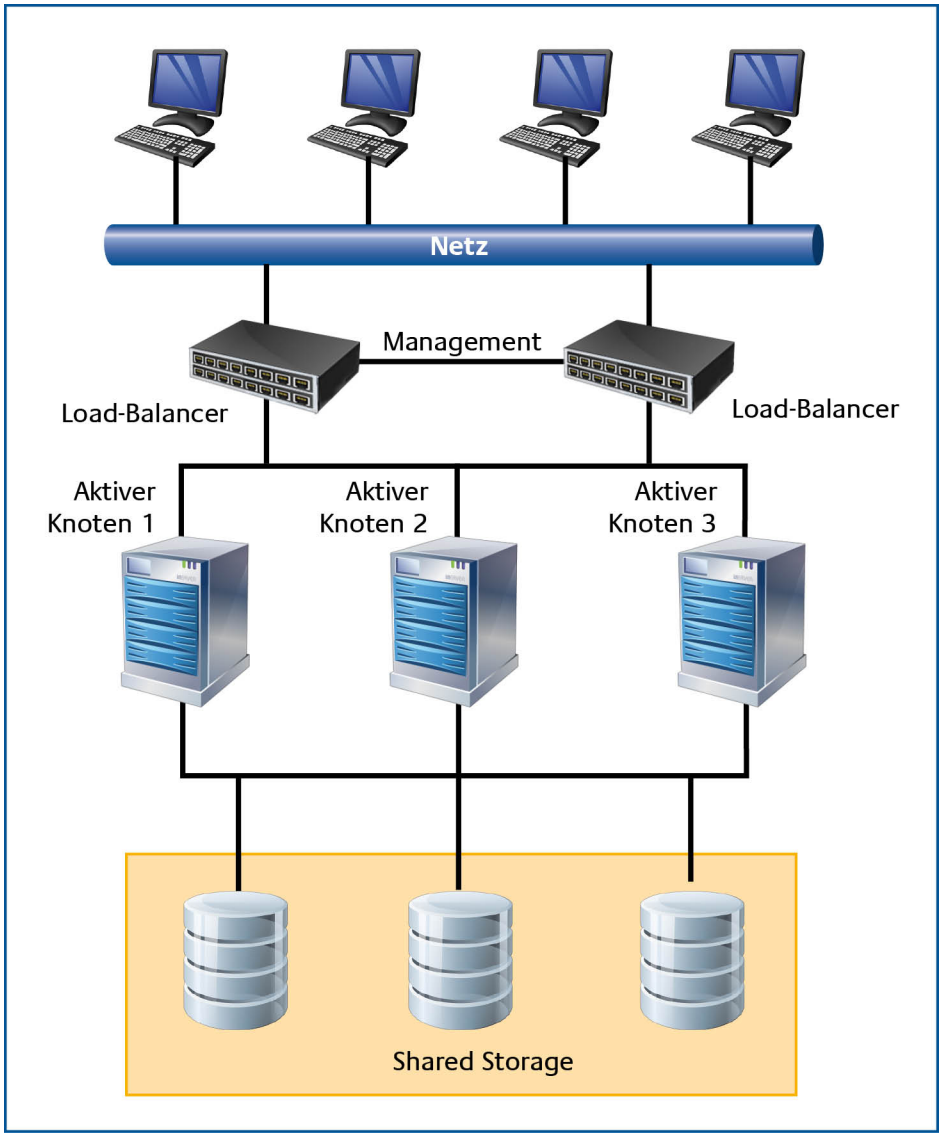


Abbildung 6: Baum-Cluster

Neben der einfachen Ausprägung von Baum-Clustern existieren symmetrische Baum-Cluster, die entweder über eine eingehende und eine ausgehende Seite verfügen oder bidirektional betrieben werden können. Diese Cluster zeichnen sich durch eine Dispatcher-Einheit auf jeder Seite des Clusters aus.

Baum-Cluster bestehen aus funktional unterschiedlichen Komponenten. Die Dispatcher-Stufe besteht typischerweise aus einer vergleichsweise einfach aufgebauten Appliance, die keine langfristigen Zustände verwaltet und sich ausschließlich auf die Arbitration beschränkt. Die Cluster-Stufe hingegen kann aus Paar-Clustern bestehen. Es besteht aber auch die Möglichkeit, dass die einzelnen Systeme der Cluster-Stufe gar nicht miteinander kommunizieren und selbst von ihrer Zugehörigkeit zu einem Cluster gar nichts „wissen“.

Es sollte bei der Konzeption einer HV-Lösung darauf geachtet werden, dass auch die Dispatcher-Stufe redundant ausgelegt werden kann. Oft trifft man in diesen Bereichen eine Cold-Stand-By-Architektur an, die eine active-passive-Kombination ohne automatisches Fail-Over darstellt. Dadurch sinkt bei dieser Cluster-Topologie die Verfügbarkeit.

Der Vorteil einer Baum-Struktur ist die bessere Skalierbarkeit. Baum-Cluster können leicht erweitert werden und somit können sowohl die Verfügbarkeit als auch Qualitätsmerkmale wie der garantierte Durchsatz angepasst werden. Allerdings wird die Performance durch die Anbindung der Dispatcher-Stufe des Clusters limitiert. Ein über eine Gigabit-Leitung angeschlossener Baum-Cluster kann maximal einen Gigabit-Durchsatz erreichen. Häufig wird dieser kleine aber wesentliche Aspekt bei der Skalierbarkeitsbetrachtung übersehen.

Die Hauptfunktion des Dispatchers liegt somit in der Verteilung der Daten oder Aufträge auf die einzelnen Knoten. Wenn diese Verteilung noch zusätzliche qualitative Merkmale besitzt, wie die Fähigkeit Sessions gleichmäßig zu verteilen oder auf die Auslastung der Knoten zu reagieren, spricht man auch von einem Loadbalancer.

Baum-Cluster können über solche Loadbalancer zu heterogenen mehrstufigen Baum-Clustern zusammengeschaltet werden, um höhere Durchsatzleistungen zu erreichen. Für die Durchsatzbetrachtung ist aber zwischen dem Gesamtdurchsatz und dem Einzeldurchsatz zu unterscheiden. Insbesondere bei stark skalierten Clustersystemen kann der Unterschied zwischen Gesamtdurchsatz und Einzeldurchsatz ganz erheblich sein. Typische Beispiele dafür sind Dienste, die einen hohen Rechenaufwand erfordern wie Spam- oder Viren-Filter.

Bei einem Baum-Cluster erfolgt die Überwachung der Clusterelemente durch den Dispatcher. Es ist Aufgabe des Dispatchers eine Fehlfunktion zu erkennen und das als fehlerhaft erkannte Teilsystem aus dem Clusterverbund auszuschließen.

Eine Synchronisation der Cluster-Elemente findet nicht notwendigerweise statt, da bei bestimmten Anwendungen die Cluster-Elemente vollkommen autonom arbeiten und keine Zustandsinformation mit den anderen Cluster-Elementen austauschen. Solche Cluster zeichnen sich durch sogenannte Sessionverluste im Falle einer Fehlfunktion aus. Bei transaktionsbasierten Protokollen bedeutet dies, dass die zum Zeitpunkt des Ausfalls durchzuführende Transaktion nach Deaktivierung des defekten Cluster-Elements neu aufgesetzt werden muss, um von einem anderen Cluster-Element bearbeitet werden zu können.

Daneben gibt es aufwendigere Clustervarianten, bei denen der Dispatcher Zustandsinformationen mitprotokolliert und im Falle eines Ausfalls die gerade durchgeführte Aktion auf ein verbliebenes System umleitet.

1.3.3 Bus-Cluster

Ein Bus-Cluster zeichnet sich dadurch aus, dass alle Cluster-Elemente über einen Bus miteinander kommunizieren. Das bedeutet, dass der für den Clusterbetrieb notwendige Kommunikations-Overhead auf die Nutzdaten aufgeschlagen werden muss. Typischerweise übernimmt ein System am Bus die Arbitration der Cluster-Kommunikation. Häufig erfolgt das auf eine Weise, die eine Verdoppelung der Nutzdaten nach sich zieht. Da ein System das primäre System ist, werden die Daten von diesem System angenommen und auf die übrigen Systeme, die die so genannten Sekundären Systeme sind, verteilt. Der Verteilvorgang erfolgt gegebenenfalls durch erneutes Senden der Nutzdaten über den gemeinsamen Bus. Dadurch halbiert sich mindestens die verfügbare Bandbreite eines solchen Clusters, wenn man nicht den Zusatz-Overhead durch die Mehrfachbenutzung des Busses, etwa durch Kollisionen oder durch Bus-Management, berücksichtigt. Bus-Cluster sind somit nur begrenzt skalierbar. Ein Hinzufügen eines zusätzlichen Gerätes zum Cluster hat typischerweise einen geringeren Einfluss auf die Leistungsfähigkeit des Clusters als bei einem Baum-Cluster.

Bus-Cluster verfügen über die Möglichkeit über den Bus, der die einzelnen Cluster-Elemente verbindet, sowohl Synchronisationsdaten als auch Überwachungsdaten auszutauschen. Typischerweise wird dabei nicht die eigentliche Funktion des jeweiligen Cluster-Elements überwacht, sondern es findet eine von der eigentlichen Wirkaktion unabhängige Kommunikation statt, um ein defektes System auf Basis von Time-Outs als defekt zu erkennen. Unterschreitet ein Einzelsystem gewisse Mindestantwortzeiten, wird es als defekt klassifiziert und von der Verteilung der Nutzdaten ausgeschlossen. Unter den verbliebenen Systemen finden typischerweise automatisierte Abstimmvorgänge statt, nach denen die verbliebenen Systeme sich über das neue Master-System einigen.

1.3.4 Vermaschte Cluster

Der vermaschte Cluster wird häufig für extrem rechenintensive numerische Problemlösungen eingesetzt. Es handelt sich dabei um ein teil- oder vollvermaschtes Netz, das nur für spezielle parallelisierbare Anwendungen eingesetzt werden kann. Auf den Cluster kann typischerweise nur über vom Hersteller angebotene API (Programmierschnittstellen) zugegriffen werden. Cluster dieser Art werden in rechenintensiven Bereichen eingesetzt, z. B. für Simulationen in der Klimaforschung. Sie spielen für Hochverfügbarkeitsüberlegungen eine stark untergeordnete Rolle.

Vermaschte Cluster werden typischerweise auf der Basis einer LAN-Netzwerkinfrastruktur aufgebaut. Es ist jedoch auch möglich, einen vermaschten Cluster auf der Basis einer WAN-Netzwerkinfrastruktur zu realisieren. Diese Art von Cluster wird häufig für sehr rechenintensive Aufgaben verwendet, die keine besondere Anforderung an Antwortzeiten stellen. Man spricht in diesem Zusammenhang auch von GRID-Clustern.

Eine Synchronisation von vermaschten Clustern auf LAN-Basis findet typischerweise über eine standardisierte Schnittstelle namens PVM (Parallel Virtual Machine) statt, während bei GRIDs häufig applikationsspezifische Protokolle zum Einsatz kommen. In letzterem Fall findet eine Synchronisation zwischen den einzelnen Cluster-Elementen nicht statt. Stattdessen gibt es eine Kontrollinstanz, die die Abarbeitung der verschiedenen Teilaufgaben kontrolliert. Der Ausfall einer Teilkomponente wird dadurch kompensiert, dass die für die ausgefallene Teilkomponente zu erledigende Aufgabe an ein anderes Clusterelement erneut vergeben wird. Der Ausfall wird auch dadurch erkannt, dass ein ausgefallenes Cluster-Element sich selber nicht mehr bei der Kontrollinstanz meldet, um eine neue Aufgabe zur Abarbeitung abzuholen.

1.3.5 Zusammenfassung Topologien

Die folgende Tabelle fasst die Eigenschaften der verschiedenen Cluster-Architekturen zusammen.

	Shared Memory	***Paar***	***Baum***	***Bus***	***Vermascht***
Synchro-nisation	sehr schnell	schnell	sehr langsam	langsam	sehr langsam
Skalierbar-keit	bis ca. 8 Knoten	2 Knoten	sehr gut	bis ca. 8 Knoten	quasi beliebig
Arbitration	einfach	einfach	sehr einfach	aufwendi g	sehr aufwendig

Tabelle 2: Cluster-Topologie-Eigenschaften

1.4 Strategien für die Ressourcenverteilung: Arbitration

Durch das mehrfache Vorhandensein von Elementen in einem Cluster müssen die zu verarbeitenden Daten auf ein oder mehrere Elemente verteilt werden. Die notwendige Auswahl der jeweiligen Elemente wird durch Arbitration durchgeführt. Die Arbitration kann darüber hinaus die Verteilung konkurrierender Anfragen nach Prioritäten gestaffelt vornehmen. Je nach Art der Clusterarchitektur ist es notwendig, innerhalb eines Clusters eine Arbitration zu realisieren.

Bei strangredundanten Clustersystemen gibt es nur eine Arbitration für die Stränge, aber nicht für die einzelnen Elemente innerhalb eines Strangs, da die Stränge aus korrespondierenden Strangelementen bestehen, bei denen das jeweils höher geordnete auf das untergeordnete zugreift.

Bei einer vollredundanten Architektur muss entschieden werden, welches Clusterelement die jeweilige Anfrage bearbeitet. Sowohl für die Art und Weise, wie das für die Verarbeitung der anstehenden Anfrage verantwortliche Clusterelement ausgewählt wird, als auch für die Strategien, nach denen die Anfragen auf die einzelnen Elemente verteilt werden, gibt es unterschiedliche Lösungsmöglichkeiten.

1.5 Anwendung der Ressourcenverteilung

Die zuvor genannten Cluster-Arten werden, abhängig davon, wie robust ein Prozess implementiert werden muss, angewandt.

Gegen Gefahren, die sich nur bei einer Ressource auswirken, wie Hardwarefehler, Krankheit o. Ä, reicht ein Cluster an einer Lokation (auch Single-Site-Knoten genannt).

Möchte man auch gegen großflächige Schadensereignisse (Katastrosphen wie Hochwasser, Flugzeugabsturz, Brand oder Vandalismus) Vorkehrungen vornehmen, die eine gesamte Lokation treffen, sind die Cluster geografisch zu verteilen. Dazu werden die Cluster-Knoten oft mehrere Kilometer auseinander in verschiedenen Rechenzentren platziert. Diese Art von Clustern nennt man auch „stretched Cluster“, „multiple-Site-cluster“ oder geografisch verteilte Cluster (*engl.: „geographically dispersed“*).

Bei Gefahren, die global nahezu zeitgleich auftreten, wie z. B. Computer-Viren reichen die bislang aufgeführten Anwendungsansätze nicht. Auch gegen Gefahren wie Programmierfehler, die durch Redundanz der gleichen Ressource an einem oder an mehreren Orten nur identisch vervielfältigt

werden, muss die Funktionalität redundant ausgelegt werden. Ähnlich verfährt man bei menschlichen Ressourcen wie Entscheidungsträgern. Man spricht dann von heterogenen Clustern. So werden wichtige Entscheidungen von mehreren Personen mit unterschiedlichem Erfahrungshintergrund getroffen und z. B. in der Raum- und Luftfahrt Systeme unterschiedlich gedoppelt unterschiedlich realisiert oder programmiert. Als prominentes Beispiel bzgl. unterlassener heterogener Redundanz sei die Explosion der ersten Ariane-Rakete beim Start aufgrund eines Fehlers in der Fließkommaarithmetik eines weltweit verbreiteten Rechnerchips genannt. Dieser Aufwand zur Ausfallsicherheit wird z. B. dann erforderlich, wenn Menschenleben durch fehlerhafte Entscheidungen, Handlungen, Algorithmen, Hard- oder Software in Gefahr gebracht werden können.

Alle namhaften und weitverbreiteten Betriebssysteme sowie Standard-Datenbank-Systeme beinhalten die Funktionalitäten für Single- und Multi-Site-Cluster auf homogener Basis. Im Großrechner-Bereich, stark verbreitet in der Kreditwirtschaft und im militärischen Bereich, erreicht man in der Praxis auf diesen Architekturen aufbauend Verfügbarkeitsraten von mehr als 99,999 % (sog. „five-nines“), was einer Ausfallzeit von unter 5 Minuten pro Jahr bei 7 Tagen mit 24 Stunden Betrieb entspricht.

Im Multi-Site-Bereich setzen allerdings die enorme Datenrate und die derzeitigen technischen und wirtschaftlich vernünftig darstellbaren Übertragungsmöglichkeiten eine Grenze von etwas mehr als 20 km für die räumliche Trennung von Clustern. Damit lassen sich Komplexe realisieren, die den meisten großflächigen Schadensereignissen nicht gleichzeitig ausgesetzt sind und trotzdem diese hohe Verfügbarkeit auch im Katastrophenfall gewährleisten.

Mit spezieller Cluster-Software können auch heterogene Cluster auf Betriebssystem-Ebene aufbaut werden. Heterogene Anwendungsfunktionen sind allerdings bislang speziellen Entwicklungen vorbehalten, weil die Synchronisation über Standardschnittstellen im Allgemeinen nicht ausreichend ist.

Dienste und Anwendungen, deren gesamte Datenhaltung auf clusterfähigen DB-Sytemen aufbaut und nicht direkt Betriebssystem-Funktionalitäten nutzen, sind automatisch clusterfähig, wenn sie auf einem Cluster-DB- und -Betriebssystem ablaufen. Andernfalls müssen diese Services speziell für Cluster konzipiert und realisiert werden, um auf Ausfälle geeignet zu reagieren und die Kontinuität selbst zu gewährleisten. Clusterfähige Anwendungen werden auch als „cluster aware“ bezeichnet.

Die Anwendung der Clustertechnologie für unterschiedliche Verfügbarkeitsanforderungen wird in der folgenden Grafik noch einmal anschaulich dargestellt.

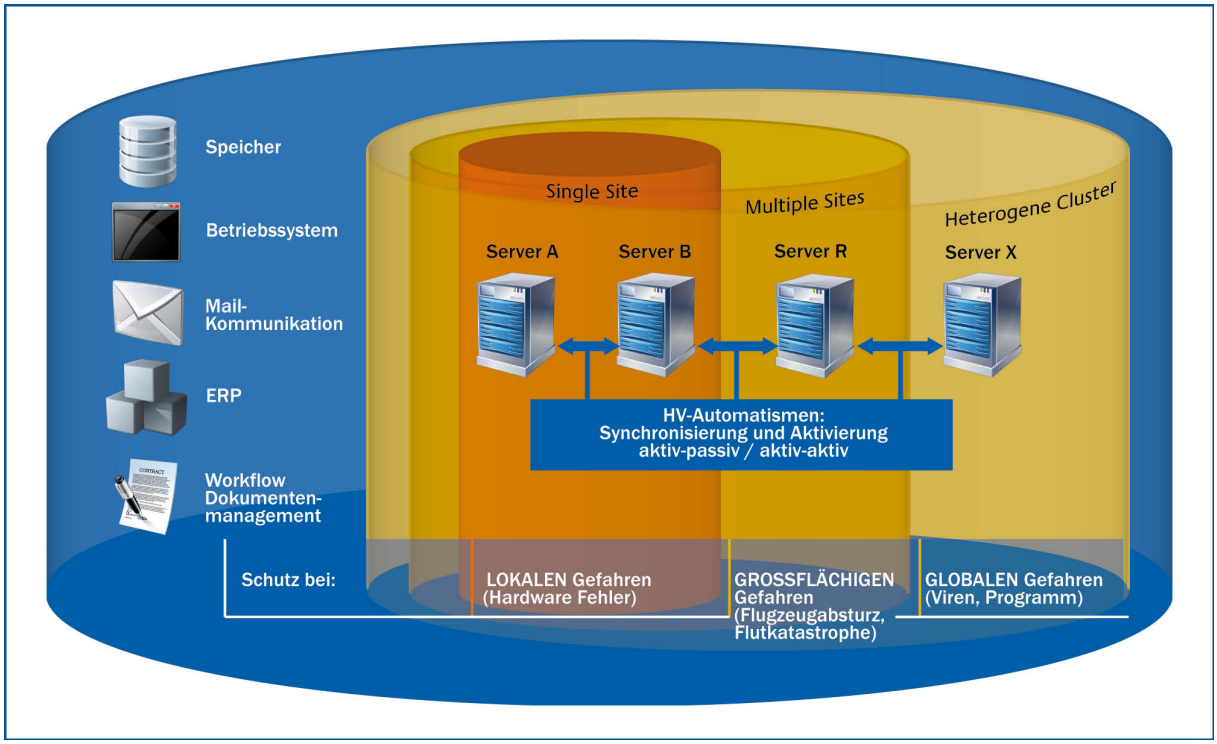


Abbildung 7: Anwendung der Cluster-Architektur

Anhang: Verzeichnisse

Abkürzungsverzeichnis

Ein komplettes Verzeichnis hierzu findet sich in Band AH, Kapitel 5

Glossar

Ein komplettes Verzeichnis hierzu findet sich in Band AH, Kapitel 6

Literaturverzeichnis

Ein komplettes Verzeichnis hierzu findet sich in Band AH, Kapitel 7