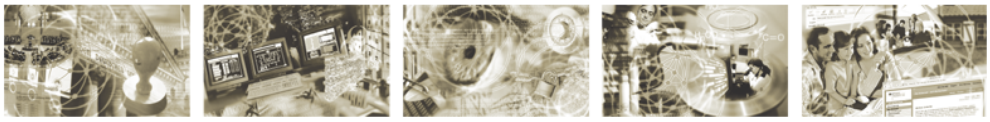
Bundesamt
für Sicherheit in der
Informationstechnik



Band 2, Kapitel 5: Server

(Hardware)

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

Inhaltsverzeichnis

1 Server (Hardware) 7
2 Definition des Begriffs "Server" 8
2.1 Server als Bezeichnung für Dienste (Services) 8
2.1.1 Client-Server-Modell 10
2.1.2 X-Window-System 12
2.1.3 Peer-to-Peer-Modell 13
2.1.4 Host-Terminal-System 15
2.1.5 Cluster-Computing 17
2.1.6 Grid-Computing 18
2.2 Server als Bezeichnung für Rechner-Hardware 19
2.2.1 Physische Server 19
2.2.2 Virtuelle Server 22
3 Server-Kategorien 23
3.1 Physische Server 23
3.1.1 Supercomputer 23
3.1.2 Mainframe 23
3.1.3 Enterprise-Server 24
3.1.4 Workstation 24
3.1.5 Blade-Server 25
3.1.6 Microcomputer 26
3.1.7 Embedded System 27
3.2 Virtualisierung 28
3.2.1 Auf Basis von Betriebssystemen 28
3.2.2 Auf Basis von Emulation 29
3.2.3 Hypervisor/Hardware-Virtualisierung 30
4 Server-Architektur 33
4.1 Verteilte Systeme 33
4.1.1 Vor- und Nachteile von verteilten und zentralen Systemen 33
4.1.2 Software Architekturen verteilter Systeme 35
4.1.3 Hardware-Architekturen verteilter Systeme 36
4.2 Betriebssystem-Architektur 37
4.2.1 Netzwerk-Betriebssystem 37
4.2.2 Verteiltes Betriebssystem 37
4.2.3 Multiprozessor-Betriebssystem 38
4.3 Rechner-Core-Architekturen 38
4.3.1 Havard-Architektur 38
4.3.2 Von-Neumann-Architektur 39
4.4 Rechner-Eingabe-Ausgabe 40
4.4.1 Netzwerk (LAN / WAN) 41
4.4.2 Storage (SCSI / SATA / Firewire) 42
4.4.3 Konsole (RS232 / V.24) 43
4.4.4 Drucker (Parallel / Seriell) 43
4.4.5 Management (In-Band / Out-of-Band) 44
4.5 Interne Rechner-Komponenten 44
4.5.1 CPU (Central Processing Unit) 44
4.5.2 Hauptspeicher 45
4.5.3 Boot-ROM / PROM / EEPROM 45

4.5.4 Backplane / Mainboard....45
4.5.5 Erweiterungsports....46
4.5.6 Netzteil....46
4.5.7 Lüfter....46
4.5.8 Inter-Prozessor-Kommunikation....47
4.5.9 Dark-Fibre / City LAN / PDH / SDH....47
4.5.10 Wavelength-division multiplexing (WDM)....48
4.5.11 ESCON Channel / FICON....49
4.5.12 Backplane / Parallele Bus Systeme....49
4.6 Kopplung von Server-Systemen....50
4.6.1 Punkt-zu-Punkt....52
4.6.2 Bus / Stern....52
4.6.3 Ring / logisch / physikalisch....53
4.6.4 Vermaschung....54
5 Geografische Redundanz....56
5.1 Geografische Verteilung....57
5.1.1 Lokal....58
5.1.2 Regional....58
5.1.3 National....59
5.1.4 International....60
5.2 Ausstattung der Standorte....60
5.2.1 Cold Site....60
5.2.2 Warm Site....61
5.2.3 Hot Site....61
5.2.4 Kollokation....61
6 Heterogene und zeitliche Redundanz....63
7 Server und Verfügbarkeit....64
7.1 Leistungsübergabepunkte und Abhängigkeiten....64
7.1.1 Bereitgestellte Server-Dienste (Services)....64
7.1.2 Abhängigkeiten von internen Komponenten....64
7.1.3 Umwelteinflüsse und Abhängigkeiten von externen Ressourcen....65
7.2 Fehlertoleranz und Automatismen....65
7.2.1 Überwachung....66
7.2.2 Fehlerbehandlung....67
7.2.3 Betriebskonzept....68
8 Zusammenfassung....69
Anhang: Verzeichnisse....71
Abkürzungen und Akronyme....71
Glossar....71
Literaturverzeichnis....71

Abbildungsverzeichnis

Abbildung 1: Das klassische Client-Server-Modell....10
Abbildung 2: Verteilte Client-Server-Rollen....11
Abbildung 3: Peer-to-Peer-Modell....14
Abbildung 4: Host-Terminal-System....16
Abbildung 5: Virtualisierung eines physischen Servers....29

Abbildung 6: Virtualisierung auf der Basis der Hypervisor Architektur.....31
Abbildung 7: Kopplung von Server-Systemen in einer Bus Topologie.....52
Abbildung 8: Kopplung von Server-Systemen in einer Stern-Topologie.....53

Tabellenverzeichnis

Tabelle 1: Verteilte Systeme.....33
Tabelle 2: Übersicht Flynn-Klassifikation.....39

1 Server (Hardware)

Die zunehmende Automatisierung von geschäftskritischen Abläufen erfordert eine effiziente, skalierbare und ständig verfügbare IT-Infrastruktur. Sowohl bei der Realisierung und dem sicheren und hochverfügbaren Betrieb von komplexen E-Government-Umgebungen als auch bei Rechenzentren oder Web-Portalen für Unternehmen und Organisationen bildet sowohl die Server-Hardware als auch die Server-Software neben dem Netzwerk eine tragende Säule in der IT-Infrastruktur.

Der Schwerpunkt liegt in diesem Beitrag auf der Server-Hardware, wobei sich in vielen Punkten eine direkte Abhängigkeit zu der darauf aufsetzenden Server-Software ergibt. Daher wird soweit nötig auch die Server-Software betrachtet bzw. es wird auf den jeweiligen Beitrag des HV-Kompendiums verwiesen, in den die Themen ausführlich dargestellt werden.

2 Definition des Begriffs "Server"

Unter dem Begriff „Server“ (*engl. to serve = bedienen*) wird meist die Kombination aus der physischen Server-Hardware (Rechner, Computer) in Verbindung mit dem darauf aufsetzenden Betriebssystem sowie einer oder mehreren Anwendungen verstanden, welche letztendlich einem über das Netzwerk verbundenen Client (*dt. Kunde*) die Server-Dienste zur Verfügung stellen.

Ausgehend von dieser eher allgemeinen Definition werden im Folgenden nun die verschieden gebräuchlichen Ausprägungen des Begriffs „Server“ dargestellt und in Bezug auf das Thema Hochverfügbarkeit betrachtet. Dabei wird ersichtlich, dass es neben einer Typisierung hinsichtlich der angebotenen Server-Dienste auch verschiedene Ansätze für die Verteilung der Server-Dienste gibt.

Abgeleitet aus dieser noch sehr abstrakten Sichtweise auf das Thema Hochverfügbarkeit werden in den weiteren Kapiteln konkrete Lösungen dargestellt, mit denen eine Hochverfügbarkeit der Server-Hardware, fallbasiert auch unter Verwendung der darauf aufsetzenden Server-Software oder einer entsprechender Ausgestaltung der Peripherie erreicht werden kann.

2.1 Server als Bezeichnung für Dienste (Services)

Unter dem Begriff „Server“ kann, abhängig von dem Kontext in dem der Begriff „Server“ verwendet wird, entweder die Kombination aus einem physischen Server und der darauf ablaufenden Software oder auch nur die Server-Software oder nur die Server-Hardware verstanden werden. In diesem Kapitel soll darunter eine Software-Komponente für die Erbringung von Diensten (Services) verstanden werden, die auf einer nicht näher spezifizierten Rechner-Hardware abläuft.

Eine Betrachtung der verschiedenen Ausprägungen von Rechner-Hardware als Ablaufumgebung für die Server-Software erfolgt in dem Kapitel „4.1.2 Software Architekturen verteilter Systeme“ und „4.1.3 Hardware Architekturen verteilter Systeme“, wobei sowohl auf die real existierende und physisch vorhandene Server-Hardware als auch auf die Virtualisierung von real existierender Server-Hardware eingegangen wird.

Im allgemeinen Sprachgebrauch wird für eine Typisierung eines Servers häufig der angebotene Dienst vorangestellt, sodass sich hier Bezeichnungen wie z. B. Mail-Server, Web-Server, Datei-Server, Datenbank-Server, usw. ergeben. Diese Konstellation bezeichnet meist die Gesamtheit von Server-Hardware, netzwerkfähigen Betriebssystem und der Anwendung, die diesen Dienst erbringt.

Dieser Dienst kann entweder von einer Software-Instanz auf einem physischen Server oder von mehreren Software-Instanzen, die wiederum auf einen oder aber auch auf verschiedenen, eventuell auch geografisch verteilten, physischen Servern laufen, erbracht werden.

Im Sinne der Hochverfügbarkeit ist es wichtig, dass aus der Sicht des Clients mindestens eine dieser Software-Instanzen erreichbar ist und die Anfrage des Clients mit einem akzeptablen Antwortzeitverhalten bedient wird. Ein weiteres wichtiges Kriterium neben der generellen Erreichbarkeit der Server-Dienste stellen die insgesamt zur Verfügung stehenden Server-Ressourcen dar. Nur wenn die Rechner über ausreichende Ressourcen (Prozessor, Hauptspeicher, Storage, usw.) verfügen, können die Anfragen der Clients innerhalb einer vorgegebenen Reaktionszeit bearbeitet werden.

Wird der Server-Dienst von mehreren, physischen Servern erbracht, so sollte bei einem Ausfall eine für den Client möglichst unbemerkte Übernahme durch einen noch verfügbaren Server erfolgen. Dazu müssen die einzelnen physischen Server untereinander über ein Netzwerk in Verbindung

stehen, um auf gemeinsame Datenbestände zuzugreifen oder die aktuelle Sitzung eines Clients zu übernehmen. Eine Betrachtung dieser Server-Verbindungen unter Hochverfügbarkeitsaspekten erfolgt in den Kapiteln 4.5.8 „Inter-Prozessor-Kommunikation“ und 4.6 „Kopplung von Server-Systemen“.

Eine Sonderstellung bei den gemeinsam genutzten Server-Ressourcen ergibt sich bei der Ankopplung an einen gemeinsam genutzten Massenspeicher, da hierfür unter anderem eigene Netzwerke in Form eines Storage Area Network (SAN) zur Verfügung gestellt werden können. Für eine Betrachtung der hochverfügbaren Ankopplung von verteilten Massenspeichern wird hier auf den Beitrag „Speichertechnologien“ des HV-Kompendiums verwiesen.

Für eine sinnvolle Verteilung der insgesamt zur Verfügung stehenden Server-Ressourcen können verschiedene Ansätze gewählt werden. So können Server-Ressourcen z. B. unter dem Aspekt der Lastverteilung zugeordnet werden, um eine gleichmäßige Verteilung der Server-Anfragen zu gewährleisten. Weitere Ansätze sind in dem Kapitel 2.1.5 „Cluster-Computing“ beschrieben. Allen Ansätzen gemeinsam ist die Tatsache, dass nur die Ressourcen (Rechenleistung, Speicher, usw.) zur Verfügung gestellt werden können, die über das Netzwerk erreichbar und mit einer real existierenden und betriebsbereiten Server-Hardware unterlegt sind.

Sofern der Client nicht lokal auf dem gleichen physischen Server wie der Server-Dienst abläuft, muss eine ausfallsichere Netzwerkverbindung zwischen allen Clients und den Server-Diensten existieren. Die Netzwerkverbindung muss dabei so ausgelegt sein, dass sie den erforderlichen Durchsatz (messbar z. B. in Megabit pro Sekunde) mit einer maximal tolerierbaren Latenzzeit erbringt. Für eine genauere Betrachtung zu dem Thema Netzwerk und Hochverfügbarkeit wird auf den Beitrag „Netzwerk“ des HV-Kompendiums verwiesen. Für eine Betrachtung der hochverfügbaren Ankopplung der Server-Hardware an die zugrundeliegende Netzwerkumgebung sei hier auf das Kapitel 4.4.1 „Rechner Eingabe-Ausgabe“ verwiesen.

Bei den Clients, die entweder lokal oder über ein Netzwerk auf die Server-Dienste zugreifen, kann es sich zum einen um Endgeräte handeln, die für eine Mensch-Maschine-Kommunikation ausgelegt sind (Mail-Clients, Web-Browser). Zum anderen kann es sich bei diesen Clients aber auch um automatisierte Agenten handeln, die eine Anfrage an einen Server senden, um von diesem die notwendigen Informationen für die Bewältigung einer Teil-Aufgabe innerhalb eines größeren Gesamtsystems zu erhalten. Als Beispiel für eine solche Maschine-Maschine-Kommunikation sei hier das DNS (Domain Name System) angeführt, mit der ein Client eine Anfrage an eine auf einen Server laufenden Dienst (Service) stellt, um z. B. die IP-Adresse zu einen Domainnamen herauszufinden.

Bei den von einem oder mehreren Servern zur Verfügung gestellten Server-Diensten ist neben der generellen Verfügbarkeit auch eine Reaktionszeit innerhalb der vorgegebenen Grenzen von großer Bedeutung. Handelt es sich bei der Client-Server-Verbindung z. B. um eine Mensch-Maschine-Kommunikation mit einer interaktiven Anwendung, so wird von den Nutzern in der Regel eine Reaktionszeit von wenigen Sekunden erwartet. Anders verhält es sich hier wiederum bei einer Stapelverarbeitung, wo die Aufträge in eine Warteschlange des Servers gestellt werden. Dabei kann die Verbindung zwischen Client und Server für den Zeitraum der Abarbeitung unterbrochen werden und erst wieder zum Zeitpunkt der Fertigstellung der Arbeitsergebnisse aktiviert werden.

Während bei der Stapelverarbeitung die Arbeitsaufträge, die Auslastung der Rechner-Ressourcen und die notwendigen Redundanzen für eine ausreichende Fehlertoleranz noch vorausschauend geplant werden können, sind die Anforderungen an eine hohe Verfügbarkeit für eine interaktive Anwendung deutlich höher gesteckt. Für eine laufende Überwachung der tatsächlich zur Verfügung stehenden Server-Ressourcen bieten sich hier geografisch verteilte Messsonden an, die die

Anfragen eines Clients simulieren und die Ergebnisse hinsichtlich Reaktionszeit an einen zentralen Leitstand übermitteln. Für weitere Hinweise hinsichtlich der Überwachung von Server-Diensten wird hier auf den Beitrag „Überwachung“ im HV Kompendium verwiesen.

2.1.1 Client-Server-Modell

Populär wurde das klassische Client-Server-Modell (siehe Abbildung 1) in den 1980er Jahren im Rahmen des damals weitverbreiteten Downsizing-Trends. Mit dem Downsizing wurde das Ziel verfolgt, die meist teuren und vielfach unflexiblen Großrechner und Minicomputer durch kostengünstigere Client-Server-Architekturen abzulösen.

Die zugrundeliegende Architektur und Technologie wurde dabei in vielen Punkten an die bestehenden Ansätze aus der UNIX-Welt angelehnt, da hier schon verschiedene Ansätze für eine verteilte Rechnerarchitektur verfügbar waren. Neben den grundlegenden Prinzipien einer verteilten Client-Server-Architektur wurde später auch das TCP/ IP-Netzwerkprotokoll für die Kommunikation zwischen dem Client und dem Server aus der UNIX-Welt übernommen.

Ein zentraler Punkt bei dem damaligen Downsizing war die möglichst vollständige Ablösung der Großrechner und Minicomputer durch entsprechend leistungsstarke Rechner auf Basis eines Personal Computers (PC). Ausgestattet mit leistungsstarken Prozessoren und einem ausreichend dimensionierten Speicher konnten diese PCs die Funktion eines Servers wahrnehmen, welcher den Clients z. B. eine zentrale Dateiablage, eine Datenbank oder auch eine Anwendung zur Verfügung stellt. Als Clients wurden die damals grade populär gewordenen PCs verwendet, die teilweise auch schon über eine graphische Benutzeroberfläche verfügten.

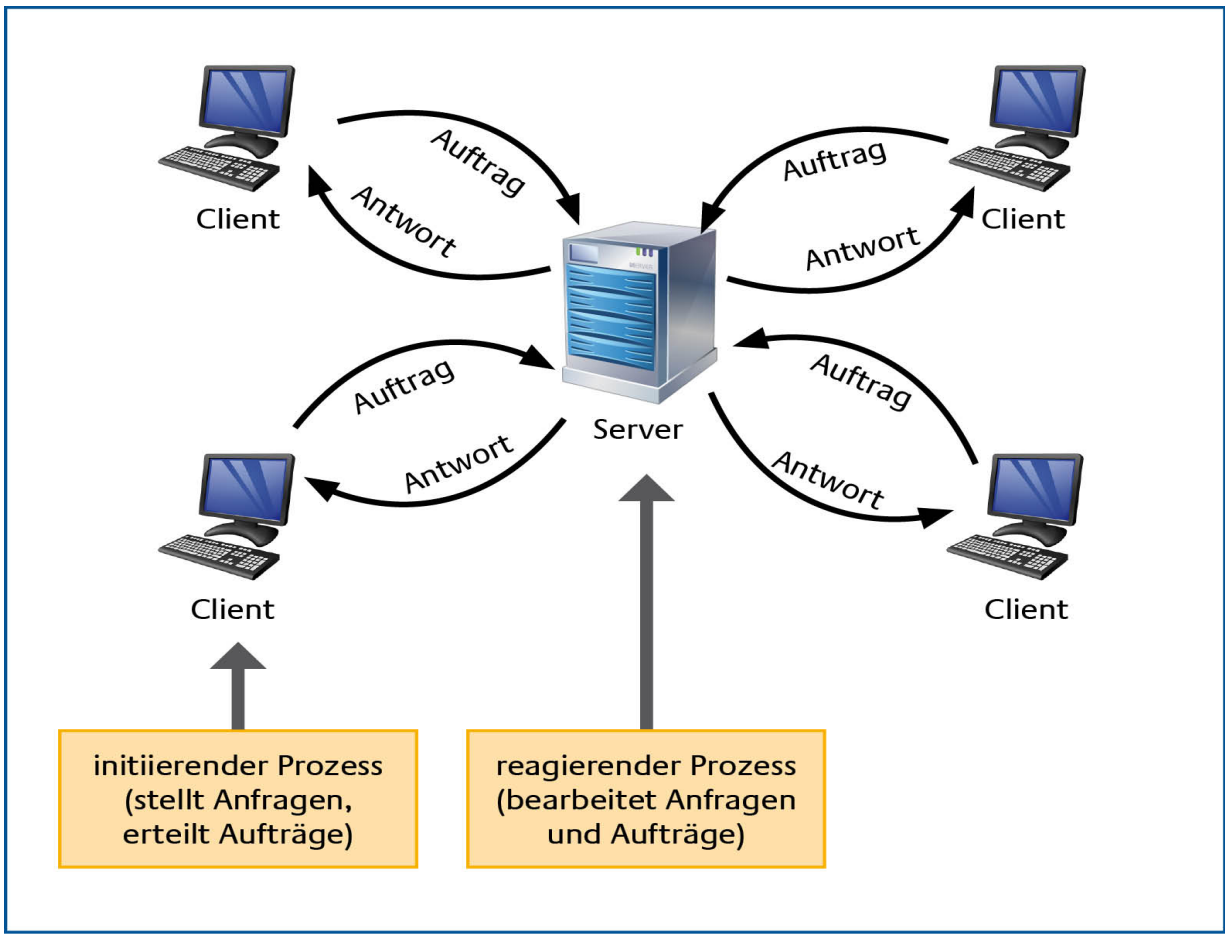


Abbildung 1: Das klassische Client-Server-Modell

Während das „klassische“ Client-Server-Modell noch von einem Server und einer bestimmten Anzahl von physisch getrennten Clients ausgeht, sind die Kommunikationsbeziehungen und Abhängigkeiten bei den heutigen Client-Server-Architekturen () deutlich komplexer.

So kann es für einen Server z. B. notwendig sein, gleichzeitig bzw. nacheinander die Rolle eines Servers und eines Clients einzunehmen. Wie in der nachstehenden “ ersichtlich, nimmt der Datei-Server in Bezug auf die Anfrage des Client im ersten Schritt die Rolle eines Servers ein. Den Ausgangspunkt bildet hier die Anfrage eines Clients, der auf eine auf dem Datei-Server abgelegte Datei zugreifen möchte.

Auf dem Datei-Server läuft nun ein Server-Dienst, welcher zum einen die Anfrage des Clients entgegennimmt und zum anderen prüfen muss, ob der Client überhaupt berechtigt ist auf die auf dem Datei-Server abgelegte Datei zuzugreifen. Für diese Prüfung benötigt ein Datei-Server in der Regel noch weitere Informationen, die dem Datei-Server von weiteren Servern zur Verfügung gestellt werden.

Um diese Informationen zu erhalten, kann der Server-Dienst auf dem Datei-Server auch die Rolle eines Clients annehmen und in dieser Funktion wiederum Anfragen sowohl an den Zeit-Server als auch an den Authentifizierungs-Server stellen. Sobald der Datei-Server eine Reaktion auf seine Anfragen erhalten hat, kann der Server-Dienst auf dem Datei-Server mit der Bearbeitung der ursprünglichen Client-Anfrage fortfahren. Dabei wird dem Client der Zugriff auf die Datei gestattet, sofern die Authentifizierungsmerkmale zum Zeitpunkt des Zugriffs auf die Datei gültig waren und mit der vorgegebenen Identität des Clients übereinstimmen.

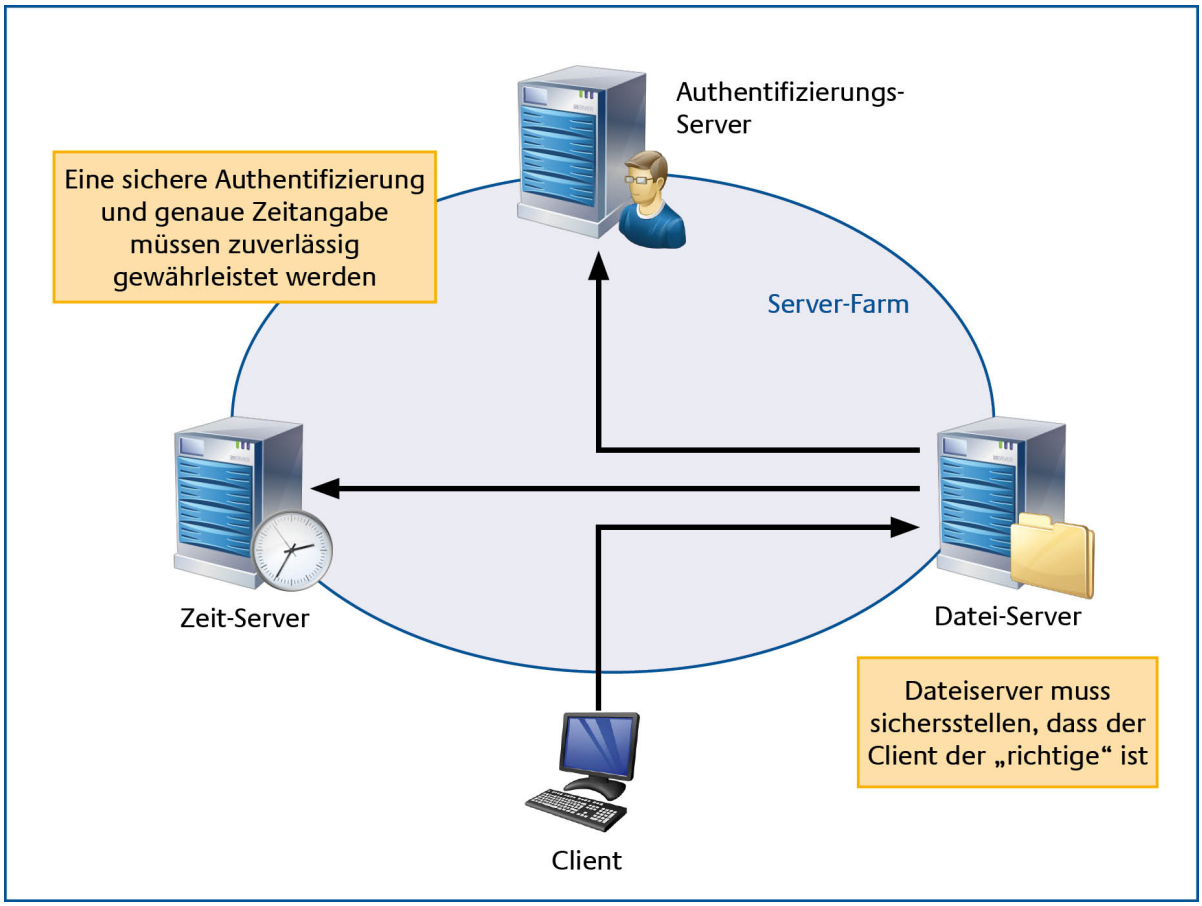


Abbildung 2: Verteilte Client-Server-Rollen

Client- und Server-Funktionalitäten müssen sich nicht zwangsläufig auf physischen getrennten Maschinen befinden. Auch eine Vereinigung der Server- und Client-Rolle auf einen physischen Rechner ist häufig anzutreffen, um eine sinnvolle Strukturierung des Programmcodes zu erreichen.

Für eine Betrachtung der Hochverfügbarkeit müssen alle Abhängigkeiten zwischen den Client- und Server-Diensten einer verteilten Anwendung identifiziert werden, die sich auf einen gegebenen Geschäftsvorfall zurückführen lassen und durch eine Client-Anfrage initiiert werden. Nur wenn alle für den Geschäftsvorfall notwendigen Client- und Server-Instanzen die notwendigen Ressourcen sowie die Kommunikationsbeziehungen zwischen den Client- und Server-Instanzen verfügbar sind, kann der Geschäftsvorfall abschließend bearbeiten werden.

Um die Abhängigkeiten von den diversen und eventuell auch noch geografisch verteilten Server-Diensten zu verringern, können die Server- und Client-Dienste auch auf einer hochverfügbaren Server-Farm zusammengefasst werden. Dabei wird die Komplexität des Clients so weit reduziert, dass dieser nur noch eine Eingabe- und Ausgabefunktion wahrnimmt und alle Abhängigkeiten in zentralen Server-Farmen konzentriert werden. Bei dieser Variante wird der Client auch häufig Thin-Client genannt, da auf dem Client praktisch keine Rechnerressourcen benötigt werden.

Im Gegensatz dazu können in dem Client-Server-Modell auch so genannte „Fat-Clients“ verwendet werden, die sich dadurch auszeichnen, dass Teile der Programmlogik auf den Client ausgelagert werden und somit zu einer deutlichen Entlastung des Servers beitragen können. Als Nachteil kann sich hier – abhängig von der konkreten Anwendung – der hohe Kommunikationsbedarf und damit die Abhängigkeit zu diversen, verteilten Servern erweisen. Da die absolute Anzahl der Clients meist höher ist als die Anzahl der Server, wird die Wahrscheinlichkeit eines Ausfalls von einzelnen oder Gruppen von Clients deutlich erhöht, da hier sowohl die Komplexität des Gesamtsystems als auch die Anzahl der Kommunikationsbeziehungen deutlich höher ist als bei der „Thin-Client-Variante“.

2.1.2 X-Window-System

Das X-Window-System (auch X Version 11, X11 oder X genannt) ist ein Verfahren, welches eine grafische Benutzeroberfläche über ein zugrundeliegendes Netzwerk quasi fernsteuern kann. Dabei wird durch einen sogenannten X-Server eine grafische Oberfläche für eine Mensch-Maschine-Kommunikation bereitgestellt, die wiederum von einem X-Client ferngesteuert wird.

Das X-Window-System wurde 1984 im Rahmen des Projektes Athena in einer Zusammenarbeit mit dem MIT (Massachusetts Institute of Technology) und einigen kommerziellen Anbietern von Computern entwickelt und basiert auf den Prinzipien des zuvor dargestellten Client-Server-Modells:

- Der X-Server läuft auf dem Arbeitsplatzrechner eines Anwenders und stellt seine (grafischen) Dienste den X-Clients zur Verfügung. Er enthält den Grafikkartentreiber sowie Treiber für Tastatur, Maus und andere Eingabegeräte (wie z. B. Grafik-Tablets) und kommuniziert mit dem X-Client über das Netzwerk.
- Der X-Client ist das Anwendungsprogramm, das die grafischen Ein-/ Ausgabe-Dienste des X-Servers benutzt. Er kann auf demselben, oder auch auf irgendeinem entfernten Rechner laufen (sofern eine Netzwerkverbindung zwischen beiden besteht). Der X-Client benutzt die Dienste des X-Servers, um eine grafische Darstellung zu erreichen und empfängt von ihm die diversen Ereignisse (*engl. events*) wie Tastenanschläge, Mausbewegungen, Klicks usw.

Im Gegensatz zu dem zuvor dargestellten Client-Server-Modell fällt auf, dass hier der X-Server „nur“ für die grafische Benutzerschnittstelle benötigt wird, während der X-Client die Programmlogik für die Abbildung der Geschäftsprozesse enthält. Abweichend von der Client-Server-Architektur bildet hier der X-Client die zentrale Instanz muss daher unter Hochverfügbarkeitsaspekten als kritische Ressource angesehen werden. Der Ausfall eines X-

Servers ist in der Regel unkritisch, da jederzeit die grafische Benutzeroberfläche eines anderen X-Servers übernommen werden kann.

Bei dem Ausfall des X-Client hingegen fallen auch alle von dem X-Client zur Verfügung gestellten Dienste für die Unterstützung der Geschäftsprozesse aus. Diese Terminologie aus der X11-Sicht führt vielfach zu Missverständnissen, da hier der X-Client die zentrale Ressource darstellt, wobei die X-Server nur für die grafische Benutzerschnittstelle benötigt werden.

Ebenso wie die Server in einer Client-Server-Architektur sind die X-Clients weitestgehend unabhängig von einer bestehenden Verbindung zu einem X-Server. Für den Wiederanlauf nach einer Störung der Netzwerkverbindung zwischen dem X-Client und dem X-Server ist in der Regel ein erneuter Aufbau der Verbindung notwendig. Der eigentliche Programmablauf auf dem X-Client wird dadurch in der Regel nicht beeinflusst und der Anwender kann nach Behebung der Netzwerkstörung und einem erneuten Verbindungsaufbau meist wieder uneingeschränkt weiterarbeiten.

2.1.3 Peer-to-Peer-Modell

Im Gegensatz zu dem zuvor dargestellten Client-Server-Modell besteht das Peer-to-Peer-Modell () aus einer losen Kopplung von einer (theoretisch) beliebigen Anzahl von physisch getrennten Rechnern. Die Bezeichnung Peer-to-Peer (*engl. Peer für „Gleichgestellter“, „Ebenbürtiger“*) soll dabei zum Ausdruck bringen, dass es sich hier um gleichberechtigte Rechner handelt, wobei jeder Rechner sowohl Dienste in Anspruch nehmen als auch Dienste zur Verfügung stellen kann. Die Rechner können somit sowohl als Arbeitsstationen genutzt werden als auch Anfragen aus dem Netz beantworten.

Die Rechner in einem Peer-to-Peer-Modell arbeiten meist völlig autonom und sind in der Regel auch keiner gemeinsamen administrativen Hoheit unterstellt. Im Gegensatz zu dem zuvor dargestellten Client-Server-Modell, in dem einzelne Server unter Umständen einen „Single Point of Failure“ darstellen können, haben Ausfälle einzelner Rechner in dem Peer-to-Peer-Netzwerk oder partielle Störungen in dem zugrundeliegendem Netzwerk, praktisch keine Auswirkungen auf die Verfügbarkeit.

Die einzelnen Rechner in dem Peer-to-Peer-Modell nutzen ein, auf dem TCP/IP-Protokoll basierendes, Netzwerk (vorwiegend das Internet), um darauf aufsetzend ein sogenanntes Overlay-Netzwerk aufzubauen. Dieses Overlay-Netzwerk beschreibt die Netzwerktopologie der einzelnen Peers, die über das Peer-to-Peer-System verbunden sind und stellt den einzelnen Peer-Rechnern grundlegende Server-Dienste, wie z. B. Lookup und Suche, zur Verfügung.

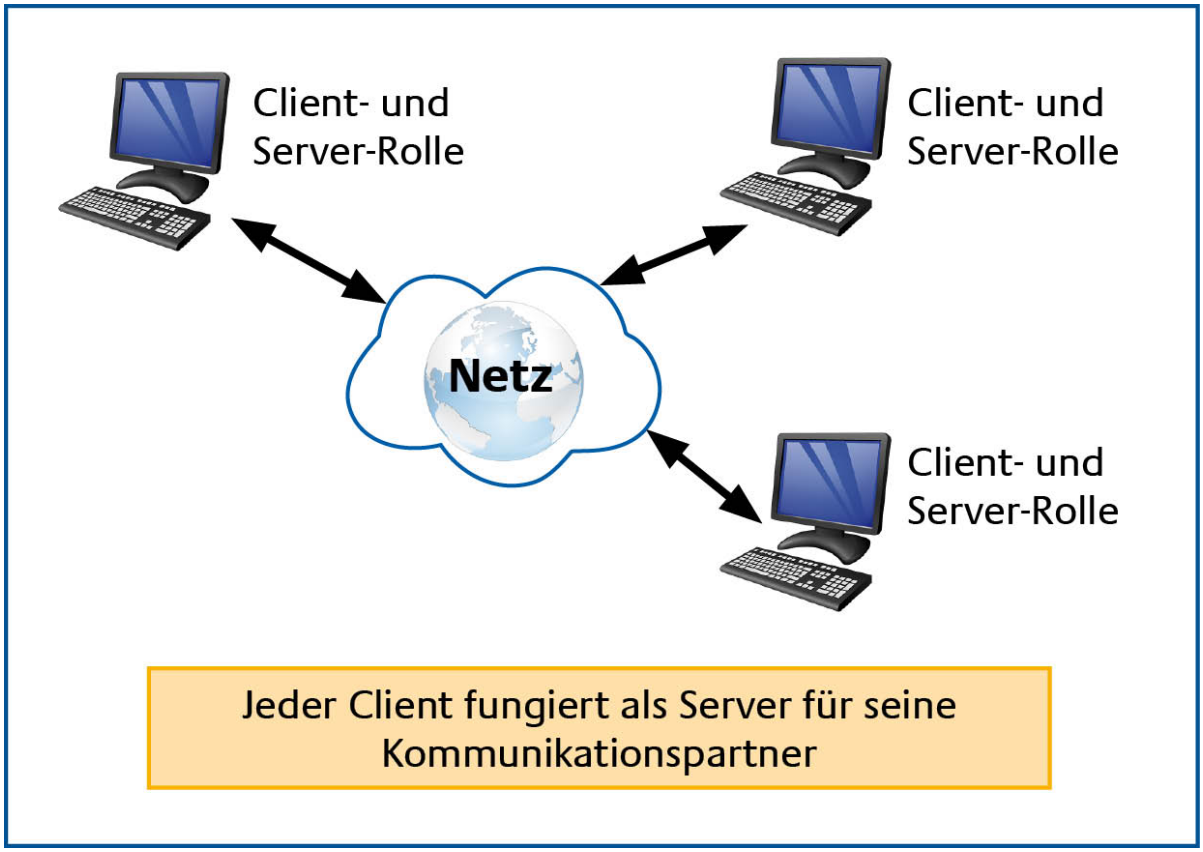


Abbildung 3: Peer-to-Peer-Modell

Mit der Lookup-Operation können Peers im Netzwerk diejenigen Peers identifizieren, die für eine bestimmte Objektkennung (Object-ID) zuständig sind. In diesem Fall ist die Verantwortlichkeit für jedes einzelne Objekt mindestens einem Peer fest zugeteilt, man spricht von strukturierten Overlays. Mittels der Such-Operation können die Peers nach Objekten im Netzwerk suchen die gewisse Kriterien erfüllen (z. B. Datei- oder Buddynamen-Übereinstimmung). In diesem Fall gibt es keine Zuordnungsstruktur für die Objekte im P2P-System; man spricht also von unstrukturierten Overlays.

Sobald die Peer-Rechner identifiziert wurden, die die gesuchten Objekte bereitstellen, bzw. die gesuchten Dienste anbieten, erfolgt eine direkte Kommunikation zwischen den beiden Peer-Rechnern. Dabei nimmt der Peer-Rechner, der das Objekt bereitstellt bzw. den Dienst anbietet, die Rolle des Servers und der Peer-Rechner, der das Objekt oder den Dienst angefragt hat, die Rolle des Clients ein.

Hinsichtlich der Verfügbarkeit können die folgenden drei Varianten unterschieden werden:

- Zentral organisierte Peer-to-Peer Netzwerke
  - Bei dieser Variante gibt es einen zentralen Server, der ein stets aktuelles Verzeichnis der derzeit verfügbaren Peer-Rechner bereitstellt. Sofern ein Peer-Rechner eine Server-Ressource benötigt, stellt dieser eine Anfrage an den zentralen Server und erhält als Antwort eine Liste der Peer-Rechner, die über die gewünschte Ressource verfügen. Ein zentrales Peer-to-Peer-Netzwerk hat den Nachteil, dass es nicht skalierbar ist und über einen Single Point of Failure verfügt.
- Dezentral, unstrukturierte Peer-to-Peer-Netzwerke

  - Bei dieser Variante gibt es weder ein zentrales Verzeichnis noch eine präzise Kontrolle über die Netzwerk-Topologie oder den Ort der Ressourcen. Das Netzwerk wird geformt durch Peer-Rechner, die sich am Netzwerk anmelden, ohne dabei genau festgelegte Regeln beachten zu müssen. Die gängigste Anfragemethode ist das sog. Flooding *(engl. Flooding für „fluten")*, wobei die Anfrage an alle Peer-Rechner in einem bestimmten Radius weitergeleitet wird. Diese unstrukturierten Konzepte sind extrem widerstandsfähig gegenüber sporadisch ein- und austretenden Peers. Jedoch stellen die Suchmechanismen ein großes Problem hinsichtlich der Skalierbarkeit der Peer-to-Peer-Netzwerke dar, da die Suche eine hohe Netzwerklast erzeugt und ein garantiertes Auffinden einer Ressource in einer bestimmten Zeit unmöglich macht.

- Dezentrale, strukturierte Peer-to-Peer-Netzwerke
  - In einem dezentralen, strukturierten Peer-to-Peer-Netzwerk ist ebenfalls keine zentrale Verzeichnisstruktur vorhanden, sie verfügen aber über eine signifikante Struktur. Die Topologie des P2P-Netzwerks und die Platzierung der Ressourcen basieren auf einem, allen Peers bekannten, Verfahren, welches das Auffinden von Ressourcen wesentlich vereinfacht. In wenig strukturierten Netzwerken können nur Hinweise auf die Platzierung einer Ressource erhoben werden, während in hoch strukturierten Netzwerken der Ort einer Ressource präzise bestimmt werden kann. Als eine typische Realisierung dieser Variante sei hier das Peer-to-Peer-System „Chord" angeführt, welches von dem Massachusetts Institute of Technology (MIT) entwickelt wurde und auf dem Distributed-Hash-Table (DHT) Verfahren basiert.

Während sich in dezentral strukturierten Netzwerken Verfügbarkeitsaussagen zum Teil allein aus der Struktur ableiten lassen, ist in einem dezentralen und unstrukturierten Peer-to-Peer-Netzwerk eine pauschale Verfügbarkeitsaussage generell nicht möglich. In strukturierten Peer-to-Peer-Netzwerken können aus Messwerten allgemeine statistische Aussagen abgeleitet werden, sodass eine grobe Schätzung über Verfügbarkeitskriterien (z. B. Lookup, Suche usw.) gemacht werden kann.

Dabei beruhen die meisten Annahmen und Rechenmodelle für die Verfügbarkeit eines Peer-to-Peer-Netzwerkes auf der Voraussetzung, dass aus einer zufälligen Auswahl von Peers zu einem gegebenen Zeitpunkt nur einzelne Rechner und nicht alle Peer-Rechner simultan ausfallen.

Bei einem direkten Vergleich mit dem Client-Server-Modell wird deutlich, dass der Ausfall eines Peer-Rechners die Verfügbarkeit nicht zwangsläufig in dem Maße reduziert, wie dies bei einem Ausfall eines Servers in dem Client-Server-Modell der Fall wäre. Allerdings wird jedoch bei dem Ausfall oder der Nicht-Erreichbarkeit von mehreren Peer-Rechnern zu einem bestimmten Zeitpunkt deutlich, dass die Verfügbarkeit des Gesamtsystems in einen wesentlich größeren Umfang beeinträchtigt wird.

2.1.4 Host-Terminal-System

Bevor sich das Client-Server-Modell durchsetzen konnte, war das Host-Terminal-System () das beherrschende Arbeitsprinzip in der Datenverarbeitung. Dabei wurde ein leistungsfähiger Zentralrechner (Host, Mainframe, Großrechner) eingesetzt, der über vorgeschaltete Konzentratoren oder Vorrechner eine meist größere Anzahl (>1000) an Terminals bedienen konnte. Dabei wurden auf dem Zentralrechner die notwendigen Ressourcen bereitgestellt, um die Daten vorzuhalten und die Applikationen typischerweise im Online-, Time-Sharing- oder Batch-Betrieb abzuarbeiten.

Aus Sicht des Terminals wurden die Tastatureingaben entweder einzeln oder als Zusammenfassung, für z. B. eine Eingabemaske oder eine Bildschirmseite, vom Terminal zum Host übertragen. Für die

Rückrichtung vom Zentralrechner zu dem Terminal, wurden Informationen für eine strukturierte Bildschirmausgabe, meist mit zusätzlichen Steuerzeichen, versehen. Im Gegensatz zu den Clients in dem zuvor vorgestellten Client-Server-Modell waren die Terminals hier reine Ein-/Ausgabe-Geräte ohne eigene Speicher- oder Rechenkapazität. Seit sich die PCs durchgesetzt haben, werden diese Terminals in der Regel durch sogenannte Terminalemulationen simuliert.

Auch die Verbindung zwischen dem Zentralrechner und den Terminals war von den Herstellern genau vorgegeben und hierarchisch strukturiert. Befanden sich etwa bis Ende der 1960er Jahre die Terminals meist ausschließlich in räumlicher Nähe zu dem Zentralrechner, wurde es in den 1970er Jahren möglich, die Terminals über ein hierarchisches Netzwerk aus Datenfernübertragungsleitungen mit dem zentralen Großrechner zu verbinden.

Der zentrale Großrechner bildete, zusammen mit den Vorrechnern (Front End Controllern), die Spitze der Netzwerk-Hierarchie. In den geografisch entfernten (*remote*) Standorten wurden häufig auch sogenannte Konzentratoren (auch Cluster-Controller genannt) eingesetzt, welche die Aufgabe hatten, eine Anzahl von Terminals zusammenzufassen und den Kommunikationsaufwand zu den zentralen Großrechner möglichst gering zu halten. Die Kommunikation funktionierte damals typischerweise in einem Polling-Verfahren, wobei das Großrechner-/Vorrechner-Gespann die Aufgabe hatte, entweder bei den Konzentratoren oder bei den Terminals direkt anzufragen, ob Daten vorliegen, die zum Zentralrechner übertragen werden sollen.

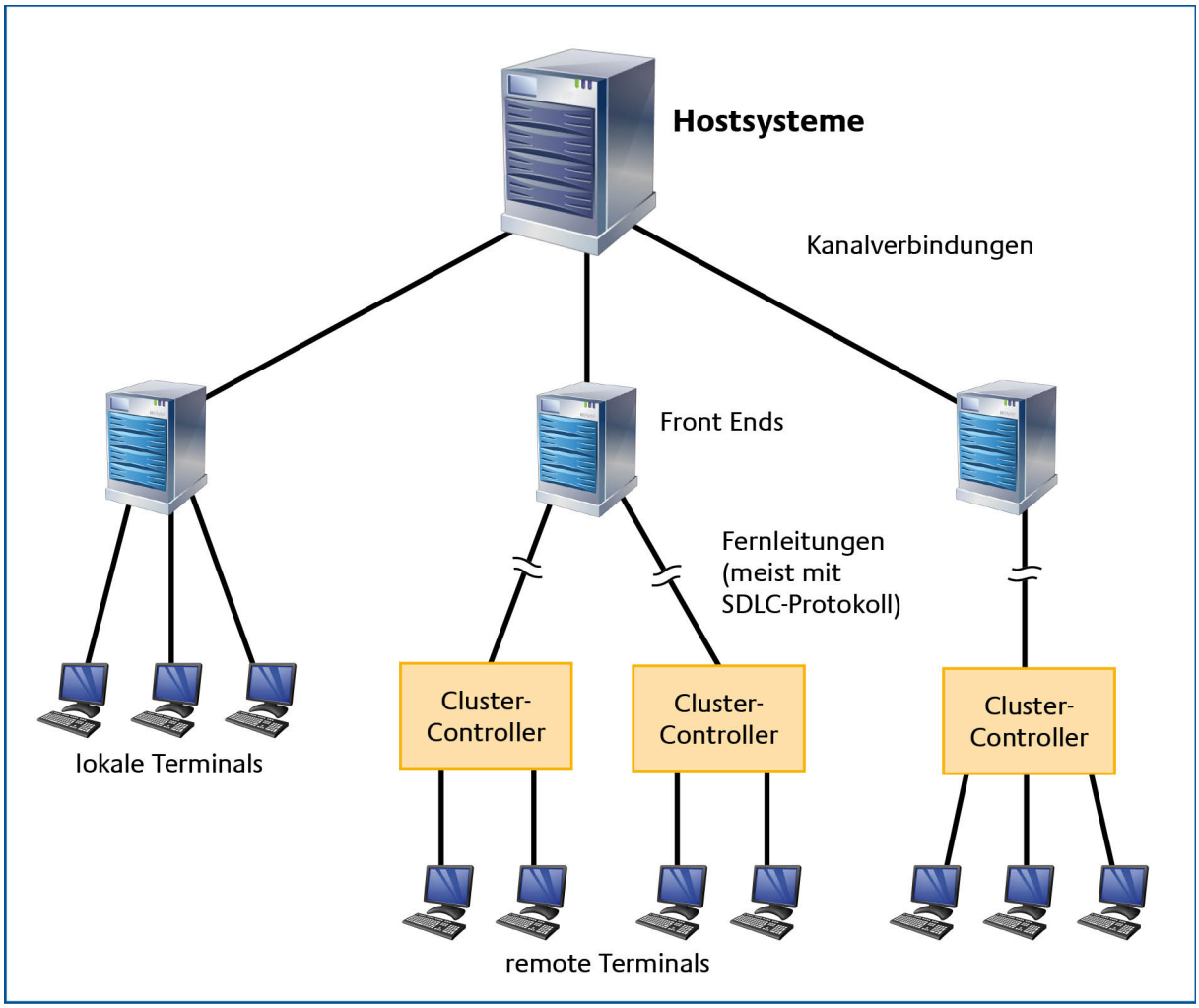


Abbildung 4: Host-Terminal-System

Diese ursprüngliche Aufteilung, mit einem zentralen Großrechner und einer Anzahl von Terminals, ist heute praktisch nicht mehr anzutreffen und als historisch anzusehen. Da von dem Zentralrechner in diesem ursprünglichen Host–Terminal-System unter Umständen mehrere Tausend Terminal-Arbeitsplätze abhängig waren, wurde bei der Auslegung der Hardware-Komponenten für den Zentralrechner schon seit jeher eine hochverfügbare Auslegung angestrebt.

Auch wenn das ursprüngliche Host–Terminal-System der Vergangenheit angehört, so bilden die Zentralrechner, in der Form von hochverfügbaren Großrechnern und Mainframes, aufgrund ihrer Leistungsfähigkeit und Robustheit in vielen Versicherungen, Banken, Institutionen und Behörden immer noch das Rückgrat für die Abbildung von unternehmenskritischen Geschäftsprozessen.

Anstatt über ein hierarchisches Netzwerk mit einfachen Terminals zu kommunizieren, werden diese Zentralrechner heutzutage meist als hochverfügbare Server eingesetzt, die in der Regel über ein TCP /IP-basiertes Netzwerk ihre Server-Dienste zur Verfügung stellen. Dabei werden, durch den Einsatz von virtuellen Maschinen und der damit verbundenen hohen Skalierbarkeit, dem Anwender verschiedene Wege zu individuellen Rechnerlösungen eröffnet. So können z. B. mehrere Betriebssysteme völlig abgeschottet voneinander auf einem Mainframe betrieben werden.

Auch können die Zentralrechner an geografisch weit auseinander liegenden Standorten platziert werden, sodass bei einem schwerwiegenden Zwischenfall (Katastrophen-Fall oder K-Fall) an einem gegebenen Standort, die Server-Dienste von einem weiteren Standort mit minimalen Ausfallzeiten, übernommen werden können. Auf die Möglichkeiten einer Kopplung von mehreren Großrechnern wird in dem Kapitel 4.5.8 „Inter-Prozessor-Kommunikation“ und in dem Kapitel 4.6 „“ näher eingegangen. Die Verteilung der Rechnerressourcen auf verschiedene Standorte ist Thema des Kapitels 5: „Geographische Redundanz“.

Im Gegensatz zu PC-basierenden Servern werden in einem Großrechner meist nur sorgfältig aufeinander abgestimmte Komponenten verbaut, die zudem hochgradig redundant ausgelegt sind, sodass sich hier schon von Haus aus eine relativ hohe Verfügbarkeit ergibt. Aus der Historie der Host-Terminal-Systeme war es bei den Großrechnern auch früher schon üblich, dass die Wartung dieser Rechner im laufenden Betrieb durchgeführt werden konnte. Häufig konnten sogar Aufrüstungen und Hardware-Austausch ohne Unterbrechung des Betriebs durchgeführt werden.

Diese traditionellen Stärken in Verbindung einer zukunftsweisenden Technologie sind sicher ein Grund dafür, dass sich die Großrechner noch für eine längere Zeit behaupten können.

2.1.5 Cluster-Computing

Der Begriff Cluster (von engl. Cluster – Schwarm, Gruppe, Haufen) steht in der Regel für eine Anzahl von vernetzten Rechnern, die sich gegenüber den Benutzern (Clients) wie ein einzelner Rechner darstellen. Mit dem Aufbau eines Clusters werden grundsätzlich zwei Zielsetzungen verfolgt. Eine der Zielsetzungen besteht darin, durch das Zusammenschalten mehrerer Einzelrechner eine Erhöhung der Rechenkapazität zu erreichen. Die andere Zielsetzung verfolgt die Erhöhung der Fehlertoleranz und Robustheit (z. B. bei dem Ausfall von einzelnen Computern) durch die Zusammenschaltung von mehreren, eventuell auch geografisch separierten Einzelrechnern.

Darüber hinaus kann zwischen den so genannten homogenen und heterogenen Clustern unterschieden werden. Die Rechner von homogenen Clustern laufen typischerweise unter dem gleichen Betriebssystem und auf der gleichen Hardware. In einem heterogenen Cluster können verschiedene Betriebssysteme und unterschiedliche Hardware eingesetzt werden.

Die bekannteste Form eines Clusters besteht meist aus einem homogenen Cluster, wo entweder das Betriebssystem oder ein Server-Dienst oberhalb der physischen Hardware virtualisiert wird. So ist

es z. B. möglich, ein Betriebssystem auf mehrere, real existierende Rechner aufzuteilen und den darauf aufsetzenden Anwendungen die Illusion eines einzelnen physischen Rechners mit einem jeweils eigenen Betriebssystem zu vermitteln.

Ein ähnliches Verfahren ist ebenso mit nahezu beliebigen Server-Diensten möglich. So kann z. B. eine Datenbank-Instanz auf mehrere, über ein Netzwerk verbundene physische Rechner-/Betriebssystem-Kombinationen verteilt werden. Auch hier ist aus der Sicht des Benutzers (Clients) der Zugriff auf die Datenbank völlig transparent, da für den Benutzer nicht ersichtlich ist, auf welchem konkreten physischen Rechner seine Datenbankabfrage abgearbeitet wird.

Neben dieser zuvor dargestellten Form einer Cluster-Bildung sind nahezu beliebig viele weitere Varianten möglich, welche in dem Beitrag „Cluster-Architekturen“ im HV-Kompendium ausführlich beschrieben sind. Ebenso wird in dem Beitrag „Datenbank“ des HV-Kompendiums auf den Aufbau eines hochverfügbaren Datenbank-Clusters eingegangen.

2.1.6 Grid-Computing

Die physischen Rechner in einem Cluster-Verbund sind (wie meist auch einzelne Rechner) in der Praxis nur selten wirklich ausgelastet. Das heißt, ein großer Teil der zur Verfügung stehenden Rechnerressourcen wird gar nicht genutzt. Daraus entwickelte sich die Idee, diese bisher ungenutzten Rechenkapazitäten für einen größeren Nutzerkreis über das Internet verfügbar zu machen.

Aus dieser ersten Idee entstand in den vergangenen Jahren eine neue Technologie mit der Bezeichnung „Grid-Computing“. Dabei wurde der Begriff „Grid“ ursprünglich aus einem Vergleich mit dem Stromnetz (*engl. Power Grid*) abgeleitet, wo die elektrische Energie von den Kraftwerken bereitgestellt und über das Leitungsnetz bis zum Endverbraucher übertragen wird.

Analog dazu soll das Grid einem Benutzer ebenso einfach Ressourcen, wie z. B. Rechenleistung oder Speicherplatz, über das Internet zur Verfügung stellen, wie es möglich ist, Strom aus einer Steckdose zu beziehen. Bei dem „Grid-Computing“ müssen sich die teilnehmen Rechner nicht nur auf einzelne, innerhalb einer Organisation verfügbaren Rechner beschränken, sondern können auch nahezu beliebige über das Internet erreichbare Rechner mit einbeziehen.

Im Gegensatz zu einem Cluster, der aus einer Zusammenschaltung von einer zuvor meist genau festgelegten Menge von Rechnerkomponenten besteht, wird mit dem Grid-Computing das Ziel verfolgt, die Rechenleistung und Ressourcen, entsprechend der Anforderung der Verbraucher (Clients), auf die im Grid verfügbaren physischen Rechner zu verteilen.

Das Kernstück des Grid-Computing bildet eine Reihe von offenen Standards und Protokollen, z. B. die „Open Grid Services Architecture“ (OGSA) für die Kommunikation zwischen heterogenen und geografisch voneinander getrennten Systemen. Für den Zugriff auf eine vom Grid zur Verfügung gestellte Ressource übergibt der Benutzer seinen Auftrag über eine genormte Schnittstelle an das Grid, woraufhin die Ressourcenallokation automatisch erfolgt.

Die typischen Aufgaben, für die sich das Grid-Computing heutzutage als Strategie anbietet, sind solche die, die Leistung einzelner Computer überfordern. Dazu gehören z. B. die Auswertung und Darstellung von sehr großen Datenmengen aus der naturwissenschaftlichen und medizinischen Forschung. Häufig werden Grid-Architekturen auch in der Meteorologie und rechenintensiven Simulationen in der Industrie angewandt. Im Vergleich zu den Ansätzen aus Wissenschaft und Forschung haben sich die Grid-Computing-Ansätze bei der klassischen IT-gestützten Abwicklung von Geschäftsprozessen noch nicht durchsetzten können.

Durch den Einsatz von Konzepten des Autonomic-Computing [AuCo] wird es in der Zukunft möglich sein, selbst optimierende Grid-Computing-Systeme zu entwerfen. Autonome Verfügbarkeitsdienste überwachen das Vorhandensein einer ausreichenden Anzahl von Dienstekopien und instanziieren bei steigender Last selbstständig neue Kopien. Da keine globale Sicht des Gesamtsystems, wie in den Peer-to-Peer-Systemen, existiert, entscheidet jede Instanz anhand ihrer lokalen Information. Der Informationsaustausch geschieht indirekt durch Beobachtung und Reaktion auf Veränderungen des Systems, was zu gegenläufigen Regelprozessen (thrashing) führen kann.

In Bezug auf die Hochverfügbarkeit ergibt sich bei dem Grid-Computing ein ähnliches Bild wie bei einem strukturierten Peer-to-Peer-Netzwerk. Abhängig von der Grid-Architektur und der Menge an verfügbaren physischen Rechnern und Ressourcen sind die Auswirkungen bei Ausfällen von einzelnen physischen Rechnern praktisch zu vernachlässigen .Server als Bezeichnung für Rechner-Hardware

2.2 Server als Bezeichnung für Rechner-Hardware

Wie bereits in dem Kapitel Fehler: Referenz nicht gefunden „Fehler: Referenz nicht gefunden“ erwähnt, kann unter dem Begriff „Server“ entweder die Kombination aus einem physischen Server und der darauf ablaufenden Software oder aber auch nur die Server-Software oder die Server-Hardware verstanden werden. In diesem Kapitel soll unter dem Begriff „Server“ nun die reale oder virtuelle Hardware verstanden werden, die die Ablaufumgebung für die Server-Dienste (Services) bereitstellt.

2.2.1 Physische Server

Ein physischer Server besteht aus einer Reihe von Hardware-Komponenten und elektronischen Bauteilen, die über Leiterplatten (auch PCB = Printed Circuit Boards genannt), Bussysteme und Backplanes (Rückverdrahtungsplatten) miteinander verbunden sind. Über Steckverbinder stehen meist noch diverse weitere Schnittstellen für die Kommunikation mit externen Massenspeichern, Anwendern (Mensch-Maschine-Kommunikation über Tastatur und Monitor), Druckern (über eine parallele Schnittstelle), Modems (über eine serielle Schnittstelle) sowie weiteren Computern (über eine Netzwerk-Schnittstelle) zur Verfügung.

Alle diese Hardware-Komponenten und elektronischen Bauteile werden in der Regel mit einem oder mehreren Netzteilen für die Stromversorgung in einem Gehäuse zusammengefasst, wobei über die zuvor genannten Steckverbinder die Kommunikation mit der Außenwelt ermöglicht wird. Durch die in dem Gehäuse vorhandenen Hardware-Komponenten und elektronischen Bauteile wird die gesamte zugeführte Energie in Wärme umgewandelt, so dass für die Einhaltung der maximal zulässigen Betriebstemperatur innerhalb eines Server-Gehäuses entweder herkömmliche Lüfter oder aber auch spezielle „Heat Pipes“ (*dt. Wärmerohre*) zum Einsatz kommen.

Von-Neumann-Architektur

Die einem physisches Server zugrundeliegende Hardware-Architektur basiert in der Regel auf der so genannten „Von-Neumann-Architektur“, welche auf einer Beschreibung durch John von Neumann aus dem Jahre 1946 zurückgeht. Diese Architektur definiert für einen Computer fünf Hauptkomponenten:

1. die Recheneinheit (Arithmetisch-Logische Einheit (ALU)),
2. die Steuereinheit,
3. die Buseinheit

4. den Speicher und
5. die Eingabe- und Ausgabeeinheit(en).

Recheneinheit und Steuereinheit

In den heutigen Computern sind die ALU und die Steuereinheit meistens zu einem Baustein verschmolzen, der sogenannten CPU (Central Processing Unit, zentraler Prozessor). Mit dem Begriff Buseinheit sind die parallelen Datenleitungen (8/16/64/128 und mehr parallele Verbindungen) gemeint, die die CPU mit dem Hauptspeicher, aber auch mit den Peripheriegeräten, verbindet.

2.2.1.1 Buseinheit

Diese Verbindungen sind bei kleineren Servern meist direkt auf dem Mainboard realisiert und für Server-interne Erweiterungen über interne Steckplätzen verfügbar. Bei Midrange- oder Mainframe-Computern können sich diese Buseinheiten auch über mehrere Steckkarten oder sogar über mehrere, eng beieinander stehende Server erstrecken. Die maximal mögliche Entfernung dieser Buseinheiten ist aufgrund der Taktung moderner CPUs im Gigahertz Bereich und den sich durch die Verbindungen ergebenen Laufzeiten – auf Entfernungen von wenigen Metern beschränkt.

2.2.1.2 Speicher

Der Speicher ist eine Anzahl von durchnummerierten „Zellen“. Jede von ihnen kann ein kleines Stück Information aufnehmen. Wesentlich in der Von-Neumann-Architektur ist, dass sich Programme und Daten einen Speicherbereich teilen. Dem gegenüber stehen in der Harvard-Architektur Daten und Programm eigene (physikalisch getrennte) Speicherbereiche zur Verfügung, wodurch zum einen der Durchsatz gesteigert werden kann und zum anderen auch bei Schreiboperationen im Datenbereich keine Programme überschrieben werden können.

Für den Speicher eines Servers werden in der Regel Dynamic-Random-Access-Memory (DRAM)-Bausteine verwendet, die auf einem Trägermodul montiert und zu Modulen zusammengeschaltet werden. Das speichernde Element ist dabei ein Kondensator, der entweder geladen oder entladen ist. Über einen Schalttransistor wird er zugänglich und entweder ausgelesen oder mit neuem Inhalt beschrieben. Der Speicherinhalt ist flüchtig (volatil), das heißt, die gespeicherte Information geht bei fehlender Betriebsspannung oder zu später Wiederauffrischung verloren.

2.2.1.3 Memory-Management-Unit

Die meisten der heutzutage eingesetzten Betriebssysteme verwenden den physisch vorhandenen Speicher nicht direkt, sondern verwenden eine zwischengeschaltete Memory-Management-Unit (Speicherverwaltungseinheit MMU). Bei der MMU handelt es sich meist um eine in Hardware realisierte Funktionseinheit, die das Übersetzen von virtuellen Adressen in physische Adressen bewerkstelligt. Jede durch einen Prozess angeforderte virtuelle Adresse wird zuerst durch die MMU in eine physische Adresse umgerechnet, bevor sie auf den Adressbus geschrieben wird.

Das Wort „virtuell“ steht für die gesamte Anzahl von einzigartig ansprechbaren Speicherplätzen, die dem Server-Betriebssystem oder einer auf dem Server laufenden Anwendung zur Verfügung stehen. Dieser virtuelle Speicher kann aus Sicht des Betriebssystems größer sein als der tatsächlich in dem Server installierte physische Speicher. Sofern von dem auf dem Server ablaufenden

Programm eine virtuelle Adresse angesprochen wird, die von der MMU nicht auf eine physische Speicherzelle abgebildet werden kann, wird durch die MMU ein sogenannter Seitenfehler (*engl.: "page fault", „page miss"*) ausgelöst. Als Reaktion auf diesen Seitenfehler werden von dem Betriebssystem zunächst die nicht benötigten Speicherbereiche aus dem physischen Speicher auf ein externes Speichermedium ausgelagert.

2.2.1.4 Auslagerungsprozess

Durch diesen Auslagerungsprozess (auch Swapping oder Paging genannt) wird in dem physischen Speicher Platz geschaffen, um den Speicherbereich in den physischen Speicher zu laden, welcher die ursprünglich angeforderte virtuelle Speicherzelle enthält. Die Auslagerung und Einlagerung erfolgt in enger Zusammenarbeit zwischen der MMU und dem auf dem Server laufenden Betriebssystem. Aus Sicht einer auf dem Server ablaufenden Anwendung ist dieser Vorgang, abgesehen von den deutlichen längeren Zugriffzeiten bei einem Swapping- oder Paging-Vorgang, völlig transparent.

Durch die ständig steigenden Taktraten der CPUs sind die aktuell verfügbaren Speicherbausteine (DRAMs) und die Buseinheiten meist nicht mehr in der Lage, mit den Taktraten der CPU mitzuhalten. Um die CPU bei der Befehlsausführung und dem Zugriff auf die Speicherinhalte nicht „auszubremsen", wird dem Hauptspeicher ein sogenannter Cache vorgeschaltet.

Dieser Cache Speicher verfügt in der Regel über deutlich weniger Speicherkapazität als der mit DRAM-Bausteinen realisierte Hauptspeicher. Dem gegenüber liegen aber die Zugriffszeiten des Cache-Speichers im Bereich der Taktzyklen der CPU, sodass die Befehlsabarbeitung ohne Wartezyklen erfolgen kann. Voraussetzung dafür ist, dass sich die von der CPU angeforderten Befehl- oder Datenworte im Cache befinden.

Eingabe- und Ausgabeeinheit(en)

Ausgehend von der dargestellten Von-Neumann-Architektur soll hier eine weitere Komponente eines physischen Servers, die Eingabe- und Ausgabeeinheit vorgestellt werden. Diese Komponenten sind, ebenso wie der Hauptspeicher, über ein Bussystem mit der CPU bzw. den CPUs verbunden und bilden die Schnittstelle zwischen dem physischen Server und der Außenwelt (Peripherie).

Die Ausgestaltung der Eingabe- und Ausgabeeinheit(en) reicht von speziellen Halbleiterbauteilen, die zusammen mit der CPU auf einem gemeinsamen Mainboard angeordnet sind, bis hin zu abgesetzten „Front End Rechnern" oder dedizierten „Kommunikations- oder Vorrechnern". Die Rechner, die der CPU vorgeschaltet sind, sollen verhindern, dass die CPU durch die weit langsameren Eingabe- und Ausgabeoperationen ausgebremst wird.

Ein physischer Server kann somit als die Menge der vorab genannten Hardware-Komponenten aufgefasst werden. Durch das koordinierte Zusammenwirken der verschiedenen Komponenten ist ein Server in der Lage, die auf einem Speichermedium abgelegten Maschinenbefehle abzuarbeiten und durch die Nutzung der Eingabe- und Ausgabeeinheit(en) mit der Umwelt zu interagieren. Im Sinne einer Hochverfügbarkeit, ist der physische Server in der Regel das kleinste Element, auf dem die dargestellten Maßnahmen sinnvoll angewandt werden können.

Der Grund hierfür ist unter anderen, das der Ausfall oder die Störung einer einzelnen Komponente aus der oben dargestellten Von-Neumann-Architektur – die Befehlsausführung zum Erliegen bringen kann. Weiterhin ist bei allen Hardware-Komponenten eine stabile Stromversorgung und der Betrieb innerhalb der vom Hersteller spezifizierten Temperaturbereiche zwingend erforderlich, da nur so die engen Toleranzgrenzen der Halbleiterbauelemente eingehalten werden können.

2.2.2 Virtuelle Server

Im einfachsten Fall werden die angebotenen Server-Dienste in der Form erbracht, dass jeder Serverdienst auf genau einem physischen Server abläuft. Diese Architektur könnte auch als 1:M-Relation zwischen einem physischen Server und einer Anzahl von Server-Diensten dargestellt werden. Sollte die Leistungsfähigkeit eines einzelnen Hosts nicht ausreichen um die Aufgaben eines Servers zu bewältigen, können mehrere physische Server zu einem Verbund (auch Server-Cluster genannt) zusammengeschaltet werden. Eine ähnliche Vorgehensweise kann auch angewandt werden, wenn durch den Ausfall eines Servers eine Verbindungsunterbrechung in der Client-Server-Kommunikation verhindert werden soll. Dabei stellt sich der Server-Cluster gegenüber den Clients wie ein einzelner (virtueller) Server dar. Dem Benutzer, der über seinen Client mit dem Server verbunden ist, bleibt verborgen, welcher der physischen Server die jeweilige Anfrage abarbeitet. Diese Variante ist allerdings nicht Gegenstand dieses Kapitels und wird in dem Beitrag „Cluster-Architekturen“ des HV-Kompendiums ausführlich betrachtet.

Es gibt aber auch den umgekehrten Fall, wo ein einziger physischer Server in mehrere, virtuelle Server aufgeteilt wird. Auf jeden dieser virtuellen Server kann ein separates Betriebssystem installiert werden, auf dem wiederum die eigentlichen Server-Dienste ablaufen. Den Benutzern, die über die Client-Rechner auf die Server-Dienste zugreifen, bleibt dabei verborgen, dass es sich bei den verschiedenen Servern in Wirklichkeit nur um einen einzigen physischen Server handelt.

Für die zuvor dargestellte Virtualisierung der physischen Server-Hardware sind mehrere Varianten möglich, diese sind in dem Kapitel 2.2.2 „Virtuelle Server“ näher beschrieben. Dort werden auch zwei Varianten vorgestellt, die für einen Einsatz in einer Hochverfügbarkeitsarchitektur geeignet sind. Der Vorteil eines virtualisierten Servers besteht darin, dass dieser innerhalb kürzester Zeit von einem ausgefallenen physischen Server auf einen betriebsbereiten physischen Server transferiert werden kann. Auch können virtuelle Server auf physischen Server-Clustern installiert werden, wodurch die Fehlertoleranz gegenüber dem Ausfall eines physischen Servers nochmals erhöht werden kann.

Den zahlreichen Vorteilen einer Server-Virtualisierung stehen allerdings auch eine Reihe von Nachteilen gegenüber. So wird nicht jede, auf dem physischen Server zur Verfügung stehende Hardware von den virtuellen Servern unterstützt. Auch können die in den physischen Servern zur Verfügung stehenden Ressourcen teilweise nicht im vollen Umfang genutzt werden. Abhängig von dem zugrunde liegenden Betriebssystem und der Virtualisierungssoftware, kann der nutzbare Hauptspeicher z. B. auf 3,6 GB beschränkt sein, auch wenn in dem physischen Server mehr Hauptspeicher vorhanden ist. Es bestehen teilweise auch Einschränkungen bei der Nutzung der zur Verfügung stehenden CPUs. Dabei kann es möglich sein, dass der physische Server mit vier Prozessoren ausgestattet ist, die Virtualisierungssoftware aber nur maximal eine CPU nutzen kann. Bei der Virtualisierung bildet die zugrunde liegende Virtualisierungssoftware auch meist einen Single Point of Failure für mehrere virtuelle Server und Dienste. Somit können Störungen oder Ausfälle in der Virtualisierungssoftware leicht zu ungewollten Nebeneffekten führen, die möglicherweise zu schwerwiegenden Ausfällen führen.

3 Server-Kategorien

Dieses Kapitel enthält eine Darstellung der aktuell gebräuchlichen Server-Varianten. Dabei wird sowohl auf die physischen Server als auch die virtuellen Server eingegangen. Bei jeder Server-Kategorie erfolgt, neben einer kurzen Darstellung der Unterscheidungsmerkmale und Besonderheiten, auch eine kurze Betrachtung hinsichtlich des Einsatzes in einer Hochverfügbarkeitsumgebung.

3.1 Physische Server

In dem Kapitel 2.2.1 „Physische Server“ wurde der grundsätzliche Aufbau eines physischen Servers im Sinne eines Rechnerknotens dargestellt. In den folgenden Abschnitten werden nun einige typische Ausprägungen von Rechnerknoten vorgestellt, die als physischer Server zum Einsatz kommen.

3.1.1 Supercomputer

Ein Supercomputer ist ein Hochleistungsrechner, der meist zu Simulationszwecken eingesetzt wird. Die Haupteinsatzgebiete für einen Supercomputer sind meist rechenintensive Problemstellungen aus den Bereichen Chemie, Physik, Geologie, Luft- und Raumfahrt, Wetter- und Klimaforschung. Die typischen Merkmale eines modernen Supercomputers sind meist eine große Anzahl von Prozessoren, die auf gemeinsame Peripheriegeräte und einen teilweise gemeinsamen Hauptspeicher zugreifen.

Das primäre Ziel für die Architektur eines Supercomputers ist das Erreichen einer maximalen Rechenleistung. Da sich aufgrund von physikalischen Grenzen nicht beliebig schnelle Prozessoren bauen lassen, sind alle heutigen Supercomputer als Parallelrechner ausgeführt. Dabei wird zwischen den Vektorrechnern und den Skalarrechnern unterschieden. Während die Vektorprozessoren Berechnungen auf vielen Daten gleichzeitig durchführen, können Skalarprozessoren dagegen nur ein Operandenpaar pro Befehl bearbeiten. Bei allen Supercomputern spielen die Prinzipien für eine hochverfügbare Auslegung aber meist nur eine untergeordnete Rolle (von Ausnahmen abgesehen), da das primäre Ziel für einen Supercomputer die Durchführung einer maximalen Anzahl von Rechenoperation (meist angegeben in FLOPS = Floating Point Operations Per Second) in einem gegebenen Zeitabschnitt ist.

3.1.2 Mainframe

Im Gegensatz zu Supercomputern, die auf hohe Rechenleistung ausgelegt sind, ist ein Mainframe auf Zuverlässigkeit und hohen Datendurchsatz optimiert. Neben der Bezeichnung Mainframe werden die Server-Systeme in dieser Kategorie auch häufig als Großrechner oder als „Host“ bezeichnet. Die englische Bezeichnung Mainframe geht zurück auf die frühen 1960er Jahre, wo die Zentraleinheit der damaligen Großrechner noch in großen Einbaurahmen (*engl: Mainframe*) untergebracht war. In Bezug auf ihre Leistungsfähigkeit sind die Mainframes zwischen den Supercomputern und den Minicomputern (Enterprise Server) anzusiedeln.

Durch die ständig fortlaufende Miniaturisierung können die heutigen Großrechner schon auf einer Standfläche von wenigen Quadratmeter und einer Höhe von ca. zwei Metern untergebracht werden. Die typischen Anwendungen eines Großrechners sind in Banken, Versicherungen, großen Unternehmen und in der öffentlichen Verwaltung gegeben.

Ein typischer Großrechner aus der heutigen Zeit verfügt in der Regel über mehrere Gbyte an Arbeitsspeicher und Festplattenkapazitäten im zweistelligen TeraByte-Bereich. Auch wenn diese

Ressourcen teilweise schon in modernen Heimcomputern vorhanden sind, wird durch sorgfältig aufeinander abgestimmte Komponenten und optimal auf die Server-Hardware abgestimmte Betriebssysteme erreicht, dass mit einem Mainframe problemlos mehrere tausende Terminals bedient werden können.

Die Hardware- und Software-Komponenten eines Mainframes sind meist hochgradig redundant ausgelegt und für einen unterbrechungsfreien Non-Stop-Betrieb konzipiert. So kann meist auch die Wartung, die Aufrüstungen und sogar ein Hardware-Austausch ohne Unterbrechung des laufenden Betriebs durchgeführt werden. Auch können auf einem Großrechner meist virtuelle Maschinen gestartet werden, in denen jeweils wieder ein eigener Server ablaufen kann. Dadurch können die Mainframes auch für eine Konsolidierung von Server-Farmen verwendet werden.

Weiterhin sind Großrechner für die Verarbeitung von großen Datenmengen ausgelegt. Ihre Eingabe- und Ausgabefähigkeiten werden meist nur noch von den Supercomputern übertroffen. Dies ist ein weiterer Grund, warum Mainframes in Rechenzentren noch immer ihren Platz haben, wobei nicht nur die Geschwindigkeit, sondern auch die hohe Sicherheit hinsichtlich Integrität und Vertraulichkeit der verarbeiteten Informationen eine große Rolle spielt.

3.1.3 Enterprise-Server

Ein Enterprise-Server entspricht in etwa der Kategorie der Minicomputer aus früheren Tagen. Diese Server bieten nahezu die Verfügbarkeit und Skalierbarkeit eines Mainframes, basieren aber meist auf einer Ansammlung von Servern der Workstation-Kategorie, die in einem gemeinsamen Gehäuse bzw. Rack zusammengefasst werden.

Die Server in der Kategorie der Enterprise-Server verfügen heutzutage typischerweise über 8 bis 64 Prozessoren, die zusammengenommen über bis zu zwei TeraByte Hauptspeicher verfügen können. Die Prozessoren, der Hauptspeicher und die Eingabe- und Ausgabeeinheiten können dabei wiederum dynamisch in sogenannten Domains zusammengefasst werden, wobei jede dieser Domains einen eigenständigen physischen Server darstellen kann. Die meisten der Komponenten innerhalb einer Domain können dabei in der Regel mehrfach redundant ausgelegt werden, da typischerweise eine relativ flexible Zuordnung der innerhalb des Enterprise-Server-Racks installierten und verfügbaren Eingabe- und Ausgabekanälen, CPUs und Hauptspeicher zu den Domains möglich ist.

So kann mit den Enterprise-Servern, ähnlich wie mit den Mainframes, eine außergewöhnliche Skalierbarkeit, Zuverlässigkeit und Verfügbarkeit erreicht werden. Auch können die auf der Hardware der Enterprise-Server aufsetzenden Betriebssysteme zu Clustern zusammengeschaltet werden, wobei mit den einzelnen Domains jeweils ein physischer Server abgebildet werden kann.

Im Gegensatz zu den Mainframes werden auf den Enterprise-Servern allerdings keine speziell für die Hardware konzipierten Betriebssysteme verwendet. Vielmehr werden die erprobten und weitverbreiteten Betriebssysteme verwendet, wie sie auch bei Workstations zum Einsatz kommen. Die Portierung von bestehenden Anwendungen auf Enterprise-Server wird dadurch gegenüber den Mainframes deutlich erleichtert, da die typische Betriebssystemarchitektur von Mainframes deutlich von denen der sonstigen Server-Betriebssysteme abweicht.

3.1.4 Workstation

Der Begriff „Workstation“ entwickelte sich in den 1980er Jahren und stand für ein Computersystem, welches als ein besonders leistungsfähiger Rechner für anspruchsvolle technisch-wissenschaftliche Anwendungen konzipiert war und über hervorragende Grafikfähigkeiten verfügte. Der Hintergrund für die Entwicklung von dedizierten Workstations bestand darin, dass zu

dieser Zeit die Rechner in der Regel als Timesharing-Systeme ausgelegt waren, wobei den einzelnen Benutzern nur eine sehr begrenzte Rechenzeit zugestanden werden konnte.

Die Leistungsfähigkeit einer Workstation lag in dem Bereich der in dieser Zeit weitverbreiteten Minicomputer, wobei die Minicomputer meist als Timesharing-Systeme für kleinere Unternehmen oder auch Abteilungen verwendet wurden. Die klassischen Minicomputer wurden im Laufe der Zeit durch die immer leistungsfähigeren Microcomputer verdrängt und sind aus heutiger Sicht als historisch anzusehen. Von der Leistungsfähigkeit ist eine Workstation oberhalb der eines PCs und unterhalb der eines klassischen Großrechners anzusiedeln.

Mit der Entwicklung von immer schnelleren PCs, welche typischerweise der Kategorie Microcomputer zuzuordnen sind, verwischen die Unterschiede zwischen einer Workstation und einem PC immer mehr. Allerdings liegt die Zuverlässigkeit, Speicherkapazität, Verarbeitungsgeschwindigkeit sowie der Datendurchsatz einer Workstation auch heute noch in Regionen, die eher von speziell als Servern ausgelegten Microcomputern erreicht werden.

Die im Vergleich zu Arbeitsplatzrechnern höhere Leistung wird unter anderem durch die Verwendung von speziellen Mikroprozessoren und optimal aufeinander abgestimmten Hardware-Komponenten, sowie optimal auf die Hardware abgestimmten Betriebssystemen und Anwendungen erreicht. So basieren die heutigen Workstations häufig auf einer RISC-(Reduced Instruction Set Computer) Architektur, wobei Befehle in nur einem einzigen Taktzyklus abgearbeitet werden können.

Aufgrund der hohen Zuverlässigkeit und der Leistungsfähigkeit werden Workstations vorzugsweise als Server für geschäftskritische Anwendungen eingesetzt. So sind Workstations eine bevorzugte Hardware-Plattform für z. B. Datenbanken, Applikations-Container oder EAI-Hubs (Enterprise Application Integration, der EAI-Hub ist der zentrale Knoten für den Nachrichtenaustausch**)**. Im Gegensatz zu den Mainframes oder den Enterprise Servern sind auf der Ebene der Server-Hardware meist keine besonderen, über den Standard hinausgehende Vorkehrungen vorhanden, um auch bei dem Ausfall von einzelnen Hardware-Komponenten einen Non-Stop-Betrieb zu gewährleisten.

Um hier dennoch eine Hochverfügbarkeit gewährleisten zu können, werden die physisch eigenständigen Workstations meist über Cluster zu einem größeren Verbund zusammengeschaltet. Diese Cluster werden entweder auf der Ebenen des Betriebssystems oder der Anwendung realisiert und bilden in Zusammenhang mit redundanten Netzteilen, Lüftern und redundanten Netzwerkkarten, die wiederum mit getrennten Network-Switches verbunden sind, ein für die meisten Anwendungen akzeptables Maß an Hochverfügbarkeit. Weitere Information zu dem Thema Server-Cluster finden sich in dem Beitrag „Cluster-Architekturen“ in diesem HV-Kompendium.

3.1.5 Blade-Server

Haben die physischen Server vor wenigen Jahrzehnten noch Räume von dem Ausmaß einer Turnhalle gefüllt, so ist es heute durch die zunehmende Miniaturisierung möglich, die gleiche Rechenleistung auf der Größe einer Eurokarte (Standardformat für 19-Zoll-Einschubtechnik, 100 x 160 mm**)** zu realisieren. Diese zunehmende Miniaturisierung führte zu der Entwicklung von sogenannten Blade-Servern, wobei beispielsweise ein Standard-42HE – 19“-Rack bis zu 84 Blades aufnehmen kann. Die Leistungsfähigkeit eines jeden einzelnen, auf einer Einsteckkarte realisierten Servers entspricht in etwa dem eines im nächsten Kapitel beschriebenen Microcomputers.

Die Blade-Architektur ist dadurch gekennzeichnet, dass ein vollständiger physischer Server (CPU, Hauptspeicher, E/A) auf einer Hauptplatine zusammengefasst wird. Diese Einheit kann dann in ein vertikales, sogenanntes Sub-Chassis, gesteckt werden. Diese Anordnung – wie Messer in einem

Messerblock – könnte auch zu der Bezeichnung Blade-Server geführt haben, da das englische Wort „Blade“ im deutschen die Bedeutung (Messer-)Klinge hat.

Die einzelnen Blade-Server besitzen entweder keine, eine oder zwei Festplatten. Sofern lokale Festplatten auf den einzelnen Blade-Servern vorhanden sind, werden diese meist für das Betriebssystem genutzt. Wird mehr Plattenplatz benötigt, kann dieser entweder über Adapter im übergeordneten Blade-Chassis (SAN, NAS) oder über ein Erweiterungsmodul mit Festplatten erreicht werden, das dann allerdings selbst wieder einen Einschubplatz im Blade-Chassis benötigt.

Der Vorteil der Blade-Server liegt in der kompakten Bauweise, der hohen Leistungsdichte, der Skalierbarkeit und Flexibilität sowie der einfacheren Verkabelung mit wesentlich geringerem Kabelaufwand und der schnellen und einfachen Wartung. So wird nur ein einziger Tastatur-Grafik-Mauscontroller für den gesamten Baugruppenträger benötigt. Auch kann die Administration meist remote über einen dedizierten Netwerkanschluss in dem übergeordneten Chassis erfolgen, sodass mit einer einzigen Konsole alle physischen Server administriert werden können.

Da die einzelnen physischen Blade-Server bei einem Ausfall einfach und schnell ausgetauscht werden können – durch einfaches Ziehen der Steckkarte – ergeben sich deutliche Vorteile gegenüber den herkömmlichen physischen Servern, die jeweils in einem eigenen Gehäuse untergebracht sind.

Bei den Blade-Servern kann der Austausch eines ausgefallenen physischen Servers teilweise sogar über das Bedienprogramm des Blade-Chassis gesteuert werden, wobei alle E/A-Kanäle des ausgefallenen physischen Servers, mit einem weiteren Blade-Server im Blade-Chassis verknüpft werden. Nach einem Neustart, eines als Backup-Blade-Servers konfigurierten Systems, können somit nach einem Hardware-Ausfall die vom ausgefallenen Server erbrachten Dienste innerhalb kürzester Zeit wieder zur Verfügung gestellt werden.

3.1.6 Microcomputer

Der Begriff Microcomputer geht zurück auf die zweite Hälfte der 1970er Jahre, wo eine Reihe von Herstellern Computer für Heim- und Büroanwendungen auf den Markt brachten. Diese Rechner basierten Anfangs auf der Basis von preiswert verfügbaren 8-Bit-CPUs, die Anfang der 1980er Jahre allerdings von 16-Bit CPUs abgelöst wurden. Im Gegensatz zu den damals dominierenden Minicomputern und Mainframes, wo die Zentraleinheit oft mehrere Baugruppen umfasste, wurde die Recheneinheit und die Steuereinheit bei dem Microcomputer in einem einzigen integrierten Schaltkreis, dem sogenannten Mikroprozessor, zusammengefasst.

Einen bedeutenden Wendepunkt gab es Anfang der 1980er Jahre, als ein bisher auf Großrechner spezialisiertes Unternehmen, einen Microcomputer unter der Bezeichnung Personal Computer (PC) auf dem Markt brachte. Unter anderem durch die Marktstellung des Unternehmens wurde diese Ausprägung eines Microcomputers zum De-Facto-Standard. Seit der Einführung des Begriffs Personal Computer Anfang der 1980er Jahre hat der Gebrauch des Begriffs Mikrocomputer stark abgenommen, obwohl eine ständig zunehmende Anzahl von mikroprozessor-basierten Geräten unseren Alltag dominieren.

Auch wenn die Mikrocomputer kleiner und weniger leistungsfähig sind als die Mainframes, Minirechner oder Workstations, so können diese aber aufgrund der geringen Größe und des geringeren Anschaffungspreises meist vielseitiger und flexibler eingesetzt werden. Allerdings sind Ausstattung, Stabilität und Wartungsfreundlichkeit meist nur zweitrangig, da Microcomputer entweder nur als „Standalone“ Rechner oder als Clients in einer Client-Server Anwendung eingesetzt werden.

Nichtsdestotrotz konnten sich Microcomputer aber Mitte der 1980er Jahre behaupten, als es darum ging, die einzelnen PCs mit einem Netzwerk zu verbinden. Dabei wurden die einzelnen PCs über ein LAN (Local Area Network) verbunden, um – entweder auf Basis eines Peer-to-Peer-Netzwerkes oder einer Client-Server-Architektur – zentrale Dienste zu nutzen oder Daten auszutauschen oder zentral abzulegen. Die anfänglichen Netze waren meist auf einzelne Abteilungen oder Arbeitsgruppen beschränkt, wobei die Folgen eines Server-Ausfalls in der Regel überschaubar waren. Der primäre Grund für den Einsatz von Microcomputern als Server war und ist auch heute sicherlich noch der relativ günstige Anschaffungspreis.

Wurde Anfangs noch keine Unterscheidung zwischen einem dedizierten Arbeitsplatzrechner und einem Server vorgenommen, so werden heutzutage auch Microcomputer angeboten, die für einen Einsatz als Server optimiert sind. Auch wenn die Hardwarearchitektur im Wesentlichen noch kompatibel mit der eines PCs aus den 1980er Jahren ist, so werden die Rechner dennoch, in Hinblick auf eine optimale Wartungsfreundlichkeit und Zuverlässigkeit konstruiert.

Wie bereits bei den Workstations erwähnt, müssen die auf Microcomputer basierenden Server auch über Cluster zu einem größeren Verbund zusammengeschaltet werden, um eine Unabhängigkeit von Hardware-Ausfällen zu gewährleisten. Wie auch bei den Workstations werden diese Cluster entweder auf den Ebenen des Betriebssystems oder auf der Anwendungsebene realisiert und bilden im Zusammenhang mit redundanten Netzteilen, Lüftern und redundanten Netzwerkkarten, die wiederum mit getrennten Netzwerk-Switches verbunden sind, eine für die meisten Anwendungen akzeptables Maß an Hochverfügbarkeit. Weitere Information zu dem Thema Server-Cluster befinden sich in dem Beitrag „Cluster-Architekturen“ in diesem HV-Kompendium.

3.1.7 Embedded System

Unter einem „Embedded-System“ versteht man Hard- und Software-Systeme, die eingebettet in umgebende technische Systeme, komplexe Steuerungs-, Regelungs- und Datenverarbeitungsaufgaben übernehmen. Die Einsatzgebiete für ein Embedded-System befinden sich vorwiegend in der Luft- und Raumfahrttechnik, der Automations- und Automobil-Technik, der Telekommunikation sowie im Bereich der Konsumer- und Haushaltsgerätetechnik. Bedingt durch die vielfältigen Einsatzmöglichkeiten ist eine generelle Charakterisierung von Embedded-Systemen nicht möglich. Im Gegensatz zu den zuvor dargestellten Personal Computern sind Embedded-Systeme nicht mit den üblichen Peripheriegeräten wie Maus, Tastatur oder Festplatten ausgestattet, sondern nutzen abhängig vom Einsatzgebiet funktionelle Tasten, Drehschalter und LCD-Anzeigen. Ein Embedded-System basiert meistens auf einem Mikroprozessor, ein Microcontroller mit der darauf aufsetzenden Software oder auf einer komplett in Hardware abgebildeten State-Machine, die völlig ohne Software auskommt. Abhängig vom Verwendungszweck und der vorgesehenen Einsatzumgebung, wird ein Embedded-System meist maßgeschneidert konzipiert. Soll ein Embedded-System z. B. sicherheitskritische Steuerungsaufgaben übernehmen, so wird es üblicherweise mehrfach redundant ausgelegt. Dabei laufen mehrere Embedded-Systeme im Parallelbetrieb und werden durch eine unabhängige Instanz überwacht. Ergeben sich abweichende Ergebnisse bei diesen parallel laufenden Systemen, kann durch die Überwachungsinstanz z. B. ein Neustart des abweichenden Systems veranlasst werden.

Eine weitere wichtige Eigenschaft von Embedded-Systemen ist, neben der Fehlertoleranz und der Hochverfügbarkeit, auch häufig noch die Einhaltung von maximal definierten Antwortzeiten. So muss z. B. bei einem Antiblockiersystem die elektronisch gesteuerte Bremse nahezu unverzögert im Millisekundenbereich reagieren. Eine Überschreitung der definierten Latenzzeit ist nicht tolerierbar.

Daneben sind Embedded-Systemen meist rauen Umweltbedingungen wie Hitze, Kälte, Staub oder Erschütterungen ausgesetzt. Um hier die Verfügbarkeit der zur Verfügung gestellten Dienste zu

gewährleisten, muss die Hardware häufig hermetisch gekapselt und entsprechend robust ausgelegt sein. So werden häufig Steckverbindungen eingespart und auf die Verwendung von beweglichen Komponenten, wie Festplattenlaufwerke oder Lüfter, verzichtet.

Weitere Einflüsse, die auf ein Embedded-System einwirken können, sind kurzfristige Spannungsspitzen oder Einbrüche in der Stromversorgung. Auch hier müssen die Embedded-Systeme zuverlässig reagieren und bei einer Störung innerhalb kürzester Zeit den erwarteten Dienst wieder zur Verfügung stellen. Ebenso kann sich Radioaktivität negativ auf die Funktion eines Embedded-Systems auswirken, da z. B. Alpha-Strahlen die Inhalte von Speicherzellen verändern können. Gegen sporadische Störungen in der Spannungsversorgung enthalten Embedded-Systeme meist spezielle Detektoren, die eine Störungen in der Spannungsversorgung erkennen und anschließend die Ausführung einer Prüfroutine veranlassen. Extern bedingte Veränderungen an den Speicherinhalten werden meist durch zusätzliche Speicherzellen erkannt und korrigiert.

Im Gegensatz zu allen zuvor dargestellten Server-Kategorien bildet eine robuste und hochverfügbare Auslegung meist die oberste Priorität bei der Konzeption eines Embedded-Systems. Abhängig von den sich aus dem vorgesehenen Einsatzgebiet ergebenen Gefährdungen und dem potentiellen Schadenspotential, kommen bei den Embedded-Systemen praktisch alle in dem Beitrag „Prinzipien der Verfügbarkeit“ des HV-Kompendiums dargestellten Prinzipien zur Anwendung.

3.2 Virtualisierung

Wie bereits im Kapitel 2.2.2 „Virtuelle Server“ kurz dargestellt, gibt es verschiedene Möglichkeiten für eine Virtualisierung der physischen Server-Hardware. In den folgenden Abschnitten werden die derzeit gebräuchlichsten Varianten für eine Virtualisierung der Server-Hardware dargestellt.

3.2.1 Auf Basis von Betriebssystemen

Diese Variante der Virtualisierung kommt häufig beim Web-Hosting zum Einsatz, wobei sich mehrere Kunden einen real existierenden, physischen Server teilen. Durch die Trennung der Kundenumgebung durch eine sogenannte „chroot“-Umgebung (oder ein ähnlich gelagertes Verfahren) werden die Ablaufumgebungen voneinander abgeschottet. Der Begriff „chroot“ steht für „change root“ und ist eine Funktion auf Unix-Systemen um das Rootverzeichnis zu ändern.

Aus der Sicht eines einzelnen Kunden sieht es so aus, als würde ihm ein eigener physischer Server zu Verfügung stehen, da er innerhalb seiner eigenen Ablaufumgebung einen uneingeschränkte Zugang hat und über Administrative (Root-) Rechte verfügt. Tatsächlich werden aber nur bestimmte Bereiche des globalen Dateisystems den jeweiligen Kunden zugeteilt und auch alle von den Anwendungen des Kunden gestarteten Prozesse werden von einem übergeordneten, globalen Prozess abgeleitet, der sich unter der Kontrolle eigentlichen Server-Betriebssystems befindet.

Somit nutzen alle auf dem Server laufenden Anwendungen das nur einmal vorhandene Betriebssystem, wobei durch die Abschottung der Ablaufumgebungen der Eindruck entsteht, dass für jeden Kunden ein eigener physischer Server zur Verfügung steht. Da diese Variante der Virtualisierung hinsichtlich der Verfügbarkeit keine Vorteile bietet, soll sie hier nicht weiter betrachtet werden. Auch hinsichtlich der weiteren Sicherheitsziele wie – Vertraulichkeit und Integrität – bietet diese Variante gegenüber mehreren physischen Servern nur einen geringen Schutz, da alle Anwendungen auf einem gemeinsamen Betriebssystem aufsetzen und daher die logische Trennung der Anwendungen nur im Benutzer-Kontext und nicht im sicheren Kernel-Kontext erfolgen kann.

3.2.2 Auf Basis von Emulation

Eine weitere Möglichkeit für mehrere, logisch getrennte Server auf einem einzigen physischen Server zur Verfügung zu stellen besteht darin, mehrere Betriebssysteme zu installieren und diese auch gleichzeitig auf dem physischen Server ablaufen zu lassen. Dies kann dadurch erreicht werden, dass auf der Server-Hardware und dem darauf aufsetzenden Betriebssystem eine Virtualisierungssoftware installiert wird. Diese Virtualisierungssoftware bildet anschließend das Verhalten von mehreren physischen Servern nach, wobei auf jeden dieser nachgebildeten (emulierten) virtuellen Server auch wiederum ein Betriebssystem installiert werden kann.

Wie in der ersichtlich, werden durch die Virtualisierungssoftware die Hardwareressourcen eines physischen Servers nachgebildet, sodass sich der (emulierte) virtuelle Server, aus der Sicht des darauf aufsetzenden Betriebssystems, wie ein real existierender, physischer Server darstellt. Während sich sowohl die CPU als auch die Speicherbereiche mit den von Betriebssystemen her bekannten Mechanismen auf die Gast Betriebssysteme verteilen lassen, sind für die Virtualisierung von physischen Hardwarekomponenten in der Regel zusätzliche Maßnahmen notwendig.

So werden zumeist alle Zugriffe des Gast-Betriebssystems auf die virtuelle Hardware sowie auf die Eingabe- und Ausgabeeinheiten (unter anderem Netzwerk, Speicher, Konsole) abgefangen und über die Virtualisierungssoftware an das Betriebssystem des physischen Servers weitergeleitet.

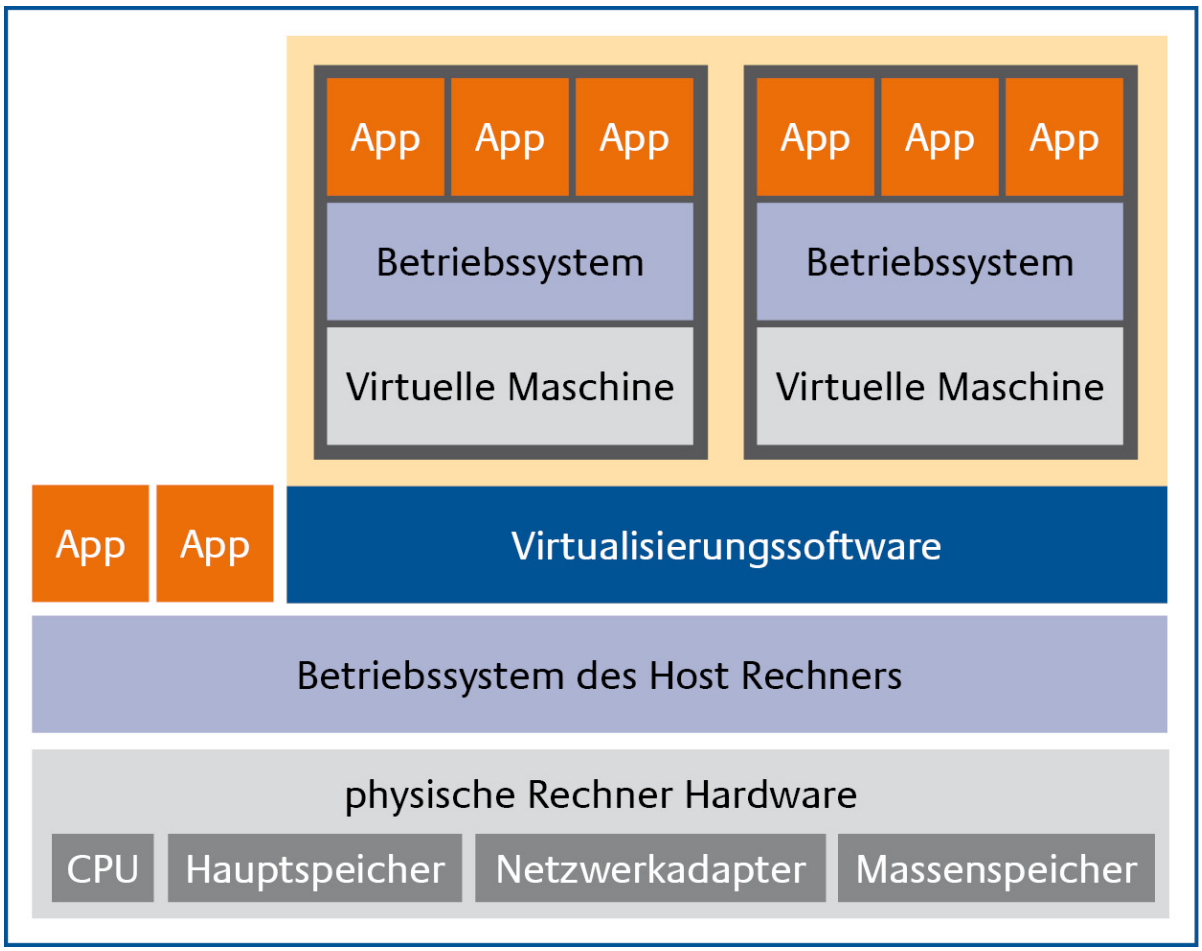


Abbildung 5: Virtualisierung eines physischen Servers

Ein weiterer Aspekt ergibt sich durch die Architektur moderner CPUs, die eine Trennung zwischen dem Betriebssystem und der Anwendungsumgebung realisieren. Dabei werden kritische Teile des Betriebssystems in einem Kernel Mode (Ring 0) verarbeitet, während die Anwendungen im User Mode (Ring > 0) ablaufen. Durch diese Separierung sollen die Auswirkungen von Anwendungen auf die Stabilität und Sicherheit des Betriebssystems vermindert werden. Dieser, auch als Kontext-Switch bezeichnete Übergang vom User-Mode in den Kernel-Mode, muss durch die Virtualisierungssoftware abgefangen und entsprechend behandelt werden.

Da die virtuellen Server meist binärkompatibel mit dem physischen Server sind, erfolgt die Abarbeitung der Befehle der virtuellen Server meist direkt durch den zugrunde liegenden physischen Server. Sowohl die Verwaltung der Zeitscheiben für die CPU als auch die Zuteilung des physischen Hauptspeichers erfolgt entsprechend den Mechanismen wie sie auch in Betriebssystemen implementiert sind.

Gegenüber der ersten Variante kann bei der Virtualisierung auf Basis einer Emulation ein höheres Sicherheitsniveau in Bezug auf die Sicherheitsziele – Vertraulichkeit und Integrität – erreicht werden, da die Anwendungen in einem weitestgehend isolierten Container (dem virtuellen Server) ablaufen. Bei der Virtualisierung auf Basis einer Emulation stellt die Virtualisierungssoftware ein potenzielles Angriffziel dar, da diese ebenso wieder im Benutzerkontext läuft und durch diese alle Betriebssystemaufrufe der darauf aufsetzenden virtuellen Maschinen transformiert und weitergeleitet werden. Im Bezug auf Hochverfügbarkeit kann die Virtualisierungssoftware wiederum auf einen Server-Cluster aufgesetzt werden oder teilweise auch mit auf weiteren physischen Servern laufenden Instanzen der Virtualisierungssoftware zu einem Cluster zusammengeschaltet werden. Die virtuellen Server können somit fast völlig isoliert werden von der zugrunde liegenden Hardware.

3.2.3 Hypervisor/Hardware-Virtualisierung

Bei den sogenannten Hypervisor-Architekturen wird das normalerweise auf der physischen Server-Hardware aufsetzende Betriebssystem, sowie die normalerweise auf dem Betriebssystem aufsetzenden Virtualisierungssoftware in einer Programmkomponente vereint. Da diese Variante direkt auf der physischen Server-Hardware (bare-metal) aufsetzt, wird zum einen eine höhere Performance erzielt, zum anderen aber auch eine größere Abhängigkeit von der zugrunde liegenden physischen Server-Hardware sowie deren Konfiguration erkauft.

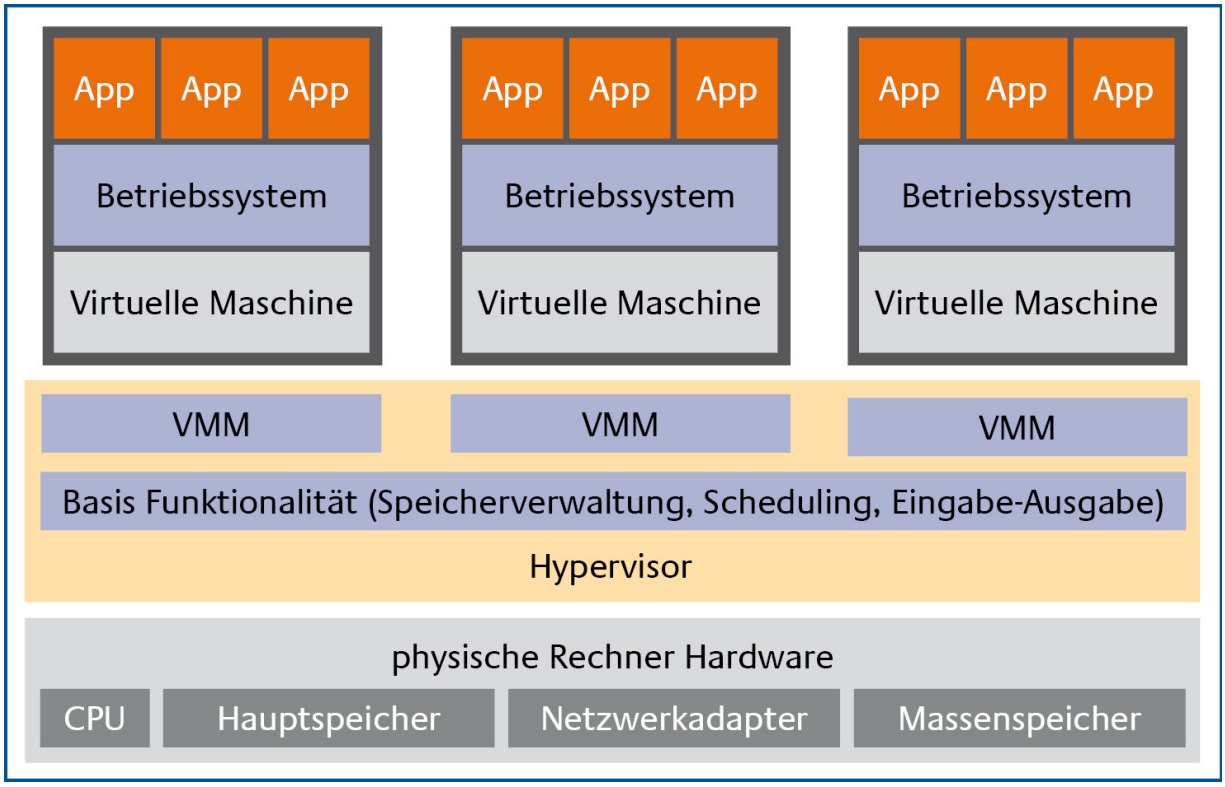


Abbildung 6: Virtualisierung auf der Basis der Hypervisor Architektur

Dabei wird zwischen der physischen Server-Hardware und den in den virtuellen Servern laufenden Betriebssystemen eine zusätzliche Software-Schicht eingezogen, die jedem darauf aufsetzenden Betriebssystem den Eindruck vermittelt, dass es alle Ressourcen der zugrunde liegenden physischen Server-Hardware exklusiv nutzen kann. Diese zusätzliche Software-Schicht wird meist als Hypervisor bezeichnet. Dem Hypervisor kommt die Aufgabe zu, die real zur Verfügung stehenden Ressourcen des physischen Servers (CPU, Speicher, Eingage- und Ausgabe) den virtuellen Servern zuzuweisen sowie den Zugriff auf die Ressourcen entsprechend zu koordinieren. Die Zuweisung der Ressourcen und der Verwaltung der virtuellen Maschinen geschieht meist über VMM (Virtual Machine Monitors), welche über eine in dem Hypervisor angesiedelte Software-Komponente abgebildet werden.

Da der Hypervisor direkt auf der physischen Server-Hardware aufsetzt, brauchen die direkten Hardware- oder BIOS (Basic Input Output System) -Zugriffe von dem auf den virtuellen Servern laufenden Betriebssysteme nicht mehr aufwändig abgefangen und in Betriebssystemaufrufe des zugrunde liegenden Betriebssystems übersetzt werden. Der Hypervisor kann die Virtualisierung dabei auf zwei verschieden Arten unterstützen. Zum einen mit einem Verfahren mit der Bezeichnung Paravirtualization, wobei ein modifiziertes Gast-Betriebssystem zum Einsatz kommt, welches bei Hardware- oder BIOS-Zugriffen direkt mit dem Hypervisor kommuniziert. Zum anderen sind mittlerweile Prozessoren verfügbar, die spezielle Mechanismen für die Virtualisierungssoftware enthalten. Dabei werden die Hardware- oder BIOS-Zugriffe des Gast-Betriebssystem vom Prozessor erkannt und durch einem speziellen Kontext-Switch direkt an den Hypervisor übergeben.

Diese Variante bietet hinsichtlich der Sicherheitsziele – Vertraulichkeit und Integrität – sicherlich das höchste Niveau bezogen auf die bisher vorgestellten Virtualisierungslösungen. Auch wird mit dieser Variante in naher Zukunft eine optimale Ausnutzung der zur Verfügung stehenden

Ressourcen des zugrunde liegenden physischen Servers möglich sein. Diese Vorteile werden allerdings mit einer höheren Abhängigkeit von der Hardware bzw. dem Gast-Betriebssystem erkauft.

Für Hochverfügbarkeit kann der, auf einen physischen Server ablaufende Hypervisor, auch mit anderen Hypervisoren auf weiteren physischen Servern gekoppelt werden, sodass sich hier eine Architektur, ähnlich der eines Server-Clusters, ergibt. Die einzelnen physischen Server agieren hier jeweils als zustandslose Rechenknoten, wobei allerdings für eine Hochverfügbarkeit meist auch die Eingabe- und Ausgabekanäle auf allen Rechnern im Cluster-Verbund mit der Außenwelt verbunden sein müssen.

4 Server-Architektur

4.1 Verteilte Systeme

Nach der Definition von Andrew Tannenbaum ist ein Verteiltes System ein Zusammenschluss unabhängiger Computer, welcher sich aus der Sicht des Benutzers als ein einzelnes System präsentiert.

Die spezifischen Eigenschaften eines verteilten Systems können wie folgt beschrieben werden:

- Entferntheit:
  Die Komponenten eines Verteilten Systems sind meist auch räumlich voneinander getrennt.
- Asynchronität:
  Kommunikations- und Verarbeitungsprozesse werden nicht durch einen zentralen Taktgenerator synchronisiert. Änderungen und Prozesse sind demzufolge nebenläufig und können parallel abgearbeitet werden.
- Unabhängigkeit der Komponenten:
  Die Komponenten sind unabhängig voneinander. Der Ausfall einer Komponente beeinträchtigt andere Elemente des Verteilten Systems nur insofern, als diese auf die ausgefallene Komponente zugreifen wollen. Es entsteht nur ein „Partieller Systemausfall“.
- Selbstverwaltung:
  Es gibt keine zentrale Einheit, die sämtliche Management- und Steuerungsfunktionen übernimmt. Somit besitzen die einzelnen Komponenten ein gewisses Maß an Autonomie.
- Dynamische Rekonfiguration:
  Um die Leistungsfähigkeit eines Verteilten Systems zu erhöhen, können Programme und Daten zwischen verschiedenen Orten bewegt werden. Auch muss ein Verteiltes System in der Lage sein, dynamisch auf äußere Einwirkungen zu reagieren und Umstrukturierungen vorzunehmen, also zum Beispiel bei einem Server-Ausfall die Verbindung neu aufbauen.
- Heterogenität:
  Verteilte Systeme bestehen meist aus Hardware-Komponenten unterschiedlicher Hersteller.

In der werden die möglichen Architekturen verteilter Systeme klassifiziert:

Hardware	***Software***	
	lose gekoppelt	fest gekoppelt
fest gekoppelt	-	Multiprozessorbetriebssystem
lose gekoppelt	Netzbetriebssystem	Verteiltes Betriebssystem

Tabelle 1: Verteilte Systeme

4.1.1 Vor- und Nachteile von verteilten und zentralen Systemen

In den Anfängen der elektronischen Rechner existierten aufgrund der Anschaffungskosten und Größe ausschließlich zentrale Systeme, die von einem kleinen Team von Spezialisten bedient werden konnten. Mit der zunehmenden Miniaturisierung und den ständig fallenden Anschaffungskosten wurde es spätestens seit Mitte der 1980er Jahre möglich, mehrere kleinere Rechner einzusetzen, um damit z. B. einen bestehenden Mainframe abzulösen.

Durch dieses Vorgehen konnten deutliche Einsparpotenziale realisiert werden, da die von einem Mainframe erbrachte Rechenleistung in der Regel auch durch eine Zusammenschaltung von einer nahezu unbegrenzten Anzahl von Mikrocomputern aufgebracht werden konnte.

Allerdings wurde durch diese große Anzahl an dezentralen Systemen auch die Anfälligkeit gegen Störungen aller Art deutlich erhöht, sodass die verteilten Systeme – zumindest in den Anfängen – nur bedingt die Ausfallsicherheit der klassischen Großrechner erreichen konnten.

Im Folgenden werden nun die allgemein zutreffenden Vor- und Nachteile für eine Server-Architektur, auf der Basis von verteilten Systemen, gegenübergestellt. Abhängig vom speziellen Verwendungszweck, werden die Argumente mehr oder weniger zutreffend sein.

Verteilte Systeme zeichnen sich durch die folgenden Vorteile aus:

- Skalierbarkeit:
  Verteilte Systeme ermöglichen eine dynamische Anpassung oder Lastverteilung entsprechend der aktuellen Anzahl von Benutzer, Prozessen, Zugriffen oder auch der Menge an Daten.
- Erweiterbarkeit bzw. Integrierbarkeit:
  Bestehende Systeme können die neu hinzugekommenen Server-Systeme und die Dienste nutzen, ohne dass ein System gleicher Funktionalität neu entwickelt werden muss.
- Fehlertoleranz bzw. Ausfallsicherheit:
  Die einzelnen Bestandteile eines Verteilten Systems sind weitestgehend autonom. Im Falle eines Fehlers oder sogar Ausfalls einer Systemkomponente, können die übrigen Einheiten im Idealfall unbeeinflusst weiterarbeiten und ggf. den Störfall überbrücken.
- Kosteneffizienz:
  Die Flexibilität und Anpassbarkeit von Verteilten Systemen führen gegenüber zentralen Systemen zu einem leichteren und damit günstigeren Management der IT-Infrastruktur.
- Teilweise Dezentralität des Managements:
  Der Eigentümer einer Ressource hat die Möglichkeit, das Management dieser Komponente selbst zu übernehmen. Dies hat, neben eventuell niedrigeren Managementkosten, weitere Vorteile wie Schnelligkeit, mögliche optimale Anpassung an die Bedürfnisse des Einzelnen und automatische Skalierung der mit dem Management beauftragten Personenzahl an den Managementbedarf (hoher Servicegrad bei hoher Kosteneffizienz).

Verteilte Systeme haben allerdings auch eine Reihe von Nachteilen:

- Komplexere Software:
  Die Realisierung eines Verteilten Systems erfordert komplexere Softwarelösungen als die eines zentralen Systems. Dies äußert sich nicht nur in der Komplexität der einzelnen Softwarekomponenten und ihres Zusammenwirkens, sondern auch in der hohen Komponentenanzahl, die die betriebswirtschaftlichen Funktionen durch den Erwerb und die Verwaltung zahlreicher Lizenzen belastet.
- Eventuell inkonsistente Datenbestände und Zustände:
  In Verteilten Systemen kann es vorkommen, dass derselbe Sachverhalt mehrmals gespeichert vorliegt und sich die Datensätze aufgrund einer mangelhaften Synchronisation widersprechen.
- Schwierige Fehlerbestimmung:
  Die Lokalisierung von Fehlern kann sich in Verteilten Systemen als schwieriges Unterfangen herausstellen. Durch umfangreiche Abhängigkeiten zwischen den Systemen oder bei dem hinzufügen oder herausnehmen von Systemen können nicht vorhersehbare Fehler entstehen.

– Datenschutzprobleme:
  Datenbestände in Verteilten Systemen ermöglichen generell einen einfacheren Zugriff auf sensible Daten als dies bei separater Datenhaltung der Fall ist.
– Schwierigere Überwachbarkeit:
  Die teilweise Dezentralität des Managements kann bei unzureichend geschulten Benutzern zu Managementfehlern führen. Die Qualität des IT-Managements ist somit in Verteilten Systemen schwieriger zu gewährleisten als in zentralen Systemen.

4.1.2 Software Architekturen verteilter Systeme

Wie in der “ dargestellt wird, ist die Softwarearchitektur für den Einsatz in verteilten Systemen grundsätzlich abhängig von der zugrunde liegenden Hardwarearchitektur des physischen Servers. So kommt beispielsweise bei einer fest gekoppelten Rechnerarchitektur in der Regel ein Multiprozessor-Betriebssystem zum Einsatz. Das Multiprozessor-Betriebssystem hat die Aufgabe, eine optimale Lastverteilung auf die vorhandenen Prozessoren vorzunehmen.

Neben der Hardware, sind meist auch die weiteren Komponenten des Multiprozessor-Betriebssystems in einer monolithischen Software-Komponente zusammengefasst. Diese Eigenschaft wird auch als eine feste Kopplung der Software bezeichnet. Die für ein Multiprozessor-Betriebssystem relevanten Hochverfügbarkeitsaspekte werden im Kapitel 4.2.3 „Multiprozessor-Betriebssystem“ betrachtet.

Eine weitere Softwarearchitektur-Variante für verteilte Systeme ergibt sich, wenn keine gemeinsame Nutzung von Rechner-Ressourcen (z. B. Speicher, Prozessor, Bussystem) vorliegt. Bei dieser Variante handelt es sich meist um mehrere physisch getrennte Rechner, die über ein Netzwerk miteinander verbunden sind. Um diese physisch getrennten Server gegenüber den Benutzer als ein transparentes System darzustellen, wird meist ein verteiltes Betriebssystem eingesetzt.

Eine tiefer gehende Darstellung zu dem Thema; verteiltes Betriebssystem und eine Betrachtung hinsichtlich der Hochverfügbarkeitsaspekte erfolgt in dem Kapitel 4.2.2 „Verteiltes Betriebssystem.“ Dabei gelten die Anmerkungen und Betrachtungen bzgl. Eines verteilten Betriebssystems ebenso für die auf dem Betriebssystem aufsetzenden Software-Schichten.

Als letzte grundlegende Software-Architektur Variante für verteilte Systeme soll hier das Netzbetriebssystem – stellvertretend für alle denkbaren Client-Server, Peer-to-Peer- und Grid-Computing-Anwendungen – aufgeführt werden. Wie in der e“ ersichtlich, handelt es sich um eine lose Kopplung sowohl der Hardware als auch der darauf aufsetzenden Software. Bei dieser Variante handelt es sich meist um physisch völlig separate Server-Systeme, die über eine serielle Verbindung gekoppelt sind. Dabei läuft typischerweise auf dem einen Teil der physischen Server eine Server-Software-Komponente und auf den verbleibenden physischen Rechnern eine Client-Komponente.

Eine tiefer gehende Darstellung zu dem Thema Netzbetriebssystem und die Betrachtung hinsichtlich der Hochverfügbarkeitsaspekte erfolgt in dem Kapitel 4.2.1 „Netzwerk-Betriebssystem“. Dabei gelten die Anmerkungen und Betrachtungen für ein Netzwerk-Betriebssystem analog für alle sonstigen denkbaren Client-Server-, Peer-to-Peer- und Grid-Computing-Anwendungen, die auf dem Prinzip einer losen Kopplung von Software-Komponenten aufbauen.

4.1.3 Hardware-Architekturen verteilter Systeme

Wie in der „e“ kurz dargestellt wird, kann bei der Hardware-Architektur eines verteilten Systems grundsätzlich zwischen einer lose gekoppelten und einer fest gekoppelten Architektur unterschieden werden. In der Praxis finden sich manchmal auch Mischformen dieser beiden Extreme, die dann als nahe Rechnerkopplung bezeichnet wird.

Bei der festen Kopplung (Multiprozessor-Ansatz) teilen sich die Prozessoren Hauptspeicher, Bussystem und meist auch die Eingabe- und Ausgabeeinheiten. Das Betriebssystem sowie die auf dem Betriebssystem aufsetzenden Anwendungsprogramme werden nur einmal im Hauptspeicher geladen. Dieser Ansatz wird häufig auch als „Shared-Memory“- bzw. „Shared-Everything“-Ansatz bezeichnet.

Einer der Vorteile dieser festen Kopplung ist eine effiziente Kooperation zwischen den Prozessoren über den gemeinsamen Hauptspeicher. Auch kann bei einem Multiprozessor-Betriebssystem eine effektive Lastverteilung erreicht werden, indem gemeinsame Auftragswarteschlangen im Hauptspeicher gehalten werden, sodass alle Prozessoren darauf zugreifen können.

In Hinblick auf den Einsatz in einer Hochverfügbarkeitsumgebung sind fest gekoppelte Hardwarearchitekturen in der Regel eher weniger geeignet, da der gemeinsame Hauptspeicher nur eine geringe Fehlerisolation bietet und die Software (z. B. das Betriebssystem) und die darauf aufsetzenden Anwendungen nur einmal im Hauptspeicher vorliegen (Single Point of Failure).

Auch die Skalierbarkeit einer fest gekoppelten Hardwarearchitektur ist in der Regel begrenzt, da mit wachsender Prozessoranzahl sowohl das Bussystem als auch der Hauptspeicher leicht zum Engpass wird. Aus diesen Gründen sind derzeitige Multiprozessor Systeme meist auf 2 bis 30 Prozessoren beschränkt, wobei bereits ab 10 Prozessoren in der Regel nur noch geringe Leistungssteigerungen erzielt werden können.

Bei der losen Kopplung kooperieren mehrere, voneinander unabhängige Rechner, die jeweils über einen eigenen Hauptspeicher, eigene Bussysteme sowie eigene Eingabe- und Ausgabesysteme verfügen. Weiterhin laufen auf jedem Rechner eigenständige Instanzen sowohl des Betriebssystems als auch der darauf aufsetzenden Anwendungen. Mit dieser Architektur kann eine weitaus bessere Fehlerisolation erreicht werden als bei der festen Kopplung. Auch lassen sich von der Anzahl der CPUs her gesehen wesentlich leistungsfähigere Konfigurationen realisieren, als dieses bei einer engeren Kopplung möglich ist, insbesondere wenn bei der losen Kopplung jeder Rechnerknoten selbst wieder ein Multiprozessor-Rechner ist.

Ein wesentlicher Nachteil der losen Kopplung ist ihre aufwendige Kommunikation, die ausschließlich durch den Nachrichtenaustausch über ein Verbindungsnetzwerk erfolgt. Neben der Kopplung über eine „normale“ Netzwerkverbindung, existiert in der Praxis eine nahezu unüberschaubare Vielfalt von Topologien, Protokollen und Übertragungsmedien.

Dabei ist für die Hochverfügbarkeit die möglichst vollständige und ausfallsichere Vermaschung aller beteiligten CPUs ein wichtiges Kriterium. Zum anderen dürfen die für den Austausch der Nachrichten verwendeten Verbindungen zwischen den Prozessoren nur minimale Verzögerungen aufweisen, da teilweise schon Latenzzeiten von einigen Millisekunden die Abarbeitung transaktionsorientierter Anwendungen unmöglich machen. Weiterführende praktische Hinweise für eine Kopplung der Prozessoren finden sich in dem Kapitel 4.5.8 „Inter-Prozessor-Kommunikation“.

4.2 Betriebssystem-Architektur

Auf Systemen mit nur einer CPU ist es üblicherweise die Aufgabe des Betriebssystems, Benutzern und Anwendungen einen geregelten Zugriff auf CPU, Speicher sowie Eingabe- und Ausgabegeräte zu ermöglichen. Die Anforderung, dass auf einem Prozessor mehrere Betriebssysteme ablaufen müssen, kann durch den Einsatz von virtuellen Maschinen gelöst werden. Für jedes Betriebssystem wird dabei der Anschein erweckt, dass alle vorhandenen Ressourcen eines physischen Servers ausschließlich dem aufrufenden Betriebssystem zur Verfügung stehen.

Bei einem verteilten System mit mehreren Prozessoren ist es einer der wichtigsten Aspekte, die Ressourcen der einzelnen Rechner Knoten gemeinsam zu nutzen. Dabei verwaltet üblicherweise jeweils ein lokales Betriebssystem die zur Verfügung stehende Hardware wie CPU, Speicher, Netzwerk und angeschlossene Geräte.

Um diese lokalen Ressourcen in einem verteilten System ebenfalls zu nutzen, muss das jeweils lokale Betriebssystem entsprechende Mechanismen etablieren, die eine Kommunikation mit den weiteren Komponenten des verteilten Betriebssystems ermöglichen.

4.2.1 Netzwerk-Betriebssystem

Unter dem Begriff Netzwerk-Betriebssystem wird im Allgemeinen eine Sammlung von System-programmen verstanden, die den Benutzern oder den auf dem Server ablaufenden Anwendungen die Möglichkeit gibt, auf die Dienste und Ressourcen eines über das Netzwerk erreichbaren Knotens zuzugreifen. Die Benutzer müssen dabei in der Regel wissen auf welchen Server sich die gesuchte Ressource befindet und sich meist auch vor der Nutzung authentisieren.

Das Netzwerk-Betriebssystem läuft dabei meist auf dem als Server bezeichneten physischen Server und erlaubt es den Benutzern an den angeschlossenen Arbeitsstationen bzw. Clients, Nachrichten und Daten auszutauschen, sowie Dateien und Peripheriegeräte gemeinsam zu nutzen.

Das Netzwerk-Betriebssystem folgt dabei vollständig den Prinzipien einer losen Kopplung, da sowohl die Hardware im Sinne der physischen Server, als auch das auf der Hardware ablaufende Betriebssystem, weitestgehend autonom agieren.

Zu real existierenden Nutzern eines Netzwerk-Betriebssystems zählen viele verfügbare Client-Server, Grid- und Peer-to-Peer-Anwendungen, da diese auf die grundlegenden Dienste eines Netzwerkbetriebssystems aufsetzen.

4.2.2 Verteiltes Betriebssystem

Ein verteiltes Betriebssystem läuft in der Regel auf mehreren, physisch autonomen Server-Rechnern ab, die jeweils wiederum mit mindestens einer CPU ausgestattet sind. Zwischen den Rechnern muss ein Verbindungsnetzwerk vorhanden sein, über das die Knotenrechner kommunizieren können.

Alle beteiligten Knotenrechner verfügen oft über die gleiche Hardware-Ausstattung und auf allen Knotenrechnern kommt in der Regel derselbe Betriebssystemkern zum Einsatz. Bei dem Ausfall eines physischen Rechners oder bei einem schwerwiegenden Fehler innerhalb eines Betriebssystems kann ein Anwender meist über einen anderen Rechner aus dem Verbund in transparenter Weise weiterarbeiten.

Das Gesamtsystem verliert dabei nur soviel an Leistung, wie der ausgefallene Rechner durch seine CPU dazugegeben hatte. Durch die fortlaufende Synchronisation des verteilten Betriebssystems kann die Single Point of Failure Problematik ausgeschlossen werden.

Durch die feste Kopplung eines verteiltes Betriebssystems wird weiterhin erreicht, dass ein Prozess, der auf einen beliebigen Knotenrechner abläuft, nicht explizit angeben muss, wo sich ein anderer Prozess, mit dem er kommunizieren möchte, innerhalb des verteilten Betriebssystems befindet.

Eine typische Ausprägung eines solchen, zuvor beschriebenen verteilten Betriebssystems, ist ein Server-Cluster, der unter anderem in dem Beitrag „Cluster-Architekturen“ des HV-Kompendiums näher beschrieben wird.

4.2.3 Multiprozessor-Betriebssystem

Bei einem System mit nur einem Prozessor werden die einzelnen Prozesse in Zeitscheiben aufgeteilt und somit „quasi-parallel“ abgearbeitet. Bei jedem Umschalten werden die Register-Inhalte des alten Prozesses auf dem Kellerspeicher (engl: Stack) abgelegt und die Register mit den entsprechenden Werten aus dem Kellerspeicher des neuen Prozesses initialisiert. In einem Multi-Prozessor System laufen die einzelnen Prozesse nicht sequentiell ab, sondern können von den im Rechner vorhandenen Prozessoren tatsächlich zur gleichen Zeit ausgeführt werden.

Die Aufgabe des Multiprozessor-Betriebssystems ist es, die parallel laufenden Prozesse über eine Manipulation der gemeinsam genutzten Speicherbereiche so zu koordinieren, dass die Abarbeitung der einzelnen Prozesse geordnet abläuft. Dabei ist die Anzahl der tatsächlich vorhandenen CPUs für die nutzenden Prozesse in der Regel völlig transparent. Um einen gleichzeitigen Zugriff auf die Speicherbereiche zu verhindern, werden meist Kontrollstrukturen wie Semaphore und Monitore genutzt, die bereits bei Einprozessorsystemen zum Einsatz kommen.

Durch den gemeinsam genutzten Speicher und der durch die Hardware-Architektur bedingten räumlichen Nähe der einzelnen Prozessoren, bringt der Einsatz eines Multiprozessor-Betriebssystems in einer Hochverfügbarkeitslösung praktisch keine Vorteile. Auch leistet die Komplexität eines Multiprozessor-Betriebssystems nicht unbedingt einen Beitrag zur Robustheit des Gesamtsystems.

4.3 Rechner-Core-Architekturen

In diesem Kapitel werden einige grundlegende Rechner-Core-Architekturen vorgestellt und mit dem Thema Hochverfügbarkeit in Verbindung gebracht. Es werden die am meisten verbreiteten Rechner-Core-Architekturen sowie ein Klassifizierungsschema für Rechner Architekturen vorgestellt.

4.3.1 Havard-Architektur

Bei der Harvard-Architektur ist der Befehlsspeicher physisch vom Datenspeicher getrennt und beide Speicher werden über getrennte Bussysteme angesteuert. Der Vorteil dieser Architektur besteht darin, dass Befehle und Daten gleichzeitig geladen, bzw. geschrieben werden können. Bei einer klassischen Von-Neumann-Architektur sind hierzu mindestens zwei aufeinanderfolgende Buszyklen notwendig.

Weiterhin sorgt die physische Trennung von Daten und Programm dafür, dass bei Softwarefehlern kein Programmcode überschrieben werden kann. Nachteilig ist allerdings, dass nicht benötigter Datenspeicher nicht als Programmspeicher (oder umgekehrt) genutzt werden kann.

Die Harvard-Architektur wurde zunächst überwiegend in RISC-Prozessoren konsequent umgesetzt. Viele moderne Prozessoren verwenden eine Mischform aus der Harvard- und der von-Neumann-Architektur, wobei die Befehle und Daten extern in einem gemeinsamen Speicher liegen, aber innerhalb des Prozessors in getrennten Caches und Pipelines verwaltet werden.

Wie im nächsten Kapitel dargestellt, entspricht die Harvard-Architektur dem SISD-Rechner (Single Instruction, Single Data) Konzept, wobei die Befehle sequentiell abgearbeitet werden.

4.3.2 Von-Neumann-Architektur

Dass Von-Neumann Rechnermodell (siehe hierzu auch Kapitel 2.2.1 "Physische Server") gilt als das grundlegende Rechnermodell, und unterteilt einen Rechner in eine Zentraleinheit, einen Speicher und eine Eingabe/Ausgabeeinheit. Die Zentraleinheit kann wiederum in ein Rechen- und ein Steuerwerk unterteilt werden, womit letztendlich die sequentielle Abarbeitung von Instruktionen realisiert wird.

Wie im nächsten Kapitel dargestellt, entspricht die Von-Neumann-Architektur dem SISD-Rechner (Single Instruction, Single Data) Konzept, wobei die Befehle sequentiell abgearbeitet werden.Flynn-Klassifikation

Die Unterteilung von Rechnerarchitekturen erfolgt häufig nach der von Michael J. Flynn [Flynn1972] vorgeschlagenen Taxonomie (). Dabei werden die Rechner Architekturen nach der Anzahl der vorhandenen Befehls- (Instruction Streams) und Datenströme (Data Streams) unterteilt.

	Einfacher Instruktionsstrom	***Mehrfacher Instruktionsstrom***
einfacher Datenstrom	SISD von Neumann	MISD
mehrfacher Datenstrom	SIMD Vektorprozessoren	MIMD Mehrprozessorrechner

Tabelle 2: Übersicht Flynn-Klassifikation

Unter der Bezeichnung SISD-Rechner (Single Instruction, Single Data) werden traditionelle Einprozessor-Rechner verstanden, welche die Befehle sequentiell abarbeiten. SISD-Rechner sind z. B. Personal-Computer oder Workstations die nach der Von-Neumann- oder der Harvard-Architektur aufgebaut sind. Bei der Von-Neumann-Architektur wird für die Daten und die Befehle der gleiche Speicher verwendet, bei der der Harvard-Architektur werden für die Daten und die Befehle jeweils voneinander unabhängige Speicher eingesetzt.

Die auf der SIMD (Single Instruction, Multiple Data) Architektur basierenden Rechner sind häufig Supercomputern oder Mainframes, wobei mit einem Befehl gleichzeitig mehrere gleichartige Datensätze verarbeiten werden können. Auch Array- oder Vektorprozessor sind in der Regel Ausprägungen der SMID-Architektur.

Die MISD (Multiple Instruction, Single Data) Architektur wird in der Literatur häufig kontrovers diskutiert. Auf den ersten Blick und unter dem Gesichtspunkt einer Leistungssteigerung, ist dieser Architektur-Ansatz in der Regel wenig sinnvoll wenn mehrere Prozessoren parallel mit gleichen Eingangsdaten operieren. Allerdings kann diese Architektur, im Sinne einer Fehlertoleranz unter Umständen vorteilhaft sein, wenn die resultierenden Daten nach der Befehlsabarbeitung der parallel laufenden Rechner auf die Übereinstimmung geprüft werden.

Die MIMD (Multiple Instruction, Multiple Data) Architektur war in der Vergangenheit häufig den Supercomputern oder Mainframes vorbehalten. Mit der fortschreitenden Miniaturisierung und der Erhöhung der Packungsdichte von Halbleiterbauelemente, steht diese Architekturvariante nun seit einiger Zeit auch für den Bereich der Workstations und Microcomputer zur Verfügung.

Das Besondere an der MIMD-Architektur ist die Tatsache, dass gleichzeitig verschiedene Operationen auf verschieden Eingangsdatenströmen durchgeführt werden können. Die Verteilung

der Aufgaben auf die zur Verfügung stehenden Ressourcen erfolgt dabei zur Laufzeit durch einen der beteiligten Prozessoren in dem Rechnerverbund.

Bei der MIMD-Architektur kann weiterhin zwischen fest gekoppelte und lose gekoppelte Rechnersysteme unterschieden werden. Zu den fest gekoppelten Systemen zählen Multiprozessorsysteme, die sich einen gemeinsamen Speicherbereich teilen. Lose gekoppelte Systeme bestehen oft aus mehreren, voneinander unabhängigen Rechnern, die in einem Rechnerverbund agieren. Eine weiterführende Beschreibung zu den möglichen Varianten einer Rechnerkoppelung findet sich unter dem Kapitel 4.1.3 „Hardware-Architekturen verteilter Systeme“.

4.4 Rechner-Eingabe-Ausgabe

Über die Eingabe- und Ausgabeeinheiten kann ein physischer Rechner Daten von den umgebenen Peripheriegeräten aufnehmen oder Daten an diese Peripheriegeräte senden. Unter dem Begriff „Peripheriegerät“ kann jede Einrichtung verstanden werden, die in der Lage ist einem gegebenen Rechner Daten auszutauschen. Typische Peripheriegeräte sind bei einem Arbeitsplatzrechner die Eingabegeräte – Tastatur und Maus – sowie ein Datensichtgerät (Bildschirm) für die Ausgabe. Abhängig vom Einsatzgebiet können an dem physischen Rechner auch lokale Drucker oder DFÜ-Komponenten (z. B. Modem) für die Datenübertragung angeschlossen sein.

Sofern es sich bei dem Rechner um einen Server handelt, gehören auch externe Speichereinheiten oder sonstige über das Netzwerk erreichbaren Ressourcen zu den typischen Peripheriegeräten. Aufgrund der unüberschaubaren Vielfalt von möglichen Peripheriegeräten kann die Aufzählung niemals vollständig sein. Diese vorhergehende Aufzählung ist daher auch nur als eine lose Zusammenstellung von möglichen Peripheriegeräten anzusehen.

Bei den Eingabe-Ausgabeeinheiten kann weiterhin noch zwischen der physikalischen und der logischen Schnittstelle unterschieden werden. Unter einer Schnittstelle (engl.: Interface) versteht man allgemein eine Verbindungsstelle zwischen zwei Systemen oder einen Übergabepunkt von einem System zum anderen. Im Allgemeinen ist eine Schnittstelle der Punkt, an dem eine Verbindung zwischen zwei Elementen hergestellt wird, sodass eine Zusammenarbeit der beiden Elemente möglich wird.

Die physikalische Schnittstelle umfasst in der Regel eine, wie auch immer geartete, Steckverbindung am Gehäuse oder auf dem Mainboard des Rechners.

Neben den klassischen elektrischen oder optischen Übertragungsmedien, kommen bei Arbeitsplatzrechnern und Notebooks auch häufig drahtlose Übertragungsmedien – auf Basis von Infrarot Verbindungen oder Funkwellen im GHz Bereich – zum Einsatz.

Im Gegensatz zur physikalischen Schnittstelle, ist die logische Schnittstelle eine Software-Komponente, welche die Verbindung zwischen dem Rechner und den Peripheriegeräten herstellt. Die logische Schnittstelle dient häufig der Übergabe von Steuerinformationen an Peripheriegeräten und ist in der Regel als Programmierschnittstelle ausgeführt.

Mit der logischen Schnittstelle können z. B. bestimmte Sektoren auf einer Festpatte adressiert, Transportverbindungen über das Netzwerk initiiert oder eingegebenen Zeichen von der Tastatur abgefragt oder angezeigt werden.

Für eine Hochverfügbarkeitslösung müssen - aus der Sicht eines Servers – zum einen die Eingabe- und Ausgabeeinheiten sowie die physikalischen Verbindungen zwischen dem Server und dem Peripheriegerät möglichst redundant vorhanden sein, sodass bei einem Ausfall immer noch eine

weitere physikalische Verbindung existiert. Sinnvoll kann es ebenfalls sein, wichtige Peripheriegeräte redundant auszulegen um einem möglichen SpoF vorzubeugen.

Neben einer Redundanz auf der physikalischen Ebene, müssen auch auf der logischen Ebene Mechanismen für eine redundante Wegefindung zur Verfügung gestellt werden. Dazu werden z. B. bei Netzwerken über intelligente Routing-Mechanismen entsprechende Funktionen bereitgestellt, die ein Datenpaket bei einem Ausfall von Verbindungen auch über alternative Wege zum Ziel-System transportieren können.

Im Folgenden werden exemplarisch einige Eingabe- und Ausgabeeinheiten vorgestellt, die typischerweise bei einen Server vorhanden sind.

4.4.1 Netzwerk (LAN / WAN)

Ein Server stellt in der Regel Dienste für weitere, mit dem Server verbundene Rechnerknoten zur Verfügung. Damit diese Rechnerknoten die Dienste des Servers nutzen können, muss eine Verbindung zwischen dem Server und den Nutzern existieren. Die Verbindung zwischen den Server und den Nutzern erfolgt dabei in der Regel nicht direkt sonder über ein sogenanntes Transportnetzwerk.

Das Transportnetzwerk kann dabei auf nahezu beliebigen Technologien, Topologien und Protokollen beruhen, sofern der Zugangspunkt zu diesem Netzwerk eine von der Server Netzwerkschnittstelle unterstütztes Medium sowie ein in der Server-Software implementiertes Protokoll beherrscht.

Die Verbindung des Servers mit dem Transportnetzwerk erfolgt dabei in der Regel über eine leitungsgebundene Verbindung auf der Basis eines metallischen Leiters oder einer Glasfaser. Durch diese Verbindung wird die Kommunikation zwischen der meist als Netzwerkkomponente (z. B. Hub, Switch, Router usw.) realisierten Übergabeschnittstelle des Transportnetzes mit der physischen Netzwerkschnittstelle des Servers ermöglicht. Die Datenübertragung auf der leitungsgebundenen Verbindung basiert dabei typischerweise auf der Modulation von elektrischen oder optischen Signalen, welche entweder im Basisband oder im Breitbandformat erfolgt.

Auch wenn die Anschaltung eines Servers prinzipiell auch über Funknetze oder optische Sichtverbindungen erfolgen könnte, kommt diese Variante bei der Anschaltung von Servern an ein Transportnetzwerk aufgrund der hohen Störanfälligkeit praktisch nicht zu Einsatz. Als Ausnahmen können hier vielleicht Notebooks gesehen werden, die über eine eingebauten WLAN-Karte verfügen und über eine stationäre WLAN-Station mit den Nutzern kommunizieren.

Häufig werden auch Funkverbindungen eingesetzt, um bei dem Ausfall einer primären Verbindung einen alternativen Weg zur Verfügung zu haben. Das Vorhandensein eines solchen alternativen Weges ist für den Server in der Regel allerdings meist völlig transparent, da der Endpunkt sowohl für den primären als auch den alternativen Weg durch die Netzwerkkomponenten gebildet wird.

Ebenso erfolgt auch die Umschaltung auf den Alternativweg durch die Netzwerkkomponente was durch den Server nicht wahrgenommen werden kann, da aus der Sicht des Servers die Netzwerkkomponenten die nächste Station für die Kommunikation mit dem Ziel darstellt.

Die physikalische Schnittstelle zwischen dem physischen Server und dem Netwerk wird in der Regel durch eine Steckverbindung gebildet, die sich meist am Gehäuse des Servers befindet und von außen zugänglich ist. Die Aufbereitung der für die Übertragung verwendeten elektrischen oder optischen Signale erfolgt meist mit den Eingabe- und Ausgabeeinheiten, welche den internen Systembus des physischen Servers mit der physischen Schnittstelle verbinden.

Neben der Umwandlung der parallel vorliegenden Datenworte in ein serielles Format, erfolgt durch die Eingabe-Ausgabeeinheiten in der Regel auch die Aufbereitung der Informationen für die Übertragung auf einem elektrischen oder optischen Medium. Diese Aufbereitung umfasst zum einen die Transformation in das verwendete Übertragungsformat sowie zum anderen die Anpassung an die elektrischen oder optischen Anforderungen entsprechend den in verschiedenen Standards festgelegten Eigenschaften für die physikalische Schnittstelle.

Für eine hochverfügbare Auslegung sollten daher immer mehrere Verbindungen zwischen den physischen Server und den Netzkomponenten existieren, sodass bei dem Ausfall von einer Verbindung immer mindestens noch eine weitere zur Verfügung steht. Diese redundante Verbindung sollte dabei möglichst mehrere, in dem physischen Server installierten Eingabe- und Ausgabeeinheiten umfassen, die jeweils wieder mit mehren physisch getrennten Netzwerkkomponenten verbunden sind.

Für weitere detaillierte Betrachtungen zu einer hochverfügbaren Ankopplung des Servers und einer hochverfügbaren Auslegung des Netzwerks soll an dieser Stelle auf den Beitrag „Netzwerk“ in dem HV-Kompendium verwiesen werden.

4.4.2 Storage (SCSI / SATA / Firewire)

Auch wenn es prinzipiell möglich ist, das Betriebssystem und alle Anwendungen für einen beliebigen Server komplett aus einem mit dem internen Bussystem verbundenen Halbleiter Speicher (z. B. einen EEPROM) zu laden, wird dieses Verfahren praktisch nur bei den Embedded-Server-Systemen angewandt. Bei allen sonstigen Server-Systemen werden das Betriebssystem und alle notwendigen Anwendungen in der Regel auf externe Massenspeicher abgelegt, die über eine Verbindung mit dem Server verbunden sind.

Die Anbindung eines Servers an einen externen Massenspeicher hat viele Gemeinsamkeiten mit der Verbindung eines Servers an ein allgemeines Netzwerk. Neben den speziell für die Übertragung von Speicherblöcken ausgelegten Netzwerken – wie einem Storage-Area-Network (SAN) – kann die Verbindung zwischen Server und Massenspeicher auch über ein herkömmliches Netzwerk erfolgen.

Unter entsprechenden Voraussetzungen ist es z. B. für einen Server möglich, den IPL (initial Programm Load) oder Boot Vorgang komplett über das Netzwerk abzuwickeln. Die Voraussetzung dafür ist allerdings ein im Netzwerk erreichbarer BOOTP-Server der die benötigten Speicherblöcke zur Verfügung stellt. Auch kommen häufig Network Attached Storage (NAS) Systeme zum Einsatz, bei denen der Server ebenfalls über das Netzwerk auf einen externen Massenspeicher zugreifen kann.

Auch wenn die Anbindung eines externen Massenspeichers über ein allgemeines Netzwerk möglich ist, so erfolgt die Verbindung zu dem Massenspeicher meist über eine separate, vom allgemeinen Netzwerk unabhängige Verbindung. Für weitere detaillierte Betrachtungen zu einer hochverfügbaren Ankopplung des Servers an einen Massenspeicher sei auf den Beitrag „Speichertechnologien“ im HV-Kompendium verwiesen. In diesem werden die hier erwähnten Komponenten unter Hochverfügbarkeitsaspekten umfassend betrachtet.

Die physikalische Schnittstelle zwischen dem physischen Server und dem Massenspeicher wird in der Regel durch eine Steckverbindung gebildet, die sich entweder innerhalb des Server-Gehäuses (bei lokalen bzw. internen Massenspeicher) oder am Server-Gehäuse herausgeführt ist (für die Verbindung mit einem externen Massenspeicher). Wie auch beim allgemeinen Netzwerk, erfolgt durch die Eingabe- und Ausgabeeinheiten eine Anpassung an die, von dem Massenspeicher bzw.

dem Storage Netzwerk erwarteten Datenformate sowie eine Aufbereitung der elektrischen oder optischen Signale.

Wie bereits im vorhergehenden Kapitel dargestellt, sollten für eine hochverfügbare Auslegung immer mehrere Verbindungen zwischen den physischen Server und dem Massenspeicher bzw. dem Speichernetzwerk vorhanden sein. Dies betrifft sowohl die in dem physischen Server installierten Eingabe- und Ausgabeeinheiten als auch die Verbindungsleitungen zwischen dem physischen Server und dem Massenspeichern bzw. Speichernetzwerk. Auch sollten Eingabe- und Ausgabeeinheiten wiederum mit physisch getrennten Massenspeicher- oder Speichernetzwerk-Komponenten verbunden werden, so dass bei einem Ausfall immer noch ein weitere Massenspeicher mit einem gespiegelten Inhalt oder einer weiteren Speichernetzwerkkomponente für die Aufrechterhaltung der Verbindung zwischen dem Server und dem externen Massenspeicher zur Verfügung steht.

4.4.3 Konsole (RS232 / V.24)

Die Begriff „Konsole“ bezeichnet in der Regel ein Datensichtgerät, welches über eine direkte Punkt-zu-Punkt-Verbindung mit dem Server verbunden ist. Die Eingabe- und Ausgabeeinheiten für die Konsolen-Schnittstelle werden meist schon in einem sehr frühen Stadium nach einem Neustart des Servers initialisiert, sodass eventuelle Fehlermeldungen des Servers dargestellt werden können. Zum anderen bildet die Konsole auch einen wichtigen Eingabekanal für die Administration des Servers, da über diese Schnittstelle bereits vor dem Starten des Betriebssystems oder der Initialisierung des Netzwerks kommuniziert werden kann.

Die Server werden meist in Rechenzentren oder Server-Räumen untergebracht, die für einen längeren und regelmäßigen Aufenthalt von Personen eher weniger geeignet sind. Daher werden die Konsolen-Schnittstellen der Server meist mit sogenannten Terminal-Servern verbunden, welche wiederum über das allgemeine Netzwerk erreichbar sind. Ein Terminal-Server ist dabei ein eigenständiger Rechner, der auf der einen Seite mehrere Schnittstellen für den Anschluss der Server-Konsolen besitzt und auf der anderen Seite eine Netzwerkschnittstelle für eine Verbindung mit dem allgemeinen Netzwerk bereitstellt.

Da die Konsole in der Regel nur für die Einrichtung eines Servers oder bei einer erweiterten Fehlersuche benötigt wird, ist diese Schnittstelle meist auch nur einfach vorhanden. Bei Servern mit unternehmenskritischen Services sowie bei weit entfernten und schwer zugänglichen Servern muss der Konsolenzugang einschließlich der Netzwerkstrecke redundant ausgelegt werden (mehr im Kapitel 4.4.5 „Management (In-Band / Out-of-Band)“).

4.4.4 Drucker (Parallel / Seriell)

Für die Druckerausgabe in einem Netzwerk werden meist spezielle Server eingesetzt, die Druckaufträge der Rechner entgegennehmen und diese an die entsprechenden Drucker, Kopierer oder Plotter weiterleiten. Der Server und der an dem Server angeschlossene Drucker bestanden noch in den 1990er Jahren aus physisch separaten Einheiten. Dabei wurde der Drucker über eine parallele oder serielle Schnittstelle mit der Server verbunden. Auch hier war die physikalische Schnittstelle zwischen dem Server und Drucker als Steckverbindung ausgeführt, die in der Regel durch eine Aussparung am Server-Gehäuse zugänglich ist. Wie auch bei dem allgemeinen Netzwerk erfolgt hier die Anpassung an die von dem Drucker erwarteten Datenformate sowie die Aufbereitung der elektrischen Signale durch die im Server installierten Eingabe- und Ausgabeeinheiten.

Die Druckerschnittstelle wird in der Regel nicht redundant ausgeführt. Sofern der Server mit dem angeschlossenen Drucker über ein Netzwerk erreichbar ist, wird dieser Server auch als Druckserver bezeichnet. Der Druckserver nimmt dabei Druckaufträge über das Netzwerk entgegen und leitet

diese an die angeschlossenen Drucker, Druckwerke und sonstige andere Endgeräte, wie z. B. Plotter, weiter. Als Druckserver werden auch Server bezeichnet, die Druckaufträge über das Netzwerk zentral anzunehmen und sie dann dezentral über das Netzwerk wieder an die entsprechenden (netzwerkfähige) Drucker zu verteilen. Da die angeschlossenen Druckausgabegeräte meist langsamer sind als der eingehende Datenstrom für die Druckausgabe werden die Druckaufträge meist auch noch auf dem Druckserver solange zwischengelagert, bis der Drucker bereit ist, diese Daten entgegenzunehmen.

Im Zuge der zunehmenden Miniaturisierung werden die früher getrennten physischen Einheiten – der mit dem Netzwerk verbundenen Server und der daran angeschlossenen Drucker – heutzutage auch häufig zu einer Einheit zusammengefasst, wobei der Server dann direkt in dem Drucker eingebaut ist. Diese Kombination wird dann in der Regel auch als Netzwerkdrucker bezeichnet.

4.4.5 Management (In-Band / Out-of-Band)

Der Begriff „In-Band-Management“ bezeichnet den administrativen Zugriff auf einen Server über eine bereits bestehende Kommunikationsverbindung. Bei einem Server mit einem Ethernet-Anschluss könnte dies z. B. bedeuten, dass ein Administrator über die bestehende Netzwerkverbindung eine ssh-Verbindung (ssh = Secure Shell) mit dem Server aufbaut.

Dabei besteht jederzeit die Gefahr, dass die Netzwerkschnittstelle von dem Arbeitsplatzrechner des Administrators, des Servers oder gar das ganze Netzwerk z. B. aufgrund eines Konfigurationsfehlers ausfällt und der Administrator keine Chance mehr hat den Server „In-Band“ zu erreichen. Als einziger Ausweg bleibt hier meist nur noch ein direkter Zugriff auf den Server über die Konsole.

Ein direkter Zugriff über die Konsole kann sich als äußerst schwierig erweisen, sofern die zu administrierenden Servern weltweit verteilt sind und eine Hochverfügbarkeit angestrebt wird. Für solche Einsatzszenarien kann eine „Out-of-Band-Management“ Lösung sinnvoll sein. Im Gegensatz zu dem „In-Band-Management“ erfolgt der Zugang zu dem Server hierbei über ein vollständig separates Verbindungsnetz. Als Beispiel könnte hier von dem Arbeitsplatz des Administrators eine Modemverbindung zu der Konsole des Servers aufgebaut werden, sobald ein Zugang über die „In-Band“ nicht mehr möglich ist.

Allerdings muss auch hierbei berücksichtigt werden, dass das Verbindungsnetz für den „Out-of-Band-Management“ Ansatz tatsächlich unabhängig von „In-Band-Management“ ist. Daher macht es z. B. wenig Sinn, wenn die Modem-Verbindung über ein Voice-over-IP Netzwerk erfolgt, dessen Grundlage wiederum das ausgefallene „In-Band-Management“ Netzwerk ist.

4.5 Interne Rechner-Komponenten

In diesem Kapitel erfolgt eine zusammenfassende Betrachtung unter dem Gesichtspunkt der Hochverfügbarkeit in Bezug auf die elementaren Bauteile einer autonomen Rechnerzelle.

4.5.1 CPU (Central Processing Unit)

Die CPU (Central Processing Unit) bildet das Herzstück eines jeden Rechners und ist äußerst empfindlich gegenüber elektrostatischer Aufladung sowie nicht ausreichender Kühlung. Auftretende Spannungs- oder Stromspitzen stellen für die CPU eine große Gefahr dar, da diese Bauteile nicht für hohe Schwankungen ausgelegt sind. Ebenso kann eine unzureichende Kühlung die internen Strukturen der CPU dauerhaft schädigen, sodass die CPU durch ein neues Bauteil ersetzt werden muss. Aufgrund der hohen Integrationsdichte moderner Mehrkern Prozessoren stellen diese keine Lösung für die oben angeführten Gefährdungen dar.

Gegen mögliche Spannungsschwankungen muss in einer Hochverfügbarkeitsumgebung eine VFI-USV (Unterbrechungsfreie Stromversorgung) zwischen der Zuleitung des Stromversorgers und dem Server installiert werden. Da auch alle anderen in einem Server verwendete Halbleiterbauelemente extrem empfindlich gegen auftretende Spannungs- oder Stromspitzen und alle Bauteile von der gleichen Stromversorgung abhängig sind, wird auf eine wiederholte Darstellung dieser Maßnahmen in den weiteren Kapiteln verzichtet.

Für eine ausreichende Kühlung muss ein entsprechend dimensionierter Lüfter auf der CPU montiert werden, wobei eine Managementanwendung die Temperatur überwacht.

4.5.2 Hauptspeicher

Ebenso wie die CPU, ist auch der Hauptspeicher ein unverzichtbarer Bestandteil eines jeden Rechners. Speichermodule reagieren ebenfalls äußerst empfindlich gegenüber elektrostatischer Aufladung und unzureichender Kühlung. Im Gegensatz zu einer CPU wird bei Speichermodulen deutlich weniger elektrische Energie in Wärmenergie umgewandelt, sodass hier auf einen eigenen Lüfter in der Regel verzichtet werden kann. Dennoch ist für den Hauptspeicher auch eine ausreichende Umluft notwendig, die aber meist durch einen, im Server-Gehäuse, angebrachten Lüfter erfolgt. Auch dieser Lüfter sollte in einer Hochverfügbarkeitsumgebung fortlaufend durch eine Management Anwendung überwacht werden.

4.5.3 Boot-ROM / PROM / EEPROM

Das Boot-ROM / PROM / EEPROM enthält zum einen die notwendigen Maschinenbefehle, die ein Rechner nach einem Neustart abarbeitet. Zum anderen enthält das Boot-ROM / PROM / EEPROM in der Regel Programmelemente für eine Abstraktion der Hardware, sodass ein darauf aufsetzendes Betriebssystem von einem definierten Funktionsumfang ausgehen kann.

Daher bildet auch das Boot-ROM / PROM / EEPROM einen unverzichtbaren Bestandteil eines jeden Rechners, da ohne dieses Bauteil ein Server nicht funktionieren würde. Viele moderne Rechner verwenden wieder beschreibbare EEPROM Bausteine als Boot-ROM / PROM, wobei im Anschluss an einen Löschvorgang, ein neues Programm aufgebracht werden kann.

In einer Hochverfügbarkeitsumgebung kann dieser Vorgang problematisch werden, wenn der Programmiervorgang gestört wird oder aus anderen Gründen nicht erfolgreich beendet werden kann. Sollte dieser Fall eintreten, wird der Server nach einem Neustart nicht mehr Starten und muss ausgetauscht werden. Daher sollte dieses Vorgehen in einer Hochverfügbarkeitsumgebung nur durchgeführt werden, wenn ein Ersatzrechner innerhalb kürzester Zeit bereitgestellt werden kann.

4.5.4 Backplane / Mainboard

Die elektronischen Bauteile aus denen ein physischer Server besteht, werden in der Regel auf einer sogenannten Leiterplatte zusammengefasst. Die Leiterplatte besteht dabei meist aus mehreren Lagen eines elektrisch nichtleitenden Materials. Auf jeder Lage sind weiterhin sogenannte Leiterbahnen aufgebracht, die die elektronischen Bauteile miteinander verbinden.

Bei Servern aus den Kategorien Microcomputer und Workstation sind die wesentlichen Komponenten eines Servers wie die CPU, der Hauptspeicher, der BIOS-Chip mit der integrierten Firmware, die Eingabe- und Ausgabekomponenten für die Anbindung unter anderem des Netzwerks oder des Massenspeichers sowie Steckplätze für Erweiterungskarten meist auf einer sogenannten Hauptplatine (*engl. Mainboard, oder motherboard*) untergebracht.

Bei Rechnern aus den Kategorien – Enterprise-Server, Mainframe und Supercomputer – die in der Regel über mehrere CPUs und einen Hauptspeicher im zweistelligen Gbyte Bereich verfügen, werden die Komponenten meist auf mehrere Leiterplatten verteilt. Sie werden in einem

Einschubrahmen untergebracht und über eine sogenannte Rückverdrahtungsplatte (*engl. Backplane*) verbunden.

Bei beiden Varianten stellt das Bussystem das zentrale „Nervensystem“ eines Servers dar. Sobald auch nur eine Leitung auf dem Adress- oder Datenbus z. B. durch eine defekte Einschubkarte gestört wird, fallen in der Regel der gesamte Server sowie alle drauf laufenden Anwendungen aus.

In einer Hochverfügbarkeitsumgebung kann das Aufrüsten – z. B. mit einer neuen Einsteckkarte - unter Umständen problematisch werden, wenn durch das Stecken der neuen Komponente weitere Bausteine beschädigt werden, die auf das gemeinsame Bussystem zugreifen.

Daher sollte das Aufrüsten oder die Durchführung von Änderungen an der Hardwarekonfiguration in einer Hochverfügbarkeitsumgebung nur dann durchgeführt werden, wenn ein kompletter Ersatzrechner bereitsteht oder zumindest innerhalb kürzester Zeit beschafft werden kann.

4.5.5 Erweiterungsports

Bei Servern aus den Kategorien Microcomputer und Workstation befinden sich auf der Hauptplatine häufig auch noch sogenannte Erweiterungsports. Diese Erweiterungsports dienen der Erweiterung von grundlegenden Hardwarekonfigurationen über eine flexible Aufnahme von Steckkarten. Zu den typischen Erweiterungskarten zählen Grafik-, Sound- und Netzwerkkarten. Aber auch Hardware Security Module (HSM), Modems oder ISDN-Adapter sind abhängig von den Anwendungsgebieten anzutreffen.

Da diese Erweiterungskarten meist direkt mit dem Systembus verbunden sind, sollten bei dem hinzufügen bzw. dem entfernen von Erweiterungskarten die gleichen Vorsichtsmaßnahmen angewandt werden, wie sie bereits in dem „Kapitel Backplane / Mainboard“ beschrieben wurden.

4.5.6 Netzteil

Eine ausreichende und stabile Stromversorgung ist eine elementare Voraussetzung für den störungsfreien Betrieb eines physischen Servers. Daher sollten Server mit mindesten zwei Netzteilen ausgestattet sein. Die Netzteile sollten dabei fortlaufend überwacht werden, wobei der Ausfall eines Netzteils umgehend an einen zentralen Leitstand gemeldet werden sollte.

Auch sollten die Netzteile über jeweils separate Zuleitung und einer getrennten Absicherung von einer oder mehreren Online USV gespeist werden, sodass Störungen in dem Stromnetz möglichst ohne Auswirkungen bleiben. Für eine weitere Betrachtung des Themas Netzversorgung soll an dieser Stelle auf den Beitrag „Infrastruktur“ im HV-Kompendium verwiesen werden, in dem das Thema Netzversorgung unter Hochverfügbarkeitsaspekten umfassend betrachtet wird.

4.5.7 Lüfter

Wie bereits in dem Kapitel 4.5.1 „CPU (Central Processing Unit)“ dargestellt wurde, sind insbesondere die CPU, aber auch alle sonstigen Halbleiterbauelemente, äußerst empfindlich gegenüber einer Überschreitung der maximal zulässigen Betriebstemperatur. Bei einer Überschreitung können die Halbleiterbauelemente im schlimmsten Fall irreparabel zerstört werden.

Daher sollten in einer Hochverfügbarkeitsumgebung sowohl die Temperatur innerhalb des Server Gehäuses, als auch die CPU Gehäusetemperatur und die Umgebungstemperatur, fortlaufend überwacht und an einen zentralen Leitstand gemeldet werden.

4.5.8 Inter-Prozessor-Kommunikation

Ein wesentliches Merkmal von Mehrprozessorsystemen ist das zugrundeliegende Verbindungsnetz. Unter dem Verbindungsnetz werden in den nachfolgenden Ausführungen diejenigen Komponenten verstanden, welche für den Austausch von Informationen (Daten und Steuersignale) zwischen den Knoten eines Systems notwendig sind. Dem Verbindungsnetz kommt eine große Bedeutung zu, da sowohl die Leistung der Inter-Prozessor-Kommunikation als auch die Fehlertoleranz durch die geografische Wegeführung des Verbindungsnetzes maßgeblich beeinflusst werden können.

Entsprechend dem OSI-Referenzmodell beziehen sich die nachfolgenden Ausführungen hinsichtlich der Verbindungsnetze ausschließlich auf die Schicht 1 oder auch die physikalische Schicht (*engl. Physical layer*) entsprechend dem OSI-Referenzmodell, welches die niedrigste Schicht darstellt und auch als Bitübertragungsschicht bezeichnet wird.

Bei den Verbindungsnetzen kann unter anderem zwischen statischen und dynamischen Verbindungsnetzen unterschieden werden. Ein statisches Verbindungsnetz ist aus einer Menge von fest installierten Verbindungsleitungen zwischen den Knoten eines Netzes aufgebaut. Einzelne Knoten können dabei unterschiedlich viele Knoten als Nachbarn besitzen.

Bei dynamischen Netzen können Verbindungen zwischen Knoten je nach Bedarf hergestellt werden. Für die Übermittlung von Daten werden hier die Verbindungen jeweils neu über Schaltelemente hergestellt. Dynamische Netze enthalten aus diesem Grund eine Komponente „Schaltnetz“, an die alle Knoten des Netzes über Ein- und Ausgänge angeschlossen sind.

Dieses „Schaltnetz“ kann z. B. als Kreuzschienenverteiler (*engl: crossbar switch*) ausgeführt sein, wobei mehrere Prozessoren gleichzeitig und blockierungsfrei miteinander kommunizieren können. Im Gegensatz zu den statischen Verbindungsnetzen bestehen daher bei den dynamischen Netzen keine direkten Verbindungen zwischen den Knoten, da hier immer ein Schaltnetz zwischengeschaltet ist.

Auch unterscheiden sich statische und dynamische Verbindungsnetze in der Art, wie die Verbindung zwischen einzelnen Knoten hergestellt wird. Bei den dynamischen Netzen sind alle notwendigen Funktionen zur Steuerung und Kontrolle des Netzes in der Komponente „Schaltnetz“ konzentriert. Bei den statischen Netzen liegt die Kontrolle und Steuerung des Verbindungsaufbaus in den Netzknoten.

4.5.9 Dark-Fibre / City LAN / PDH / SDH

Dark-Fibre (*dt. „dunkle Faser“*) ist eine LWL-Leitung (Lichtwellenleiter-Leitung) die unbeschaltet, also ohne Endgeräte, verkauft oder vermietet wird. Die Faser erlaubt dabei eine Punkt-zu-Punkt Verbindung zwischen zwei Standorten. Für die Nutzung dieser Verbindung müssen an den beiden Standorten entsprechende Endgeräte angeschlossen werden, die eine Datenübertragung über Lichtwellenleiter erlauben.

Die Dark-Fibre besteht meist aus einer Singlemode-Faser (auch Monomode genannt), wobei der Kern einen solch kleinen Durchmesser besitzt, dass sich nur eine Mode (ein „Strahl“) ausbreiten kann. Dadurch sind Singlemode-Faser optimal für lange Strecken und sehr hohe Übertragungsraten geeignet.

Für die Überbrückung von kurzen Distanzen oder für die Inhouse LAN Verkabelung werden statt den Singlemode-Fasern vorwiegend sogenannte Multimode-Fasern verwendet. Diese Fasern verfügen über einen deutlich größeren Kerndurchmesser, sodass hier die optische Ansteuerung in den Endgeräten auch mit herkömmlichen LEDs (Light Emitting Diode, Leuchtdioden) statt den deutlich teureren Laser Dioden, die für die Singlemode-Faser erforderlich sind, verwendet werden können.

Die maximale Reichweite ergibt sich zum einen aus dem verwendeten Faser Typ, den optischen Verlusten auf der gesamten Strecke. (Dämpfung), der Leitung des Senders bzw. der Empfindlichkeit des Empfängers sowie den für die Übertragung verwendeten Wellenlängenbereich.

Dabei sind typischerweise die folgenden Reichweiten ohne einen zwischengeschalteten Repeater möglich:

- Multimode (MM): 850 nm, bis 1,5 km
- Singlemode (SM): 1310 nm, 10 bis 45 km
- Singlemode (SM): 1550 nm, bis 100 km

Abhängig von dem verwendeten Codierungsverfahren (z. B. 4B/5B, NRZ, NRZI, MFM), der Datenübertragungsrate, der optischen Sendeleistung der Laserdioden, des Rauschabstandes des Empfängers und der Anzahl der in dem Verbindungsweg eingefügten Spleiße (um hier nur einige wenige Parameter zu nennen) können die in der Praxis erreichbaren Längenangaben abweichen.

Unter einem Begriff City-LAN wird ein Weitverkehrsnetz mit einer lokalen Ausdehnung auf ein Stadtgebiet oder einen entsprechenden Ballungsraum verstanden. Da innerhalb eines Stadtgebietes meist ein hoher Bedarf an ausreichend dimensionierter Kommunikationsinfrastruktur existiert, gibt es in der Regel eine Vielzahl von Anbietern die Hochgeschwindigkeitsnetze auf der Basis von ATM, MPLS, SDMS, WDM oder weiteren Protokollen und Technologien anbieten.

Aufgrund dem ausreichendem Angebot von Hochgeschwindigkeitsleitungen, insbesondere in den Ballungsgebieten, bestehen insgesamt gute Voraussetzungen für die Einrichtung von redundanten Rechenzentren. Neben einer direkten Punkt-zu-Punkt Verbindung zwischen zwei geografisch verteilten Rechenzentren über eine Dark-Fibre besteht häufig auch noch die Möglichkeit einer Anschaltung an die bestehende City-LAN-Infrastruktur eines Providers. Durch die in Ballungsgebieten relativ kurzen Wege < 15km zu einen City LAN Provider kann das Risiko eines Ausfalls oder einer Störung bei einer längeren Punkt-zu-Punkt-Verbindung minimiert werden, da die Verbindungen innerhalb des City-LANs meist als mehrfach redundante Ringe realisiert sind.

Eine weitere Option für Weitverkehrsnetze ist die SDH – Synchrone Digitale Hierarchie. SDH ist ein Übertragungsverfahren auf Basis von redundanten Glasfaserringen mit denen ausfallsichere Transportverbindungen realisiert werden können. Das SDH Verfahren ist eine Weiterentwicklung des früheren PDH – Plesiochrone Digitale Hierarchie Verfahrens, welches ursprünglich für die Digitalisierung des öffentlichen analogen Telefonnetzes eingeführt wurde.

Mit SDH können neben Sprachdaten auch ATM-Zellen und IP-Daten über ein international verknüpftes Netz übertragen werden. In der Netzstruktur sind meist vordefinierte Ersatzstrecken für den Fehlerfall definiert, wobei innerhalb von Sekunden nach dem Ausfall eines Verbindungsabschnitts automatisch auf eine Alternativstrecke umgeschaltet werden kann.

4.5.10 Wavelength-division multiplexing (WDM)

Das Wellenlängenmultiplexverfahren (*engl: Wavelength Division Multiplexing*) ist ein optisches Frequenzmultiplexverfahren, das bei der Übertragung von Daten (Signalen) über ein Glasfaserkabel verwendet wird. Beim dem Wellenlängenmultiplexverfahren werden die aus verschiedenen Spektralfarben bestehende Lichtsignale zur Übertragung in einem Lichtwellenleiter verwendet.

Jede dieser so erzeugten Spektralfarben bildet somit einen eigenen Übertragungskanal, auf den man die Daten (Signale) eines Senders modulieren kann. Die so modulierten Daten (Signale) werden dann durch optische Koppelelemente gebündelt und gleichzeitig, unabhängig voneinander, übertragen. Am Ziel dieser optischen Multiplexverbindung werden die einzelnen optischen

Übertragungskanäle durch optische Filter oder wellenlängensensible opto-elektrische Empfängerelemente wieder getrennt.

Die Interface-Module der WDM Endgeräte unterstützen meist alle gängigen Schnittstellen. Damit können alle bisherigen Systeme, die sowohl optisch als elektrisch arbeiten, ihre Signale über WDM-Strecken und Ringe übertragen. Auch lassen sich damit Verbindungen über Fibre Channel und ESCON-Systeme über sehr weite Entfernungen realisieren.

Obwohl dieses Verfahren z. B. eine angemietete Dark-Fibre zwischen zwei Standorten optimal nutzen kann, bringt dieses Verfahren in Hinblick auf eine Hochverfügbarkeit keine Vorteile. Durch dieses Verfahren kann unter Umständen auch das Bewusstsein dafür verloren gehen, dass es sich bei dem physischem Medium immer nur noch um eine einzige Glasfaser handelt, obwohl an den Endgeräten das vorhanden sein von mehrere Verbindungen suggeriert wird.

4.5.11 ESCON Channel / FICON

ESCON (Enterprise Systems Connection) ist ein Protokoll, das eine Übertragungsrate von 17 MB/s pro Kanal auf der Basis von Lichtwellenleitern ermöglicht. Diese Technologie wurde von einem führenden Hersteller von Mainframe Systemen entwickelt, um einen schnellen Datenaustausch zwischen dem zentralen Rechnereinheiten und dessen Peripheriegeräten (z. B. Plattensubsysteme, Bandlaufwerke) zu ermöglichen.

ESCON wurde 1990 eingeführt und war bis zu der Markteinführung von FICON (Fibre CONnection) die führende, allerdings auch proprietäre Technologie für die Kommunikation zwischen Mainframe-Rechnern des besagten führenden Herstellers und dessen Peripherie Geräten. Für die Einrichtung der auf der ESCON Technologie basierenden Verbindungsnetze wurden sogenannte ESCON-Direktoren eingesetzt, wodurch eine statische Zuordnung der verwendeten Kanäle ermöglicht wurde. Während mittlere Mainframe-Rechnersysteme typischerweise zwischen 4 und 40 ESCON-Kanäle nutzen, werden von großen, verteilten Mainframe Systeme teilweise über 1000 Kanäle genutzt. Dabei können Entfernungen bis zu 43 Kilometer je Kanal überbrückt werden.

Im Gegensatz zu der proprietären ESCON Technologie nutzt die FICON Technologie den standarisierten Fibre Channel als Transportschicht, wodurch die (gegenüber ESCON) höheren Datenraten von Fibre Channel auch für den Mainframe Einsatz nutzbar wurden. Weiterhin ist es durch die FICON Technologie möglich geworden, Storage Area Networks auf der Basis von Fibre Channel aufzubauen in denen sich z. B. klassische Großrechner und Workstations die externen Speicher Ressourcen teilen. Die Übertragungsrate eines FICON Kanals beträgt abhängig von den angeschlossenen Systemen zwischen 1 und 4 Gbyte pro Sekunde und je FICON Kanal.

Ähnlich wie der ESCON-Direktor in der ESCON Technologie bildet der FICON-Switch (FICON Director) In der FICON-Architektur die Kerneinheit. Er implementiert eine Switched Point-to-Point-Topologie für E/A-Kanäle und Steuereinheiten, wobei ein FICON-Switch bis zu 60 Kanäle und Steuereinheiten dynamisch und nicht-blockierend (Crossbar Switch) miteinander verschalten kann.

Bei der FICON Technologie sind typische Entfernungen von bis zu 3 km für optische Übertragungen möglich. Die zulässigen Entfernungen können durch den Einsatz von sogenannten Extended Distance Laser Link-Einrichtung auf 20, 40 oder 60 km erhöht werden. Dies ist insbesondere für Unternehmen wichtig, die aus Katastrophenschutzgründen zwei getrennten Rechenzentren betreiben.

4.5.12 Backplane / Parallele Bus Systeme

Praktisch alle heute eingesetzten Rechner sind modular aufgebaut und unterstützen mindestens einen standarisierten Bus für Interfaces und Peripheriegeräte. Dadurch wird eine optimale

Konfiguration der Systeme ermöglicht. Bei Industrierechnern oder Rechnern in der Kategorie Enterprise-Server werden auch häufig Bussysteme verwendet, um die teilweise als „System Boards“ oder „Cell Boards“ bezeichneten SMP-Knoten zu einem größeren Server-System zusammenzuschalten. Auf jedem dieser „Boards“ befinden sich eine oder mehrere CPUs, Hauptspeicher-Module sowie ein Steckverbindung für die Verbindung mit dem Backplane.

Das über die Backplane realisierte parallele Bussystem verbindet somit die angeschlossenen Rechner-Knoten über eine gemeinsame Übertragungsleitung, wobei diese jedem der angeschlossenen Rechner-Knoten für ein bestimmtes Zeitintervall abwechselnd zur Verfügung gestellt wird. Die Vorteile einer solchen Verbindung sind: eine sehr einfache Erweiterung des Systems um zusätzliche Rechner, die dadurch relativ niedrigen Kosten und der vergleichsweise geringe Hardware-Aufwand durch eine geringe Zahl von Verbindungen und Komponenten.

Diesen Vorteilen steht aber auch eine Reihe gravierender Nachteile gegenüber. Ein einzelner Systembus weist eine sehr geringe Fehlertoleranz auf, da ein Ausfall des Systembusses das gesamte Server-System lahmlegen kann. Die Erweiterung des Systems um weitere Rechner erhöht die Zahl der Zugriffskonflikte und senkt dadurch die Systemleistung. Die Leistung eines Systems mit einem einzelnen Bus wird dabei also fast vollständig durch die Leistung dieses Busses bestimmt.

Um die Nachteile einer solchen Busstruktur abzuschwächen, existieren verschiedene Ansatze. Ein möglicher Ansatz besteht darin, statt des nur einmal vorhandenen, bidirektionalen Busses zwei unidirektionale Busse zu verwenden. Eine wirkliche Verbesserung wird aber in der Regel erst durch den Einsatz zusätzlicher bidirektionaler Busse zu erreichen.

Jeder Rechner Knoten ist dabei mit jedem im System vorhandenen Bus verbunden. Dadurch wird die Fehlertoleranz des Systems erheblich gesteigert und auch die Zahl der Zugriffkonflikte stark vermindert, da mehrere alternative Pfade zur Herstellung einer Kommunikationsverbindung existieren.

4.6 Kopplung von Server-Systemen

Im Gegensatz zu den im vorhergehenden Kapitel 4.5.8 „Inter-Prozessor-Kommunikation“ vorgestellten typischen und weit verbreiteten Ansätzen für eine Inter Prozessor Kommunikation erfolgt in diesem Abschnitt eine mehr generische Betrachtung für eine Kommunikation von Server-Systemen. Die vorgestellten generischen Ansätze sind dabei prinzipiell sowohl für eine direkte Inter Prozessor Kommunikation als auch für eine Server-Server oder Server-Client anwendbar.

Die Server-Dienste auf der Sender- und Empfängerseite müssen nach festgelegten Regeln arbeiten, damit sie sich einig sind, wie die empfangenen Daten zu verarbeiten sind bzw. in welchen Format die Daten für eine Übertragung aufbereitet werden müssen. Die Festlegung dieser Regeln wird in einem Protokoll beschrieben, welches die Kommunikation zwischen zwei Endpunkten abwickelt.

Die Kommunikationsbeziehung zwischen zwei Endpunkten muss des Weiteren einer Reihe von Anforderungen gerecht werden. Die Probleme, die dabei gelöst werden müssen, reichen von Fragen der elektronischen Übertragung der Signale über eine geregelte Reihenfolge in der Kommunikation bis hin zu abstrakteren Aufgaben, die sich innerhalb der kommunizierenden Anwendungen ergeben.

Da sich diese Anforderungen meist nicht mit einem einzigen, monolithischen Protokoll erfüllen lassen, wird die Kommunikationsbeziehung in mehrere Schichten aufgeteilt. Neben herstellerspezifischen Schichtenmodellen existieren auch einige öffentlich verfügbaren Standards für die Einteilung einer Kommunikationsbeziehung in verschiedene Protokollschichten. Zu den bekanntesten dieser Standards gehört sicherlich das OSI-Modell (Open Systems Interconnection

Reference Model) sowie das TCP/IP-Referenzmodell, welches die Grundlage für die Internet-Protokoll-Familie bildet.

Bei den folgenden Betrachtungen von möglichen Netzwerk Topologien muss prinzipiell zwischen einer physikalischen Verbindung und einer logischen Verbindung unterschieden werden. Bei einer physikalischen Verbindung besteht in der Regel eine direkte Verbindung zwischen der sendenden Station und einer oder mehreren empfangenden Stationen. Bei einer logischen Verbindung können sich nahezu beliebige weitere Stationen zwischen der sendenden Station und der oder den empfangenden Stationen befinden. Abhängig von dem verwendeten Übertragungsmedium und dem verwendeten Protokoll auf der physikalischen Ebene ist mit der physikalischen Verbindung entweder nur eine Punkt-zu-Punkt Verbindung oder aber auch eine Punkt-zu-Mehrpunkt Verbindung möglich.

Bei dem Übertragungsmedium handelt es sich entweder um eine leitungsgebundene Verbindung (z. B. Kupferkabel, Glasfaser) oder um eine wie auch immer geartete drahtlose Verbindung (z. B. WLAN, Richtfunk, Infrarot). Auch wenn bei jedem Übertragungsmedium prinzipiell eine physikalische Punkt-zu-Mehrpunkt Datenübertragung möglich ist, gibt es Verfahren, die eher wenig geeignet sind (z. B. bei einer Glasfaser, einer Richtfunkstrecke), und andere, die deutlich besser geeignet sind (z. B. bei Kupferkabel bei Ethernet, ungerichteten Funkverbindungen bei Wireless LAN).

Weiterhin kann bei der Nutzung des Übertragungsmediums noch zwischen Basisbandverbindungen, Breitbandverbindungen und einer Fülle von verschiedenen Multiplexvarianten unterschieden werden. Bei Basisbandverbindungen werden die zu übertragenen Signale in ihrer ursprünglichen frequenzmäßigen Lage, also ohne Trägerfrequenzen auf dem Übertragungsmedium aufgebracht. Das Frequenzspektrum des Sendesignals und die von ihm eingenommene Bandbreite sind dabei direkt abhängig von der Übertragungsgeschwindigkeit. Basisbandsysteme bieten also nur einen Kanal, der z. B. über einen Multiplexer wiederum logisch in verschiedene Kanäle aufgeteilt werden kann.

Der Begriff Breitbandverbindung hat unterschiedliche Bedeutungen. In der Regel handelt es sich aber bei einer Breitbandverbindung um ein Verfahren bei der durch die Modulation von Trägerfrequenzen mehreren Signalen gleichzeitig in unterschiedlichen Frequenzbereichen übertragen werden können.

Die folgenden Ausführungen in Bezug auf eine mögliche Kopplung von Server-Systemen sind nun eher ein allgemeiner Ansatz, da vollständige und umfassende Betrachtung den Rahmen des Kapitels sprengen würde. So beziehen sich die Beschreibungen nahezu ausschließlich auf die physikalische Verbindung zwischen zwei oder mehren Endpunkten, wobei hier ein Übertragungsmedium existieren muss, welches für die Übermittlung von Nachrichten geeignet ist. In der Regel handelt es sich bei diesem Übertragungsmedium um eine leitungsgebundene Verbindung (z. B. Kupferkabel, Glasfaser) oder um eine wie auch immer geartete drahtlose Verbindung (z. B. WLAN, Richtfunk, Infrarot).

Jede dieser Verbindungen kann nun, abhängig vom Übertragungsmedium, den Modulationsverfahren sowie den darauf aufsetzenden Protokollen prinzipiell entweder als eine Punkt-zu-Punkt, Punkt-zu-Mehrpunkt oder Mehrpunkt-zu-Mehrpunkt Verbindung genutzt werden. Auch kann die auf dem physikalischen Medium aufsetzende logische Topologie von der physischen Topologie abweichen. So kann z. B. Ethernet physisch als Stern oder als Bus aufgebaut sein – logisch gesehen ist es eine Bus-Topologie, da der Datenfluss von einem Endgerät gleichzeitig zu allen anderen Endgeräten erfolgt. Token Ring wird physisch als Stern über einen

Ringleitungsverteiler (MAU) realisiert, ist jedoch eine logische Ring-Topologie, da der Datenfluss logisch gesehen von Endgerät zu Endgerät läuft.

4.6.1 Punkt-zu-Punkt

Bei einer Punkt-zu-Punkt-Verbindung sind genau zwei Kommunikationspartner beteiligt. Dabei muss noch weiter unterschieden werden zwischen einer physischen und einer logischen Punkt-zu-Punkt-Verbindung. Bei einer physischen Punkt-zu-Punkt-Verbindung existiert genau eine einzige Übertragungsleitung zwischen der Datenquelle und der Datensenke. Störungen oder Unterbrechung der Übertragungsleitung führen zu einem Totalausfall der Kommunikationsbeziehung.

Im Gegensatz dazu kann eine logische Punkt-zu-Punkt-Verbindung mit keiner, einer oder mehreren redundanten, physischen Verbindungen unterlegt sein. Die Redundanzen können dabei wiederum die gesamte physische Verbindung oder auch nur Teilabschnitte der gesamten Verbindung umfassen. Bei dem Ausfall einer physischen Verbindung ist entweder bei verbindungsorientierten Protokollen typischerweise ein neu Aufbau der Verbindung notwendig. Bei verbindungslosen Protokollen werden die Datenpakete in der Regel über eine intelligente Wegefindung (über Routing Protokolle) über alternative physische Verbindungen zum Ziel geleitet.

4.6.2 Bus / Stern

Eine Bus Verbindung entspricht im Wesentlichen einer Punkt-zu-Punkt-Verbindung, wobei der Bus als physische Verbindung zwischen mehreren Stationen von diesen entweder gleichzeitig bzw. nacheinander genutzt werden kann. Die Stationen können dabei hintereinander, nebeneinander oder in Reihe angeordnet sein. Im Gegensatz zur physischen Punkt-zu-Punkt-Verbindung, die nur eine Unicast (pro Sender und Empfänger) Kommunikation erlauben, ist bei einem Bus auch Multicast (ein Sender und eine Gruppe von Empfängern) und Broadcast (ein Sender und alle aktiven Station als Empfänger) Kommunikation möglich. Wie auch bei einer physischen Punkt-zu-Punkt-Verbindung bildet der physische Bus einen Single Point of Failure. Bei einem Ausfall des Bussystems ist keine Kommunikation möglich.

Bei der klassischen Bus-Topologie wurde der Bus meist durch ein Koaxialkabel realisiert, wobei die teilnehmenden Stationen über eine sogenannte Media Attachment Units (MAUs) mit dem Koaxialkabel verbunden wurden. Bei dieser ursprünglichen Variante war keine aktive Netzwerk-Komponenten notwendig, da die Kommunikation und der Zugriff auf das Übertragungsmedium eigenverantwortlich durch die in an das Bussystem angeschlossenen Stationen erfolgte.

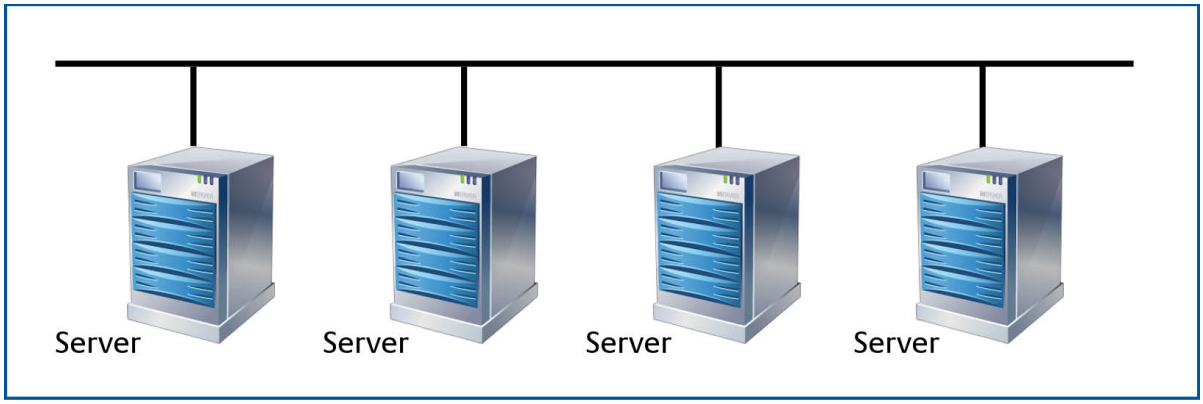


Abbildung 7: Kopplung von Server-Systemen in einer Bus Topologie

Bei der Stern-Topologie gibt es eine zentrale Komponente, die eine Verbindung zu allen angeschlossenen Stationen unterhält. Jede Station ist über eine eigene physikalische Leitung an die

zentrale Netzwerk-Komponente angebunden. Bei der zentralen Netzwerk-Komponente handelt sich im Regelfall um einen Hub oder einen Switch. Der Hub oder Switch übernimmt die Verteilfunktion für die Datenpakete. Dazu werden die Datenpakete auf elektronischem Weg entgegen genommen und an das Ziel weitergeleitet. Auch wenn prinzipiell eine passive Auslegung des Sternverteilers möglich wäre, werden in der Praxis jedoch meist aktive Komponenten eingesetzt.

Die Stern-Topologie kann auch als ein zusammengefaltetes Bus-System, manchmal auch als „Collapsed Backbone“ bezeichnet, aufgefasst werden. Durch die physisch separaten Zuleitungen zu jeder der an dem Stern angeschlossenen Station können die angeschlossenen Stationen ausreichend entkoppelt werden, sodass bei einer ausgefallenen oder störenden Station diese einfach von der Kommunikation ausgeschlossen werden kann. Allerdings stellt bei der Stern-Topologie der Sternverteiler ebenso wie der Bus bei der Bus-Topologie einen Single Point of Failure dar.

Für die Erhöhung der Fehlertoleranz wird der Sternverteiler meist auch mindestens doppelt ausgeführt, wobei weiterhin jede Station eine separate physische Verbindung zu jeder der zentralen Sternverteiler hat. Somit kann sowohl bei einem Ausfall einer der zentralen Sternverteiler als auch bei dem Ausfall von einer der physischen Verbindung zwischen einer Station und dem zentralen Sternverteiler die Kommunikation insgesamt aufrechterhalten werden.

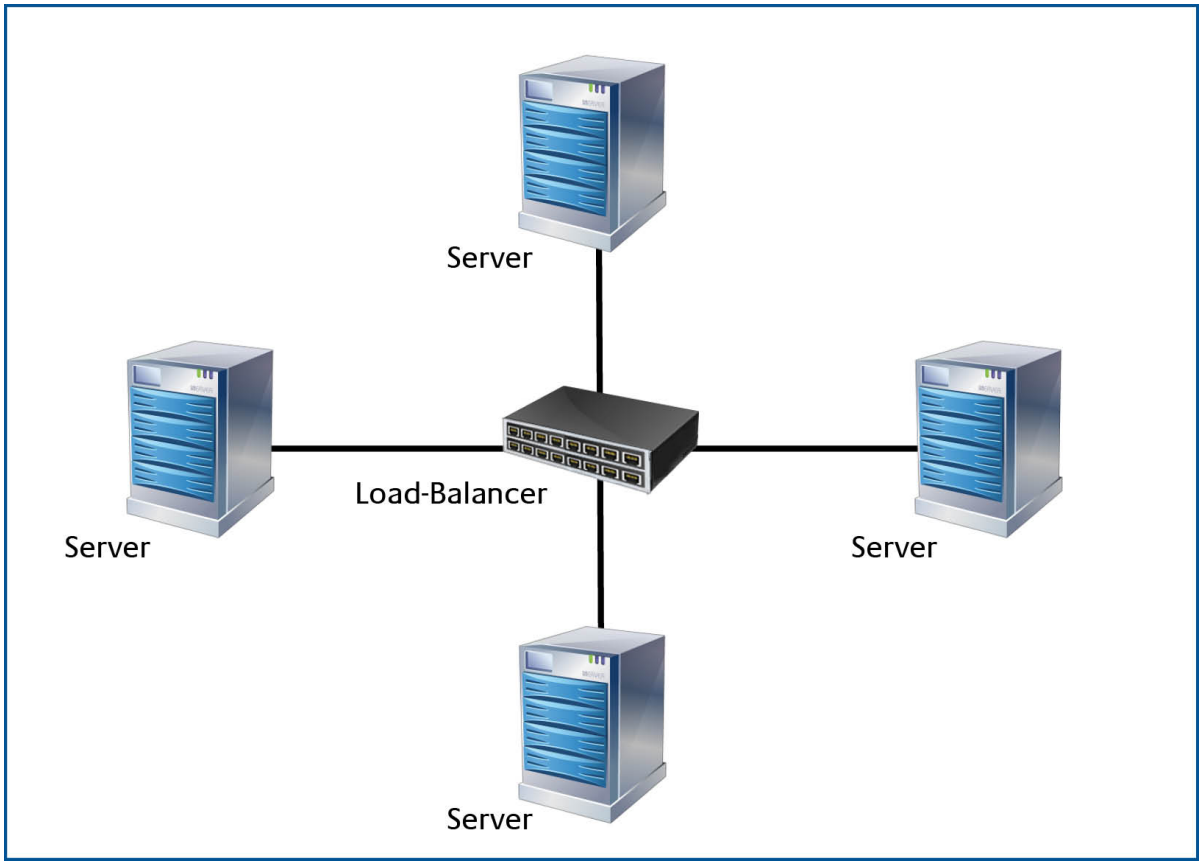


Abbildung 8: Kopplung von Server-Systemen in einer Stern-Topologie

4.6.3 Ring / logisch / physikalisch

Bei einer physikalischen Ring-Topologie sind alle Netzwerk-Stationen im Kreis angeordnet und über eine Kabelstrecke miteinander verbunden. Das bedeutet, dass an jeder Station ein Kabel ankommt und ein Kabel abgeht. Die physikalische Verbindung der Stationen erfolgt über einen

Ringleitungsverteiler (MAU), der allerdings keine aktiven Komponenten enthält. Somit benötigt die klassische Ring-Topologie keinerlei aktive und übergeordnete Netzwerk-Komponenten in dem Ring, da die gesamte Kommunikation und der Zugriff auf das Übertragungsmedium eigenverantwortlich durch die in dem physikalischen Ring angeschlossenen Stationen erfolgt.

Bei dieser einfachen Struktur fallen die Verbindungen zischen den angeschlossenen Stationen teilweise oder auch vollständig aus, wenn der Ring an einer beliebigen Stelle unterbrochen wird.

Um diesen Single Point of Failure vorzubeugen, kommen in der Praxis auch häufig Doppel Ringstrukturen mit gegenläufigen Kommunikationsverbindungen zum Einsatz. Durch dieses Verfahren ist der Ring unempfindlich gegenüber einer einzelnen Unterbrechung an einer beliebigen Stelle im physischen Ring. Dabei werden die verbleibenden Fragmente des ursprünglichen physischen Ringes mit Hilfe des redundant vorhandenen Ringes zu einem neuen Ring verknüpft.

Mit diesem Verfahren kann allerdings nur eine Fehlertoleranz bei einen Single-Fault (einer einfachen Unterberechnung) erreicht werden. Bei mehrfachen Unterbrechungen (Double-Faults or Multiple-Faults) der Ringstruktur ist dieses Verfahren nicht brauchbar. Aufgrund der hohen Fehleranfälligkeit durch die in dem Ring eingefügten Stationen und der nur einfachen Redundanz ist die klassische Ring Topologie in dem Bereich von Local Area Netzwerken (LAN) heutzutage nur noch selten anzutreffen.

Im Gegensatz zu den Local Area Netzwerken (LAN) ist die Ring Struktur in dem Bereich der auf der SDH (Synchronous Digital Hierarchy) basierenden City LANs weit verbreitet. Der Zugang zu der Ring Struktur erfolgt dabei über so genannte ADMs (Add-Drop-Multiplexer), die die Datenströme in den Ring einphasen, bzw. die Datenströme aus dem Ring entnehmen.

Ein wesentliches Merkmal der SDH-Technik ist dabei die automatische Umschaltung im Fehlerfall. Der redundante Ring dient dabei als Ersatzweg. Bei einer Störung des primären Ringes schaltet das APS (Automatic Protection System) vom primären Ring auf den redundanten Ring. Diese Topologie ist unter der Bezeichnung 4-Faser MS-SPRing (Multi-Section-Shared-Protection-Ring) ab STM-16 aufwärts standardisiert. Nach dem SDH Standard müssen die Ersatzschaltmaßnahmen nach dem Erkennen einer Störung automatisch innerhalb von 50 Millisekunden abgeschlossen sein.

Um die Nachteile der klassischen Ringstruktur abzuschwächen, existieren teilweise auch Ansätze, wobei zusätzliche Verbindungen zwischen nicht direkt benachbarten Knoten eingefügt werden. Damit kann die Fehlertoleranz erhöht werden. Es erhöht sich dabei allerdings auch der Verbindungsaufwand. Ein Ring mit zusätzlichen Verbindungen wird oft als „Chordaler Ring“ bezeichnet.

Neben einem physikalischen Ring, bei dem die physischen Verbindungen als geschlossener Ring ausgeführt werden, existiert auch noch das Konzept eines logischen Rings. Der logische Ring setzt in der Regel auf einer beliebigen Stern -, Bus -, oder Stern–Bus Topologie auf, wobei ein so genanntes „Token“ (*engl. Zeichen*) zwischen den angeschlossenen Stationen herumgereicht wird. In Bezug auf die Verfügbarkeit ist eine logische Ringstruktur dabei vollständig abhängig von der zugrundeliegenden physischen Verbindungsstruktur.

4.6.4 Vermaschung

Im Idealfall würde zwischen jeden Kommunikationspartner eine dedizierte Verbindung bestehen, was einer vollständigen Vermaschung entspricht. Bei dem Ausfall einer beliebigen Verbindung gibt es im Regelfall somit immer mehrere alternative Strecken, um den Datenverkehr fortzuführen. Da bei einer vollen Vermaschung aller Stationen (n2-n)/2 Verbindungen geschaltet werden müssen, ist eine volle Vermaschung nur bei Verbindungsnetzen mit relativ wenigen Stationen sinnvoll einsetzbar.

Als Alternative zu einer vollständigen Vermaschung kommen bei der Kopplung von Server-Systemen in der Regel alternative Ansätze zum Einsatz, welche unter anderem z. B. eine möglichst hohe Fehlertoleranz mit einer möglichst geringen Anzahl an physischen Verbindungen kombinieren können. Zu diesen teilweise oder indirekt vermaschten Verbindungsnetzen können unter anderem verschiedene Ausprägungen einer Baum-, Gitter-, 2D-Torus-, 3D-Torus- oder Hypercube Topologie verstanden werden. Dabei hat jede Topologie ihre Vor- und Nachteile und sollte für das individuelle Einsatzgebiet ausgewählt werden. Die Auswahlkriterien umfassen unter anderem den Grad der Fehlertoleranz, der Kosten und die zur Verfügung stehenden physischen Medien für die Datenübertragung.

5 Geografische Redundanz

Die bisherigen Ausführungen erläuterten die Konzeption von Serverlösungen gegen Gefahren und Fehler, die einzelnen Bauteilen innewohnt oder nur eine begrenzte räumliche Ausdehnung um den Standort haben. Das Kapitel Netzwerk zeigt Lösungen und deren Grenzen auf, um geografisch verteilte Serverarchitekturen robust gegen Schadensereignisse mit einer großflächigen Ausdehnung konstruieren zu können. Solche Realisierungen sind technisch sehr komplex und finanziell sehr aufwendig. Die Notwendigkeit ergibt sich allerdings, wenn z. B. zur Aufrechterhaltung der Ordnung, der Versorgung größerer Bevölkerungsgruppen, Rettung unwiederbringbarer Sachwerte oder extrem kritischer Unternehmensziele, die servergestützten Prozesse auch in solchen allgemein als Katastrophe bezeichneten Lagen aufrecht erhalten werden müssen.

Daneben gibt es eine Klasse von Anwendungen, bei denen quasi-statische Daten weltweit von Clienten abgerufen werden (dazu zählen zum Beispiel Suchmaschinen und Download-Server). Aus Gründen der wirtschaftlichen und technischen Effizienz bietet sich hierbei eine geografische Verteilung der Server an. Mit dieser Lastverteilung wird automatisch eine höhere Verfügbarkeit bei höherer Robustheit gegen großflächige Schadensereignisse erzielt.

Für den Betrieb eines physischen Servers müssen bestimmte Umgebungsbedingungen und Grenzwerte eingehalten werden, damit ein fehlerfreier Betrieb gewährleistet werden kann. Die für den Betrieb eines physischen Server erwarteten Umgebungsbedingungen und Grenzwerte sind in der Regel abhängig von der vorgesehenen Einsatzumgebung und werden meist mehr oder wenig ausführlich von den Herstellern der Server spezifiziert. Diese Vorgaben können entweder den physischer Server insgesamt oder aber auch nur bestimmte Teilkomponenten des physischen Servers umfassen.

Werden die vorgegebenen Umweltbedingungen nicht eingehalten oder sogar die vom Hersteller vorgegebenen Grenzwerte überschritten muss mindestens mit vorübergehenden Ausfällen, meist aber mit einem permanenten Totalausfall des physisches Server gerechnet werden. Abhängig von der Server-Architektur müssen bei einem permanenten Ausfall eines physischen Servers entweder der komplette Server oder aber auch nur die ausgefallen Teilkomponenten ersetzt werden.

Da die meisten Schadensereignisse eine bestimmte lokale Ausdehnung haben, kann durch einen ausreichenden Abstand zwischen Servern erreicht werden, dass bei einem gegebenen Schadensereignis nur ein Teil der Server betroffen ist. Abhängig von dem Umfang und der Ausdehnung eines Schadensereignis kann sich die geografische Separation als ausreichend erweisen oder aber auch nicht. Z. B. kann sich die Platzierung der Server in einem separaten Brandabschnitt bei einem lokal begrenzten Feuer als ausreichend erweisen. Bei einem regionalen Erdbeben oder einer Flutkatastrophe wird sich dieser geringe Abstand wahrscheinlich eher als unzureichend erweisen.

Für eine Unabhängigkeit von den lokalen Schadensereignissen werden die Server häufig an mehreren Standorten verteilt, wobei die Standorte untereinander eine bestimmte Mindestentfernung aufweisen sollten. Diese Verteilung der Server auf verschiedene Standorte wird häufig auch als geografische Redundanz bezeichnet, da die physischen Server-Ressourcen geografisch verteilt platziert sind.

5.1 Geografische Verteilung

Die Ergebnisse einer Risikoanalyse können als Grundlage für eine Abschätzung herangezogen werden, welche Mindestentfernung zwischen den Server Standorten als ausreichen erachtet werden kann. In Bezug auf den sicheren Betrieb der Server Hardware können die im Folgenden dargestellten Umgebungsbedingungen als Sicherheitsziele für die Server-Hardware herangezogen werden.

Ausgehend von diesen Sicherheitszielen können nun zum Einen wiederum Bedrohungen ermittelt werden, die diesen in der Form von Umweltbedingungen für die Server-Hardware bezeichneten Sicherheitszielen entgegenstehen. Des Weiteren können auf Basis dieser Sicherheitszielen auch bestehende oder noch umzusetzende Maßnahmen identifiziert werden, die diesen Bedrohungen entgegenstehen bzw. die geeignet sind, diese eliminieren zu können.

Bei dem Betrieb eines physischer Server müssen bestimmte Umgebungsbedingungen und Grenzwerte eingehalten werden, damit ein fehlerfreier Betrieb gewährleistet werden kann. Die für den Betrieb eines physischen Server erwarteten Umgebungsbedingungen und Grenzwerte sind in der Regel abhängig von der vorgesehenen Einsatzumgebung und werden meist mehr oder wenig ausführlich von den Herstellern der Server spezifiziert. Diese Vorgaben können entweder den physischer Server insgesamt oder aber auch nur bestimmte Teilkomponenten des physischen Servers umfassen.

Werden die vorgegebenen Umweltbedingungen nicht eingehalten oder sogar die vom Hersteller vorgegebenen Grenzwerte überschritten muss mindestens mit vorübergehenden Ausfällen, meist aber mit einem permanenten Totalausfall des physisches Server gerechnet werden. Abhängig von der Server-Architektur müssen bei einem permanenten Ausfall eines physischen Servers entweder der komplette Server oder aber auch nur die ausgefallen Teilkomponenten ersetzt werden.

Zu den wesentlichen Umweltbedingungen zählt in erster Linie sicherlich die Einhaltung eines bestimmten Temperaturbereichs, da bei einer Überschreitung der gegebenen Grenzwerte die empfindlichen Halbleiterbauelemente irreparabel zerstört werden können. Ein weiterer wichtiger Punkt ist die Einhaltung einer maximalen Luftfeuchtigkeit sowie eines maximalen Staubgehaltes, da durch die Verunreinigungen der Umgebungsluft in Zusammenhang mit einer hohen Luftfeuchtigkeit leitfähige Verbindungen entstehen, die die auf elektronische Informationsverarbeitung stören können.

Neben den zuvor genannten Umwelteinflüssen gibt es aber auch noch weitere durch die Umwelt bedingte Einflüsse bei Natur- und Technikkatastrophen, die die Funktion eines Servers entweder temporär oder auch dauerhaft beeinflussen können. Zum einen zählen hierzu mechanische Einwirkungen wie z. B. Erschütterungen oder mechanische Verformungen, die oft fatale Auswirkungen auf die Funktion eines Servers haben.

Weiterhin können auch elektromagnetische Wellen (E-Gefahren) die Funktionsfähigkeit eines physischen Servers beeinträchtigen. Zu diesen elektromagnetischen Wellen, welche einen physischen Server beeinflussen können, gehören unter anderem Radiowellen, Mikrowellen oder Röntgen- und Gammastrahlung im Falle von nuklearen Katastrophen (A-Gefahren). Auch hier gibt es Grenzwerte, welche unter anderem in der Europäischen EMV-Richtlinie festgelegt sind.

Die Elektromagnetische Verträglichkeit (EMV) kennzeichnet dabei den üblicherweise erwünschten Zustand, dass sich technische Geräte einander nicht wechselseitig mittels ungewollter elektrischer oder elektromagnetischer Effekte störend beeinflussen. So müssen die Hersteller eines physischen Servers diesen so konzipieren, dass dieser zum einen bis zu einem gewissen Grenzwert

unempfindlich ist, gegenüber von außen einwirkenden elektromagnetische Wellen als auch das die von einem physischen Server ausgehenden Störimpulse innerhalb der vorgegebenen Grenzwerte liegen.

Als letzte Voraussetzung für den störungsfreien Betrieb eines physischen Servers soll hier noch die direkte und mittelbare Abhängigkeit eines Servers von einer störungsfreien Stromversorgung angeführt werden, die bei einem großräumigen Stromausfall im Allgemeinen nur über einen kurzen Zeitraum durch die eigene Organisation aufrecht gehalten werden kann.

Wie aus den vorhergegangenen Ausführungen ersichtlich gibt es eine nahezu unüberschaubare Vielzahl von Bedrohungen, die die Funktion eines physischen Servers gefährden können. Auch wenn die in dem Kapitel (siehe Beitrag „Infrastruktur“ im HV-Kompendium) genannten Maßnahmen umgesetzt werden, verbleiben immer noch eine Vielzahl von Bedrohungen, die immanent an den geografischen Standort des Servers gebunden sind. Um diesen aus der Umgebung eines gegebenen Servers resultierenden oder aus den Abhängigkeit von lokalen Infrastruktur resultierenden Bedrohungen entgegenzuwirken, werden physische Server meist auf verschieden geografische Standorte verteilt, so dass bei einer Störung an einem gegebenen Standort die weiteren Standorte nicht betroffen sind.

5.1.1 Lokal

Bei einer lokalen Verteilung der physischen Server befinden sich diese meist in dem gleichen Server-Raum, in dem gleichen Gebäude oder sonstigen eng benachbarten Gebäuden oder Gebäudeteilen.

Sofern sich die redundanten Server in einem gemeinsamen Gebäude befinden, werden diese häufig in getrennten Brandabschnitten untergebracht. Bei der Unterbringung in getrennten Gebäuden z. B. auf einem Campus oder einen Werkgelände werden diese verteilten Server meist über ein lokales Hochgeschwindigkeitsnetzwerk miteinander verbunden.

Diese räumliche Nähe bietet allerdings nur einen sehr bedingten Schutz gegenüber lokalen Ereignissen wie z. B. Stromausfällen, Erdbeben, Hochwasser oder Terroranschlägen, um hier nur einige potentielle Bedrohungen anzuführen. Allerdings bietet die lokal begrenzte Verteilung der Server eine ideale und kostengünstige Voraussetzung für ein Ressourcen Sharing bzw. eine enge Kopplung der Server.

5.1.2 Regional

Bei einer regionalen Verteilung der physischen Server befinden sich diese meist in der gleichen Stadt bzw. Ballungsraum oder sonst wie in einer räumlichen Abstand von < 100km. Die Server können hierbei z. B. in getrennten Rechenzentren untergebracht sein, die über ein wie auch immer geartetes Metropolitan Area Network (MAN) verbunden sind. Ein Metropolitan Area Network (MAN) bezeichnet dabei meist ein breitbandiges Telekommunikationsnetz, das überwiegend in ringförmiger Struktur aufgebaut ist und die wichtigsten Bürozentren einer Großstadt miteinander verbindet.

Neben den auf der SDH (Synchronous Digital Hierarchy) basierenden City LANs kommen auch die klassischen Protokollen wie ATM, MPLS, SMDS in Zukunft verstärkt auch Ethernet oder FireWire Technologien zum Einsatz. Diese Technologien können entweder auf bestehende SDH Systeme als Transportschicht oder aber auch direkt auf die zugrundeliegenden Übertragungsmedien aufsetzen.

Die MANs werden dabei häufig von international tätigen Telekommunikationsfirmen aufgebaut, die dann auf diese Weise verkabelte Metropolen wiederum in einem Wide Area Network (WAN) national oder sogar international Global Area Network (GAN) weiter vernetzen.

Als Alternative können die Rechenzentren auch über eine Punkt-zu-Punkt Verbindung auf Basis einer angemieteten Dark-Fibre verbunden werden, wobei hier im Sinne einer Hochverfügbarkeit zusätzliche Verbindungen zwischen den Rechenzentren existieren sollten, die bei einem Ausfall der Dark-Fibre genutzt werden können.

Im Gegensatz zu der zuvor dargestellten lokalen Verteilung der physischen Server bietet der Ansatz mit einer regionalen Verteilung in der Regel eine größere Sicherheit gegenüber lokalen Ereignissen wie z. B. Stromausfällen, Erdbeben, Hochwasser oder Terroranschlägen. Allerdings muss die erhöhte Sicherheit mit meist nicht unerheblichen Kosten für die Telekommunikationsverbindungen bezahlt werden, da die Verbindungen von externen Providern angemietet werden müssen. Die monatlich anfallenden Kosten sind dabei meist auch wiederum direkt abhängig von der räumlichen Entfernung sowie dem benötigten Transferrate bzw. Datendurchsatz.

Durch die möglichen Einschränkungen aufgrund des Datendurchsatzes (aufgrund der anzusetzenden Kosten) und der physikalisch bedingten Laufzeiten der einzelnen Datenelemente ist eine enge Kopplung der Server teilweise nur bedingt möglich, da durch eine enge Kopplung unter Umständen Wartezyklen in die Befehlsabarbeitung eingefügt werden müssen, die die Performance reduzieren.

5.1.3 National

Die nationale Verteilung der physischen Server bezeichnet die Verteilung der Server innerhalb der territorialen Ausdehnung eines Staates. Dabei können mehrere Rechenzentren über das gesamte Hoheitsgebiet eines Staates verteilt werden, so dass sich räumliche Abstände zwischen den Rechenzentren von üblicherweise >100km realisieren lassen. Die Rechenzentren sind dabei in der Regel über ein so genanntes Wide Area Network (WAN) miteinander vernetzt, welche sich im Unterschied zu einem LAN oder MAN typischerweise über einen sehr großen geografischen Bereich erstrecken. Ebenso wie die MANs werden auch die WANs häufig von international tätigen Telekommunikationsfirmen aufgebaut und basieren auf einer länderübergreifenden und teilweise auch weltweit vermaschten Struktur. Ähnlich wie die Internet Exchange Points (IX or IXP) für das Internet existieren auch für die weltweite WAN Infrastruktur definierte Übergabepunkte, die für den Austausch der Datenpakete zwischen den verschiedenen Anbietern verwendet werden.

Im Gegensatz zu der zuvor dargestellten Varianten bietet der Ansatz mit einer nationalen Verteilung die größtmögliche Sicherheit gegenüber lokalen Ereignissen wie z. B. Stromausfällen, Erdbeben, Hochwasser oder Terroranschlägen unter Berücksichtigung von Hoheitlichen Interessen.

So könnte z. B. aufgrund abweichender rechtlicher und politischer Rahmenbedingen eines Drittstaates die Vertraulichkeit und Integrität nicht in dem Maße gewährleistet werden, wie dies im Heimatland selbstverständlich ist. Auch ist die für den Betrieb eines physischen Servers benötigte Infrastruktur in einem gegebenen Staat meist auf einem relativ einheitlichen Niveau, so dass sich auch hier recht brauchbare Risikobetrachtungen, z. B. für die Häufigkeit eines Stromausfalls, vornehmen lassen.

Ebenso wie bei regionalen Verteilung sind aufgrund des Datendurchsatzes und der physikalisch bedingten Laufzeiten der einzelnen Datenelemente in der Regel Einschränkungen bei der Kopplung der Server zu berücksichtigen. Die wesentlichen Kriterien, die für eine regionale Verteilung sprechen, sind sicherlich die größtmögliche Fehlertoleranz gegenüber großräumigen Schadensereignissen jeglicher Art bei einer gleichzeitig überschaubaren Entwicklung hinsichtlich der rechtlichen Rahmenbedingungen.

5.1.4 International

Eine Internationale Verteilung der physischen Server-Ressourcen ermöglicht aus der Sicht einer hochverfügbaren Lösung sicherlich die größtmögliche Fehlertoleranz gegenüber Umwelteinflüssen jeglicher Art, politischen Entwicklungen, Unruhen oder Ressourcenengpässen in einzelnen Staaten.

Für die Verbindung der weltweit verteilten Rechenzentren bietet das Internet bei kleinen oder mittleren Unternehmen oder Organisationen aufgrund der geringen Kosten und der gleichzeitig relativ hohen Verfügbarkeit in vielen Fällen eine ausreichende Plattform. Größere und international aufgestellte Unternehmen und Organisationen werden eher auf die von verschiedenen internationalen Providern angebotenen Private-IP-Dienste zurückgreifen, die zwar auf der gleichen physischen Infrastruktur wie das Internet aufsetzten, aber logisch von diesen getrennt sind.

Durch die weltweite Verteilung der physischen Server-Ressourcen und einer optimalen Vermaschung durch das Internet oder eine Private-IP-Lösung kann in vielen Fällen eine ausreichende Verfügbarkeit erreicht werden. Gleichzeitig aber lassen sich andere wichtige Grundwerte wie z. B. Vertraulichkeit, Integrität, Authentizität oder Nachvollziehbarkeit, nur mit einem deutlich höheren Aufwand realisieren. Bedingt durch die großen Entfernungen, die meist durch Seekabel oder auch Satellitenstrecken überbrückt werden, treten hier teilweise Verzögerungszeiten im Bereich von einigen Sekunden auf, die nur eine relativ lose Kopplung der verteilten Server zulassen.

Auch bildet ein weltumspannendes Verbindungsnetz mit mehreren Millionen angeschlossenen Rechnerknoten ein relativ komplexes System, welches unter dem Einfluss von externen Störeinflüssen teilweise zu einer nichtvorhersehbaren Reaktion neigen kann. So kann es durch den Ausfall von Seekabeln zu massiven Performanceproblemen kommen, wie es in der Vergangenheit schon öfter der Fall war. Auch sind gezielte Angriffe auf zentrale Internet Dienste oder Internet Exchange Points denkbar, wodurch die Internet-Infrastruktur in weiten Regionen zum Erliegen kommen könnte.

5.2 Ausstattung der Standorte

Für den Betrieb einer physischen Server-Ressource an einem gegebenen Standort muss in der Regel eine Umgebung bereitgestellt werden, die den Anforderungen an einen sicheren Server-Betrieb genügt. Dazu zählen typischerweise der obligatorische Zutrittsschutz und ausreichende Brandschutzmaßnahmen für den physikalischen Schutz des Servers. Weiterhin sollten auch eine ausreichend dimensionierte und unterbrechungsfreie Stromversorgung, ein ausreichend dimensionierter Netzwerkzugang und entsprechende Vorkehrungen für eine Kühlung bzw. Klimatisierung des Server-Raums vorhanden sein.

Bei der Ausgestaltung dieser Server-Räume kann typischerweise zwischen mehreren Varianten unterschieden werden. Die Varianten unterscheiden sich dabei hinsichtlich der Kosten für die initiale Inbetriebnahme des Standortes und den nachfolgenden laufenden Kosten für den Standort. Zum anderen unterscheiden sich die Varianten in dem Zeitraum, der zwischen dem Ausfall eines Standortes und der Übernahme der Server-Dienste durch einen redundanten Standort vergeht.

5.2.1 Cold Site

Mit dem Begriff “Cold Site” wird ein Standort bezeichnet, der bis auf die Büroräume für die Mitarbeiter, den Stellflächen und der benötigten Infrastruktur für den Betrieb von physischen Servern keine weiteren Komponenten oder Einrichtungsgegenstände enthält. Der Gedanke hierbei ist, mit minimalen laufenden Kosten einen vorbereiteten Standort verfügbar zu haben, an dem der

Geschäftsbetrieb bei einem Ausfall des primären Standortes innerhalb einer absehbaren Zeit wieder aufgenommen werden kann. Den geringen Kosten einer „Cold Site“ steht allerdings ein Zeitraum von mehreren Tagen bis zu mehreren Wochen gegenüber, in dem weder die Server-Dienste zur Verfügung stehen, noch die IT-unterstützten Geschäftsprozesse abgewickelt werden können.

5.2.2 Warm Site

Der Begriff “Warm Site” bezeichnet einen Standort, der neben den Büroräumen für die Mitarbeiter, den Stellflächen und die benötigten Infrastruktur für den Betrieb von physischen Servern auch einige oder alle der unternehmenskritischen Server-Ressourcen enthält. Im Gegensatz zu der „Cold Site“ können die wesentlichen und kritischen Geschäftsprozesse innerhalb relativ kurzer Zeit wieder zur Verfügung gestellt werden. Dazu muss die letzte Generation der Offline Sicherung aus dem ausgefallenem Standort auf die von der „Warm Site“ bereitgestellten Server zurückgesichert werden.

Die „Warm Site“ stellt somit einen Mittelweg zwischen der „Cold Site“ und der nachfolgend beschriebenen „Hot Site“ dar. Zum einen werden die initialen und die laufenden Kosten möglichst niedrig gehalten, da nur die wirklich kritischen Systeme vorgehalten werden müssen. Zum anderen können die unternehmenskritischen Geschäftsprozesse und Server-Dienste innerhalb eines relativ kurzen Zeitraums von wenigen Stunden bis zu wenigen Tagen wieder zur Verfügung gestellt werden.

5.2.3 Hot Site

Bei einer „Hot Site“ wird ein Server-Standort komplett dupliziert und entspricht in seiner Ausstattung praktisch vollständig dem des primären Standortes. Der konzeptionelle Grundgedanke bei der Einrichtung einer Hot Site ist das Vorhandensein eines alternativen Standortes mit der entsprechenden Infrastruktur, die im Notfall oder Katastrophenfall den Geschäftsbetrieb und die Server-Dienste innerhalb kürzester Zeit übernehmen kann. Durch die vollständig eingerichtete und betriebsbereite „Hot Site“ kann sichergestellt werden, dass die Geschäftsunterbrechung so kurz wie möglich ausfällt. Im optimalen Fall ist für die Nutzer der Server-Dienste keine Unterbrechung wahrzunehmen.

Aus technischer Sicht werden auf der Basis einer „Hot Site“ auch häufig Server-Cluster oder Hot Standby Mechanismen aufgesetzt, so dass sowohl bei Totalausfall als auch bei einen teilweisen Ausfall der Server-Ressourcen die Dienste von dem gespiegelten Server übernommen werden können.

Der hohen Fehlertoleranz und Ausfallsicherheit einer “Hot Site” stehen auf der anderen Seite aber auch relativ hohe Kosten gegenüber, da alle Hardware-Ressourcen und Server-Software-Lizenzen doppelt vorhanden sein müssen. Ebenso sind alle an dem primären Standort vorgenommenen Konfigurationsänderungen exakt auf dem alternativen Standort zu übertragen, was zu einem deutlich erhöhten Administrationsaufwand führt. Allerdings übernehmen Software-Clusterlösungen inzwischen davon einen großen Teil.

Nur hohe Sicherheitsanforderungen an die Verfügbarkeit rechtfertigen die hohen Investitions- und laufenden Kosten dieser Betriebsweise.

5.2.4 Kollokation

Unter einer Kollokation (*von lat. Con = zusammen und locus = Ort*) wird das Mitbenutzen von Ressourcen in einem Rechenzentrum verstanden. Dabei werden einzelne Einschubplätze in einen bestehenden Server-Rack bis hin zu abgeschlossenen Boxen oder Räume angeboten, in dem die Mieter ihre eigenen Server unterzubringen können. Durch die gemeinsame Nutzung der Infrastruktur des Rechenzentrums, wie z. B. einer unterbrechungsfreien Stromversorgung,

ausreichender Klimatisierung, Brandschutz, Zutrittsschutz und einer breitbandigen und redundanten Anbindungen an das Internet, ist die Kollokation meistens eine günstigste Alternative zu einem Server-Raum im eigenem Haus und einer meist zusätzlich notwendigen Standleitung zu einem Internet Provider.

Während der Fokus bei den zuvor genannten Varianten (Cold Site, Warm Site, Hot Site) mehr oder weniger auf die vollständige Redundanz eines IT-unterstützten Geschäftsbetriebes abzielt, inklusive der für den IT-Betrieb und die Geschäftsprozesse notwenigen Mitarbeiter und sonstiger für die Aufrechterhaltung der Geschäftstätigkeit notwendigen Ressourcen, beschränkt sich der Kollokation-Ansatz ausschließlich auf die Auslagerung von z. B. physischen Servern.

Abhängig von dem zugrundeliegenden Geschäftsmodell und dem Automatisierungsgrad der Prozesse kann sich eine Kollokation als ausreichend erweisen, um den Geschäftsbetrieb bei dem teilweise oder totalen Ausfall eines Standortes aufrecht zu erhalten. Im Sinne eines Outsourcing-Ansatzes kann bei einigen Anbietern von Kollokationslösungen neben der eigentlichen Stellfläche auch gleich noch das in den Rechenzentren sowieso vorhandene fachkundige Personal für eine rund um die Uhr Betreuung der untergestellten Server genutzt werden. Voraussetzung hierfür sind aber meist eine weitgehende Standarisierung der Server, der auf den Server betriebenen Anwendungen sowie der Betriebsprozesse und des IT-Service-Managements im Sinne des ITIL (IT Infrastructure Library) Ansatzes.

6 Heterogene und zeitliche Redundanz

In dem vorhergehenden Kapitel wurde mit Hilfe der geografischen Redundanz Serverarchitekturen diskutiert, die robust gegen großflächige Schadensereignisse sind. Dieser Ansatz zur Verfügbarkeitssteigerung reicht nicht, wenn sich die Gefahr sofort oder sehr schnell global ausbreitet, wie Angriffe von Schadprogrammen, oder die Gefahr in der geografisch verteilten Ressource immanent enthalten ist, wie fehlerhafte Prozessor-Chips, Software etc.

Als Lösung für diese Problematik gibt es die Diversität als besondere Ausprägung der Redundanz. In unserem Zusammenhang von Server-Hardware wird häufig auch von heterogener Hardwarestruktur gesprochen, wobei dann der negative Mehraufwand in der Administration adressiert wird. Für extrem kritische Services, die z. B. wie in der Raumfahrt und Reaktortechnik unmittelbar die Unversehrtheit von Menschen sicherstellen, muss dieser Aufwand geleistet werden.

Neben anderen Aufgaben abstrahieren Betriebssysteme die Hardware-Schicht für die darauf ablaufenden Anwendungssysteme, wodurch der Aufwand verringert wird. Zudem enthalten die führenden Betriebssysteme inzwischen Clusterlösungen und nicht proprietärer Anbieter bieten sogar Clustersoftware mit unterschiedlichen Betriebssystemen.

Natürlich birgt die Heterogenität das großes Risiko, ob die Lösung mit der Fortentwicklung der Technik überhaupt, zeitnah und wirtschaftlich darstellbar fortgeschrieben werden kann.

Der Vollständigkeit halber sollte noch erwähnt werden, dass die geografische Redundanz mit der heterogenen verbunden werden kann. Außer im erhöhten Administrationsaufwand und dem angesprochenen Innovationsrisiko ergeben sich durch die Kombination keine weiteren Probleme, als im vorhergehenden Kapitel beschrieben.

Ohne geografische Verteilung wird die heterogene Redundanz in Verbindung mit zeitlicher Redundanz (Parallelbetrieb) in der Flug- und Raumfahrt schon sehr lange eingesetzt, mit automatisierten Votern oder zur manuellen Entscheidungsunterstützung durch den Piloten.

7 Server und Verfügbarkeit

Wie bereits in dem Kapitel 2 „Definition des Begriffs "Server"“ angeführt, wird unter dem Begriff Server meist eine Kombination aus der physischen Server-Hardware in Verbindung mit dem darauf aufsetzenden Betriebssystem sowie einer oder mehrerer Anwendungen verstanden, welche letztendlich über ein Netzwerk wie auch immer geartete Dienste zur Verfügung stellt.

Diese von den Servern über ein Netzwerk zur Verfügung gestellten Dienste leisten meist einen wesentlichen Beitrag für die automatisierte Abarbeitung von geschäftskritischen Abläufen. Sind die Server-Dienste nicht verfügbar, bedeutet dies meist einen nicht vertretbaren Mehraufwand oder die auf den Server-Diensten aufsetzenden Geschäftsprozesse kommen vollständig zum erliegen.

7.1 Leistungsübergabepunkte und Abhängigkeiten

Wie in den vorhergehenden bereits dargestellt, reicht es nicht aus, die Server-Hardware zu beschaffen, um die benötigten Server-Dienste zu erbringen. Vielmehr müssen rund um die Server-Hardware noch eine Vielzahl von Bedingungen erfüllt sein, damit der Server seine Dienste zur Verfügung stellen kann. In dem nachfolgenden Unterkapitel soll nun kurz in einer Zusammenfassung dargestellt werden, welche Abhängigkeiten existieren und welche Einflüsse die Verfügbarkeit eines Servers negativ beeinflussen können.

7.1.1 Bereitgestellte Server-Dienste (Services)

Die von der Server-Hardware bereitgestellte Plattform beschränkt sich im Wesentlichen auf die Bereitstellung einer Ablaufumgebung für die in der jeweiligen Maschinensprache vorliegenden Programme. Weiterhin können auch noch die internen oder externen Speichereinheiten für die dauerhafte Ablage der Programme und Daten sowie die notwendigen Schnittstellen (Netzwerk, Konsole, usw.) für eine Kommunikation mit der Umgebung zu den von der Server-Hardware bereitgestellten Diensten gezählt werden.

Alle weiteren über das Netzwerk, die Konsole oder sonstigen Eingabe- und Ausgabeeinheiten zur Verfügung gestellten Dienste sind abhängig von der Funktion der Server-Hardware sowie von den auf dem physischen Server ablaufenden Programmen.

7.1.2 Abhängigkeiten von internen Komponenten

Die Verfügbarkeit eines physischen Servers ist mindestens abhängig von denen in Kapitel 2.2.1 „Physische Server“ dargestellten Kernkomponenten eines physischen Servers. Diese Kernkomponenten werden meist mit so genannter Glue Logicengl: Klebstofflogik) verbunden, die selbst außer der Verbindung der Komponenten keinen weiteren Nutzen hat.

Ein typisches Beispiel ist die Zusammenschaltung eines Mikroprozessors mit DRAM-Bausteinen, wobei die DRAM Bausteine eine Busstruktur als übliche Mikroprozessoren haben und zudem Inhalte der Speicherzellen in regelmäßigen Abständen aufgefrischt werden müssen. Da sich hier DRAM-Bausteine und Mikroprozessoren nicht direkt verbinden lassen, kommen hier zusätzliche Komponenten zum Einsatz, die als so genannte Glue Logic bezeichnet wird.

Für den Ablauf der Programme müssen in der Regel jede einzelne dieser Komponenten einwandfrei funktionieren, da schon der Ausfall einer einzigen Komponente den gesamten Programmablauf zum Erliegen bringen kann. Aufgrund der hohen Taktfrequenzen und den dadurch notwendigen kurzen Verbindungswegen zwischen den Komponenten wird bei praktisch allen physischen Servern auf eine redundante Auslegung der Kernkomponenten verzichtet. Bei dem Ausfall auch nur eines dieser

internen Komponenten ist in der Regel ein manueller Austausch der gesamten Baugruppe notwendig.

Um diese konkrete Abhängigkeit von der Server-Hardware zu reduzieren, werden in der Praxis mehrere Server zu Gruppen (Cluster) zusammengefasst, so dass die Server-Dienste auch bei dem Ausfall eines einzelnen Servers in der jeweiligen Gruppe weiterhin zur Verfügung stehen. Die Gruppierung kann dabei auf der Ebene der Server-Hardware (mittels Server-Virtualisierung), auf der Ebene des Server-Betriebssystems (Server-Cluster) oder auf der Ebene der Anwendungen (z. B. Datenbank-Cluster) vorgenommen werden.

7.1.3 Umwelteinflüsse und Abhängigkeiten von externen Ressourcen

Wie in dem 5.1 „Geografische Verteilung“ bereits angeführt, ist ein physischer Server abhängig von bestimmten externen Gegebenheiten. So benötigt ein physischer Server eine ausreichende Stromversorgung, wobei sowohl die Spannungswerte als auch Grenzwerte für die leitungsgebundenen Störungen innerhalb gewisser Grenzwerte liegen müssen. Bei einer Überschreitung der Grenzwerte kann es entweder zu temporären, als auch zu permanenten Ausfällen der Server-Dienste kommen.

Weiterhin ist ein physischer Server abhängig von verschiedenen Umgebungsbedingungen. So muss durch eine entsprechende Zirkulation die in einen physischen Server entstehende Wärme abgeführt werden. Zum anderen muss diese Luft innerhalb eines spezifizierten Temperaturbereichs liegen und darf gewisse Grenzwerte hinsichtlich Luftfeuchtigkeit und Staubpartikel nicht überschreiten. Auch können elektromagnetische Wellen oder radioaktive Strahlung die Funktion eines physischen Servers beeinträchtigen, so dass auch hier die jeweiligen Grenzwerte einzuhalten sind.

Abhängig von der Verwendung eines physischen Servers muss dieser auch noch mit einem wie auch immer gearteten Verbindungsnetz verbunden sein über das die Kommunikation der Server-Dienste mit den „Endverbrauchern“ stattfindet. Gleiches gilt auch für einen eventuell vorhandenen externen Massenspeicher, da ein Server bei vielen Funktionen von der temporären oder permanenten Speicherung von Programmen und Daten abhängig ist.

7.2 Fehlertoleranz und Automatismen

In der Datenverarbeitung beschreibt der Begriff „Fehlertoleranz“ im Allgemeinen die Eigenschaft eines technischen Systems, seine Funktionsweise auch bei unvorhergesehen Ereignisse oder Fehlern in der Hard- oder Software aufrechtzuerhalten. Ein technisches System ist dabei meist ein umfangreiches Hardware-Software-Gebilde mit einer beliebiger Komplexität, welches sehr viele unterschiedliche Fehlerarten ausweisen kann. Dabei ist der Begriff eines Fehlers im Allgemeinen ein sehr vager Begriff, der in der Regel eine systemspezifische Identifikation und Beschreibung erfordert.

In Bezug auf einen physischen Server könnte sich ein Fehlerzustand z. B. dadurch äußern, dass das auf dem Server ablaufenden Programm nicht oder nicht wie erwartet abläuft. Weitere Fehler könnten auch darin bestehen, dass das Programm zwar korrekt abläuft, aber dass keine Kommunikation z. B. über die Netzwerk- oder Konsolen-Schnittstelle möglich ist.

Während die Fehler in der Hardware oft sporadisch, temporär und unbestimmt auftreten, z. B. durch Alterungs- und Verschleißerscheinungen, sind Fehler in der Software meist permanent vorhanden und müssen durch eine Änderung des Programmcodes behoben werden. Bei der Software treten also keine Effekte von Alterung und Verschleiß auf. Somit ist eine Dopplung von Hardware Komponenten durchaus sinnvoll, um diese bei einer unbestimmten Störung oder Ausfall mit einer

gleichartigen Hardware Komponente zu ersetzen. Bei der Software wäre jedoch eine doppelte Auslegung einer identischen Instanz eines Programms eher sinnlos.

Die meisten Fehler, die als Folge von Hardwarefehlern auftreten, lassen sich durch Automatismen beheben oder kompensieren. Voraussetzung hierfür ist allerdings eine brauchbare Methode für die Erkennung der Fehler sowie eine wie auch immer geartete redundante Hardwarekomponente, die die Funktion und Dienste der ausgefallenen Komponente übernimmt. Auch wenn die Fehlererkennung in Hardware prinzipiell möglich ist, so wird aufgrund der Komplexität der Fehler in den meisten Fällen eine Software basierende Lösung für die Fehlererkennung, Fehlerbehandlung bzw. Alarmierung eingesetzt.

Eine klassische, in Kombination von Software und Hardware realisierte Lösung, die den korrekten Programmablauf auf einen Server überwachen kann, ist die so genannte „Watchdog“ (*engl. Wachhund*) Lösung. Dabei muss die auf dem Server laufende Software in regelmäßigen Abständen ein bestimmtes Prüfwort auf eine Eingabe-Ausgabeschnittstelle schreiben. Erfolgt dieser Schreibvorgang nicht innerhalb eines vorgesehenen Zeitfensters, so wird durch in Hardware realisierte Logik entweder eine Unterbrechung (*engl. Interrupt*) ausgelöst, welche eine Fehlerbehandlungsroutine startet oder es wird durch die Hardware Logik gleich ein Neustart (*engl. Reset*) veranlasst.

Als ein weiteres Beispiel für eine ausschließlich in Hardware realisierte Fehlertoleranz soll hier noch ein Verfahren angeführt werden, welches Einzelbitfehler z. B. in dem Hauptspeicher erkennen und automatisch korrigieren kann. Dieses so genannte Error-correcting code (ECC) Verfahren benötigt bei einem 32 Bit breiten Bussystem 7 zusätzliche Check-Bits. Bei jedem Schreibvorgang wird für die 32 Bit eine Prüfsumme errechnet und in den zusätzlich vorhandenen 7 Check Bits abgelegt.

Bei einem späteren Lesevorgang wird wiederum eine Prüfsumme gebildet und mit der zusätzlich abgelegten Prüfsumme verglichen. Sofern die Prüfsumme nicht mit dem Datenwort übereinstimmt, können mit Hilfe einer in Hardware realisierten Ablaufsteuerung das fehlerhafte Bit lokalisiert und korrigiert werden. Die Fehlererkennung und Korrektur basiert bei Speicherbausteinen meist auf dem so genannten Hamming-Code, welcher von dem US-amerikanischer Mathematiker Richard Hamming in den 1940er Jahren entwickelt wurde.

Neben der Anwendung in Speicherbausteinen wird der Hamming-Code auch noch häufig in der digitalen Signalverarbeitung und der Nachrichtentechnik zur gesicherten Datenübertragung oder Datenspeicherung verwendet. Durch die zusätzlichen Kosten für die 7 zusätzlichen Check-Bits, den potentiellen Performanceeinbußen und nicht zuletzt der geringen Eintrittswahrscheinlichkeit eines Bit-Fehlers kommt dieses Verfahren in der Praxis nur in wenigen, speziellen Servern zum Einsatz.

7.2.1 Überwachung

Die ständige Überwachung des Netzwerks, der Anwendungen, der Daten und der Hardware ist für eine hohe Verfügbarkeit unerlässlich. Mit diversen Softwareüberwachungstools und –techniken kann der Zustand eines Server-Systems jederzeit ermittelt werden. Durch Trendanalysen können sich anbahnende Probleme vielfach schon identifiziert werden, bevor ein Fehler tatsächlich auftritt.

Bei einer Überwachung der physischen Server sollten insbesondere auch die Umwelteinflüsse und Abhängigkeiten externen Ressourcen berücksichtigt werden, da hier durch eine rechtzeitige Fehlererkennung etwaige Ausfälle verhindert oder zumindest die Ausfallzeiten minimiert werden können. So kann z. B. durch die laufende Überwachung der Netzteile in einen Server festgestellt werden, ob beide der redundant vorhandenen Netzteile auch tatsächlich funktionsbereit sind.

Auch kann durch eine fortlaufende Überwachung festgestellt werden, ob sich die Temperatur innerhalb des Server-Gehäuses in den vorgeschriebenen Grenzwerten bewegt oder ob z. B. aufgrund eines Lüfterausfalls die Temperatur ständig zunimmt. Ebenso kann durch eine unterbrechungsfreie Stromversorgung signalisiert werden, dass aufgrund einer ausgefallenen Netzversorgung nur noch ein bestimmter Zeitraum für das ordnungsgemäße Herunterfahren des Servers zur Verfügung steht.

Für die Überwachung der über ein Netzwerk zur Verfügung gestellten Server-Dienste sind meist externe Agenten notwendig, die eine Überwachung der Server-Dienste an den durch die Nutzer repräsentierten Endpunkten erlauben. Dabei werden durch diese externen Agenten in regelmäßigen Abständen bestimmte Anfragen an den zu Überwachenden Server geschickt. Abhängig von der Reaktion des Servers und bei einer Abweichung von dem erwarteten Verhalten können gegebenenfalls die vordefinierten Fehlerbehandlungsroutinen eingeleitet werden. Für eine laufende Überwachung der Server-Dienste bieten sich hier geografisch verteilte Messsonden an, die Anfragen eines Clients simulieren und auftretende Fehler an einen zentralen Leitstand melden können. Für weitere Hinweise hinsichtlich der Überwachung von Server-Diensten wird hier auf den Beitrag „Überwachung“ im HV Kompendium verwiesen.

Für Server, die in dem Bereich der industriellen Steuerungstechnik verwendet werden, kommt häufig der Triple Modular Redundancy (TMR) Ansatz zur Anwendung. Dabei werden drei baugleiche Server verwendet, die durch einen so genannten Voter (*engl. Wähler, Wahlberechtigter*) überwacht werden.

Der Voter überprüft die Ergebnisse aller drei Server-Komponenten und wählt auf der Basis eines Mehrheitsentscheids das richtige Ergebnis aus. Das TMR macht somit von der strukturellen, funktionalen und statischen Redundanz Gebrauch. Beim dem Ausfall von mehr als einer Komponente versagt dieser Ansatz jedoch, weil zwei zufällig falsche Ergebnisse das richtige Ergebnis beim Mehrheitsentscheid überstimmen können. Des Weiteren wird durch Hinzufügen des Voters eine neue kritische Komponente in das System aufgenommen.

7.2.2 Fehlerbehandlung

Sofern entweder durch die Überwachungsfunktionen oder durch Meldungen von Nutzern der Server-Dienste ein Fehlerzustand erkannt wird, muss zunächst eine genaue Fehlerdiagnose und anschließend die passende Fehlerbehandlung eingeleitet werden. Sowohl die Fehlerdiagnose als auch die Fehlerbehandlung können dabei entweder vollständig manuell unter der Zuhilfenahme eines Betriebshandbuchs oder teilweise automatisiert, z. B. auf der Basis von Fehlerbäumen, erfolgen.

Dabei sollte im Rahmen der Diagnose die wahrgenommenen Fehlerauswirkungen soweit als möglich korreliert werden, um den ursächlichen Auslöser für das Fehlerbild zu bestimmen. Als mögliche Vorgehensweise könnte hierbei die Root Cause Analysis (Ursachenanalyse) eingesetzt werden, welche eine strukturierte und schrittweise Methode für das Finden der tatsächlichen Ursachen eines Problems ist.

Mit zunehmender Komplexität eines Systems kann eine vollständig automatisierte Fehlerbehandlung unter Umständen einen deutlich größeren Schaden anrichten als dies ohne das Eingreifen der automatisierten Fehlerbehandlung der Fall wäre. So könnten bei dem Ausbleiben des Heartbeat Signals, z. B. aufgrund einer temporären Störung des Netzwerks, die jeweiligen Cluster Knoten annehmen, dass der jeweils andere Cluster Knoten nicht mehr verfügbar ist. Als Folge dieser Annahme würde dann jeder Cluster Knoten versuchen, die Aufgaben und Sitzungen des ausgefallenen Knotens zu übernehmen, was im Extremfall zu einem länger andauernden Totalausfall oder nicht mehr integren Datenständen führen kann.

In der Regel erweist es sich daher als vorteilhaft, wenn komplexe Zusammenhänge durch entsprechende Programme aufbereitet und verdichtet werden und die endgültige Entscheidung für die weitere Vorgehensweise zur Fehlerbehandlung durch geschultes Personal erfolgt. Als nachteilig können sich hierbei allerdings wiederum die längere Reaktionszeit und die notwendige Verfügbarkeit von ausreichend geschultem Personal zum Fehlerzeitpunkt und am richtigen Ort erweisen

Für die Aufbereitung der Informationen für eine Fehlerbehandlung wurden im Laufe der Zeit eine Reihe von IT gestützten Werkzeugen entwickelt, die das Server-Betriebs-Personal bei der Entscheidungsfindung unterstützen und somit die Ausfallzeit im Fehlerfall reduzieren können.

Zu einem können Entscheidungsbäume eingesetzt werden, wodurch das Server-Betriebs-Personal in die Lage versetzt wird, schneller und mit weniger Fehlern eine Entscheidung zu treffen. Zum anderen können so genannte „Case based reasoning“ (*engl: fallbasierte Schließen*) Systeme zu Einsatz kommen, die die Administratoren bei der Diagnose und der Entscheidungsfindung unterstützen. Die CBR-Systeme basieren dabei auf einer so genannten Fallbasis (Falldatenbank, case memory), in der bereits gelöste Probleme als Fall gespeichert sind. Ein solcher Fall besteht mindestens aus einer Problembeschreibung und einer zugehörigen Problemlösung.

Das Ziel ist, zur Lösung eines gegebenen Problems die Lösung eines ähnlichen und früher bereits gelösten Problems heranzuziehen. Damit ahmt man eine menschliche Verhaltensweise nach: Vor ein neues Problem gestellt, erinnert sich der Mensch oft an eine vergleichbare Situation, die er in der Vergangenheit erlebt hat, und versucht, die aktuelle Aufgabe ähnlich zu meistern.

Bei allen Methoden der Entscheidungsfindung ist eine aktuelle und detaillierte Dokumentation über das hochverfügbare System unbedingte Voraussetzung. Nur wenn alle Details über das System bekannt sind, können fundiert Entscheidungen über die Fehlerbehebung getroffen werden.

7.2.3 Betriebskonzept

Das Betriebskonzept oder auch Betriebshandbuch unterstützt das für den IT-Betrieb verantwortliche Personal in allen Phasen eines IT-Systems. Bezogen auf einen physischen Server sind in einem Betriebskonzept meist konkrete Handlungsanweisungen für die Bedienung eines Systems hinterlegt.

So finden sich in einem Betriebskonzept typischerweise Vorgaben für die Inbetriebnahme, die Außerbetriebnahme, die laufende Überwachung, die Wartung und die Reaktion auf Fehlersituationen. Ergänzt werden die für eine spezifische Umgebung erstellten Betriebskonzept meist noch durch die Installationsanleitungen, die von dem Herstellern der Server-Hardware herausgegeben werden.

8 Zusammenfassung

Wie in den vorhergehenden Kapiteln dargelegt, muss die Server Hardware innerhalb der von dem Hersteller der Server Hardware spezifizierten Umgebungsbedingungen betrieben werden. Eine Abweichung von diesen Umgebungsbedingungen, führt in der Regel, zu temporären Störungen bis hin zu permanenten Ausfällen, der Server Hardware.

Da diese umweltbedingten Störungen, meist nur eine beschränkte regionale Ausdehnung haben, kann durch mehre Server Systeme und deren geografische Verteilung, einem potentiellen Ausfall des Gesamtsystems, wirksam entgegengewirkt werden. Abhängig von der maximal tolerierbaren Ausfallzeit, können die Standorte entweder alle mehr oder weniger gleich ausgestattet sein, so dass diese praktisch ohne Unterbrechung die Server-Dienste des ausgefallenen Standortes übernehmen können oder nur sehr rudimentär, so dass eine Übernahme der ausgefallenen Server Dienste innerhalb eines vorgegeben Zeitraumes gewährleistet werden kann.

Für die ausfallsichere Bereitstellung der Server Dienste bei mehreren, verteilten Standorten muss eine möglichst ebenso ausfallsichere Netzwerkinfrastruktur zugrunde liegen, so dass bei dem Ausfall einzelner Verbindungen immer noch ein Weg zwischen dem Server Dienst und dem Nutzer dieser Dienste hergestellt werden kann. Auch muss sichergestellt werden, dass die zum Zeitpunk des Ausfalls aktuellen Programm- und Datenbestände an allen Standorten gleichzeitig verfügbar sind.

Weitere Probleme bei der geografischen Verteilung von physischen Server Systemen können sich aus dem benötigten Bandbreiten und durch die Signallaufzeiten auf den Übertragungsmedien und dem Verzögerungen durch die aktiven Netzwerkkomponenten ergeben. Ausgehend von einem zentralen Serversystem muss eine ausreichend leistungsfähige Verbindung zwischen jedem potentiellen Nutzer und jedem möglichen Standort existieren, so dass die verteilten Standorte ohne Einbußen in Bezug auf die Reaktionszeiten erreicht werden können. Auch machen Signalverzögerungen und Laufzeiten im Sekundenbereich, wie dies bei global verteilten Standorten häufig der Fall ist, den Einsatz von bestimmten Verfahren, Protokollen und Technologien praktisch unmöglich.

Diese Signalverzögerungen sind auch ein wesentlicher Grund dafür, dass im Sinne einer Redundanz einzelne Komponenten der Server Hardware nicht oder nur sehr eingeschränkt über größere Entfernungen verteilt werden können. So müssen sich auf Grund der hohen Taktfrequenzen eines modernen Server Systems alle Kernkomponenten einer Von-Neuman-Architektur wie CPU und Hauptspeicher meist in unmittelbarer räumlicher Nähe befinden, damit sich die Laufzeiten auf den Leiterbahnen zwischen den einzelnen Bauteilen nicht negativ auf den Durchsatz auswirken.

Aufgrund dieser Restriktionen ist eine räumliche Verteilung von z. B. CPU und Hauptspeicher in der Regel nicht möglich, so dass geografische Redundanz nur mit vollständigen Server Systemen realisiert werden kann. Eine Ausnahme von dieser Regel bilden einige Mainframe Systeme, die durch spezielle Protokolle, Koppelelemente und Technologien eine sinnvolle Verteilung der Kernkomponenten erlauben.

Auch wird in der Regel auf eine redundante Auslegung der einzelnen Hardware-Bausteine eines physischen Servers verzichtet, da diese Maßnahmen durch die räumliche Nähe nur einen minimalen Schutz, insbesondere bei Umwelteinwirkungen, bieten würden und zum anderen durch die Fehlererkennungs- und Korrekturmaßnahmen die Rechenleistung negativ beeinflussen.

Daher dienen alle möglichen Varianten von Mehrprozessorsystemen ausschließlich der Erhöhung des Durchsatzes bzw. Rechenleistung durch Parallelverarbeitung und weniger dem Schutz vor dem

Ausfall einer einzelnen CPU. Als Vorsorge für einen Serverausfall muss daher die gesamte Server-Hardware als eine unteilbare und zusammenhängende Einheit betrachtet werden.

Auf Basis der Server Hardware können wiederum verschiedenen Virtualisierungslösungen aufgesetzt werden, welche die jeweils zugrundeliegende Schicht abstrahiert. So kann z. B. auf der Basis eines Hypervisor die reale Hardware in der Form abstrahiert werden, dass den darauf aufsetzenden Betriebssystemen das Vorhandensein von mehreren physischen Servern vorgetäuscht wird.

Auch kann die Grundlage der Virtualisierungslösung aus theoretisch beliebig vielen einzelnen physischen Servern bestehen, wobei dadurch die auf der Virtualisierungslösung aufsetzenden Betriebssysteme und Anwendungen unabhängig von dem Ausfall eines einzelnen Servers werden.

Im Gegensatz zu dem klassischen Server oder Datenbank-Clustern, dem Grid-Computing oder den Peer-to-Peer Netzwerken stellt die Abstraktion der Hardware durch eine Virtualisierungslösung die unterste Ebene dar, auf der eine Verteilung von Server Ressourcen über ein Netzwerk möglich ist.

Bei allen vorgestellten Ansätzen für die Abstraktion und Verteilung der Server Hardware und Dienste spielt neben der Erreichbarkeit über das Verbindungsnetzwerk, der fortlaufenden Synchronisation der Programme, Daten und dynamischen Zustände auch die weitestgehende Autonomie der Server Systeme eine wichtige Rolle. Nur wenn alle an dem Gesamtsystem beteiligten Server durch eine übergeordnete und verteilte Software Komponente in der Lage sind, Fehlerzustände und Störungen zu erkennen und diese selbständig zu beheben, kann eine optimale Hochverfügbarkeit erreicht werden.

Anhang: Verzeichnisse

Abkürzungen und Akronyme

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