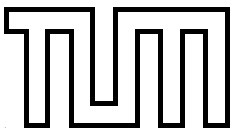


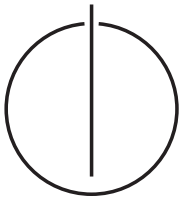
Technische Universität München

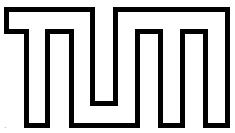
Fakultät für Informatik

Diplomarbeit in Informatik

Konzeption und Analyse von Techniken zur Virtualisierung von I/O-Kanälen

Martin Metzker





Technische Universität München

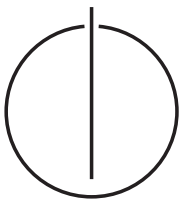
Fakultät für Informatik

Diplomarbeit in Informatik

Konzeption und Analyse von Techniken zur Virtualisierung von I/O-Kanälen

Design and analysis of techniques for virtualizing I/O-Channels

Bearbeiter:	Martin Metzker
Aufgabensteller:	Prof. Dr. Heinz-Gerd Hegering
Betreuer:	Dr. Vitalian Danciu Dr. Nils gentschen Felde Tobias Lindinger Randolph Esser (FSC) Dr. Detlef Rothe (FSC) Christoph Biardzki (LRZ)
Abgabedatum:	15. April 2009



Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. April 2009

. . . . . .
(Unterschrift des Kandidaten)

Abstract

Server in Rechenzentren sind heute meist keine Einheit mehr, sondern aufgeteilt in Speicher- und Rechenknoten. Das Speichernetz, das Speicher- und Rechenknoten verbindet, ist konfigurierbar und ermöglicht eine dynamische Ressourcenzuweisung. Dadurch können freie Ressourcen nach Bedarf zugewiesen werden, ohne physisch in die Infrastruktur einzugreifen. Server-Virtualisierung erhöht die Auslastung der Rechenknoten, indem ein Hypervisor auf einem Server viele virtuelle Server erstellt, wodurch viele Betriebssysteme auf einem einzigen realen Host betrieben werden können. Dadurch entsteht auf dem Host Aufwand zur Emulation von Hardware, der vor allem im I/O-Bereich die Leistung hemmt, da Emulation komplett in Software erfolgt und sich deshalb Rechenzeit mit allen anderen Prozessen teilen muss. Unter I/O-Virtualisierung versteht man Ansätze, mit dem Ziel den Emulationsaufwand zu minimieren und durch Hardware zu unterstützen. Die neuesten Ansätze entkoppeln I/O-Hardware von den Servern, so dass Ressourcen bereits auf I/O-Ebene dynamisch zugewiesen werden können. Ein anderer häufig verfolgter Ansatz ist I/O-Konsolidierung, bei der LAN und SAN über die selbe Infrastruktur betrieben werden, wodurch Hardware eingespart werden kann. Bei beiden Ansätzen entstehen neue Zwischenschritte und Abstraktionsebenen bei der Virtualisierung von I/O-Kanälen, die in ihrer Art sehr verschieden sein können. Die unterschiedlichen Ansätze und deren Auswirkungen auf Infrastrukturen werden in dieser Arbeit untersucht und verglichen, um eine Aussage zu treffen, in wie weit die Ansätze geeignet sind um die bestehenden Arbeitsvorgänge zu unterstützen.

Inhaltsverzeichnis

1 Einführung und Motivation

Server in Rechenzentren sind heute meist keine Einheit mehr, sondern mindestens aufgeteilt in Speicher und Rechenknoten. Die Aufteilung erfolgte vor allem aus der niedrigen Auslastung des oftmals teuren Hintergrundspeichers der einzelnen Server. Das Speichernetz, zwischen den Speicherknoten und den Rechenknoten, ist konfigurierbar und ermöglicht eine dynamische Ressourcenzuweisung, so dass freie Ressourcen bei Bedarf an Server zugewiesen werden können, ohne physisch in die Infrastruktur einzugreifen. Dadurch wird die Auslastung des Hintergrundspeichers erhöht, da nicht für jedes individuelle System Leistungsreserven vorgehalten werden müssen.

Server-Virtualisierung erhöht die Auslastung der Rechenknoten, indem ein Hypervisor auf einem Server viele virtuelle Server erstellt, wodurch es möglich wird viele Betriebssysteme auf einem einzigen realen Host zu betreiben. Dadurch entsteht auf dem Host Aufwand für die Emulation von Hardware, der vor allem im I/O-Bereich die Leistung hemmt. Dies kommt insbesondere davon, dass die Emulation komplett in Software erfolgt und sich deshalb die Rechenzeit mit allen anderen Prozessen teilen muss. Unter I/O-Virtualisierung versteht man Ansätze mit dem Ziel, den Emulationsaufwand zu minimieren und durch Hardware zu unterstützen. Zu diesem Zweck gibt es mehrere Ansätze wie per InfiniBand oder Ethernet angebundene I/O-Server, bei denen die I/O-Virtualisierung ein Zusammenspiel aus dem Dienst spezialisierter I/O-Server und lokaler Software (Treiber) ist. Die neuesten Entwicklungen in diesem Bereich sind die SR-IOV und MR-IOV Standards der PCI-SIG, die Virtualisierung bereits im PCIe-Bus ermöglichen. Diese Technologien bieten der Administration neue Handlungsmöglichkeiten und erhöhen dadurch die Flexibilität in Infrastrukturen.

Allen Ansätzen gemein ist die Entkopplung der I/O-Hardware von den Servern, so dass bereits auf I/O-Ebene Ressourcen dynamisch zugewiesen werden können. Dadurch entstehen neue Zwischenschritte und Abstraktionsebenen bei der Virtualisierung von I/O-Kanälen, die in ihrer Art sehr verschieden sein können. Diese neuen Elemente beeinflussen eine Menge von Anwendungsfällen wie zum Beispiel die Inbetriebnahme von Systemen oder das Vorgehen beim Ausfall eines Geräts. Die unterschiedlichen Ansätze und deren Eigenschaften werden in dieser Arbeit untersucht und verglichen, um eine Aussage zu treffen, in wie weit die Ansätze geeignet sind, um die bestehenden Arbeitsvorgänge zu unterstützen.

1.1 Vorgehen

In dieser Arbeit wird in Kapitel 2 ein Szenario analysiert. Das Szenario betrachtet ein Rechenzentrum, welches Server und Netze an Kunden vermietet und dabei I/O-Virtualisierung einsetzt. Gegenstand der Untersuchung ist der Dienstlebenszyklus der vermieteten Server und Netze. Aus diesem ergeben sich Anwendungsfälle (Kapitel 2.4) für den Betreiber und für den Kunden. Aus den Anwendungsfällen werden in Kapitel 2.5 Anforderungen an Virtualisierung abgeleitet. Eine vollständige Virtualisierung von I/O-Kanälen wird erreicht, wenn eine Technik alle abgeleiteten Anforderungen erfüllt.

Kapitel 3 stellt Methoden vor, die zur Virtualisierung von I/O-Kanälen eingesetzt werden können. Rechenzentren zeichnen sich durch verteilte, skalierbare Infrastrukturen aus. Aus diesem Grund müssen mehrere Methoden kombiniert werden, da mehrere Komponenten an der Virtualisierung von I/O-Kanälen beteiligt sind. Kapitel 3 schließt mit einer Produktübersicht über Implementierungen der einzelnen Methoden.

Kapitel 4 stellt Techniken zur Virtualisierung von I/O-Kanälen vor. Jede Technik ist eine Kombination aus den zuvor in Kapitel 3 vorgestellten Methoden. In Kapitel 4.2 werden die Eigenschaften der unterschiedlichen Techniken mit den Anforderungen aus Kapitel 2.5 abgeglichen. Am Ende der Arbeit geht Kapitel 5 auf Defizite und Verbesserungsmöglichkeiten ein. Kapitel 6 fasst die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick auf weiterführende Untersuchungen im Zusammenhang mit dieser Arbeit.

2 Szenarioanalyse

Um einen vollständigen Überblick über die Virtualisierung von I/O-Kanälen zu erhalten, wird ein Rechenzentrum betrachtet, das I/O-Kanäle vollständig virtualisiert. *I/O-Kanäle* sind Kommunikationswege und stellen ihre Funktionen als Dienst zur Verfügung. Ein Maximum an Virtualisierung ist genau dann erreicht, wenn jede Komponente zwischen den beiden Endpunkten eines I/O-Kanals virtualisiert ist. In einem solchen Rechenzentrum nimmt Virtualisierung bei allen Vorgängen und Arbeitsabläufen eine zentrale Position ein, da jedes Gerät an der Virtualisierung von I/O-Kanälen beteiligt ist. Diese Schichtung bedeutet, dass kein Nutzer des Dienstes (Client) Wissen über die zugrundeliegende Technik benötigt und auch keinen Einfluss darauf nehmen kann. Der I/O-Kanal als solcher ist für Clients *transparent*. Deshalb muss der Verwendungszweck von I/O-Kanälen nicht berücksichtigt werden und die Betrachtung der beiden Endpunkte sowie der Infrastruktur zwischen den Endpunkten von I/O-Kanälen hinreichend ist. Mögliche Endpunkte sind per Definition eine Speichereinheit, oder ein Treiber des Betriebssystems, wobei auf Speicherknoten und Speichereinheiten in dieser Arbeit nicht näher eingegangen wird. Da heutzutage jeder einsatzbereite Server über ein Betriebssystem verfügt, muss der gesamte Dienstlebenszyklus virtueller Infrastrukturen, insbesondere den darin enthaltenen Servern, betrachtet werden. In einem Rechenzentrum eines passenden Szenarios spielen demnach Dienstlebenszyklen von Servern und virtuellen Infrastrukturen eine grundlegende Rolle. Um ein Maximum an Virtualisierung zu erreichen, wird auf jeder Komponente der Infrastruktur Virtualisierung betrieben. In einem solchen Rechenzentrum sind die Vorgänge und Arbeitsschritte der Anwendungsfälle, die den Dienstlebenszyklus betreffen, von Virtualisierung geprägt (siehe oben). Deshalb sind die Anwendungsfälle, die sich aus einem solchen Szenario ergeben, geeignet, um darauf basierend Anforderungen an Techniken zur Virtualisierung von I/O-Kanälen abzuleiten.

Zunächst werden im Kapitel 2.1 einige Begriffe eingeführt, um die einzelnen Komponenten und Systeme benennen und unterscheiden zu können. Das Szenario in Kapitel 2.2 beschreibt den Dienstleister Molpid, der ein solches Rechenzentrum betreibt, um virtuelle Infrastrukturen an Kunden zu vermieten. Anschließend werden in Kapitel 2.3 die für die Anwendungsfälle relevanten Akteure vorgestellt. Die Anwendungsfälle des Szenarios werden in Kapitel 2.4 analysiert. Aus der Analyse resultierende Anforderungen werden in Kapitel 2.5 gebündelt.

2.1 Begriffe und Definitionen

Server in Rechenzentren sind heute meist keine Einheit mehr, sondern meistens aufgeteilt in Speicher und Rechenknoten. Die Aufteilung erfolgte ursprünglich aus der Konsolidierung des niedrig ausgelasteten Hintergrundspeichers einzelner Server.
Ein *Rechenknoten* bezeichnet den Server (Host), auf dem Betriebsysteme und Applikationen betrieben werden. Ein Server kann über Festplatten verfügen, allerdings werden diese nie als Hintergrundspeicher genutzt.
Ein *Speicherknoten* ist im Zusammenhang dieser Arbeit ein Computer, dessen einzige Auf-

gabe es ist, Speichereinheiten über ein Netz zugänglich zu machen.
Eine *Speichereinheit* ist abstrahierter Hintergrundspeicher, der Rechenknoten zugänglich gemacht wird. Auf ihr werden Daten zur dauerhaften Speicherung abgelegt.
Das **Speichernetz** (SAN), zwischen den Speicherknoten und den Rechenknoten, ist konfigurierbar und ermöglicht eine dynamische Ressourcenzuweisung. Freie Kapazitäten können dadurch nach Bedarf an Server zugewiesen, ohne physisch in die Infrastruktur einzugreifen. Dadurch wird die Auslastung des Hintergrundspeichers erhöht, da nicht für jedes System einzeln Leistungsreserven vorgehalten werden müssen. Im Gegensatz zum LAN dient das SAN lediglich dem Zweck Server mit Hintergrundspeicher zu verbinden. Deshalb werden SANs typischerweise für diese Aufgabe optimiert.
Ein *I/O-Kanal* stellt der Kommunikationsweg zwischen einem Betriebssystem und einer zugewiesenen Speichereinheit, oder zwei Betriebssystemen dar. Ein I/O-Kanal erstreckt sich vom Treiber des Betriebssystems durch I/O-Hardware über Switche bis zur Speichereinheit oder bis zu dem Treiber eines anderen Betriebssystems. Ein I/O-Kanal zeichnet sich dadurch aus, dass die Zwischenschritte zwischen den Endpunkten nicht von dem Betriebssystem beeinflussbar sind.
Als *I/O-Hardware* wird in dieser Arbeit die Hardware bezeichnet, die es einem Server ermöglicht, auf ein Netz zuzugreifen, an das auch andere Server angebunden sind. Für das LAN wird I/O-Hardware meißt als NIC (Network Interface Card) bezeichnet, während für das SAN die allgemeinere Hardware-Bezeichnung HBA (Host Bus Adapter) gängig ist.
Die *Infrastruktur* ist der physische Aufbau des Rechenzentrums mit allen Komponenten und Verbindungen. Dazu zählen Server, Switche und Speicherknoten, sowie mindestens eine Management-Station und Chassis für Bladeserver.
Eine *virtuelle Infrastruktur* bezeichnet eine Teilmenge der Infrastruktur. Hat ein Kunde mehrere Server gemietet, kann er diese in virtuellen Infrastrukturen organisieren. Eine virtuelle Infrastruktur eines Kunden umfasst dessen gemietete Server und das Netz, das die gemieteten Server verbindet.

Da Hintergrundspeicher in Speicherknoten ausgegliedert wurde, existiert der Server nicht mehr als Einheit, sondern als Konzept, bestehend aus einer dreischichtigen Architektur, wie in Abbildung 2.1 dargestellt wird.

Die Hosts der obersten Schicht sind, wie auch die Speicherknoten der untersten Schicht, an das Speichernetz (die mittlere Schicht), angeschlossen. Das Speichernetz selbst ist so aufgebaut, dass ein beliebiger Host auf den Dienst eines beliebigen Speicherknoten zugreifen kann und dadurch mit mehreren Speichereinheiten verknüpft werden kann. Die Server sind als senkrechte Säulen eingezeichnet, um zu zeigen, dass ein für den Kunden einsatzbereiter Server ein Ergebnis und kein Teil des Aufbaus ist. Server können auch als Tupel $s = (c_{host} \times c_{san} \times c_{speicher})$ von Konfigurationen für Hosts (c_{host}), SAN (c_{san}) und Speicherknoten ($c_{speicher}$) aufgefasst werden.

2.2 Szenario: Segmentierte Infrastrukturen

Ein Produkt der Firma Molpid stellt die Vermietung von Servern dar. Dabei wird zwischen physischen und virtuellen Servern unterschieden. Physische Server richten sich an Kunden, die mehrere Server mieten und Software mit sehr hohen Leistungsforderungen an die Hardware einsetzen. Für gewöhnlich werden virtuelle Server eingesetzt, um eine bessere

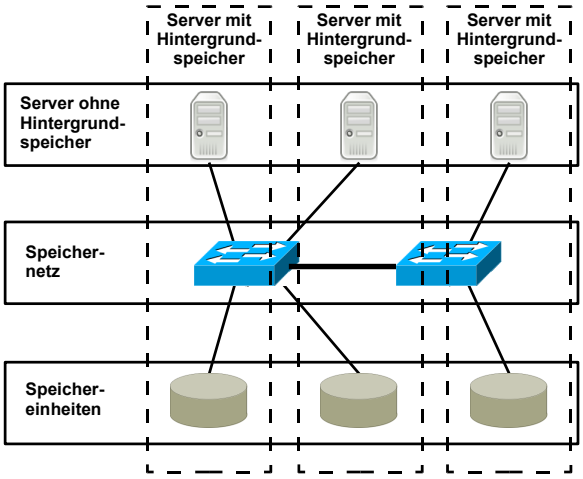


Abbildung 2.1: Die Aufteilung eines Servers in drei Schichten

Auslastung der Hardware zu erreichen. Dazu weden auf den Servern Hypervisor betrieben, die virtuelle Server erzeugen. Server können ohne Vorgaben und Voreinstellungen vermietet werden. Der Kunde kann auf diese Weise Betriebssysteme und Applikationen frei wählen. Ein Kunde, der mehrere Server mietet, kann diese in virtuellen Infrastrukturen organisieren. Virtuelle Infrastrukturen ermöglichen eine Kommunikation zwischen den Servern, identisch mit dem Verhalten eines separaten LANs. Der Datenverkehr innerhalb einer virtuellen Infrastruktur ist für Dritte unzugänglich, so dass es zu keinem ungewollten Datenaustausch zwischen den Kunden kommen kann. Zusätzlich kann mit dem Kunden in der Dienstvereinbarung (Service Level Agreement, SLA) ein bestimmtes Maß an Verlässlichkeit bezüglich Verfügbarkeit des Servers und zur Verfügung stehender Übertragungsrate sowie der Netzverzögerung vereinbart werden. Die Inanspruchnahme weiterer Dienstleistungen durch den Kunden ist möglich. Dazu zählen ein automatisches Backup der Speichereinheiten und das Installieren von Betriebssystemen auf Servern sowie Veränderungen bezüglich der Zusammensetzung von virtuellen Infrastrukturen. Die genaue Beschaffenheit des Rechenzentrums bleibt dem Kunden verborgen, so dass dieser kein Wissen über die Topologie der physischen Infrastruktur benötigt.

Der Hintergrundspeicher eines Servers wird komplett auf Speicherknoten ausgelagert. Dies ermöglicht es dem Betreiber jederzeit Anzahl, Größe und Zugriffsberechtigungen der Speichereinheiten zu verändern, ohne an der Hardware physische Veränderungen durchzuführen. Dadurch können Kundenwünsche bezüglich des Hintergrundspeichers während des produktiven Betriebs zeitnah und ohne Unterbrechungen umgesetzt werden. Benötigt ein Kunde zum Beispiel mehr Speicherplatz als ihm ursprünglich zur Verfügung steht, kann der Speicheradministrator die entsprechende Speichereinheit des Kunden vergrößern, ohne am Server des Kunden Veränderungen durchzuführen.

Eine schematische Übersicht über den Aufbau der Infrastruktur zeigt Abbildung 2.2. Als Hosts werden Bladeserver eines großen deutschen Serverherstellers eingesetzt. Die NIC und und der HBA der Blades sind redundant ausgelegt, um den Ausfall einer einzelnen Komponente aufzufangen. Neben einem gestarteten Betriebssystem erlangt man auch durch einen BMC (Baseboard Management Controller) Zugang zu den Hardwareeigenschaften eines Bla-

des. Der BMC ist ein eigenständiger Computer und direkt mit dem Chassis verbunden, in dem das Blade steckt. Auf diese Weise erlangt Molpid jederzeit Zugang zum Überwachen und Konfigurieren der Hardware, unabhängig von der virtuellen Infrastruktur und eingesetzten Software der Kunden. Jedes Chassis ist mit einem MMB (Management Blade) versehen, der nach außen die Schnittstelle des Chassis darstellt. Das MMB speichert Konfigurationen für die identifizierenden Merkmale von Bladeservern, wie zum Beispiel GUID, MAC-Adresse oder WWN. Mit diesen Konfigurationen überschreibt der MMB mit Hilfe des BMC die Werkseinstellungen des Blades. So sind zum Einen zwei baugleiche Blades gegeneinander austauschbar, da sämtliche identifizierenden Merkmale überschreibbar sind, zum Anderen wird eine Konfiguration statt einem Blade, einem Fach des Chassis zugewiesen. Bei der Aktivierung eines Blades wird die Konfiguration vom MMB auf das Blade übertragen und die Werte der jeweiligen Hardware-Komponente zugewiesen. Eine zentrale Management-Station stellt eine Verbindung zu einem MMB her, um die Switche und Blades eines Chassis zu verwalten. Alle übrigen Switche sind eigenständige Komponenten und werden als solche individuell verwaltet. Ein Management-VLAN ermöglicht der Management-Station jederzeit die Kommunikation mit jedem Blade, Chassis, Switch und Speicherknoten, so dass die Verwaltung der Geräte unabhängig von der Konfiguration der Netze für Kunden ist.

Für die Verbindung von Servern und Speicherknoten kommt ein spezialisiertes SAN zum Einsatz, während für alle übrige Kommunikation das LAN benutzt wird. Das Nutzungsprofil des Speichernetzes hängt direkt mit dem Nutzungsprofil des Hintergrundspeichers zusammen. Da der Zugriff auf Hintergrundspeicher immer über ein Dateisystem und damit blockweise erfolgt, ist die kleinste Einheit der übertragenen Nutzdaten ein Block des Dateisystems. Die Blockgröße eines Dateisystems befindet sich üblicherweise in der Größenordnung von Kilobytes. Bei Zugriffen auf den Hintergrundspeicher werden häufig ganze Dateien gelesen oder geschrieben. Das Nutzungsprofil zeichnet sich daher auch dadurch aus, dass häufig viele zusammenhängende, direkt aufeinander folgende Blöcke übertragen werden.

Für den Kunden wird ein Portal bereitgestellt, über das dieser seine virtuellen Infrastrukturen steuern kann. Die dem Kunden zur Verfügung stehenden Funktionen dienen der Steuerung und nicht der Konfiguration der Server. Dazu zählen neben dem Ein- und Ausschalten auch das Neuinstallieren von Servern. Des Weiteren kann der Kunde hier das Sichern seiner Speichereinheiten steuern.

Die Steuerung zum Sichern von Speichereinheiten beinhaltet die Verwaltung der Zeitpunkte an denen Sicherungskopien der Speichereinheiten erstellt werden, das unmittelbare Anstoßen des Erstellens von Sicherungskopien sowie das Zurückspielen von Sicherungskopien auf die Speicherheinheiten.

Außerdem wird das Portal als zentrales Kommunikationsmedium zwischen dem Kunden und dem Betreiber verwendet. So kann der Kunde über das Portal virtuelle Infrastrukturen anlegen und verändern sowie zusätzliche Server bestellen. Funktionen für virtuelle Infrastrukturen haben keinen direkten Effekt, sondern werden an Molpid weitergeleitet.

2.2.1 Sicht des Betreibers

Die Management-Station ist die Steuerzentrale der gesamten Infrastruktur. Es handelt sich hierbei um einen einzigen Computer, der von Molpid dazu benutzt wird, Switche, Chassis und Hypervisor zu konfigurieren. Da virtuelle Infrastrukturen jederzeit auf Kundenwunsch geändert werden können, gibt es eine spezielle virtuelle Infrastruktur, das Management-VLAN, welches immer existiert und dem Administrator die Kommunikation mit allen Swit-

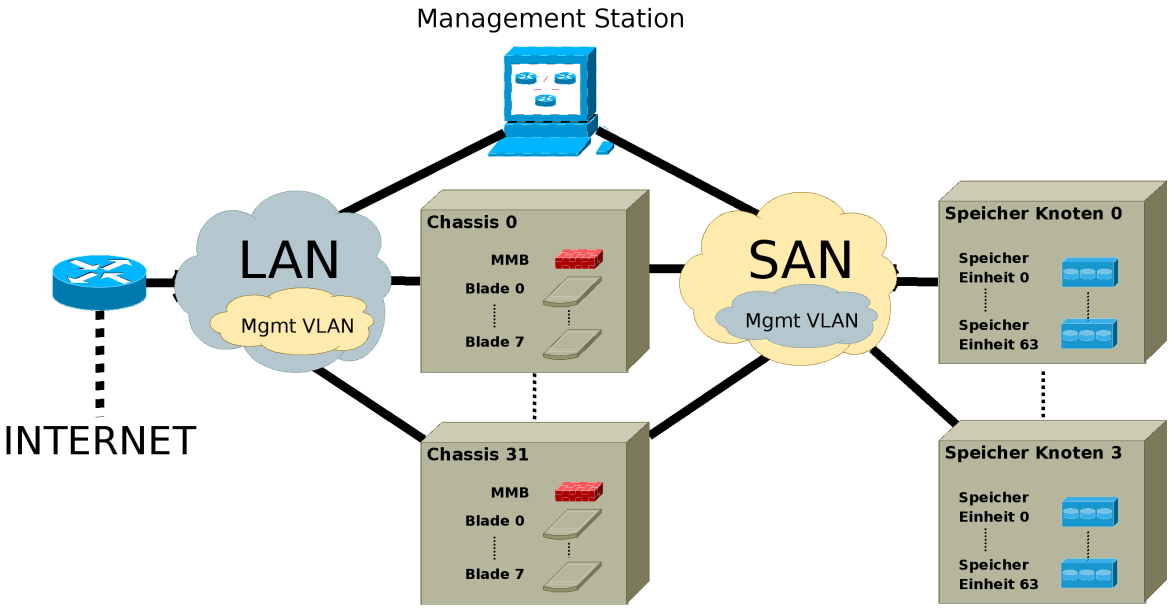


Abbildung 2.2: Sicht des Betreibers auf die Infrastruktur

chen, Chassis, Blades und den Speicherknoten ermöglicht. Die Management-Station hält eine Kopie der Konfiguration einer jeden Komponente vor. Sie kann diese nach Bedarf verändern und auf die Geräte zurückschreiben. Die Konfiguration eines Blades stellt eine Menge von identifizierenden Merkmalen dar und ist getrennt vom Betriebssystem, das auf diesem Blade betrieben werden soll. Die Konfiguration eines Switches ist die Menge aller Einstellungen, die das Verhalten des Switches steuern. So werden von zentraler Stelle aus Konfigurationen erstellt und verteilt, um die Kundenbestellungen umzusetzen. Für den Zusatzdienst des Vorinstallierens von Betriebssystemen benutzt Molpid ein Abbild einer fertigen Installation und kopiert dieses bei Bedarf auf die Speichereinheiten des Kunden. Das Abbild muss nach dem Kopieren für den Server des Kunden „personalisiert“ werden, indem das Betriebssystem mit den identifizierenden Merkmalen des Servers konfiguriert und anschließend mit den richtigen Treibern für die Hardware ausgestattet wird. Für die Backup-Funktion wird ein Programm des Herstellers der Speichereinheit benutzt. Dieses Programm wird auf den Speicherknoten betrieben und kann Abbilder von Speichereinheiten erstellen und zurückspielen.

2.2.2 Sicht des Kunden

Während der Betreiber Server und Speichereinheiten in vollständig voneinander getrennten Pools verwaltet (vgl. Abbildung 2.3), bleibt dem Kunden die Ausgliederung des Hintergrundspeichers in Speicherknoten verborgen. Ein Server und dessen zugewiesene Speichereinheiten bilden für den Kunden eine Einheit, die er als ein Server betreibt. Das Speichernetz ist für den Kunden unzugänglich. Veränderungen an Speichereinheiten und der Zuweisung von Speichereinheiten können nur indirekt über das Portal erfolgen. Chassis bleiben dem Kunden vollständig verborgen, da Chassis bzw. MMBs niemals Teil einer vermieteten virtuellen Infrastruktur sind. Ein Kunde nimmt lediglich seine gemieteten Server, aufgeteilt in virtuelle Infrastrukturen, entsprechend der Aufträge aus dem Portal, wahr.

Abbildung 2.3 zeigt den Unterschied zwischen der Sicht des Kunden und der des Betreibers auf die Infrastruktur. Der Betreiber verwaltet Server und Speichereinheiten in getrennten Pools. Für die Bereitstellung von virtuellen Infrastrukturen für Kunden müssen zusätzlich zu den Servern und Speichereinheiten auch die beiden Netze LAN und SAN verwaltet werden. Der Betreiber überwacht und konfiguriert die Komponenten und stellt so die vom Kunden gewünschte Infrastruktur zusammen. Der Kunde kann seine Server über das Portal ein- und ausschalten und den Status der Server abfragen. Alle Konfigurationsänderungen und Bestellungen werden als Anfrage an den Betreiber weitergeleitet und von diesem bearbeitet.

2.3 Akteure

Die Verwaltung des im Szenario beschriebenen Rechenzentrums wird auf vier Rollen verteilt. Diese sind *LAN*-, *SAN*-, *Server*- und *Speicheradministration*. Die Aufteilung in Rollen erfolgt anhand von Technologien. Wissen über eine bestimmte Technologie ist demzufolge komplett in einer einzigen Rolle konzentriert.

LAN-Administratoren sind verantwortlich für das Anlegen und Betreiben von Kommunikationspfaden in virtuellen Infrastrukturen. Ein wichtiger Aspekt dieser Arbeit sind die Analysen von Datenflüssen. Diese dienen als Grundlage für Entscheidungen hinsichtlich Kommunikationspfaden, um Leistungsengpässe zu vermeiden und Zusicherungen erfüllen zu können. LAN-Administratoren erstellen virtuelle Netze, welche zusammen mit den Servern virtuelle Infrastrukturen ergeben.

SAN-Administratoren verwalten das Netz, das Speichereinheiten und Server miteinander verbindet. Ihre Aufgaben unterscheiden sich nicht von denen der LAN-Administratoren. Die Motivation für die Differenzierung besteht darin, dass Verbindungen im LAN zwischen allen Servern ermöglicht werden, während das SAN Server mit Hintergrundspeicher verbindet. Im Gegensatz zum LAN ist die Nutzung des SANs sehr speziell, weshalb die Charakteristika der Nutzung des SANs bei der Konfiguration berücksichtigt werden können. Dadurch wird eine effizientere Nutzung des SANs erreicht.

Server-Administratoren beschäftigen sich hauptsächlich mit Ressourcenverwaltung. Zu ihrem Aufgabenbereich gehört das Überwachen der Chassis sowie der freien Kapazitäten. Im Vordergrund steht dabei die Planung mit welchen physischen Servern Kundenbestellungen erfüllt werden, um Fragmentierung von virtuellen Infrastrukturen zu vermeiden und vorhandene Kapazitäten effizient zu nutzen.

Speicheradministratoren sind für die Wartung von Speicherknoten und die Bereitstellung von Speichereinheiten zuständig. Diese Thematik soll nicht Gegenstand dieser Arbeit sein.

Die Verantwortungsbereiche der Server-, LAN- und SAN-Administratoren überlappen sich, da sich auch in Chassis Switche für LAN und SAN befinden. Dadurch können diese Switche nicht eindeutig als Verantwortungsbereich einer bestimmten Rolle identifiziert werden.

2.4 Anwendungsfälle

Das Tagesgeschäft bei Molpid umfasst Aktivitäten bezüglich des Dienstlebenszykluses von (virtuellen) Servern und virtuellen Infrastrukturen. Ein Dienstlebenszyklus besteht aus den Phasen Planung, Verhandlung, Bereitstellung, Betrieb, Anpassung und Auflösung [DR02].

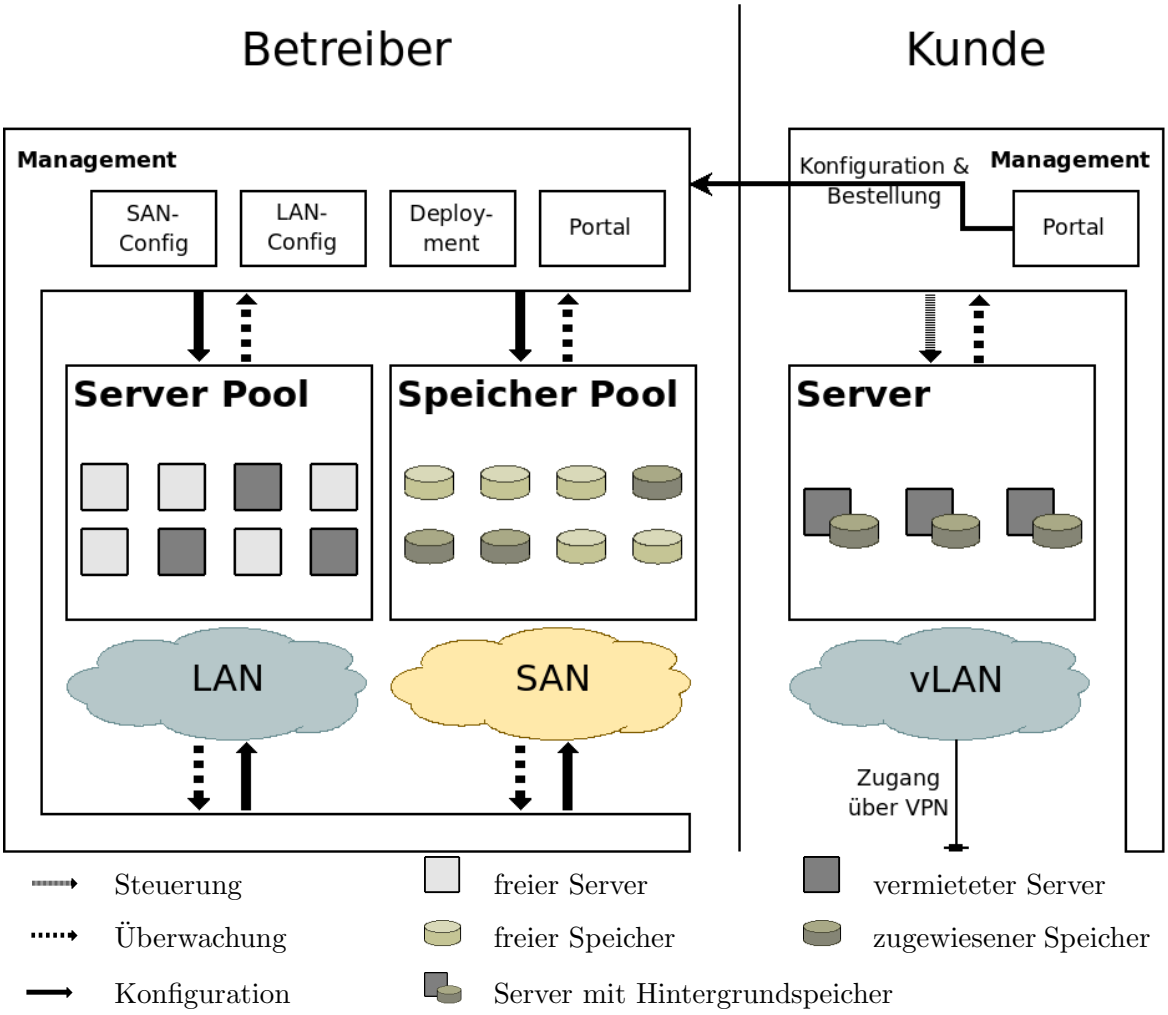


Abbildung 2.3: Managementsichten von Betreiber und Kunde

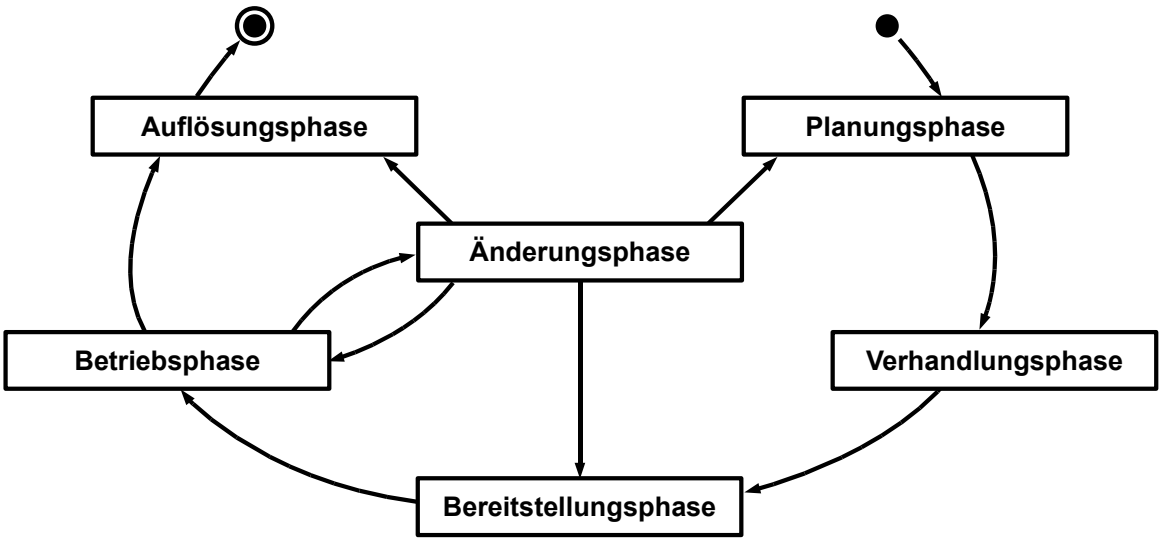


Abbildung 2.4: Der Dienstlebenszyklus nach Dreo [DR02]

Die Phasen und mögliche Übergänge zwischen den Phasen werden als Zustandsautomat in Abbildung 2.4 dargestellt.

In der *Planungsphase* werden die technischen und administrativen Aspekte der Bereitstellung und des Betriebs von virtuellen Infrastrukturen untersucht. Dazu zählen mehrere Analysen, wie die Bedarfsschwerpunkt-, Bedarfsgrößen- und Komponentenanalyse [HAN99]. Im Rahmen dieses Szenarios liefern diese Analysen Aussagen über:

- Anzahl der benötigten physischen und virtuellen Server des Kunden
- Anzahl der benötigten physischen Server, um virtuelle Server zur Verfügung zu stellen
- Abschätzung des Lastprofils für LAN und SAN

In der *Verhandlungsphase* wird eine Dienstvereinbarung zwischen dem Betreiber und dem Kunden festgelegt. Das Hauptaugenmerk dieser Phase liegt in den Zusicherungen bezüglich Verfügbarkeit, Übertragungsrate und Verzögerung. Während der *Bereitstellungsphase* instanziiert der Betreiber die zuvor festgeschriebenen virtuellen Infrastrukturen. Dabei werden die physischen Server so gewählt, dass die Netze in der Lage sind die Zusicherungen zu gewährleisten, die in der Verhandlungsphase festgelegt wurden. Die *Betriebsphase* ist die Phase, in der die virtuellen Infrastrukturen durch den Kunden genutzt werden. Die Hauptaufgabe des Betreibers liegt hier in der Sicherstellung der Betriebsbereitschaft virtueller Infrastrukturen im Rahmen der Dienstvereinbarung. In der *Anpassungsphase* werden virtuelle Infrastrukturen von Kunden verändert. Änderungen beinhalten das Zusammenfügen oder das Aufspalten virtueller Infrastrukturen. Beim Zusammenfügen werden zusätzliche Kommunikationspfade ermöglicht, während nach dem Aufspalten weniger Möglichkeiten zur Verfügung stehen. Der Dienstlebenszyklus endet mit der *Auflösungsphase*, in der die virtuellen Infrastrukturen deaktiviert werden und die Ressourcen freigegeben werden.

Das eigene Unternehmenswachstum und neue Softwareversionen können die Kundenanforderungen an die gemieteten virtuellen Infrastrukturen verändern, wodurch der Dienstlebenszyklus in die Anpassungsphase übergeht. Wie bei der Aufzählung der Phasen erwähnt,

ist die Zusammensetzung von virtuellen Infrastrukturen Gegenstand der Änderungsphase. Dies hat direkte Auswirkungen auf das Nutzungsprofil des LANs. Dabei muss sichergestellt werden, dass die Änderungen mit allen bestehenden Dienstvereinbarung des Betreibers in Einklang stehen. Es ist möglich, dass dazu ein Übergang in eine vorherige Phase notwendig wird, um andere Komponenten für die Bereitstellung zu wählen, Dienstvereinbarung anzupassen, oder in einer erneuten Planungsphase die gesamten Anforderungen des Kunden zu analysieren.

In der Planungsphase der physischen Infrastruktur werden Techniken zur Virtualisierung von I/O-Kanälen ausgewählt. Dabei steht vor allem der Investitionsschutz im Vordergrund. Aus diesem entstehen Anforderungen, die nicht unmittelbar mit virtuellen Infrastrukturen und Kunden in Zusammenhang stehen. Investitionsschutz bezieht sich im Wesentlichen auf die zeitliche Dauer des Lebenszykluses einzelner Komponenten. Sollen Komponenten wie zum Beispiel Switche und Speicherknoten den Dienstlebenszyklus der physischen Infrastruktur überdauern, nehmen diese eine besondere Rolle während der Auflösungsphase der alten Infrastruktur und der Bereitstellungsphase der neuen Infrastruktur ein.

Im Folgenden werden in Kapitel 2.4.1 Anforderungen aus der Bereitstellungs- und Auflösungsphase der physischen Infrastruktur untersucht.

Während der Dienstlebenszyklusphasen *Bereitstellung*, *Betrieb*, *Änderung* und *Auflösung* von virtuellen Infrastrukturen entstehen aus Management-Aufgaben Anwendungsfälle. An diesen sind hauptsächlich Rollen beteiligt, die vom Betreiber wahrgenommen werden. Diese Anwendungsfälle werden in Kapitel 2.4.2 untersucht. Im Anschluss daran wird in Kapitel 2.4.3 jede Dienstlebenszyklusphase betrachtet und in Bezug zum Betrieb von virtuellen Infrastrukturen und zu den bereits behandelten Anwendungsfällen gesetzt. In diesem Kapitel wird festgestellt, dass sich keine geeigneten Anwendungsfälle mit dem Kunden als Hauptrolle finden lassen. Zum Schluss werden die Anforderungen aus den verschiedenen Unterkapiteln im Abschnitt 2.5 gebündelt, ausformuliert und gewichtet. Basierend auf diesem Anforderungenkatalog werden in Kapitel 4 Techniken zur Virtualisierung von I/O-Kanälen analysiert.

2.4.1 Integration von Techniken zur Virtualisierung von I/O-Kanälen

Dieses Kapitel leitet Anforderungen aus der Änderungs- und Auflösungsphase des Dienstlebenszyklus der physischen Infrastruktur ab. Analog zu virtuellen Infrastrukturen, durchläuft die physische Infrastruktur einen Dienstlebenszyklus. Die Zyklen von virtuellen Infrastrukturen laufen während der *Betriebs-* und *Änderungsphase* der physischen Infrastruktur ab.

Die *Bereitstellungs-* und *Auflösungsphase* der physischen Infrastruktur unterscheiden sich von denen der virtuellen Infrastrukturen. Eine physische Infrastruktur wird selten vollständig bereitgestellt. Meistens wird eine bereits vorhandene Infrastruktur vergrößert und gegebenenfalls Teile davon ersetzt. Analog dazu wird eine physische Infrastruktur selten komplett aufgelöst. Statt dessen werden einige Komponenten in einer neuen Infrastruktur weiter verwendet.

Anforderungen

Die Anforderungen ergeben sich daraus, dass der Produktlebenszyklus einzelner Komponenten, wie zum Beispiel von Switchen und Speicherknoten, den Dienstlebenszyklus einer physischen Infrastruktur überdauern kann. In einer neuen physischen Infrastruktur können neue Technologien zum Einsatz kommen. Wenn vorhandene Technologien weiterverwendet

werden sollen und neue Technologien zum Einsatz kommen, müssen die Techonologien kombiniert werden. Daraus lassen sich folgende Anforderungen ableiten:

- Virtuelle Infrastrukturen müssen technologieübergreifend anlegbar sein
- Zusicherungen müssen über Technologiegrenzen hinweg durchsetzbar sein

2.4.2 Anwendungsfälle des Betreibers

Die Aufgaben des Betreibers in den Phasen der Bereitstellung, des Betriebs, der Änderung und der Auflösung sind Management-Aufgaben, mit der Absicht dem Kunden den vereinbarten Dienst entsprechend der zugesicherten Qualität zur Verfügung zu stellen. All diese Aktivitäten gehören zu einer der fünf Bereiche des Funktionsmodells der OSI Management Architektur: ***F**ault Management*, ***C**onfiguration Management*, ***A**ccounting Management*, ***P**erformance Management* und ***S**ecurity Management* (FCAPS) [HAN99].

Um den gesamten Betrieb des Rechenzentrums zu unterstützen, muss eine Virtualisierungstechnik Anforderungen erfüllen, die sich aus jedem der oben genannten Management-Gebiete ergeben können. Diese Anforderungen werden im Folgenden aus Anwendungsfällen abgeleitet, die sich je einem Management-Bereich zuordnen lassen. Accounting Management nimmt eine Sonderstellung ein und wird im Rahmen einer Anwendungsfallanalyse nicht betrachtet, da die für diese Arbeit relevanten Aktivitäten des Account Managements bereits durch andere Management-Gebiete abgedeckt werden. Das entsprechende Kapitel 2.4.2.5 wird der Vollständigkeit halber trotzdem als Anwendungsfall geführt.

Abbildung 2.5 zeigt die für die Analyse gewählten vier Anwendungsfälle und die beteiligten Akteure. Am Ende eines jeden Anwendungsfalls werden die für diesen relevanten Anforderungen aufgelistet. Erläuterungen der einzelnen Anforderungen sind in Kapitel 2.5 zusammengefasst.

Der Anwendungsfall 2.4.2.1 „Failover“ stammt aus dem Bereich Fault Management und analysiert das Vorgehen, sofern eine einzelne Komponente nicht mehr ordnungsgemäß funktioniert und deshalb ersetzt werden muss. Aus dem Configuration Management ist der Anwendungsfall 2.4.2.2 „Phyische Hardwareerweiterung“ gewählt. Dieser beschreibt wie ein zuvor inaktiver Server in die Infrastruktur integriert wird, um anschließend eine virtuelle Infrastruktur zu ergänzen. Performance Management wird in diesem Szenario im Wesentlichen durch den Anwedungsfall 2.4.2.3 „Lastverteilung“ bestimmt. Hierbei werden notwendige Aktivitäten untersucht, um sicher zu stellen, dass die Zusicherungen der Dienstvereinbarungen eingehalten werden können. Der zuletzt betrachtete Anwendungsfall „Anlegen von virtuellen Infrastrukturen“ wird in Kapitel 2.4.2.4 behandelt. Dieser Anwendungsfall lässt sich auch dem Configuration Management zuordnen, jedoch stehen hier Sicherheitsaspekte, insbesondere die Isolation des Datenverkehrs, im Vordergrund.

2.4.2.1 Fault Management

Häufig passiert es, dass Hardware, aufgrund fehlerhafter Konfiguration oder eines Defekts, nicht mehr ordnungsgemäß arbeitet. Um die Anfälligkeit gegenüber solchen Ausfällen zu reduzieren, sind die Netze und die I/O-Hardware der Blades redundant ausgelegt. Der Ausfall einer einzelnen Komponente beeinträchtigt daher nicht unmittelbar den Betrieb. Fällt ein NIC aus, kann der Betrieb über ein redundantes NIC fortgesetzt werden. Die Wahrscheinlichkeit eines Totalausfalls des Servers ist in diesem Fall erhöht, da die Redundanz nicht mehr

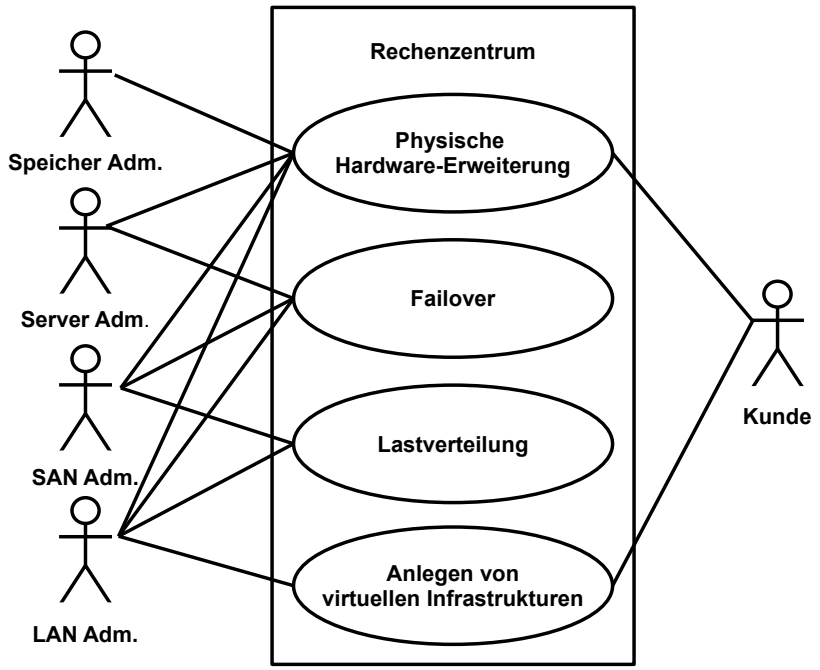


Abbildung 2.5: UML-Anwendungsfalldiagramm, zeigt die untersuchten Anwendungsfälle und die beteiligten Akteure

vorhanden ist. Wird auf dem Server ein Hypervisor betrieben, würde der Ausfall dieses physischen Servers zum Ausfall mehrerer virtueller Server führen. Im laufenden Betrieb können defekte Komponenten nicht ausgetauscht werden, sondern der Host muss heruntergefahren und ersetzt werden. Der Austausch umfasst folgende Arbeitsschritte:

1. LAN und SAN umkonfigurieren, so dass das Ersatzgerät gleichgestellt zu dem defekten Gerät betrieben werden kann
2. Ersatzgerät konfigurieren, damit es sich identisch zum defekten Gerät in die Infrastruktur integriert
3. Defekten Server herunterfahren
4. Ersatzserver starten

Herausforderungen

Um die Unterbrechung des Serverbetriebs so kurz wie möglich zu halten, werden zuerst Ersatzgerät und Netze konfiguriert, während der defekte Host weiter in Betrieb bleibt. Das Ersatzgerät kann außerhalb des produktiven Betriebs konfiguriert werden und beeinflusst diesen dadurch nicht. Da die Netze Teil der produktiven Infrastruktur sind, ist das Ändern der Konfiguration der Netze ein kritischer Eingriff in den produktiven Betrieb. Das konfigurierte Ersatzgerät wird in die Infrastruktur eingefügt, während das defekte Gerät noch in Betrieb ist. Der Ersatzserver wird hochgefahren und der Zustand des defekten Geräts auf das Ersatzgerät übertragen. Danach übernimmt der Ersatzserver die Arbeit und der defekte Server deaktiviert werden. Dieses Vorgehen reduziert den Zeitraum, in der der Server des

Kunden nicht erreichbar ist auf ein Minimum. Die beiden größten Herausforderungen liegen in dem parallelen Betrieb von defektem Server und Ersatzserver sowie die Übertragung des Zustands zwischen den Servern. Damit der Ersatzserver den defekten Server ersetzen kann, müssen die identifizierenden Merkmale überspielt werden. Befinden sich zwei Server mit identischen identifizierenden Merkmalen in der selben Infrastruktur, führt dies zu Problemen. Es ist die Aufgabe desjenigen, der die Evakuierung des Servers durchführt dafür zu sorgen, dass diese Probleme vermieden werden.

Anforderungen des Anwendungsfalls

Der Failover wird von Server-, LAN-, gegebenenfalls auch SAN-Administratoren durchgeführt. Der Server-Administrator bestimmt den Ersatzserver und konfiguriert diesen entsprechend. Da der defekte Host und der Ersatzhost für einen kurzen Zeitraum gleichzeitig betrieben werden, kann der Ersatzhost nicht den Platz des defekten Hosts im Chassis einnehmen. Deshalb müssen neue Kommunikationspfade für den Ersatzhost geschaltet werden. Durch die Überlappung der Zuständigkeitsbereiche an den Switche in den Chassis kann dies einzig durch den Server-Administrator durchgeführt werden, vorausgesetzt der Ersatzhost wird selben Chassis, wie der defekte Host eingesetzt. Anderenfalls müssen LAN- und SAN-Administrator die neuen Kommunikationspfade schalten. Für diesen Anwendungsfall sind folgende Anforderungen relevant:

- Eindeutige Bezeichner für virtuelle Infrastrukturen
- Lastverteilung über mehrere Kommunikationspfade
- Messbarkeit der Auslastung
- I/O-Hardware muss redundant auslegbar sein
- Ein einzelner Hardwaredefekt darf die Verfügbarkeit nicht beeinträchtigen
- Konfigurationsänderungen müssen sofort in Kraft treten
- I/O-Hardware muss für Administratoren eindeutig adressierbar sein
- Modifizierbarkeit aller identifizierender Merkmale eines Servers

2.4.2.2 Configuration Management

Möchte ein Kunde seine gemietete Infrastruktur durch neue Server erweitern, müssen zusätzliche Hosts in die virtuelle Infrastruktur eingefügt werden. Sind die vorhandenen Kapazitäten des Rechenzentrums ausgelastet, müssen zusätzliche physische Hosts in den Produktivbetrieb eingebracht werden. Dies erfolgt in drei Schritten:

1. Einfügen des Blades in die Topologie
2. Assoziieren des Blades mit einer Speichereinheit
3. Aufnahme des Servers in die virtuelle Infrastruktur des Kunden

Die Reihenfolge der Arbeitsschritte ergibt sich daraus, dass für das Assoziieren von Blades mit Speichereinheiten mit Pfadschaltung gearbeitet wird. Der Pfad kann erst bestimmt werden, wenn bekannt ist, in welches Chassis das Blade eingefügt wird. Abbildung 2.6 zeigt den I/O-Pfad von einem Speicherknoten zu einem neu eingefügten Blade. Der Kunde kann für neue Server zusätzliche Leistungen in Anspruch nehmen, wie z.B. das Vorinstallieren eines Betriebssystems. Ist der Server der Bestellung entsprechend vorbereitet worden, kann er in die Infrastruktur des Kunden eingefügt werden.

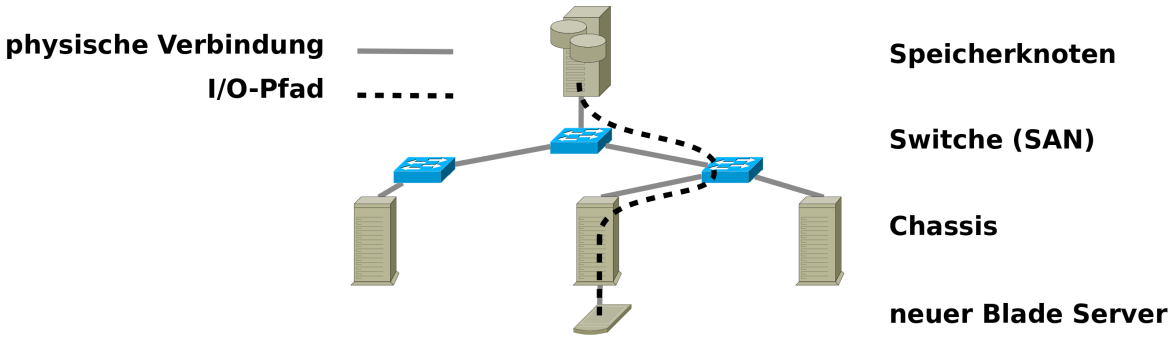


Abbildung 2.6: Der I/O-Pfad (gestrichelte Linie) von Speicherknoten zu Blade wird durch das Chassis, in das das Blade eingefügt werden soll, bestimmt

Herausforderungen

Eine der Hauptherausforderungen beim Betrieb des Rechenzentrums ist es, die häufigen Veränderungen in Anzahl und Zusammensetzung der virtuellen Infrastrukturen korrekt auf die physische Infrastruktur abzubilden. Eine virtuelle Infrastruktur kann vom Kunden über das Portal verändert werden, ohne die Interna des Rechenzentrums zu kennen. Es kann dazu kommen, dass die Server eines Kunden räumlich nicht konzentriert, sondern über das gesamte Rechenzentrum verteilt sind. In diesem Fall ist der Aufwand zum Umsetzen der Änderungswünsche des Kunden sehr hoch, da mehr Switche umkonfiguriert werden müssen. Bei der Umsetzung muss stets auf die Isolation des Datenverkehrs und die Zusicherungen aus der Dienstvereinbarungen geachtet werden.

Anforderungen des Anwendungsfalls

Eingeleitet wird dieser Anwendungsfall mit der Bestellung weiterer Server durch den Kunden. Daraufhin erstellt der Speicheradministrator auf einem Speicherknoten eine neue Speichereinheit. Der Server-Administrator bringt einen neuen physischen Server in die Infrastruktur ein. Je nach Kundenbestellung wird ein kompletter physischer Server, oder ein virtueller Server zur Verfügung gestellt. Für virtuelle Server muss der Server-Administrator zunächst noch den physischen Server vorbereiten. Ist der neue Kundenserver bereit, so kann der SAN-Administrator die Verbindung zwischen Server und Speichereinheit herstellen. Der LAN-Administrator erweitert das Management-VLAN um permanenten Zugriff auf den neuen Server sicherstellen zu können und integriert den neuen Server anschließend in eine virtuelle Infrastruktur des Kunden. Für diesen Anwendungsfall sind folgende Anforderungen relevant:

- Eindeutige Bezeichner für virtuelle Infrastrukturen
- I/O-Hardware muss für Administratoren eindeutig adressierbar sein
- I/O-Hardware muss einen Managementzugang verfügen
- Der Managementzugang der I/O-Hardware muss geschützt sein
- Virtuelle Infrastrukturen können sich überlappen
- Anzahl virtueller Infrastrukturen darf nicht begrenzt sein

2.4.2.3 Performance Management

An zentralen Stellen der Topologie können Flaschenhälse entstehen, die die Qualität der Verbindung in Übertragungsrate und Verzögerung beeinträchtigen. Die Vermeidung der Flaschenhälse ist Aufgabe des Betreibers. Dies kann auf zweierlei Arten erreicht werden. Zum Einen können als Reaktion auf einen Engpass ungesättigte physische Verbindungen hinzugefügt werden, um dadurch die Kapazität zu erhöhen. Zum Anderen kann das Netz bei der Planung und Konfiguration schon derart aufgesetzt werden, dass Engpässe seltener entstehen. Die Aktivitäten zum Zuschalten weiterer Kommunikationspfade können in drei Schritte zusammengefasst werden:

1. Ermittlung alternativer Pfade
2. Bewertung der Pfade unter Berücksichtigung aller Zusicherungen an Kunden
3. Hinzufügen neuer Pfade zur virtuellen Infrastruktur

Herausforderungen

Verfügt eine virtuelle Infrastruktur über mehr als einen Pfad zwischen zwei Knoten, muss eine Zusicherung nicht von jedem einzelnen dieser Pfade erfüllt werden, sondern die Summe der Leistungen dieser Pfade muss eine Zusicherung erfüllen. Dies erschwert die Berechnung der freien Kapazitäten, das wiederum die Bewertung von Pfaden und die Planung von neuen virtuellen Infrastrukturen erschwert.

Anforderungen des Anwendungsfalls

An diesem Anwendungsfall sind LAN- oder SAN-Administrator beteiligt, abhängig von dem Netz, in dem ein Engpass vermieden werden soll. Zuerst werden ungesättigte alternative Pfade gesucht, die keinen Flaschenhals enthalten. Diese Pfade sind geeignet, um die kritische Verbindung mit dem Engpass zu entlasten. Dieser Fall tritt ein, wenn dadurch keine bestehenden Zusicherungen an Kunden beeinträchtigt werden. Ist ein solcher Pfad gefunden, kann dieser der virtuellen Infrastrukturen hinzugefügt werden, um Last zu verteilen.

Aufgrund des bekannten Nutzungsprofils des SANs hat der SAN-Administrator zusätzliche Optimierungsmöglichkeiten. Muss ein Block erneut übertragen werden, wird dieser erneut von einer Festplatte gelesen. Aufgrund der Beschaffenheit von Festplatten ist dies sehr zeitaufwändig. Die größten Verzögerungen im SAN entstehen durch Festplattenzugriffe. Deshalb stehen bei Planung und Konfiguration des SANs die Vermeidung von Festplattenzugriffen im Vordergrund. Dies wird erreicht, indem bereits bei der Datenübertragung von Knoten zu Knoten (Link Layer des OSI Schichtenmodells) die Datenintegrität sichergestellt wird und Flaschenhälse erkannt und vermieden werden. Für diesen Anwendungsfall sind folgende Anforderungen relevant:

- Eindeutige Bezeichner für virtuelle Infrastrukturen
- Markierbarkeit von Datenströmen
- Wahl des Kommunikationspfades entsprechend der virtuellen Infrastruktur (Routing)
- Überwachbarkeit von Datenströme
- Lastverteilung über mehrere Kommunikationspfade
- Messbarkeit der Auslastung
- Regulierbarkeit von Datenströmen
- Verlustfreie Übertragung des Speichernetzes

- Blockweise Datenübertragung des Speichernetzes
- Zustellung der Datenblöcke in richtiger Reihenfolge
- Konfigurationsänderungen müssen sofort in Kraft treten
- Anzahl virtueller Infrastrukturen darf nicht begrenzt sein
- I/O-Hardware muss für Administratoren eindeutig adressierbar sein
- Modifizierbarkeit aller identifizierender Merkmale eines Servers

2.4.2.4 Security Management

Hat ein Kunde mehr als einen Server gemietet, kann er diese in virtuellen Infrastrukturen organisieren, indem er das vom Betreiber bereitgestellte Portal nutzt. Da der Kunde kein Wissen über die tatsächliche Beschaffenheit des Rechenzentrums verfügt und nur Zugriff auf seine Server hat, implementiert Molpid virtuelle Infrastrukturen für den Kunden. Dies erfolgt in drei Schritten:

1. Über das Portal erstellt der Kunde eine Konfiguration für eine virtuelle Infrastruktur und übermittelt diese an den Betreiber
2. Der Betreiber ermittelt aus der Anfrage des Kunden die notwendigen Änderungen an den Konfigurationen der einzelnen Komponenten
3. Das LAN wird neu konfiguriert

Herausforderungen

Da der Kunde die komplette Kontrolle über die von ihm gemieteten Server hat, muss die Isolation des Datenverkehrs in die Switche implementiert werden, denn diese kontrolliert alleine der Betreiber. Basierend darauf kann sichergestellt werden, dass sich Dritte keinen Zugang zu anderen virtuellen Infrastrukturen verschaffen. Da nicht nur Isolation, sondern auch Verfügbarkeit und Übertragungsrate zugesichert werden, können Kommunikationspfade nicht beliebig gewählt werden, sondern müssen mit bestehenden Pfaden und damit verbundenen Zusicherungen koordiniert werden.

Anforderungen des Anwendungsfalls

Der Kunde wählt über das Portal aus all seinen gemieteten Servern eine Teilmenge aus, die er in einer virtuellen Infrastruktur organisieren möchte. Diese Auswahl wird als Auftrag an den LAN-Administrator übergeben, der den Auftrag im LAN umsetzt. Dabei müssen analog zu Kapitel 2.4.2.3 Kommunikationspfade unter der Berücksichtigung von Zusicherungen gegenüber anderen Kunden gefunden werden.

Für diesen Anwendungsfall sind folgende Anforderungen relevant:

- Eindeutige Bezeichner für virtuelle Infrastrukturen
- Markierbarkeit von Datenströmen
- Wahl des Kommunikationspfades entsprechend der virtuellen Infrastruktur (Routing)
- Filterung von Datenströmen
- I/O-Hardware muss einen Managementzugang bieten
- Der Managementzugang der I/O-Hardware muss geschützt sein
- Virtuelle Infrastrukturen können sich überlappen

2.4.2.5 Accounting Management

Das Accounting Management stellt einen Sonderfall dar. Abgesehen von monetären Aspekten zählen zu den Aktivitäten von Accounting Management die Verwaltung von *Kundendaten*, *Zugriffsrechten* und *Kontingenten* wie auch das Sammeln und Auswerten von *Nutzungsdaten* [HAN99].

Eine Assoziation von Kunden zu virtuellen Infrastrukturen ist für den Betrieb der virtuellen Infrastruktur nicht notwendig. Alle Aktivitäten, die nicht direkt die Infrastruktur betreffen, liefern auch keine Anforderungen an diese.

In diesem Szenario wird die genaue Beschaffenheit des Rechenzentrums, insbesondere physische Verbindungen, vor dem Kunden verborgen. Der Kunde kann deshalb keine Funktionen der Switche explizit nutzen und die Dienstvereinbarung sieht vor, dass die gemieteten Server beliebig eingesetzt werden dürfen. Deshalb ist eine Verwaltung von *Zugriffsrechten* durch den Betreiber nicht notwendig.

Kontingente bezüglich Hintergrundspeicher und logischen Verbindungen in LAN und SAN werden als Teil der Dienstvereinbarung während der Verhandlungsphase festgelegt. Die Management-Aufgaben beschränken sich somit auf die Erfüllung der Dienstvereinbarung und Bereitstellung von Kontingenten, die bereits in die Bereiche Performance Management (Kapitel 2.4.2.3) und Configuration Management (Kapitel 2.4.2.2) fallen und an dieser Stelle nicht erneut betrachtet werden müssen.

Die relevanten Nutzungsdaten werden durch die verwendeten Ressourcen der physischen Server und die Auslastung der physischen Verbindungen dargestellt, da die Funktionen der Switche nie explizit von den Kunden benutzt werden und die Server-Funktionen nicht genauer spezifiziert sind. Diese Daten werden bereits im Rahmen des Performance Managements (Kapitel 2.4.2.3) erhoben um die Erfüllung der Dienstvereinbarung sicherzustellen.

Die Management-Aufgaben des Accounting Managements, welche die Infrastruktur und virtuelle Infrastrukturen betreffen, sind bereits durch Performance und Configuration Management abgedeckt. Management-Aufgaben, welche weder die Infrastruktur noch virtuelle Infrastrukturen betreffen, sind nicht geeignet, um daraus Anforderungen an Virtualisierungslösungen abzuleiten. Aus diesem Grund wird in dieser Arbeit kein zusätzlicher Anwendungsfall für das Accounting Management analysiert.

2.4.3 Anwendungsfälle des Kunden

Wie im Szenario beschrieben, soll ein Kunde eine virtuelle Infrastruktur nutzen können, wie ein separates LAN, das einzig dem Kunden zur Verfügung steht. Das Ziel ist, dass ein Kunde eine virtuelle Infrastruktur nutzen kann, wie eine speziell für ihn bereitgestellte physische Infrastruktur, ohne Virtualisierung, bestehend aus LAN und Server. Im Szenario wird dazu die Zusammensetzung der physischen Infrastruktur vor dem Kunden verborgen, so dass räumliche Anordnung und andere virtuelle Infrastrukturen keinerlei Auswirkungen haben. Es ist nicht notwendig die Aufteilung von Server in Speicher- und Rechenknoten vor dem Kunden zu verbergen, jedoch soll er Speichereinheiten lediglich benutzen können. Das heißt, der Kunde darf keinerlei Einfluss auf Beschaffenheit und Zuweisung von Speichereinheiten nehmen können.

In der *Betriebsphase* des Dienstlebenszykluses nimmt der Betreiber Management-Aufgaben wahr (siehe Kapitel 2.4.2), wohingegen der Kunde mit der betriebsbereiten virtuellen Infrastruktur arbeitet. In der *Bereitstellungs-* und der *Auflösungsphase* nimmt der Kunde keine

Aufgaben wahr. Während der Bereitstellungsphase wird die Dienstvereinbarung durch den Betreiber auf die physische Infrastruktur abgebildet. Hierbei entstehen keine Aufgaben, die der Kunde erfüllen könnte, da er keinen Zugang zur physischen Infrastruktur hat. Während der Auflösungsphase werden die durch eine virtuelle Infrastruktur gebundenen Ressourcen wieder frei gegeben. Analog zur Bereitstellungsphase entstehen hier keine Aufgaben, die der Kunde erfüllen könnte. Für den Kunden relevante Phasen des Dienstlebenszyklus sind die *Planungs-*, *Verhandlungs-* und *Änderungsphasen*.

Kapitel 2.4.2 beschreibt unter anderem, dass durch eine Änderungsphase mögliche Kommunikationspfade hinzukommen oder wegfallen. Neue Kommunikationspfade konnten im Verlauf der Verhandlungsphase nicht berücksichtigt werden und das Wegfallen bestimmter Kommunikationspfade kann das Nutzungsprofil des LANs verändern. Deswegen ist eine Änderungsphase in diesem Szenario immer mit einer erneuten Verhandlungsphase verbunden, gegebenenfalls auch mit einer erneuten Planungsphase. Dies ist eine Einschränkung gegenüber den in Abbildung 2.4 dargestellten Übergangsmöglichkeiten für Dienstlebenszyklusphasen.

Somit müssen alle Anforderungen des Kunden aus der Planungsphase und der Verhandlungsphase abgeleitet werden können. Da der Kunde jedoch nicht zwischen physischen und virtuellen Servern unterscheidet und nie explizit Funktionen der Switche nutzt (vgl. Kapitel 2.4.2.5), stehen die Anforderungen des Kunden nicht in direktem Zusammenhang mit Techniken zur Virtualisierung von I/O-Kanälen. Virtualisierung ist ein Werkzeug des Betreibers.

2.5 Anforderungen

Aus dem Szenario und den damit zusammenhängenden Herausforderungen bei den Arbeitsabläufen im Rechenzentrum werden im Folgenden Anforderungen bezüglich Virtualisierung abgeleitet. Je mehr Anforderungen eine Technik zur Virtualisierung von I/O-Kanälen erfüllt, desto höher ist der Grad an erreichbarer Abstraktion und Virtualisierung. Die Anforderungen ergeben sich entweder direkt aus dem Produkt, oder aus dem Aufbau der Infrastruktur.

#1 Virtuelle Infrastrukturen brauchen jeweils einen eindeutigen Bezeichner. Kunden können virtuelle Infrastrukturen erstellen, die durch den Betreiber auf die reale Infrastruktur abgebildet werden. Um virtuelle Infrastrukturen zentral mit der Management-Station verwalten zu können, müssen virtuelle Infrastrukturen voneinander unterscheidbar sein. Auch für die Zuordnung einer virtuellen Infrastruktur zu einem Kunden und der Visualisierung im Portal müssen virtuelle Infrastrukturen eindeutig identifizierbar sein. Ohne einen eindeutigen Bezeichner sind virtulle Infrastrukturen lediglich lose Kombinationen von Konfigurationen und können nicht als Einheit verwaltet werden. Deswegen sind eindeutige Bezeichner eine Grundvorraussetzung für die zentrale Verwaltung virtueller Infrastrukturen, wie im Szenario beschrieben.

#2 Datenströme müssen markiert werden können. Teilen sich mehrere virtuelle Infrastrukturen dieselbe physische Verbindung, kann ein Switch unmarkierte Pakete lediglich anhand von Kommunikationsendpunkten identifizieren. Kapitel 2.4.2.4 beschreibt die Problematik, dass virtuelle Infrastrukturen nicht allein durch die Kommunikationsendpunkte umgesetzt werden können. Der Grund dafür liegt darin, dass Kommunikationsendpunkte nicht unter der alleinigen Kontrolle des Betreibers stehen und so-

mit die Sicherheit und Isolation der Kommunikation nicht garantiert werden können. Deshalb muss die erste Komponente des I/O-Kanals, die alleine der Betreiber kontrolliert, in der Lage sein, Datenströme als zu einer bestimmten virtuellen Infrastruktur zugehörig zu markieren. Nur auf diesem Wege ist nachfolgenden Komponenten eine zuverlässige Zuordnung von Paketen zu virtuellen Infrastrukturen möglich. Diese Anforderung ermöglicht erst die Segmentierung der Infrastruktur in virtuelle Infrastrukturen und ist deshalb notwendig.

#3 Datenströme müssen gefiltert werden können. Eine, wie im Szenario geforderte, strikte Trennung von virtuellen Infrastrukturen zeichnet sich dadurch aus, dass Pakete nur innerhalb der virtuellen Infrastruktur zugestellt werden. Neben dem Markieren von Paketen, muss zusätzlich sichergestellt werden, dass einen Server nur solche Datenpakete erreichen, die zu dessen virtueller Infrastruktur gehören. Eine solche Filterfunktion nimmt eine Schlüsselrolle bei der Durchsetzung virtueller Infrastrukturen ein.

#4 Virtuelle Infrastrukturen müssen technologieübergreifend anlegbar sein. Werden zwei Technologien in einer physischen Infrastruktur kombiniert, muss gewährleistet sein, dass virtuelle Infrastrukturen auch über Technologiegrenzen hinweg angelegt werden können. Insbesondere benötigt dies eine Komponente, die beide Technologien implementiert und den Datenverkehr zwischen den Technologien umsetzen kann. Dabei kommt es vor allem darauf an, dass die zugesicherte Isolation des Datenverkehrs aufrechterhalten wird. Wird diese Anforderung von einem Paar aus zwei Technologien nicht erfüllt, so sind diese Technologien nicht dazu geeignet in der selben physichen Infrastruktur eingesetzt zu werden.

#5 Zusicherungen müssen über Technologiegrenzen hinweg durchsetzbar sein. Werden zwei Technologien in derselben physischen Infrastruktur eingesetzt, können zwei Server derselben virtuellen Infrastruktur angehören und gleichzeitig unterschiedliche Technologien zur Kommunikation benutzen. Die Zusicherungen der Dienstvereinbarung müssen in diesem Fall über die Grenzen einer Technologie hinaus durchgesetzt werden. Wird diese Anforderung nicht erfüllt, können keine zwei Technologien in derselben physischen Infrastruktur eingesetzt und gleichzeitig die Dienstleistungsvereinbarungen erfüllt werden.

#6 Datenströme müssen überwachbar sein. Die Dienstvereinbarung zwischen Betreiber und Kunde kann Zusicherungen über Verfügbarkeit, zur Verfügung stehender Übertragungsrate und Verzögerung bei der Datenübertragung beinhalten. Um Zusicherungen kontrollieren und messen zu können, müssen die einzelnen Komponenten Informationen über den angefallenen Datenverkehr bereitstellen. Ist diese Anforderung nicht erfüllt, lassen sich zwar virtuelle Infrastrukturen betreiben, Zusicherungen bezüglich der Dienstgüte können nicht in die Dienstvereinbarung aufgenommen werden.

#7 Die Auslastung eines Kommunikationsweges muss messbar sein. Für die Lastverteilung auf mehrere Kommunikationspfade, muss es eine Metrik geben, anhand der bestimmt werden kann, ob ein Kommunikationspfad entlastet werden muss und über wieviel freie Kapazität ein solcher verfügt. Diese Information wird auch in der Planungs- und Verhandlungsphase benötigt. Ohne diese Information können virtuelle Infrastrukturen betrieben werden, allerdings ohne Qualitätszusicherungen, wie im Szenario beschrieben.

#8 Jedes identifizierende Merkmal eines Servers muss veränderbar sein. Identifizierende Merkmale der Hardware, wie auch das Betriebssystem, gehören zum Zustand eines Servers. Soll der Zustand eines Servers übertragen werden, müssen demnach auch die identifizierenden Merkmale übertragen und auf dem Zielsystem überschrieben werden können. Ist dies nicht möglich, sind die Konsequenzen im Fehlerfall analog zu Anforderung 7.

#9 Datenströme müssen regulierbar sein. Um Zusicherungen bezüglich des Netzes umsetzen zu können, muss ein Switch kontrollieren können wieviel Ressourcen für eine bestimmte virtuelle Infrastruktur aufgewendet werden. Damit ist es möglich die maximale Übertragungsrate einzelner Verbindungen zu limitieren, um Engpässe und daraus resultierenden Datenverlust zu vermeiden. Diese Funkion ist ein wichtiges Werkzeug des Performance Managements für die Erfüllung von Dienstvereinbarungen.

#10 I/O-Hardware muss über einen Management-Zugang verfügen. Wird ein neuer Server für den Kunden zur Verfügung gestellt, müssen Parameter für die Verbindung zur Speichereinheit und dem LAN konfiguriert werden können, bevor der Server durch den Kunden aktiviert wird. Diese Parameter umfassen unter Anderem die MAC des NICs und die WWN des HBAs. Deshalb muss es einen Management-Zugang geben, der dem Betreiber die Möglichkeit gibt die Konfiguration der I/O-Hardware dauerhaft zu verändern. Dieser Zugang muss betriebssystemunabhängig sein, da der Kunde freie Wahl bei der Installation von Betriebssystemen und volle Kontrolle über einen Server während des Betriebs hat. Der Management-Zugang ist notwendig, um ein Rechenzentrum wie im Szenario beschrieben betreiben zu können.

#11 Der Management-Zugang von I/O-Hardware benötigt eine Zugangskontrolle. Da die virtuellen Infrastrukturen der Kunden voneinander isoliert sein müssen, darf ein Kunde nicht in der Lage sein seinen Server derart umzukonfigurieren, dass er aus seiner virtuellen Infrastruktur ausbrechen kann. Kunden können jedoch ganze physische Server mieten und haben somit auch den vollen Zugriff auf die Hardware des Servers. Daher muss der Zugang auf die Management-Komponente der Hardware reguliert werden können, da sonst die Isolation der virtuellen Infrastruktur nicht garantiert werden kann.

#12 Virtuelle Infrastrukturen können sich überlappen. Da ein Kunde seine Server ohne Einschränlungen in virtuelle Infrastrukturen organisieren kann, muss es auch die Möglichkeit geben einen Server in mehrere virtuelle Infrastrukturen gleichzeitig zu integrieren. Ist diese Funktionalität nicht gegeben, beschränkt dies die Möglichkeiten des Kunden.

#13 Die Anzahl der virtuellen Infrastrukturen darf nicht begrenzt sein. Virtuelle Infrastrukturen müssen auf die physische Infrastruktur abgebildet werden können. Durch häufige Veränderungen der Anzahl und Zusammensetzung der virtuellen Infrastrukturen, können diese fragmentieren, wodurch die Server einer virtuellen Infrastruktur über das Rechenzentrum verstreut platziert sein können. In diesem Fall kann es dazu kommen, dass es Switche gibt, die Teil vieler Kommunikationspfade von verschiedenen virtueller Infrastrukturen sind. Damit die Kommunikationspfade und die darüber übertragenen Daten der Kunden jederzeit isoliert bleiben, muss ein Switch praktisch beliebig viele unterschiedliche virtuelle Infrastrukturen unterscheiden können. Wird diese

Anforderung nicht erfüllt, kann die Trennung der virtuellen Infrastrukturen nicht mehr garantiert werden.

#14 I/O-Hardware muss Redundant betrieben werden können. Der Failover Anwendungsfall des Fault Managements setzt voraus, dass ein Server auch nach dem Ausfall einer NIC oder eines HBAs erreichbar bleibt. Dazu muss es möglich sein I/O-Hardware redundant zu betreiben, damit beim Ausfall einer Komponente die Erreichbarkeit gewährleistet bleibt. Wird diese Anforderung nicht erfüllt, können virtuelle Infrastrukturen erstellt und betrieben werden, jedoch kann ein Ausfall nicht aufgefangen werden. Aufgrund dessen muss häufiger physisch in die Infrastruktur eingegriffen werden und ein Ausfall resultiert unmittelbar in einem Dienstausfall.

#15 Der Ausfall einzelner Komponenten darf die Verfügbarkeit nicht beeinträchtigen. Analog zur Verfügbarkeit eines Servers durch Redundanz, müssen auch LAN, SAN und Speichereinheiten so ausgelegt sein, dass beim Ausfall einer einzelnen Komponente ein Server erreichbar bleibt. Dadurch können dessen Aufgaben und Zustand übertragen werden. Anderenfalls können Ausfälle nicht aufgefangen werden.

#16 Konfigurationsänderungen der I/O-Hardware müssen sofort in Kraft treten. Wie in Kapitel 2.4.2.2 beschrieben gehört die Modifkation der Zusammensetzung virtueller Infrastrukturen zum normalen Arbeitsablauf eines Rechenzentrums. Deshalb muss Hardware in der Lage sein, Konfigurationsänderungen zu übernehmen, ohne den Betrieb dafür unterbrechen zu müssen. Anderenfalls könnte die Modifikation einer virtuellen Infrastruktur den Betrieb vieler virtueller Infrastrukturen unterbrechen. Ist diese Anforderung nicht erfüllt, lassen sich virtuelle Infrastrukturen anlegen und betreiben. Wartungsarbeiten erzeugen jedoch häufiger kurzzeitige Ausfälle.

#17 I/O-Hardware muss für Administratoren eindeutig adressierbar sein. Im Falle eines Failovers muss der Zustand von einer Hardware Komponente von einem Server auf eine andere Komponente eines anderen Servers übertragen werden. I/O-Hardware muss daher auch über die Grenzen eines einzelnen Servers hinaus eindeutig adressierbar sein, um das Übertragen von Zuständen realisieren zu können und Kontrolle über das Verschieben von Konfigurationen zu haben. Erfüllt eine Methode diese Anforderungen nicht, fehlt ein wichtiges Werkzeug des Fault Managements. Virtuelle Infrastrukturen können in diesem Fall erstellt und betrieben werden, jedoch erhöht sich der Arbeitsaufwand und die Ausfallzeit im Fehlerfall steigt.

#18 Übertragung des Speichernetzes muss verlustfrei sein. Wird am Ende der Übertragung festgestellt, dass die Daten unvollständig übertragen wurden, oder auf dem Weg zwischen Speicherknoten und Server korrumpiert wurden, muss der komplette Datenblock erneut übertragen werden. Der Anwendungsfall „Lastverteilung“ (Kapitel 2.4.2.3) beschreibt unter anderem, dass das wiederholte Übertragen von Blöcken aufgrund des dadurch notwendigen Festplattenzugriffs sehr zeitaufwendig ist, wodurch die Effizienz des SANs sinkt. Wird die Datenintegrität bereits auf dem Link Layer, der zweiten Schicht des ISO-OSI Modells, sichergestellt, können zusätzliche Festplattenzugriffe vermieden und und dadurch die Effizienz des SANs erhöht werden.

#19 Die Rahmen des Speichernetzes müssen komplette Blöcke des Dateisystems fassen können. Wie im Szenario beschrieben wird die kleinste im Speichernetz

zu übertragende Einheit von Nutzdaten durch die Blockgröße des Dateisystems bestimmt. Um Fragmentierung zu vermeiden und den Überhang gering zu halten, muss das Speichernetz ein ganzzahliges Vielfaches der Blockgröße des Dateisystems übertragen können. Dadurch kann Fragmentierung vermieden werden, wodurch das Verhältnis von Überhang zu Nutzdaten der übertragenen Daten kleiner wird. Dies bedeutet, dass die selben Nutzdaten mit weniger Aufwand übertragen werden können und die Effizienz erhöht wird.

#20 Das Speichernetz muss Datenblöcke in der richtigen Reihenfolge übertragen. Ebenfalls aus dem Nutzungsprofil des SAN abgeleitet, müssen Pakete in der richtigen Reihenfolge übertragen werden. Werden Pakete nicht in der Richtigen Reihenfolge übertragen, müssen die einzelnen Blöcke in einem Lesepuffer gehalten werden. Die benötigte Größe des Lesepuffers ist durch die geforderten großen Rahmen und die in Kapitel 3.2.1 vorgestellte Mehrfachnutzung von Hardware durch Servervirtualisierung schwer abschätzbar. Des Weiteren erzeugt ein solcher Lesepuffer Verzögerungen im I/O-Kanal. Werden die Blöcke zuverlässig in der richtigen Reihenfolge übertragen, können Verzögerungen vermieden werden. Dies trägt ebenfalls zur Steigerung der Effizienz bei.

#21 Virtuelle Infrastrukturen beeinflussen Wahl des Kommunikationspfades. Wie im Szenario beschrieben, ist das Ziel von virtuellen Infrastrukturen, dem Kunden ein Netz zur Verfügung zu stellen, das wie ein physisch separates LAN genutzt werden kann. Kann bereits ein Switch die Markierung (siehe Anforderung 2) einer virtuelle Infrastruktur berücksichtigen, ermöglicht dies eine präzisere Steuerung der Nutzung von physischen Verbindungen. Dadurch können vorhandene Kapazitäten effizienter genuzt werden. Verfügt ein Switch nicht über dieses Mittel zur präzisen Steuerung, teilen sich die gesamten virtuellen Infrastrukturen alle vorhandenen Kapazitäten. Daraus entsteht kein Nachteil, solange die Kapazitäten ausreichen, um alle Dienstvereinbarungen zu erfüllen. Deshalb ist diese Anforderung nicht notwendig zur Umsetzung des Szenarios.

#22 Die Last muss auf mehrere physische Verbindungen verteilt werden können. Werden mehrere virtuelle Infrastrukturen auf dieselbe physische Verbindung abgebildet, muss diese Verbindung alle Zusicherungen gleichzeitig einhalten können. Übersteigen die Zusicherungen die Möglichkeiten der Verbindung, müssen im Falle von Lastspitzen I/O-Kanäle über alternative Verbindungen geleitet werden, um die Zusicherungen nicht zu verletzen. Kann die Last nicht verteilt werden, müssen entsprechende Leistungsreservern vorgehalten werden, wodurch die vorhandenen Kapazitäten weniger effizient genutzt werden können.

Eine Übersicht über die Anforderungen und die Anwendungsfälle aus denen sie resultieren, bietet Tabelle 2.1. Die Anforderungen sind unterschiedlich gewichtet. Hierbei werden folgende vier Gruppen Unterschieden:

**** Dies ist die Klasse der Anforderungen die Vorraussetzungen für alle anderen Anforderungen sind. Erfüllt eine Methode diese Anforderungen nicht, ist diese für die Implementierung virtueller Infrastrukturen ungeeignet.

*** Sind alle Anforderungen dieser Klasse erfüllt, ist eine Infrastruktur, wie im Szenario beschrieben, implementierbar. Diese Anforderungen sind Vorraussetzungen, um Dienstvereinbarungen erfüllen zu können.

Anforderung	Gewicht	Anwendungsfall				
		2.4.1	**2.4.2.1**	**2.4.2.2**	**2.4.2.3**	**2.4.2.4**
#1	****		x	x	x	x
#2	****				x	x
#3	****					x
#4	****	x				
#5	***	x				
#6	***				x	
#7	***		x		x	
#8	***		x	x	x	
#9	***				x	
#10	***			x		x
#11	***			x		x
#12	***			x	x	x
#13	***			x	x	
#14	**		x			
#15	**		x			
#16	**		x		x	
#17	**		x		x	x
#18	*				x	
#19	*				x	
#20	*				x	
#21	*				x	x
#22	*		x		x	

Tabelle 2.1: Anforderungen an Methoden zur Virtualisierung von I/O-Kanälen

** Kann eine Methode zusätzlich zu den vorrausgegangenen Anforderungen auch mit ** gewichtete Anforderungen erfüllen, kann ein deutlich höheres Maß an Qualität zugesichert werden. Zu dieser Klasse zählen insbesondere Anforderungen um das Management virtueller Infrastrukturen zu vereinfachen, so dass Wartungsarbeiten an einer virtuellen Infrastruktur andere virtuelle Infrastrukturen nicht beeinflussen.

* Anforderungen dieser Klasse betreffen die Effizienz einer Methode und Eigenschaften, die eine präzisere Kontrolle ermöglichen, als unbedingt notwendig. Sind diese Anforderungen nicht erfüllt, kann die Infrastruktur weniger effizient genutzt werden und die entsprechenden Herausforderungen können einzig durch physisches Eingreifen in die Infrastruktur bewältigt werden.

3 Stand der Technik

Kapitel 3 beschäftigt sich mit Verteilungskonzepten, Virtualisierungskonzepten und deren technische Umsetzungen. Kapitel 3.1 geht auf geschichtete Systeme und ihre Eigenschaften ein. Dabei wird festgestellt, dass geschichtete Systeme um zusätzliche Schichten erweitert werden können, wodurch weitere Funktionalität in ein System eingebracht werden kann.

Handelt es sich bei der Erweiterung um ein Kommunikationssystem, können Teilsysteme auf andere Rechner ausgelagert werden. Insbesondere ermöglicht dies die Aufteilung von Server in Speicher- und Rechenknoten, wie im Szenario (siehe Kapitel 2.2) beschrieben.

Eine andere Möglichkeit der Erweiterung sind transformierende Schichten, deren Funktion darin besteht die Interaktion zwischen zwei Schichten zu verändern. Solche Schichten ermöglichen es Probleme zu abstrahieren und sind die Grundlage für Virtualisierung. Darauf aufbauend beschreibt Kapitel 3.2 Virtualisierungskonzepte für die Hauptbestandteile eines Rechenzentrums, den Speicherknoten, Netzen und Rechenknoten (vgl. Kapitel 2.2). In jedem Unterkapitel werden heute verfügbare Technologien, die Virtualisierungskonzepte umsetzen, vorgestellt. Diese Technologien werden in Kapitel 4 zu Techniken zur Virtualisierung von I/O-Kanälen gruppiert.

3.1 Schichtung und Verteilung

Computer sind heutzutage in der Lage, aufwendige Berechnungen durchzuführen und große Mengen an Daten zu speichern. Netze, wie zum Beispiel das Ethernet, ermöglichen es Ressourcen eines Computers Benutzern an anderen Computern zugänglich zu machen. Gemeinsam genutzte Ressourcen werden meistens in Servern konsolidiert und als Dienst bereitgestellt. Die Architektur, mit der Funktionen in Systemen konsolidiert werden und als Dienst zur Verfügung gestellt werden, bezeichnet man als Klient-/Anbieter-Architektur [Brü06] (bzw. Client-/Server-Architektur). Ein Vorteil dieses Architekturstils ist die Austauschbarkeit verschiedener Systeme, vorausgesetzt diese stellen dieselben Dienste bereit. Werden Teilsysteme auf mehrere Computer verteilt, so spricht man von einem *verteilten* System. Wird diese Architektur durchgängig eingesetzt, können gemeinsam genutzte Daten und Applikationen vollständig losgelöst voneinander verwaltet werden. Wenn die Datenmenge wächst, oder Applikationen komplexer werden, kann man so gezielt den entsprechenden Server anpassen.

Eine besondere Rolle bei den Überlegungen in diesem Kapitel nehmen die *geschichteten* Systeme ein. Abbildung 3.1 skizziert wie mit dem Client-/Server-Architekturstil ein System in Teilsysteme zerlegt wird, die zu einem geschichteten System sortiert werden können. In einem geschichteten System nutzt ein Teilsystem (Schicht) nur die Dienste von genau einem anderen Teilsystem. Diese Anwendung des Client-/Server-Architekturstils erlaubt es Probleme schrittweise zu lösen. Ein Teilsystem löst ein Teilproblem und stellt diese Lösung dem nachfolgenden Teilsystem zur Verfügung. Eine Schicht nutzt die Dienste einer vorausgehenden Schicht, ohne näher auf ihre Beschaffenheit einzugehen. Insbesondere ist die Funktion einer Schicht unabhängig von der Anzahl der vorausgehenden Schichten und der Art und

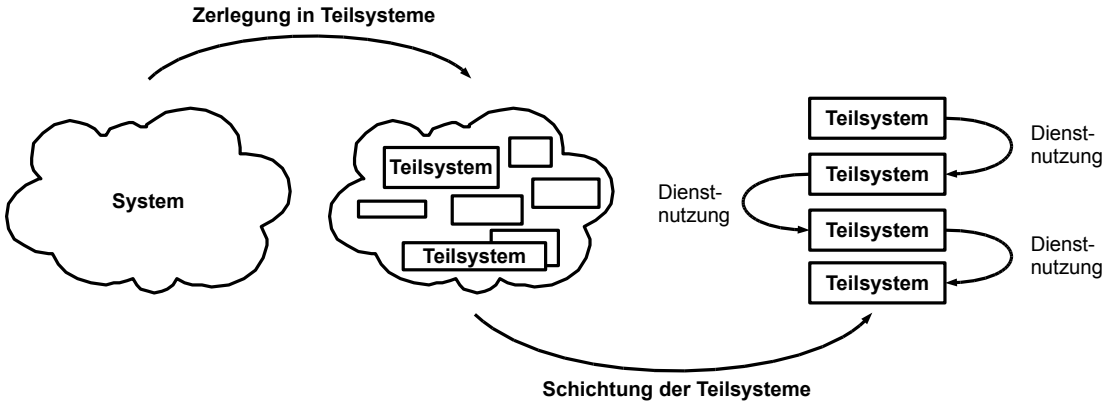


Abbildung 3.1: Zerlegung eines Systems in Teilsysteme

Weise wie diese Probleme lösen. Statt dessen können die in vorausgehenden Schichten immer als gelöst betrachtet werden. Diese Art der Problembetrachtung, bei der davon ausgegangen wird, dass bestimmte Probleme in vorausgehenden Schichten gelöst werden, nennt man auch *Abstraktion*.

Schichtung und Abstraktion finden in der Informatik in vielen Disziplinen, unter anderem in Computerarchitektur [Tan01], Rechnernetzen [Tan07], und Softwareengineering [Brü06], Anwendung und sind auch die Konzepte, die Verteilung und Virtualisierung ermöglichen. In den folgenden Unterkapiteln wird zuerst auf allgemeine Eigenschaften von geschichteten Systemen eingegangen (Kapitel 3.1.1) und anschließend auf die sich daraus ergebenden Einsatzmöglichkeiten (Kapitel 3.1.2).

3.1.1 Eigenschaften geschichteter Systeme

Dieser Abschnitt beschreibt Eigenschaften und Zusammenhänge geschichteter Systeme. Aus diesen Eigenschaften folgt die Möglichkeit zusätzliche Schichten in ein System einfügen zu können, ohne vorhandene Schichten verändern zu müssen. Diese Fähigkeit ist die Grundlage für Verteilung und Virtualisierung.

In geschichteten Systemen nutzt ein Teilsystem (Schicht) S_a nur Dienste des Teilsystems S_{a-1}. Ebenso werden die Dienste, die S_a bereit stellt, nur durch die Schicht S_{a+1} genutzt. Der wichtigste Aspekt hierbei ist, dass alle Schichten S_k mit $k > a+1$ nicht mit Schicht S_a interagieren. Deshalb hängen in einem geschichteten System die Funktionen der Schichten S_k nicht unmittelbar von der Schicht S_a ab.

Abgesehen von der speziellen Aufgabe, die eine Schicht erfüllt, nimmt sie Dienste in anspruch und stellt ihrerseits Dienste bereit. Anhand dieser drei Teilaufgaben lässt sich eine Schicht S_a in die drei Teilschichten ${S_a}^0$, ${S_a}^1$ und ${S_a}^2$ zerlegen. Die Aufgabe von ${S_a}^0$ ist die Nutzung der Dienste der Schicht S_{a-1}. ${S_a}^1$ ist die *Kernschicht* von S_a und erfüllt die eigentliche Aufgabe der Schicht S_a. Dazu nutzt sie Dienste oder Daten der vorausgehenden Schicht. Für die Interaktion mit der vorausgehenden Schicht ist die Teilschicht ${S_a}^0$ zuständig. Die Teilschicht ${S_a}^2$ stellt die Funktionen oder das Ergebnis von ${S_a}^1$ als Dienst zur Verfügung. Dieser Dienst wird von der Schicht S_{a+1} (genauer: der Teilschicht ${S_{a+1}}^0$) genutzt. Nimmt eine Schicht S_a Dienste einer Schicht S_b in Anspruch, so liegt S_a *über* S_b und S_b liegt *unter*

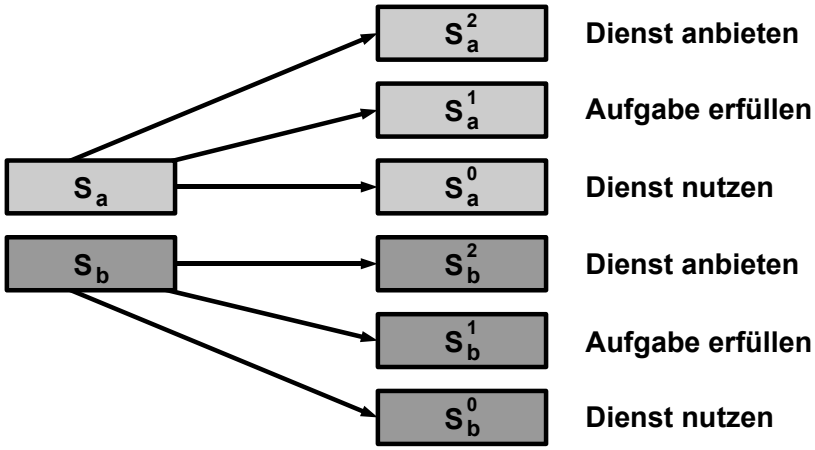


Abbildung 3.2: Zerlegung von zwei Schichten in sechs Teilschichten

S_a.

Abbildung 3.2 zeigt die Zerlegung der beiden übereinander liegenden Schichten S_a und S_b, in insgesamt sechs Teilschichten ${S_a}^2$ bis ${S_b}^0$. Die eigentliche Aufgabe von S_a wird in der Schicht ${S_a}^1$ erfüllt, während die eigentliche Aufgabe von S_b in der Schicht ${S_b}^1$ erfüllt wird. Diese Skizze verdeutlicht, dass die Erfüllung der speziellen Aufgaben der einzelnen Schichten nicht unmittelbar voneinander abhängen. In einem geschichteten System nutzt eine (Teil-) Schicht nur die Dienste der direkt darunter liegenden (Teil-) Schicht. Wie die Abbildung zeigt liegen zwischen den Kernschichten noch die beiden Teilschichten ${S_a}^0$ und ${S_b}^2$. In einem geschichteten System kann ${S_a}^1$ nicht direkt mit ${S_b}^1$ interagieren.

Bietet eine Schicht S_c genau dieselben Dienste wie S_b an (${S_b}^2 = {S_c}^2$), implementieren ${S_b}^2$ und ${S_c}^2$ dieselbe *Schnittstelle*. Ist die Schnittstelle von ${S_b}^2$ und ${S_c}^2$ identisch, so kann S_a sowohl mit S_b, als auch S_c interagieren. Die Schichten S_b und S_c können gegeneinander ausgetauscht werden, ohne dass S_a angepasst werden muss. Analog dazu kann S_a durch eine Schicht S_d ersetzt werden, wenn gilt ${S_d}^0 = {S_a}^0$.

Kombiniert man beide Eigenschaften, ermöglicht dies die Erweiterung eines geschichteten Systems. Dazu konstruiert man eine neue Schicht S_{neu}, so dass ${S_{neu}}^2 = {S_b}^2$ und ${S_{neu}}^0 = {S_a}^0$ gilt. Aufgrund der zuvor festgehaltenen Eigenschaften von geschichteten Systemen kann S_{neu} zwischen S_a und S_b eingefügt werden, ohne dass dadurch S_a oder S_b angepasst werden müssen. Durch die Wahl der Teilschichten ${S_{neu}}^0$ und ${S_{neu}}^2$ kann das Gesamtsystem um die Funktionalität von ${S_{neu}}^1$ erweitert werden.

Fasst man die spezielle Aufgabe, die in ${S_{neu}}^1$ erfüllt wird, als System auf, das in die Schichten S_f und S_g zerlegt werden kann, so können beliebig viele Schichten zwischen S_a und S_b eingefügt werden. Dadurch lässt sich ein vorhandenes System verändern und erweitern, ohne die Konzepte und Ideen der ursprünglichen Teilsysteme anpassen zu müssen. Steht S_{neu} stellvertretend für ein System, das aus den Schichten S_f und S_g besteht, so wurden S_f und S_g zu S_{neu} abstrahiert.

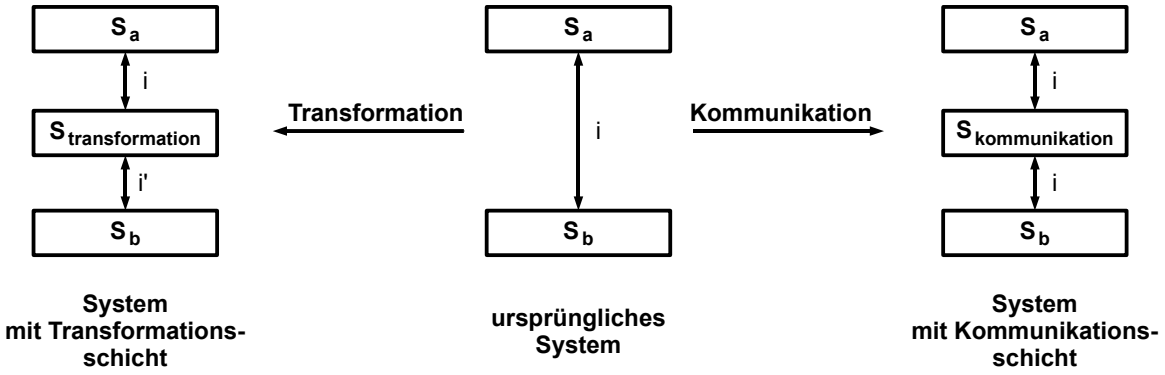


Abbildung 3.3: Unterschied zwischen Transformation und Kommunikation

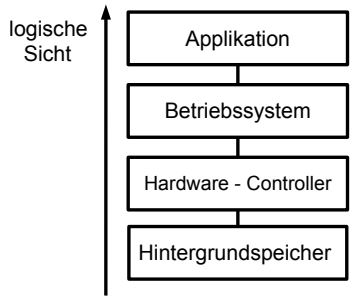
3.1.2 Einfügen zusätzlicher Schichten

Dieses Kapitel greif die im vorherigen Kapitel 3.1.1 beschriebene Möglichkeit der Erweiterung von geschichteten Systemen auf. Dabei werden die Einsatzmöglichkeiten *Transformation* und *Kommunikation* untersucht. Sowohl bei der Transformation als auch bei der Kommunikation werden zusätzliche Schichten in ein System eingefügt. Der Unterschied liegt in der Auswirkung auf die Interaktion der beiden Schichten zwischen denen die zusätzlichen Schichten eingefügt werden.

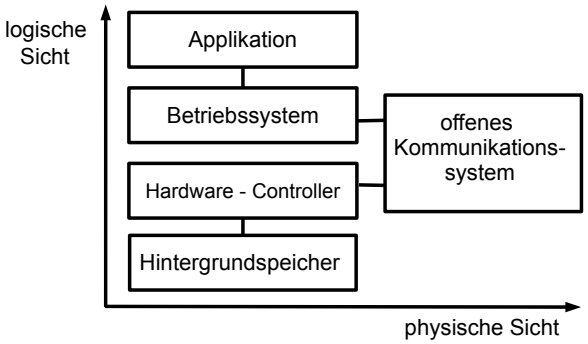
Abbildung 3.3 zeigt den Unterschied zwischen Transformation und Kommunikation. Im ursprünglichen System interagieren die Schichten S_a und S_b miteinander. Die Interaktion i ist als Pfeil zwischen den beiden Schichten eingezeichnet. Im ursprünglichen System nehmen S_a und S_b an derselben Interaktion teil. Eine Transformationschicht $S_{transformation}$ *interpretiert* die Interaktion i die von S_a ausgeht und beginnt mit der Schicht S_b eine neue Interaktion i'. S_a ist nach wie vor an der Interaktion i beteiligt, während S_b nun an der Interaktion i' beteiligt ist. S_a und S_b sind in diesem Fall keine Interaktionspartner mehr, da sie an unterschiedlichen Interaktionen beteiligt sind.

Im Gegensatz zum Verhalten von $S_{transformation}$ steht das Verhalten der Kommunikationsschicht, $S_{kommunikation}$. Die Interaktion i zwischen S_a und $S_{kommunikation}$ ist dieselbe Interaktion i, wie auch zwischen den Schichten $S_{kommunikation}$ und S_b. $S_{kommunikation}$ interpretiert die Interaktion i nicht, sondern *transportiert* sie zu S_b. Dadurch bleiben S_a und S_b Interaktionspartner, obwohl S_a und S_b nicht mehr direkt miteinander interagieren. Hier muss zwischen dem *logischen Datenfluss* und dem *physischen Datenfluss* unterschieden werden. Der logische Datenfluss ist die Interaktion i zwischen Schichten S_a und S_b. Unter dem physische Datenfluss versteht man eine Menge von Interaktionen, die nötig sind, um den logischen Datenfluss zu realisieren. In diesem Beispiel umfasst der logische Datenfluss die Interaktionen S_a mit $S_{kommunikation}$ und $S_{kommunikation}$ mit S_b.

Die Kommunikation wird in diesem Kapitel vereinfacht als einzelne Schicht modelliert, die im Gegensatz zu transformierenden Schichten, die Interaktion des ursprünglichen Systems nicht modifiziert. In aktuellen Ansätzen ist Kommunikation ein eigenes, vielschichtiges System, mit dem Zweck S_a und S_b zu entkoppeln und auf verschiedene Computer verteilen zu können. Die Kommunikationsschicht in Abbildung 3.3 nutzt die Dienste von S_b. Eine solche Kommunikationsschicht ist ein *geschlossenes System*. Aktuelle Ansätze zur Vertei-



(a) Logische Sicht, ohne Kommunikationssystem



(b) Logische uns physische Sicht, mit Kommunikationssystem

Abbildung 3.4: Vereinfachte Sicht auf den Datenfluss ohne Verteilung 3.4(a) und mit Verteilung 3.4(b)

lung von Systemen nutzen *offene Systeme* zur Interaktion. Offene Systeme übernehmen den Austausch von Informationen zwischen zwei Einheiten, unabhängig von den Computern, auf denen sie betrieben werden. „Offen“ nennt man diese Systeme, weil sie auf Standards basieren, so dass Implementierungen unterschiedlicher Hersteller zusammen verwendet werden können. Offene Systeme dienen lediglich dem Austausch von Informationen. Dabei werden die zu übertragenden Informationen nicht interpretiert [Hal96]. Offene Systeme lassen sich demnach nicht als Schicht zwischen zwei Teilsystemen S_a und S_b modellieren, da ein offenes System keine Dienste von S_b benutzt. Eine entsprechende Modellierung der Interaktion wird im nächsten Kapitel 3.1.3 eingeführt.

3.1.3 Serverinteraktion

Server in Rechenzentren interagieren nicht direkt mit Benutzern, sondern bieten Dienste über ein Netz an. Werden diese Dienste von Teilsystemen auf anderen Servern genutzt, um damit neue Dienste bereit zu stellen, handelt es sich um ein verteiltes System (siehe Kapitel 3.1). Mit den Erkenntnissen aus Kapitel 3.1.1 beschreibt Kapitel 3.1.2 wie ein bestehendes geschichtetes System durch das Einfügen von Transformations- oder Kommunikationsschichten verändert werden kann. Mit dem Einfügen der Kommunikationsschicht aus Kapitel 3.1.2 wird die Interaktion zwischen zwei Teilsystemen S_a und S_b aufgespaltet, um diese auf unterschiedliche Computer verteilen zu können. Nach diesem Prinzip wurden Server in Speicher- und Rechenknoten aufgeteilt. Hintergrundspeicher ist ein Teilsystem, welches auf einen anderen Computer ausgelagert wird. Das Speichernetz ist das Kommunikationssystem über das Speicher- und Rechenknoten miteinander interagieren.

Abbildung 3.4(a) zeigt eine vereinfachte Modellierung vom Zugriff einer Applikation auf Hintergrundspeicher. Die Interaktion der Schichten ist als Verbindungslinie zwischen diesen eingezeichnet. Der Zugriff der Applikation auf Hintergrundspeicher wird vom Betriebssystem in Steuersignale für den Hardware-Controller (HBA) umgewandelt, welcher direkt mit dem Hintergrundspeicher interagiert. Bei diesem Ablauf nutzt die Applikation das Dateisystem des Betriebssystems als Dienst, um auf Hintergrundspeicher zuzugreifen. Ein Treiber

dient dem Betriebssystem zur Interaktion mit dem HBA, der direkt auf Hintergrundspeicher zugreifen kann.

Wie in Kapitel 3.1.2 beschrieben, bleibt beim Einfügen einer Kommunikationsschicht der logische Datenfluss erhalten. Betrachtet man beim Zugriff einer Applikation auf Hintergrundspeicher lediglich logische Datenflüsse (*logische Sicht*), verändert sich das Modell des Servers aus Abbildung 3.4(a) nicht durch Verteilung. Eine Modellierung einzig durch die logische Sicht ist nicht zur Modellierung von Verteilung geeignet. Da das Modellieren von Verteilung als zusätzliche Schicht, nicht den Eigenschaften offener Systeme entspricht, ist auch dieses nicht geeignet um aktuelle Verteilungsansätze mit offenen Systemen zu modellieren. Um Verteilung mit offenen Systemen modellieren zu können, wird deshalb zusätzlich zur logischen Sicht die *physische Sicht* verwendet. Abbildung 3.4(b) beinhaltet neben der logischen Sicht die physische Sicht, die es ermöglicht den Datenfluss zwischen zwei Teilsystemen genauer darzustellen, ohne das ursprüngliche Modell verändern zu müssen. Höhere Schichten, wie die Applikation oder der Hintergrundspeicher, nutzen das Kommunikationssystem implizit durch andere Schichten. In diesem Beispiel durch das Betriebssystem und den Hardware-Controller. Die Möglichkeit der Kommunikation mit anderen Systemen wird der Applikation und dem Hintergrundspeicher vorenthalten. Da die Endpunkte der Interaktion nicht den vollständigen Datenfluss kontrollieren, spricht man von einem I/O-Kanal.

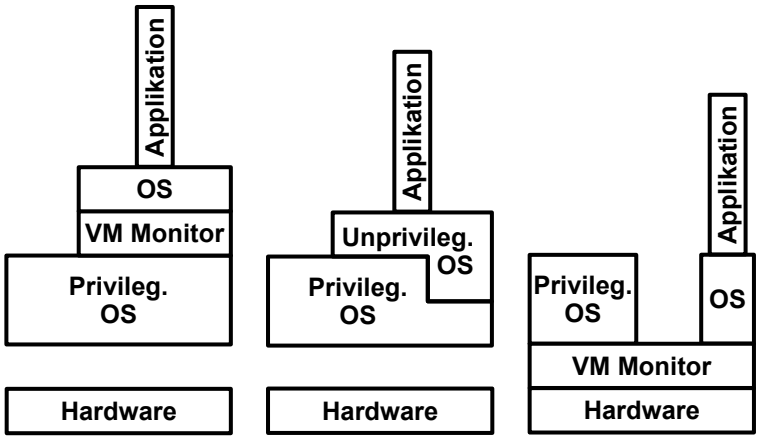
3.2 Virtualisierungskonzepte

Das vorausgegangene Kapitel 3.1 beschreibt mit seinen Unterkapiteln die Eigenschaften geschichteter Systeme und geht auf die Möglichkeit der Verteilung von geschichteten Systemen ein. Durch die so erlangte Möglichkeit Teilsysteme auszulagern, kann spezialisierte Hardware zur Erfüllung der Teilaufgaben eingesetzt werden. Die am weitesten verbreitete Anwendung dieses Konzepts ist das Auslagern von Hintergrundspeicher in Speicherknoten. Da Speicher- und Rechenknoten durch das SAN miteinander verbunden werden, entsteht ein neuer Berührungspunkt zwischen den Servern. Betreibt man zusätzlich mehrere Betriebssysteme auf einem einzelnen Server, so entstehen weitere Berührungspunkte.

Virtualisierung wird eingesetzt, um diese Berührungspunkte zu kontrollieren und dadurch Server zu konsolidieren. Ausgehend von dem Aufbau eines Rechenzentrums in Kapitel 2 kann zwischen Virtualisierung in den Bereichen Rechenknoten, Netze und Speicherknoten unterschieden werden. Die folgenden Kapitel untersuchen aktuelle Auffassungen von Virtualisierung in den einzelnen Bereichen. Ziel ist es Vorgänge einheitlich zu beschreiben und eine einheitliche Definition von Virtualisierung zu finden. Für jeden der drei Teilbereiche werden zu beginn Konzepte für Virtualisierung beschrieben und anschließend Produkte vorgestellt, die Konzepte implementieren.

3.2.1 Virtualisierung in Rechenknoten

Nach der Definition aus Kapitel 2.1 werden auf Rechenknoten Betriebssysteme und Applikationen betrieben. Rechenknoten interagieren miteinander über das LAN und sind über das SAN an Speicherknoten angebunden. Virtualisierung in Rechenknoten bezeichnet den gleichzeitigen Betrieb mehrerer unabhängiger Betriebssysteme auf demselben Host [SPF+06]. Die Möglichkeiten der Virtualisierung in Rechenknoten wird maßgeblich durch die verwendete Plattform bestimmt. In dieser Arbeit dient Intel's x86-Plattform als Grundlage, da diese am



(a) Vollvirtualisierung mit Emulation (b) OS-Virtualisierung (c) Paravirtualisierung

Abbildung 3.5: Skizzen unterschiedlicher Virtualisierungsansätze, nach [LDgFK08]

weitesten verbreitet ist. Werden andere Plattformen eingesetzt, können andere Rahmenbedingungen für den gleichzeitigen Betrieb mehrerer Betriebssysteme gelten.

Im Folgenden gelte die folgende Definition eines Betriebssystems:

> `Ein Betriebssystem ist die Gesamtheit aller Software, die für den anwendungsunabhängigen Betrieb des Rechners notwendig ist [Bra97].`

Zu einem Betriebssystem (OS) gehört unter anderem der *Betriebssystemkern*, der Organisationsprogramme zur Speicher-, Prozessor-, Geräte- und Netzverwaltung enthält. Alle Programme müssen für den Zugriff auf Hardware die Dienste des Betriebssystemkerns nutzen [Bra97]. Der direkte Zugriff auf Hardware wird durch ein Sicherheitssystem der Rechnerarchitektur geschützt. Für spezielle *privilegierte* Funktionen lässt dieses Sicherheitssystem nur die Benutzung durch einen einzigen, privilegierten Betriebssystemkern zu [Tan01]. Der gleichzeitige Betrieb mehrerer Betriebssysteme ist also nur möglich, wenn alle Aufrufe privilegierter Funktionen durch den einen privilegierten Betriebssystemkern kanalisiert werden.

Das Kanalisieren von privilegierten Zugriffen entspricht dem Einfügen einer zusätzlichen Schicht, um das Problem des privilegierten Zugriffs für Betriebssysteme zu lösen. Das Problem wird durch Abstraktion in einer zusätzlichen Schicht gelöst, so dass jedes Betriebssystem privilegierte Befehle aufrufen kann (vgl. Kapitel 3.1). Der folgende Abschnitt stellt drei Konzepte zum Kanalisieren von privilegierten Zugriffen vor. Die Implementierungen, die zu Techniken zur Virtualisierung von I/O-Kanälen kombiniert werden, nutzen meist Mischformen der vorgestellten Methoden.

Um Aufrufe privilegierter Hardwarefunktionen zu kanalisieren stehen drei verschiedene Ansätze zur Verfügung:

- Virtualisierung aller Hardware-Komponenten (Abbildung 3.5(a))
- OS-Virtualisierung (Abbildung 3.5(b))
- Paravirtualisierung (Abbildung 3.5(c))

Bei der Virtualisierung aller Hardware-Komponenten wird für jedes Betriebssystem ein vollständiger Rechner in Software nachgebildet, so dass das Betriebssystem innerhalb des *virtuellen* Rechners betrieben wird. Sämtliche Zugriffe auf die Architektur des virtuellen Rechners sind Ereignisse der Software und werden zu Aufrufen für den privilegierte Betriebssystemkern weiterverarbeitet. Aufrufe werden wenn möglich ohne Modifikation weitergereicht, um die Geschwindigkeit zu erhöhen. Dieses Konzept wird auch *Vollvirtualisierung* genannt.

Vollvirtualisierung wird häufig auch zusammen mit oder gar als Synonym für *Emulation* genannt. Der Begriff Emulation soll darauf hinweisen, dass eine spezielle Funktion vollständig in Software umgesetzt wurde. Implementierungen der Vollvirtualisierung beinhalten auch Ansätze, um Daten zwischen virtuellem Server und Hardware direkt und ohne Verarbeitung weiterzuleiten, um die Leistung des Gesamtsystems zu erhöhen. Die Leistung eines Systems setzt sich zusammen aus der weitergeleiteten Datenmenge, dem *Volumen*, und der Anzahl der Interaktionen zwischen virtuellen Servern und der Hardware, die *Transaktionsrate*. Die direkte Weiterleitung von Daten ist keine echte Emulation, da die Daten nicht verarbeitet werden. Vollvirtualisierung ist demnach eine Teilmenge von Emulation, da für den virtuellen Server immer noch eine Hardware-Schnittstelle Emuliert wird.

Bei der zweiten Technik wird der Betriebssystemkern der nicht privilegierten Betriebssysteme modifiziert, so dass dieser nicht mehr mit Hardware interagieren kann. Der modifizierte Betriebssystemkern ist darauf ausgelegt als Applikation auf dem privilegierten Betriebssystem ausgeführt zu werden, wodurch kein virtueller Rechner emuliert werden muss. Ein derart modifizierter Betriebssystemkern greift nicht mehr auf Hardwarefunktionen zu, sondern auf die Dienste des privilegierten Betriebssystemkerns.

Paravirtualisierung setzt einen Hypervisor als privilegiertes System ein. Ein Hypervisor wird häufig auch als *Virtual Machine Monitor* (VMM) bezeichnet. Die einzige Aufgabe des VMM ist die Virtualisierung der zugrunde liegenden Hardware. Im Gegensatz zu den anderen Techniken generiert der VMM keine Zugriffe, ausgehend von den Aufrufen der nicht privilegierten Betriebssysteme, sondern leitet die Aufrufe weiter (vgl. Transformation und Kommunikation in Kapitel 3.1.2). Der VMM ist nicht für den effektiven Hardwarezugriff zuständig, sondern lediglich für die Koordinierung der Hardwarezugriffe durch die Betriebssysteme. Dabei kann es notwendig sein Teile des nichtprivilegierten Betriebssystem an den VMM anzupassen [Lin06].

3.2.1.1 Produkte

Eine Virtualisierungsschicht, die durchgängig Emulation einsetzt, heißt Emulator. Häufig genannte Beispiele für einen Emulator sind Qemu [qem09] und Bochs [boc09]. Solche Lösungen emulieren auch Hintergrundspeicher, wodurch der Zugriff eines emulierten Servers auf Speichereinheiten vollständig kontrollierbar ist. Die Emulationssoftware wandelt jeden Zugriff des virtuellen Servers, in Zugriffe für das privilegierte Betriebssystem um. Das kostet Rechenleistung und Zeit, weshalb Emulatoren nicht so effizient sind, wie andere Lösungen.

Im Gegensatz zur Emulation steht die Paravirtualisierung aus Abbildung 3.5(c). Hier wird den virtuellen Servern im Wesentlichen die Schnittstellen der physischen Hardware-Komponenten zur Verfügung gestellt und der VMM koordiniert die Zugriffe. Der VMM emuliert die Schnittstellen mehrfach, wodurch jeder Hardwarezugriff der virtuellen Server vom VMM kontrolliert werden kann. Der VMM erweitert die Hardware-Plattform um die Möglichkeit der Virtualisierung. Bei manchen Implementierungen, zum Beispiel Xen [xen09], erfordert dies kleine Anpassungen am Betriebssystem des virtuellen Servers [Lin06]. Da bei

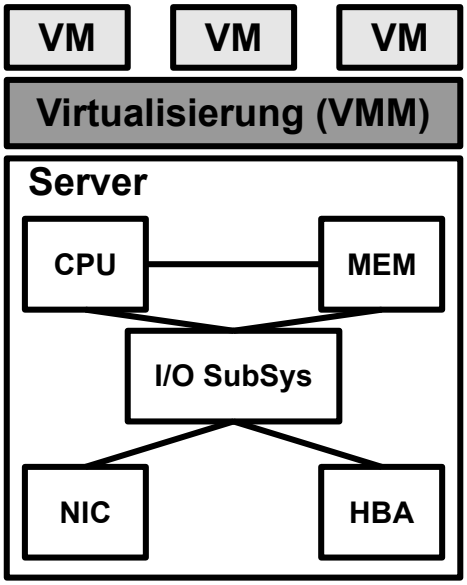


Abbildung 3.6: Detailiertere Ansicht eines Servers mit VMM

Paravirtualisierung nicht ganze Komponenten, sondern lediglich Schnittstellen emuliert werden, ist Paravirtualisierung leistungsfähiger als Emulation.

Die meisten kommerziellen Anbieter für Server-Virtualisierung setzen auf eine Kombination aus Voll- und Paravirtualisierung. Die Kombination soll ein Maximum an Leistung erzielen, ohne Veränderungen an Betriebssystemen zu benötigen. Die wichtigsten kommerziellen Ansätze zur Server-Virtualisierung sind der ESX-Server von VMware [vmw09] und HyperV von Microsoft [mic09].

3.2.1.2 Virtualisierung von Hardware-Schnittstellen

Unter Hardware-Virtualisierung versteht man Ansätze, die Funktionen der Virtualisierungsschicht in Hardware-Komponenten verlagern, um so den VMM zu entlasten. Dies ist die logische Fortsetzung von Server-Virtualisierung und wird in diesem Kapitel vorgestellt.

Aktuelle Hardware liefert meist mehr Leistung, als ein einzelner Server benötigt. Eine höhere Auslastung der Hardware kann erreicht werden, indem man die leistungsstarke Hardware in externen *I/O-Servern* konsolidiert, zu virtualisierten Einheiten abstrahiert und diese als Dienst mehreren Servern zugänglich macht. Bei dieser Art der I/O-Virtualisierung unterscheidet man zwischen „*Single Root I/O-Virtualisierung*“ (SR-IOV) und „*Multi Root I/O-Virtualisierung*“ (MR-IOV) [PCI07, PCI08]. Im Verlauf dieses Kapitels werden diese beiden Ausprägungen anhand der PCI-SIG Spezifikationen untersucht.

In vorausgegangen Kapiteln, wurde ein Server immer als eine Hardware-Schicht betrachtet, ein mit einem Betriebssystem interagiert (zum Beispiel Kapitel 3.1.2). Um Hardware-Virtualisierung zu analysieren ist diese Darstellung nicht ausreichend, da bei dieser vereinheitlichten Darstellung die Komponenten, die in I/O-Server ausgelagert werden sollen, nicht enthalten sind. Abbildung 3.6 zeigt virtuelle Server (VMs), die Virtualisierungsschicht aus Kapitel 3.2.1 und ein Modell eines physischen Servers. Die wesentlichen Teilsysteme

des Servers sind die CPU, der Hauptspeicher (MEM, Abkürzung aus dem Englischen „Memory“), das I/O-Teilsystem (I/O-Subsystem) und die I/O-Hardware. Die Verbindungslinien repräsentieren Interaktionsmöglichkeiten der Teilsysteme. Die I/O-Hardware wird in der Abbildung durch ein NIC einen HBA repräsentiert. Wie die Abbildung zeigt ist die Interaktion mit I/O-Hardware nur durch das I/O-Subsystem möglich. Als Technologie für das I/O-Subsystem wird heute meist PCI, in der aktuellen Version PCI Express, eingesetzt. Dieses Kapitel geht genauer auf PCI Express und dessen Erweiterungen für SR-IOV und MR-IOV ein.

PCI Express

PCI Express ist ein Standard, der die Anbindung von Hardware an Computer spezifiziert [PCI06]. Für diesen Standard wurden die Erweiterungen PCI SR-IOV [PCI07] und PCI MR-IOV [PCI08] spezifiziert. So kann mit Hardware, die diese Standards implementiert ein *I/O-Netz* (siehe Kapitel 3.2.1.2) aufgebaut werden.

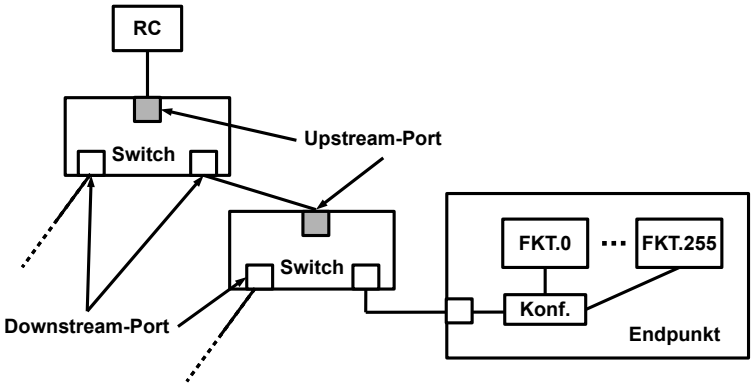
Das im Kapitel 3.2.1.1 eingeführte Modell eines physischen Servers wird in Abbildung 3.6 dargestellt. Die Abbildung zeigt, dass CPU, Hauptspeicher und I/O-Subsystem miteinander verbunden sind und so miteinander interagieren können. Das dabei verwendete Kommunikationssystem ist nicht standardisiert und von Hersteller zu Hersteller unterschiedlich. Allerdings unterstützen heutige Systeme immer auch die Anbindung von Hardware durch standardisierte Techniken der Peripheral Component Interconnect Special Interest Group (PCI-SIG). Die standardisierte Technologie ist der PCI-Bus. Mit dem PCI-Bus wird Hardware über ein standardisiertes Verfahren an das herstellerspezifische I/O-Subsystem angebunden. In früheren Versionen wurden die Geräte durch ein Bussystem verbunden waren. Die aktuelle Version des PCI-Buses, PCI Express (PCIe), ist kein Bussystem mehr, sondern die Geräte sind in einer Sterntopologie hierarchisch organisiert.

Für das Betriebssystem und die Hardware-Komponenten zeigt sich PCIe immer noch als Bussystem, um bestehende Konzepte und Implementierungen weiterhin verwenden zu können. Die Wurzel der PCIe Hierarchie ist der Root Complex (RC). Er ist die Schnittstelle zwischen der herstellerspezifischen Architektur und dem PCI-Bus. Seine Aufgabe ist das Umsetzen der Daten und Befehle vom Betriebssystem in PCIe Steuer- und Datenpakete und die Übermittlung der Pakete an die angeschlossenen Geräte.

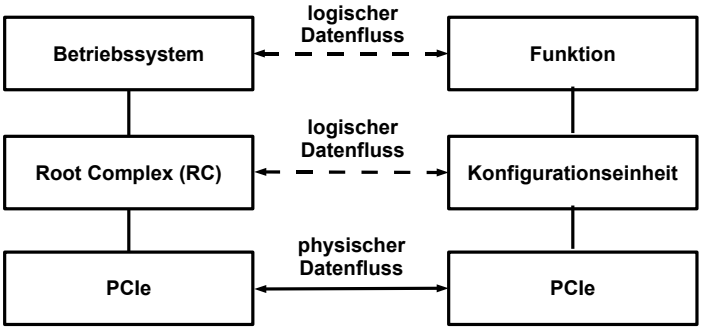
Die PCIe Hierarchie

Bevor auf die SR-IOV und MR-IOV Erweiterungen eingegangen wird, beschreibt dieses Kapitel den prinzipiellen Aufbau einer PCIe Hierarchie, ohne Erweiterungen. Der PCI-Bus ist auch ein geschichtetes Kommunikationssystem. So ist die Interaktion der Betriebssysteme mit den Geräten als logischer Datenfluss unabhängig von den darunter liegenden Schichten. Dies ermöglicht das Hinzufügen von Techniken wie SR-IOV und MR-IOV, ohne dass vorhandene Betriebssysteme oder VMMs angepasst werden müssen (vgl. Kapitel 3.1).

Abbildung 3.7(a) zeigt einen Ausschnitt einer PCIe Hierarchie. Jeder Knoten im Baum der PCIe Hierarchie ist entweder ein Switch, ein Endpunkt oder der Root Complex (RC). Jeder Switch hat genau einen Upstream-Port und mindestens einen Downstream-Port. Es sind immer je ein Upstream- mit einem Downstream-Port verbunden. Ein PCIe Endpunkt kann bis zu 256 *Funktionen* bereitstellen. Jede Funktion eines Endpunkts erscheint dem Betriebssystem als ein eigenständiges Gerät. Interagiert das Betriebssystem mit einem Gerät,



(a) Ausschnitt einer PCIe-Hierarchie



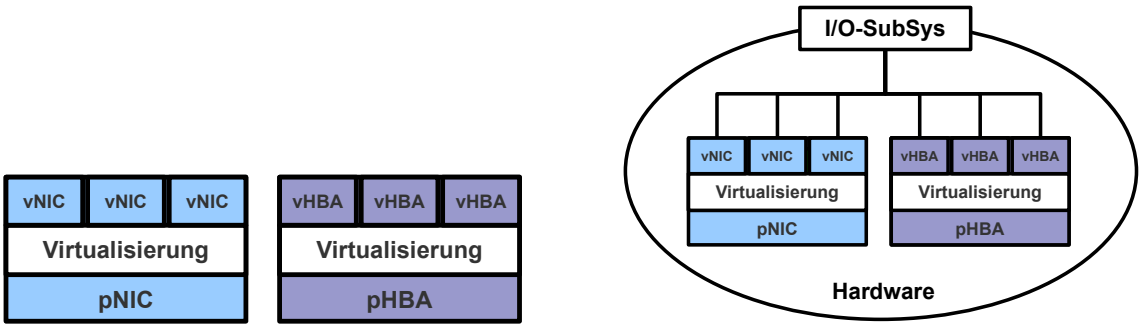
(b) Schichtung des PCI-Buses

Abbildung 3.7: Topologische Anordnung von PCIe-Komponenten und Schichtung der Kommunikation über den PCI-Bus

so interagiert es mit einer Funktion, die über den PCI-Bus zur Verfügung gestellt wurde. Jeder Endpunkt verfügt über eine Konfigurationseinheit. Die Konfigurationseinheit ist das Gegenstück zum RC. Der RC ermöglicht dem Server den Zugang zum PCI-Bus und die Konfigurationseinheit ermöglicht den Funktionen den Zugang zum PCI-Bus. Diese Schichtung wird in Abbildung 3.7(b) dargestellt. Es existiert zwei logischer Datenflüsse: einer zwischen Betriebssystem und Funktion und ein weiterer zwischen RC und Konfigurationseinheit. Die mit „PCIe“ beschrifteten Schichten übernehmen die physische Datenübertragung.

SR-IOV

SR-IOV ist die Fähigkeit einer Hardware-Komponente sich selbst zu virtualisieren. Dies wird in Abbildung 3.8(a) skizziert. Auf die physische I/O-Hardware (pNIC und pHBA) wird eine Virtualisierungsschicht gesetzt, die wie ein VMM mehrere virtuelle Instanzen der Schnittstelle zur Verfügung stellt. Die virtuellen Instanzen sind in der Abbildung mit vNIC oder vHBA benannt. Abbildung 3.8(b) zeigt je drei virtuelle NICs und HBAs am I/O-Subsystem. Der VMM aus Abbildung 3.5(c) muss dadurch nicht mehr den Zugriff mehrerer virtueller Server auf eine Hardware-Komponente koordinieren. Statt dessen kann er jedem virtuellen Server eine eigene Hardware-Komponente zuweisen. Die Koordination der Zugriffe übernimmt die I/O-Hardware selbst. Dies entlastet den VMM.



(a) Selbstvirtualisierte I/O-Hardware mit eigener Virtualisierungsschicht

(b) Jede virtuelle Instanz kann einzeln über das I/O-Subsystem benutzt werden

Abbildung 3.8: Virtualisierung einer NIC und eines HBAs mit SR-IOV

Die PCI SR-IOV Spezifikation sieht vor, dass eine Hardware-Komponente über eine eigene Virtualisierungsschicht verfügt, die es mehreren *virtuellen Funktionen* erlaubt die Ressourcen der selben Funktion (vgl. Kapitel 3.2.1.2) zu nutzen. Im Folgenden wird eine Funktion „physische Funktion“ genannt, um Verwechslungen zu vermeiden. Im Gegensatz zur Server-Virtualisierung wird nicht mit Emulation durch Software gearbeitet. Die virtuellen Funktionen sind Hardware-Bausteine, implementieren jedoch keine ganze physische Funktion, sondern lediglich den Teil der Hardware, der für die Interaktion mit dem Betriebssystem zuständig ist (Buffers, Queues, etc.). Ansonsten teilen sich virtuelle Funktionen immer die Ressourcen einer physischen Funktion. Durch die gemeinsame Benutzung von Ressourcen, sind die Mehrkosten für eine weitere virtuelle Funktion auf der Hardware-Komponente geringer als für eine weitere physische Funktion. Die Anzahl genaue virtueller Funktionen, die ein Endpunkt bereitstellt ist implementierungsspezifisch. Nach Spezifikation sind jedoch höchstens 128 virtuelle Funktionen pro physischer Funktion möglich. Die Spezifikation sieht außerdem vor, dass virtuelle Funktionen zwischen zwei physischen Funktionen migriert werden können. Dazu enthält die Konfigurationseinheit Methoden, mit denen der Inhalt der Buffer etc. einer virtuellen Funktion ausgelesen und in eine andere virtuelle Funktion geschrieben werden können. Mit SR-IOV kann die Leistung eines vorhandenen Systems gesteigert werden, da virtuelle Funktionen Aufgaben übernehmen, die vorher ein VMM oder ein Emulator erfüllen musste.

MR-IOV

MR-IOV ist die Fähigkeit einer Hardware-Komponente an mehrere I/O-Subsysteme angeschlossen zu werden [PCI08]. Diese Fähigkeit ermöglicht es I/O-Hardware in spezialisierte I/O-Server auszulagern. Durch diese Erweiterung des I/O-Subsystems können die I/O-Subsysteme der physischen Server miteinander verbunden werden um die Ressourcen des selben I/O-Servers zu nutzen. Dadurch entsteht ein neues Netz zwischen den Servern, ähnlich wie durch die Aufteilung von Server in Speicher- und Rechenknoten. Zusammen mit SR-IOV kann mit MR-IOV eine Infrastruktur wie in Abbildung 3.9 aufgebaut werden. Die I/O-Hardware ist nicht mehr Bestandteil eines Servers, sondern wird in externen I/O-Servern konsolidiert. Die virtuellen Instanzen der Schnittstellen können von unterschiedlichen phy-

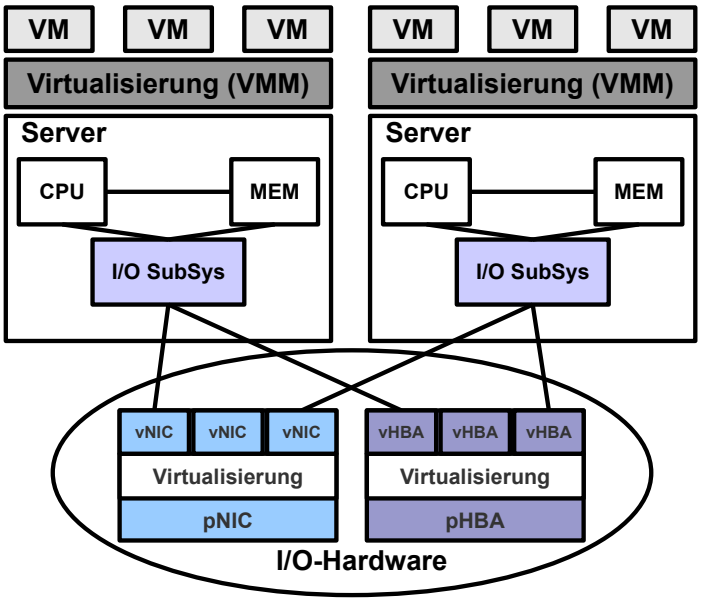


Abbildung 3.9: Durch den Einsatz von SR-IOV und MR-IOV kann I/O-Hardware durch mehrere physische Server gleichzeitig benutzt werden

sischen Rechnern genutzt werden, wodurch die Auslastung der Hardware erhöht wird. Ein Vorteil dieser Technologie für die Betreiber von Rechenzentren ist, dass ein physischer Server nicht mehr über ein NIC und einen HBA verfügen muss, sondern nurnoch eine einzelne Komponente, die den Zugang zum *I/O-Netz* ermöglicht. Dadurch muss ein physischer Server mit weniger Hardware ausgerüstet werden und auch die Anzahl der physischen Verbindungen wird dadurch reduziert, wodurch Kosten gespart werden können.

Die Spezifikation für Multi Root I/O Virtualization ist die jüngste Erweiterung zum PCIe-Standard. Für den topologischen Aufbau ergeben sich hierbei einige Veränderungen gegenüber der Darstellungen aus Kapitel 3.2.1.2. In einer PCIe Hierarchie kann nun mehr als ein RC vorhanden sein, wodurch die Hierarchie kein Baum mehr ist, da es keine Wurzel mehr gibt. Eine andere Veränderung ist, dass ein Switch über mehr als einen Upstream-Port verfügen kann, denn dies ist notwendig, um einen Switch an mehr als einen RC anzuschließen. Die MR-IOV Spezifikation geht hier noch einen Schritt weiter. PCIe Switche mir MR-IOV Fähigkeiten haben keine expliziten Upstream- und Downstream-Ports mehr. Um das bestehende Kommunikationskonzept des PCI-Buses beibehalten zu können, werden virtuelle Hierarchien (virtual hierarchy, VH) erstellt und jedem RC eine eigene VH zugewiesen. Die Switche Leiten die Datenpakete dann entsprechend dem RC, von dem das Paket stammt, weiter. Die logischen Datenflüsse aus Abbildung 3.7(b) können dadurch erhalten werden. Des weiteren sieht die Spezifikation vor, dass auch Hardware die nicht die MR-IOV Spezifikation implementiert zusammen mit MR-IOV Switchen verwendet werden kann. Ist dies der Fall, muss der darüber liegende Switch dies berücksichtigen und darf diese Komponente nur einer einzigen VH zuordnen.

Zur Zeit gibt es noch keine Hardware auf dem Markt, die MR-IOV 1.0 implementiert. Dies hat mehrere Gründe. Die Version 1.0 dieser Spezifikation ist sehr neu und die Entwicklung entsprechender Produkte lässt noch auf sich warten. Ein großes Problem dabei ist, dass

MR-IOV keine Technologie ist, die in einem einzelnen Gerät implementiert wird. Zum Einen benötigt man PCIe Switches, die MR-IOV-Fähigkeiten implementieren und zum Anderen impliziert diese Technologie eine Veränderung in der Infrastruktur. Man benötigt I/O-Servern und andere phyische Verbindungen zwischen den Servern. Die Firmen halten sich mit ihren Fortschritten bei dieser Entwicklung sehr bedeckt. Außer Neterion nenn niemand MR-IOV in den Produktbeschreibungen.

3.2.2 Virtualisierung in Netzen

Netze sind meist durch viele Schichten aufgebaut. Schichtung selbst beinhaltet bereits ein hohes Maß an Abstraktion, da tiefer liegende Schichten implizit benutzt werden (vgl. Kapitel 3.1.1). Ein Versuch die Virtualisierung von Netzen mit Abstraktion, analog zur Virtualisierung in Speicherknoten zu beschreiben würde daher in einem Modell ähnlich dem ISO-OSI-Modell enden, ohne eine konkrete Identifizierung der Virtualisierung, als Erweiterung der Möglichkeiten des ISO-OSI-Modells. Nach Tanenbaum findet sich Virtualisierung in diesem Bereich in „virtual private network“ (VPN) und „virtual local area network“ (VLAN) [Tan07].

VPNs sind eine Klasse von Netzen, die als logische Netze auf einer Netzinfrastruktur aufgebracht werden [HD09]. In der Anwendung bezeichnen VPNs meist Ansäzte, um getrennte Netze in einem logischen Netz zu vereinen. In dieser Arbeit werden VPNs betrachtet, die eine Infrastruktur in isolierte, logische Einhieten segmentieren. Diese Unterart von VPNs heißt VLAN und wurde bereits vom 802-Komitee standardisiert [Tan07]. Durch die Standardisierung werden VLANs wird meistens mit dem entsprechenden IEEE-Standard-802.1Q assoziiert [iee06].

Ein Netz kann auf mehrere Arten segmentiert werden. Man kann mehrere IP-Subnetze innerhalb einer Infrastruktur betreiben und so das Netz auf IP-Ebene segmentieren. Dies entspricht einer Segmentierung des Netzes auf Schicht 3 des ISO-OSI-Referenzmodells. Der zuvor erwähnte IEEE-Standard-802.1Q sieht vor, dass ein Rahmen vor der Übertragung mit einer Nummer markiert wird. Diese Markierung identifiziert das Netzsegment, zu dem der Rahmen gehört und so kann das Netz auf Schicht 2 des ISO-OSI-Modells segmentiert werden. Mit entsprechenden Paketfiltern können analog dazu Netze anhand von anderen Eigenschaften und anderen Protokollen und OSI-Schichten segmentiert werden. Segmentierung alleine ist also keine hinreichende Eigenschaft von Virtualisierung in Netzen, da die Möglichkeit der Segmentierung bereits eine Eigenschaft einer geschichteten Kommunikation ist.

Nach Tanenbaum zeichnen sich VLANs dadurch aus, dass die Segmentierung innerhalb der OSI-Schicht 2 geschieht. Kommunikationssysteme, die nach dem ISO-OSI-Referenzmodell entworfen wurden, sind geschichtete Systeme. In geschichteten Systemen nutzen alle Schichten implizit die Funktionen der darunterliegenden Schichten (vgl. Kapitel 3.1). Demnach gilt die Segmentierung implizit für alle Schichten, die über der Schicht liegen, die die Segmentierung umsetzt. Eine Implementierung der Segmentierung in OSI-Schicht 2 bewirkt also, dass es keine Möglichkeit der Kommunikation zwischen zwei Systemen gibt, die die Segmentierung ignorieren kann.

Für die Virtualisierung von I/O-Kanälen ist außerdem wichtig, dass für die Segmentierung keine Abstraktion benutzt wird. Dadurch sind den Switchen des Netzes alle Informationen der Segmentierung zugänglich und so können Switche die Segmentierung durchsetzen, ohne auf entsprechende Funktionen in den Kommunikationsendpunkten angewiesen zu sein (vgl. Anforderung 2 auf Seite 19). Für die Server ist die Infrastruktur bereits in mehrere VLANs

segmentiert.

Dieses Verständnis von Virtualisierung deckt sich mit der aus dem Englischen übersetzten Definition für Virtualisierung:

> `Virtualisierung ist eine Methode physischer Systeme sich wie mehrere, voneinander unabhängige, logische Systeme zu verhalten [Cis08c].`

Dieses Kapitel untersucht die Fähigkeiten von Netzen, Kommunikationspfade zu kontrollieren. Der Fokus liegt in diesem Zusammenhang auf der Segmentierung der Infrastruktur in voneinander isolierte VLANs, sowie auf der Identifizierung von Kommunikationsendpunkten. Da Kommunikationspfade nur innerhalb eines VLANs aufgebaut werden können nennt man dies*Pfadvirtualisierung*. Bei der Virtualisierung in Bezug auf die Identifizierung von Kommunikationsendpunkten spricht man von *Adressvirtualisierung*. Diese Methoden sind nach der Server-Virtualisierung (Kapitel 3.2.1) die zweite Klasse von Technologien, die zur Virtualisierung von I/O-Kanälen eingesetzt werden. Aktuell werden für Netze in Rechenzentren die Technologien Fibre Channel, Ethernet InfiniBand verwendet. Aus diesem Grund werden in dieser Arbeit jene drei Technologien betrachtet.

3.2.2.1 Fibre Channel

Die Fibre Channel Technologie wird von der Fibre Channel Industry Association (FCIA) verwaltet. Die technischen Spezifikationen von Fibre Channel (FC) und deren Veröffentlichung übernimmt das Technical Comittee T11 [t11]. Einige dieser Spezifikationen wurden als Standard vom American National Standards Institute (ANSI) angenommen. Zur FC Technologie gehören mehrere Schichten und andere Dienste, so dass die Datenübertragung zwischen zwei FC Geräten gesichert wird und Eigenschaften der Verbindung gesteuert und überprüft werden können [fco].

Ein FC-Netz kann auf unterschiedliche Art und Weise aufgebaut werden. Das kleinste FC-Netz besteht aus zwei Rechnern die mit einer direkten physischen Verbindung (Punkt-zu-Punkt Verbindung) verbunden sind. Komplexere Methoden sind Arbitrated Loop (FC-AL) [T1195] und Switched Fabric (FC-SW) [T1108c], mit denen mehrere Rechner miteinander verbunden werden können. Allen Ausprägungen gemein ist, dass Rahmen immer nur von einem Gerät zum Nächsten übertragen werden. Über Anzahl und Art der Weiterleitung entscheidet jedes Gerät einzeln. Die Entscheidungsfindung wird maßgeblich von den Fibre Channel Generic Services (FC-GS) [T1108a] beeinflusst. Die FC-GS sind eine Menge von Diensten, von denen die Endpunkte und Switche Informationen und Konfigurationen beziehen.

Die Spezifikation für FC-AL beschreibt ein Netz mit Ringtopologie, während FC-SW ein Netz mit Sterntopologie spezifiziert. Die Interaktion zwischen zwei Endpunkten (Nodes) ist unabhängig von der vorliegenden Topologie, da die Kommunikation von FC-Geräten ein geschichtetes System ist. Die FC Kommunikationstechnologie besteht aus den fünf Schichten FC-0, FC-1, FC-2, FC-3 und FC-4.

FC-4 , die oberste Schicht des FC Protokollturms, fungiert als Anpassungsschicht. Die Aufgabe dieser Schicht ist es die Kommunikation von Protokollen höherer Schichten (upper layer protocols), zum Beispiel Version 4 des Internet Protokolls (IPv4), auf eine FC-Kommunikation abzubilden. Dazu muss jedes Protokoll ein Gegenstück in der FC-4 Schicht haben. Dieses Gegenstück muss die Eigenschaften einer Kommunikation, wie

zum Beispiel die Adressierung, auf Eigenschaften der FC-Kommunikation abbilden. Für das Beispiel IPv4 existiert der IETF Draft IPv4FC, der die Umsetzung von IPv4 auf FC spezifiziert [uCC04].

FC-3 stellt allgemeine Funktionen zur Verfügung. Diese können von allen Protokollen, unabhängig von der verwendeten Topologie, gleichermaßen benutzt werden. Ein Beispiel für solche Dienste ist Verschlüsselung nach der Fibre Channel Security Protocols-Spezifikation (FC-SP) [T1108b]. Zusätzlich werden auf dieser Schicht Verwaltungsdienste bereitgestellt. Zu erwähnen sind hier der *Directory-Server* und der *Management-Server*. Der Directory-Server dient unter anderem der Namensauflösung von WWNs zu N_Port IDs, welche für die Weiterleitung von Rahmen benötigt werden. Der Management-Server verwaltet Konfigurationen für die Schichten FC-1 und FC-2. Diese Dienste werden in der FC-GS-Spezifikation definiert [T1108a].

FC-2 ist zuständig für die Rahmenbildung und Rahmenübertragung in einem FC-Netz. Die Spezifikation für Verbindungsdienste, Fibre Channel Link Services (FC-LS), definiert die Dienste, die FC-2 für die Schicht FC-3 zur Verfügung stellt [T1106]. Die Spezifikation für Fibre Channel Fabric Services (FC-FS) [T1109] umfasst die möglichen Zusammensetzungen eines Rahmens und die Möglichkeiten zur Adressierung. FC-AL und FC-SW benutzen diese Spezifikationen um eine Kommunikation zwischen zwei Nodes zu ermöglichen.

FC-1 übernimmt die Datenkodierung. Dies ist eine extra Schicht, da FC unterschiedliche Kodierungen und Rahmenbegrenzungen zulässt, um das Nutzungsprofil des Übertragungsmediums anpassen zu können. Die Möglichkeiten zur Datenkodierung sind Teil der Fibre Channel Physical and Signaling Interface-Spezifikation (FC-PH) [T1194].

FC-0 ist die phyische Übertragung von Daten über ein Übertragungsmedium. Die physischen Eigenschaften von Übertragungsmedien, Sendern und Empfängern ist ein weiterer Teil Teil von FC-PH [T1194].

Um FC vielseitig einsetzen zu können, enthält eine Spezifikation meist mehrere Alternativen für die Lösung eines bestimmten Problems oder einer bestimmten Aufgabe. Welche Alternativen eine Komponente implementiert bleibt den Herstellern dabei in weiten Teilen freigestellt. FC sieht vor, dass sich ein Gerät immer am Netz anmelden muss, sobald eine physische Verbindung besteht. Während der Anmeldung werden die Verbindungsparameter für die physische Verbindung vereinbart [T1196]. Die gewählten Einstellungen werden auch durch entsprechende Einträge im Directory Server oder Management-Server beeinflusst. Als identifizierendes Merkmal für FC-HBAs bei der Netzanmeldung wird der *N_Port Name*, eine 8 Byte lange, weltweit eindeutige Zahl, benutzt [T1106]. Der N_Port Name ist auch als World Wide Name (WWN) bekannt [T1008]. Jeder Port eines HBAs (N_Port) hat einen eigenen, bei der Herstellung festgelegten, WWN. Während des Anmeldevorgangs wird einem N_Port eine *N_Port ID* zugewiesen, die für alle weitere Kommunikation innerhalb des Netzes benutzt wird.

Um zwischen verschiedenen virtullen Servern zu unterscheiden, muss jeder virtuelle Server eine eigene N_Port ID zugewiesen bekommen. Dies ist keine Aufgabe, die alleine von der Virtualisierungsschicht wahrgenommen werden kann, da die N_Port ID bei der Anmeldung des N_Ports an das Netz generiert wird. Um eine eigene N_Port ID zu bekommen, müsste sich jeder virtuelle Server einzeln am Netz anmelden. Da sich der HBA selbst am Netz anmeldet

und nicht der Server, bzw. das Betriebssystem, muss das Netz das mehrfache Anmelden eines N_Ports unterstützen. Die entsprechende Technologie heißt *N_Port ID Virtualization* (NPIV) [T1102]. Zur Identifikation von virtuellen Servern verfügen diese über einen WWN, der von der Virtualisierungsschicht zugewiesen wird.

Zur Aufteilung eines Netzes in isolierte, logische Teilnetze sieht FC die Segmentierung des Netzes in *Zonen* (Zones) vor [T1108a]. Die Zonenzuweisung geschieht über den Management-Server. Die Spezifikation sieht eine Segmentierung in Zonen anhand von N_Port Namen, N_Port IDs oder anderen Adressierungen, entsprechend FC-FS, vor. FC verfügt über diverse Mechanismen zur Steuerung der Datenflüsse innerhalb der Zonen. Einzelne Datenflüsse werden anhand der Adressierung identifiziert und kontrolliert. Jeder Switch kann über Regeln verfügen, die vorgeben auf welche Weise eine bestimmte Verbindung behandelt werden soll. Die Regeln können sich auf eine einzelne physische Verbindung zwischen Geräten beziehen, oder auf die Kommunikationsendpunkte. Mit diesem Mechanismus kann FC Zusicherungen über die Verfügbarkeit und Verbindungsqualität implementieren. Auch die Wahl der physischen Verbindungen für die Weiterleitung von Rahmen kann dadurch beeinflusst werden. Dies ermöglicht eine präzise Steuerung der Datenflüsse in einem FC-Netz.

Fibre Channel ist im Wesentlichen eine Implementierung der Schichten 1 und 2 des ISO-OSI-Referenzmodells. Aufgrund der vielen Eigenschaften und Möglichkeiten von FC, wie die präzise Steuerung der Datenflüsse zwischen zwei Endpunkten, kann mit FC zusätzlich Schicht 3 des OSI-Modells implementiert werden.

3.2.2.2 Ethernet

Die bekannteste und älteste Netztechnologie ist Ethernet. Der Standard, der die heute gültige Datenübertragung für Ethernet beschreibt ist seit 1983 der IEEE-802.3-Standard [Tan07]. Dieser wurde im Lauf der Zeit immer weiter entwickelt, um höhere Übertragungsraten zu erreichen. Zur Zeit ist die größtmögliche, standardisierte Übertragungsrate für Ethernet 10 Gigabit pro Sekunde (10GbE). Der entsprechende Standard ist trägt die Bezeichnung 802.3ae.

Der Hauptaspekt bei der Entwicklung des Ethernet war es, die Kommunikation von Computern, die das selbe Übertragungsmedium benutzen, zu koordinieren [IEE05]. In der aktuellen Fassung der Spezifikation des 10 GbE wird dieser Aspekt nicht mehr berücksichtigt. Statt dessen muss jeder Rechner über eine eigene physische Verbindung zum Switch verfügen. Dies ist schon seit früheren Versionen gängige Praxis, da bei einer eigenen Verbindung und Kollisionsdomäne pro Rechner das Netz effizienter genutzt werden kann [Tan07]. Der ursprüngliche Aufbau eines Ethernet-Rahmens zur Datenübertragung ist dabei erhalten geblieben.

Ethernet dient lediglich der Sicherung einer physischen Verbindung, entsprechend der Medium-Access-Control Teilschicht der Schicht 2 des ISO-OSI-Referenzmodells. Weitere Funktionen werden bei einem auf Ethernet basierenden Netz in höheren Schichten, die nicht mehr ausschließlich zur Ethernet-Technologie gehören implementiert.

Ein Ethernet-Rahmen besteht im Wesentlichen aus einer Zieladresse, einer Quelladresse, einem Typfeld, Nutzdaten und einer Prüfsumme. Ziel- und Quelladresse dienen der Identifizierung der Verbindungsendpunkte. Bei Ethernet werden Endpunkte durch eine eindeutige 48 Bit *MAC-Adresse* identifiziert. Das Typfeld speichert die Art der übertragenen Nutzdaten. Die Prüfsumme dient der Fehlererkennung. Ein Switch speichert die MAC-Adressen der angeschlossenen Rechner und kann eingehende Rahmen anhand der Zieladresse weiterleiten. Erreicht ein Rahmen mit einer unbekannten Zieladresse den Switch, leitet er den Rahmen

über alle Verbindungen weiter. Außerdem können bei Ethernet Rahmen gezielt über alle Verbindungen weitergeleitet werden (Broadcast). Dazu wird ein Rahmen an die spezielle MAC-Adresse FF:FF:FF:FF:FF:FF (alle Bits auf 1) adressiert und verschickt. Empfängt ein Switch einen Rahmen, der so adressiert wurde, leitet er den rahmen über alle Verbindungen weiter. Daraus entsteht die Anforderung, dass ein Ethernet-Netz kreisfrei sein muss, denn wird ein Rahmen per Broadcast in einem nicht kreisfreien Netz verschickt, so wird ein Switch einen Broadcast-Rahmen immer wieder empfangen und verschicken. Dadurch wird über die Zeit die komplette zur Verfügung stehende Übertragungsrate des Netzes durch Broadcast-Rahmen aufgebraucht und es können keine weiteren Daten übertragen werden [Tan07]. Um Kreise zu vermeiden, wird häufig das Spannung-Tree-Protokoll (STP) oder dessen Weiterentwicklung, das Rapid-Spanning-Tree-Protokoll eingesetzt. Mit diesem Protokoll werden Kreise erkannt und physische Verbindungen deaktiviert, um Kreise zu unterbrechen. STP ist Teil des IEEE-802.1D-Standards [iee04].

Ein Rahmen kann nach der Quelladresse mit einem „VLAN-Tag“, entsprechend dem IEEE-802.1Q-Standard, markiert werden. Mit dieser Markierung werden bei Ethernet Netzsegmente unterschieden, unabhängig von Ziel- und Quelladresse. Im VLAN-Tag sind 12 Bit für die VLAN-Kennung vorgesehen. Ein einzelner Switch kann also zwischen 2^{12} Netzsegmenten unterscheiden. Ein Ethernet-Switch implementiert VLANs, indem er Rahmen nur an die Ports weiterleitet, denen ebenfalls eine entsprechende VLAN-Kennung zugewiesen wurde [Tan07].

Um zwischen unterschiedlichen virtuellen Servern zu unterscheiden, muss eine Virtualisierungsschicht also jedem virtuellen Server eine eigene MAC-Adresse zuweisen, um den Datenverkehr trennen zu können. Durch den Einsatz von VLANs lässt sich ein Ethernet Netz in getrennte logische Netze aufteilen.

3.2.2.3 InfiniBand

Die dritte Kommunikationstechnologie, die in diesem Kapitel untersucht wird, ist InfiniBand (IB). Die Verwaltung und Spezifikation von IB ist die Aufgabe der InfiniBand Trade Association [inf09]. Dieser Abschnitt basiert auf der InfiniBand Architecture Spezifikation [inf07], in der die Übertragung und der zugehörige Protokollstapel dieser Technologie festgelegt werden.

Die InfiniBand Architektur (IBA) wurde entworfen um *Prozessorplattformen*, *I/O-Plattformen* und *I/O-Geräte* miteinander zu verbinden [inf07]. Eine Prozessorplattform ist eine Einheit aus CPUs und Hauptspeicher, während I/O-Geräte HBAs entsprechen. I/O-Plattformen sind nicht näher spezifiziert. Abbildung 3.10 zeigt, wie die IBA das I/O-Subsystem (vgl. Kapitel 3.2.1.2), das LAN und das SAN vereint. Das resultierende Einheitsnetz heißt *System Area Network*. Die Verbindung zwischen zwei Endpunkten heißt in der IBA *Channel*. Ein IB-HBA, der auf Prozessorplattformen eingesetzt wird, nennt man deshalb *Host Channel Adapter* (HCA). Der Channel Adapter (CA) anderer Komponenten heißt *Target Channel Adapter* (TCA). In der Spezifikation unterscheiden sich TCA und HCA in den Anforderungen welche Funktionen implementiert werden müssen. Viele Funktionen, die für einen HCA zwingend erforderlich sind, wie zum Beispiel einige Dienste der Transportschicht, sind optional für einen TCA.

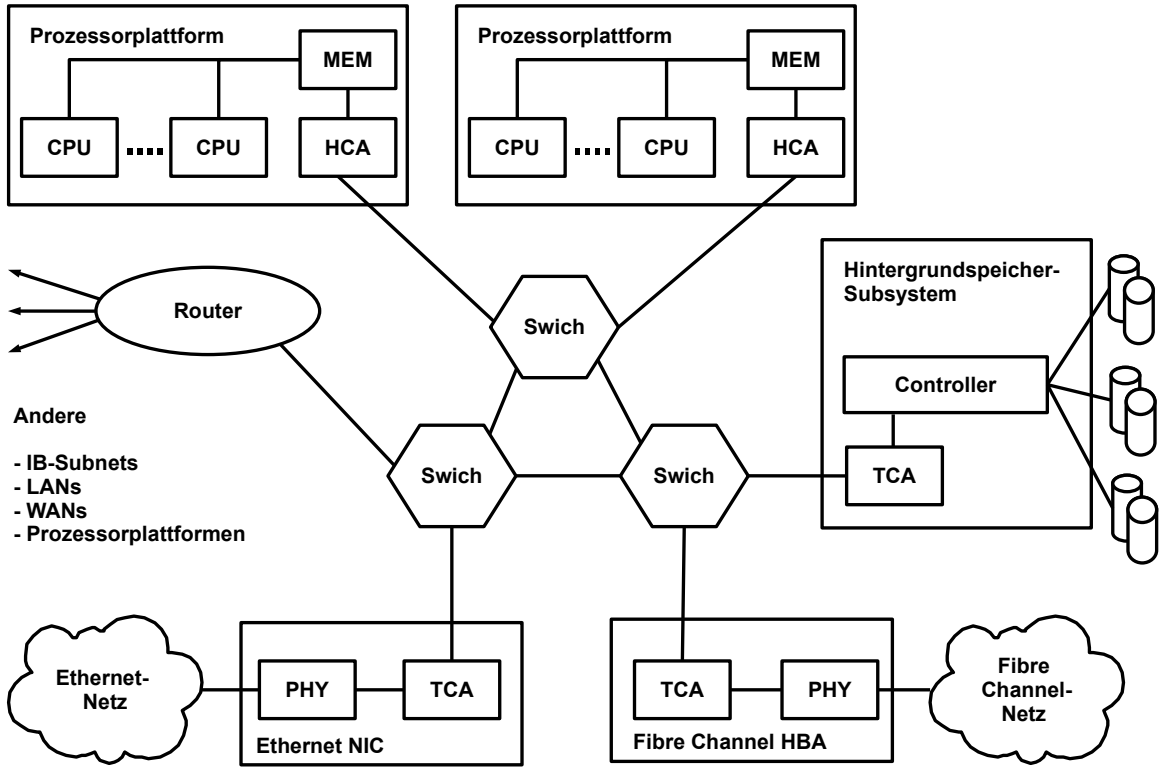


Abbildung 3.10: Skizzierung der IBA nach [inf07]

IB-Sicherungsschicht

Ein IB-Netz besteht aus einer Menge von physischen Punkt-zu-Punkt Verbindungen. Auf der Sicherungsschicht des ISO-OSI-Referenzmodells ist ein Endpunkt ein *Port*. Jeder Port muss sich am Netz anmelden, sobald eine physische Verbindung besteht, analog zu FC. Anstatt eines WWN wird jeder Port durch einen *Globally Unique Identifier* (GUID) eindeutig identifiziert. Bei der Anmeldung wird jedem Port eine lokale Adresse (Local Identifier, LID), zugewiesen, die für jede weitere Kommunikation verwendet wird, ebenfalls analog zu FC. Die Ports eines Switches erhalten keine Adresse, da diese nicht adressiert werden. Die einzige Ausnahme ist ein Management-Port pro Switch, über den ein Switch verwaltet wird. Meldet sich ein Port am Netz an, so wird er Mitglied eines *Subnets*. Ein IB-Rahmen beginnt immer mit dem *Local Routing Header* (LRH), mit dem ein Port jeden anderen Port des Subnets adressieren kann. Der LRH enthält auch ein Feld *Service Level*, das die Bezeichnung der Dienstgüteklasse der Verbindung enthält.

IB-Vermittlungsschicht

Als weiterer Teil der IBA, zusätzlich zu einer Sicherungsschicht, ist eine Vermittlungsschicht spezifiziert. Jeder Port erhält außer einer lokalen Adresse, die den Port innerhalb des Subnets adressierbar macht, eine globale Adresse (global Identifier, GID), die den Port von jedem anderen Port aus adressierbar macht. Die globale Adresse hat das Format einer Internet Protocol Version 6-Adresse (IPv6-Adresse) und besteht aus einem *Subnet Prefix* und dem

GUID des Ports. Während jeder Endpunkt immer eine IPv6-Adresse in dieser Form hat, so können einem Endpunkt optional weitere globale Adressen zugewiesen werden. Soll ein Port außerhalb des eigenen Subnets adressiert werden, so muss direkt hinter dem LRH ein *Global Routing Header* (GRH) in den Rahmen eingefügt werden. Ein Vergleich des Aufbaus des GRH mit dem Aufbau eines IPv6-Headers nach der IPv6-Spezifikation in RFC 2460 [DH98] zeigt, dass ein GRH genau ein IPv6-Header ist. Dadurch ist es möglich IPv6-Datenverkehr nativ durch ein IB-Netz zu routen.

Subnets werden über *Router* miteinander verbunden. Um einen Port global zu adressieren muss als Zieladresse im LRH ein Router eingetragen werden und im GRH die globalen IPv6-Adressen der Quell- und Zielports. Ein Router empfängt einen Rahmen, ersetzt den LRH und leitet den Rahmen entsprechend seiner Routingtabelle weiter. Bei der Ersetzung des LRH adressiert der Router den Rahmen entweder an den endgültigen Port, sofern er diesen erreichen kann, oder den nächsten Router. Die IBA Spezifikation legt nicht fest wie Routingtabellen generiert und zwischen den Routern ausgetauscht werden sollen. Statt dessen wird auf IPv6 verwiesen, für das mehrere Routingprotokolle verfügbar sind.

IB-Transportschicht

Die IBA enthält auch eine Spezifikation für eine Transportschicht. Diese erlaubt es direkt den Prozess oder Controller hinter dem Zielport zu adressieren. In der Transportschicht kann ein HCA kann zusätzlich sicher stellen, dass Daten erfolgreich übertragen wurden und gegebenenfalls eine erneute Übertragung anfordern. Aufbauend auf diesen Fähigkeiten der Transportschicht, sieht die IBA den Betriebsmodus *Remote Direct Memory Access* (RDMA) für Channels vor. RDMA erlaubt es dem Endpunkt, der die Interaktion beginnt (Initiator), direkt lesend oder schreibend auf den Hauptspeicher des entfernten Endpunkts zuzugreifen, ohne den Umweg über CPU und Betriebssystem des entfernten Endpunkts. Ebenso können empfangene Daten, direkt in den eigenen Hauptspeicher des entsprechenden Prozesses geschrieben werden. Das Betriebssystem des Initiators muss lediglich die Datenübertragung anstoßen. Die effektive Datenübertragung übernimmt vollständig der HCA und benachrichtigt das Betriebssystem, sobald die Datenübertragung abgeschlossen ist. Diese Methode der Datenübertragung entlastet zum Einen die CPU und das Betriebssystem und zum Anderen sind weniger Zwischenschritte für den Datenaustausch notwendig. Durch diese beiden Effekte wird mit dem Einsatz von RDMA eine hohe Transaktionsrate und eine niedrige Verzögerung erreicht. Um diese Vorteile voll ausnutzen zu können werden Protokolle festgelegt, die für die Datenübertragung auf RDMA setzen. Im Anhang der IBA Spezifikation sind SDP und iSER geführt. SDP ist ein Byte-Strom orientiertes Transportprotokoll, welches das Verhalten von TCP imitiert und RDMA zur Datenübertragung nutzt. Das iSER Protokoll kombiniert die Vorteile der RDMA-Datenübertragung mit iSCSI.

Außer RDMA bietet die IB-Transportschicht auch den verbindungsorientierten oder verbindungslosen Dienst zur Datenstromübertragung zwischen zwei Kommunikationsendpunkten, entsprechend der Schicht 4 des ISO-OSI-Referenzmodells. Zusätzlich zu diesen drei Dienstformen sind auch zwei *RAW*-Dienste möglich, die nicht alle möglichen Schichten der IBA benutzen. Die RAW-Dienste sind *Raw IPv6 Datagram* und *Raw Ethertype Datagram*. Ein Rahmen zur Übertragung von einem Raw IPv6 Datagram, besteht aus einem LRH, einem GRH, Nutzdaten und einer Checksumme. Da der GRH genau dem IPv6 Header entspricht, kann über eine solche Verbindung IPv6 nativ über IB betrieben werden. In diesem Fall werden nur die IB-Implementierungen der Schichten 1-3 des ISO-OSI-Referenzmodells verwen-

det. Ein Raw Ethertype Datagram besteht aus einem LRH, einem *Raw Header* (RWH), Nutzdaten und einer Checksumme. Der RWH enthält lediglich ein Typfeld, analog zu dem Typfeld eines Ethernet-Rahmens. Da kein GRH eingesetzt wird, ist es mit dem Raw Ethernet Datagram nur möglich Daten zu einem Zielport innerhalb des eigenen Subnets zu übertragen. Bei einer Datenübertragung im Raw Ethertype Format werden nur die IB-Implementierungen der Schichten 1-2 des ISO-OSI-Referenzmodells verwendet.

Im Anhang der Spezifikation der InfiniBand Architektur befindet sich eine Beschreibung für ein *Dienstgüterahmenwerk* (Quality of Service framework). Dieses sieht eine extra Management-Komponente vor (QoS-Manager), die Datenströme überwacht. Der QoS-Manager wertet die gesammelten Daten aus und kann anhand eines Regelwerks die Konfiguration der Switche anpassen. Die durch den QoS-Manager veränderbaren Konfigurationen umfassen die Bandbreitenzuweisung und die Weiterleitungsregeln. Bei einer vollständigen Umsetzung der Spezifikation können Zusicherungen sowohl für physische Verbindungen, als auch für die Verbindung zwischen zwei Endpunkten festgelegt werden.

Die Segmentierung einer IB-Infrastruktur in mehrere logische Infrastrukturen nennt man *Partitionierung*. Dazu wird ein Rahmen mit einer *Partitionskennung* (Partition Key) markiert. Jeder Switch und jeder Router kann die Partitionskennung auslesen und bei der Weiterleitung des Rahmens berücksichtigen. Die Partitionskennung wird in einem speziell dafür vorgesehenen Feld im *Base Transport Header* (BTH) geführt. Wie der Name schon andeutet, implementiert IB die Segmentierung von Infrastrukturen in der Transportschicht. Da dieses Feld immer im BTH vorhanden ist, gibt es eine Standardpartition, zu der jeder Endpunkt gehört.

3.2.2.4 I/O-Konsolidierung

Nach Server- und Speicherkonsolidierung ist I/O-Konsolidierung der nächste Schritt in der Entwicklung von Technologien zum Einsatz in Rechenzentren. Bisher waren LAN und SAN zwei physisch getrennte Netze im Rechenzentrum. Als *I/O-Konsolidierung* bezeichnet man Ansätze um LAN und SAN auf dem selben phyischen Medium zu vereinen.

Die bisher typische Technologie für das SAN ist Fibre Channel. Da FC eine sehr präzise Steuerung der Datenflüsse ermöglicht, kann ein FC Netz gut an das Nutzungsprofil des SANs angepasst werden. Außerdem konnte mit FC lange Zeit eine höhere Übertragungs- und Transaktionsrate erzielt werden als mit Ethernet. Da für das LAN kein spezielles Nutzungsprofil erstellt werden kann, benötigte man hier keine so präzise Steuerungsmöglichkeit wie im SAN. Die typische für das LAN eingesetzte Technologie ist daher das kostengünstigere Ethernet.

FCoE ist das erste verfügbare Produkt, welches diesem neuen Trend bei der Vernetzung im Rechenzentrum folgt. Durch die Kapselung von FC-Rahmen ermöglicht FCoE ein auf Ethernet basiertes SAN. Dadurch kann für LAN und SAN dieselbe Technologie eingesetzt und so dieselbe physische Infrastruktur verwendet werden. LAN und SAN werden auf eine Infrastruktur konsolidiert. Der Vorteil hier bei ist, dass weniger Hardware angeschafft werden muss und der Platz- und Stromverbrauch gegenüber einem Rechenzentrum mit physisch getrenntem LAN und SAN geringer ist. Da FCoE weder definitiv zu Ethernet noch zu FC gehört, wird FCoE in diesem Kapitel näher beschrieben.

Die MR-IOV PCI Express Technologie aus Kapitel 3.2.1.2 ist ebenfalls ein Ansatz zur I/O-Konsolidierung. Allerdings wird hier nicht der gesamte Datenverkehr von Endpunkt zu Endpunkt konsolidiert, wie bei FCoE, sondern nur bis zu einem I/O-Server. Der I/O-Server

teilt den Datenverkehr schließlich auf unterschiedliche Netze auf.

Betrachtet man nur die technischen Fähigkeiten, so eignet sich auch FC zur I/O-Konsolidierung. Wie bereits zu Beginn des Kapitels erwähnt, wird FC bereits häufig als SAN Technologie eingesetzt. Im LAN wird meistens IP als Vermittlungsprotokoll eingesetzt und RFC 4338 spezifiziert den Betrieb von von IP auf Fibre Channel [DCN06]. Dadurch könnten „normale“ LAN-Verbindungen auch über FC aufgebaut werden. Da Fibre Channel Hardware jedoch um einiges teurer ist als Ethernet, existiert ein solches Betriebszenario nur selten.

Fibre Channel over Ethernet Mit Fibre channel over Ethernet (FCoE) wird versucht das Beste von Ethernet und Fibre Channel zu vereinen. Die hohe Präzision bei der Kontrolle von Datenflüssen und die dadurch ermöglichten Zusicherungen bezüglich der Verbindungsqualität von FC sollen mit dem kosteneffizienten Ethernet verschmolzen werden. Bei FCoE werden FC-Rahmen als Nutzdaten in Ethernet-Rahmen übertragen. Dieses Vorgehen nennt man *Kapselung*. Ein Switch, der sowohl Zugang zu einem Ethernet Netz als auch zu einem FC Netz hat, kann den FC-Rahmen extrahieren und per FC weiterleiten. Dies ermöglicht es Ethernet und FC Komponenten gleichzeitig zu nutzen. Der Vorteil dabei ist, dass beim Wechsel vom vergleichsweise teuren FC auf das billigere Ethernet nicht sofort die gesamte Infrastruktur ausgetauscht werden muss, sondern die FC Komponenten weiterhin verwendet werden können. Da der FC Anteil von FCoE vollständig in den Ethernet-Rahmen eingebettet wird, kann jeder Ethernet Switch die Rahmen weiterleiten. Deswegen können hier auch vorhandene Komponenten weiterbenutzt werden und die Umstellung auf FCoE kann schrittweise erfolgen.

Um die Zusicherungen von FC auch bei Ethernet gewährleisten zu können, müssen die Ethernet Switche verschiedene FCoE-Funktionen implementieren. Unbedingt notwendig sind jedoch lediglich Jumbo Frames, welche bereits von einer breiten Basis an Hardware unterstützt wird und so kann man FCoE auch ohne den vollen Funktionsumfang betreiben. Für eine vollständige Umsetzung der Zusicherungen von FC auf Ethernet sind eine Reihe von Erweiterungen nötig:

Jumbo Frames FC-Rahmen können über 2kB groß werden. Ein normaler Ethernet-Rahmen ist lediglich 1517 Bytes groß. Jumbo Frames können bis zu 16kB groß sein, wodurch es ermöglicht wird FC-Rahmen vollständig zu Kapseln, ohne diese auf mehrere Ethernet-Rahmen aufteilen zu müssen.

IEEE 802.1Qau Gewöhnliche Ethernet Geräte verwerfen Pakete, wenn sie nicht schnell genug weitergeleitet bzw. verarbeitet werden. Dies steht stark im Konflikt mit den FC Zusicherungen. IEEE 802.1Qau (Congestion Notification) ist ein Mechanismus um Datenstau kommunizieren zu können. Wird ein Endpunkt über einen Datenstau informiert, so kann er seine Übertragungsrate verlangsamen und so das Netz entlasten und den Datenstau vermeiden.

IEEE 802.1Qaz Mit diesem Standard bekommt Ethernet die Möglichkeit Bandbreite dynamisch zuzuweisen um immer die maximale Übertragungsrate einer physischen Verbindung nutzen zu können. Die Klassifikation und Steuerung der einzelnen Klassen erfolgt über diverse andere IEEE 802.1QaX Standards.

IEEE 802.1Qbb Dieser Standard ist als PFC (priority-based flow control) bekannt. Dies ist die Kernkomponente die mittels der anderen Standards dafür sorgt, dass auch bei starkem Verkehrsaufkommen die Übertragung gewährleistet bleibt.

DBX Das Data Center Bridging Capability Exchange Protocol (kurz: DBX) wird zur Konfiguration und Verwaltung von FCoE Geräten eingesetzt [Int08c]. Mit DBX können Geräte synchronisiert und Verbindungen konfiguriert werden, unabhängig von der eingesetzten Technologie. Dieses Protokoll wird bei FCoE zur Steuerung und Kontrolle von Datenflüssen zwischen FC-Komponenten und Ethernet-Komponenten eingesetzt.

3.2.3 Virtualisierung in Speicherknoten

Microsoft definiert Virtualisierung in Speicherknoten als ein Hilfsmittel um mehrere physische Speichergeräte als eine einzelne, nummerierte, logische Speichereinheit (LUN) betrachten zu können [Mic08]. Nach dieser Definition ist Virtualisierung eine Methode zur Abstraktion. Die wahre Anzahl und Beschaffenheit der vorhandenen Speichergeräte bleibt den höheren Schichten verborgen, da diese nur logische Einheiten nutzen, die unter Umständen aus mehreren Speichergeräten bestehen. Ein Beispiel für eine solche Abstraktion ist ein RAID-Verbund. Die Funktionalität aktueller Speicherknoten ist umfangreicher, als das Zusammenfassen von Festplatten zu einem RAID-Verbund. Heutige Lösungen sind auf Schichtung und Verteilung ausgelegt. Das heißt, dass aktuell verfügbare Speicherknoten in mehrere Schichten segmentiert wurden, um unabhängig vom Rest des Rechenzentrums Skalierungseffekte erzielen zu können und um den tatsächlichen Zugriff auf Hintergrundspeicher präziser steuern zu können.

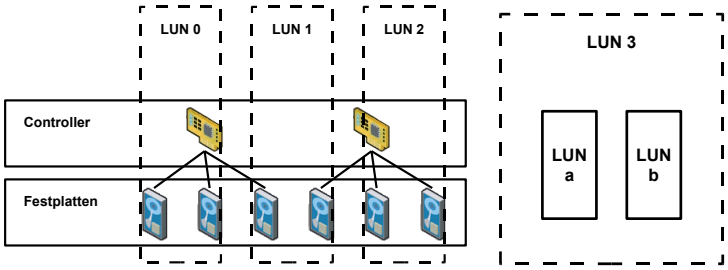


Abbildung 3.11: LUNs können beliebig weit von der Hardware abstrahiert werden

Abbildung 3.11 illustriert das Konzept eines Speicherknotens. LUNs sind der Dienst, den ein Speicherknoten anbietet. Tatsächlich verbirgt sich hinter einer LUN mindestens ein Controller, der eine Festplatte kontrolliert, also genau der Teil eines Servers, der bei der Aufteilung von Servern in Speicher- und Rechenknoten ausgelagert wurde. Das Konzept des RAID-Verbunds, bei dem eine LUN auf mehrere Festplatten des selben Controllers verteilt werden, ist nicht neu und schon länger verfügbar [Tan01]. Die erweiterte Schichtung von Speicherknoten ermöglicht es LUNs über die Festplatten mehrere Controller zu verteilen, auch über die Grenzen eines Speicherknotens hinweg. Für die Organisation und Verteilung von LUNs gibt es mehrere Strategien, die ineinander verschachtelt werden können. Die resultierenden Möglichkeiten und deren Vor- und Nachteile sind aufgrund ihres Umfangs und der teilweise sehr herstellerspezifischen Techniken nicht Teil dieser Arbeit. Der wesentliche Aspekt der Virtualisierung in Speicherknoten ist, dass die Eigenschaften einer LUN nicht direkt von der Größe und der Übertragungsrate einer bestimmten Festplatte abhängen. Eine LUN wird als Einheit über den Dienst eines Speicherknotens zur Verfügung gestellt.

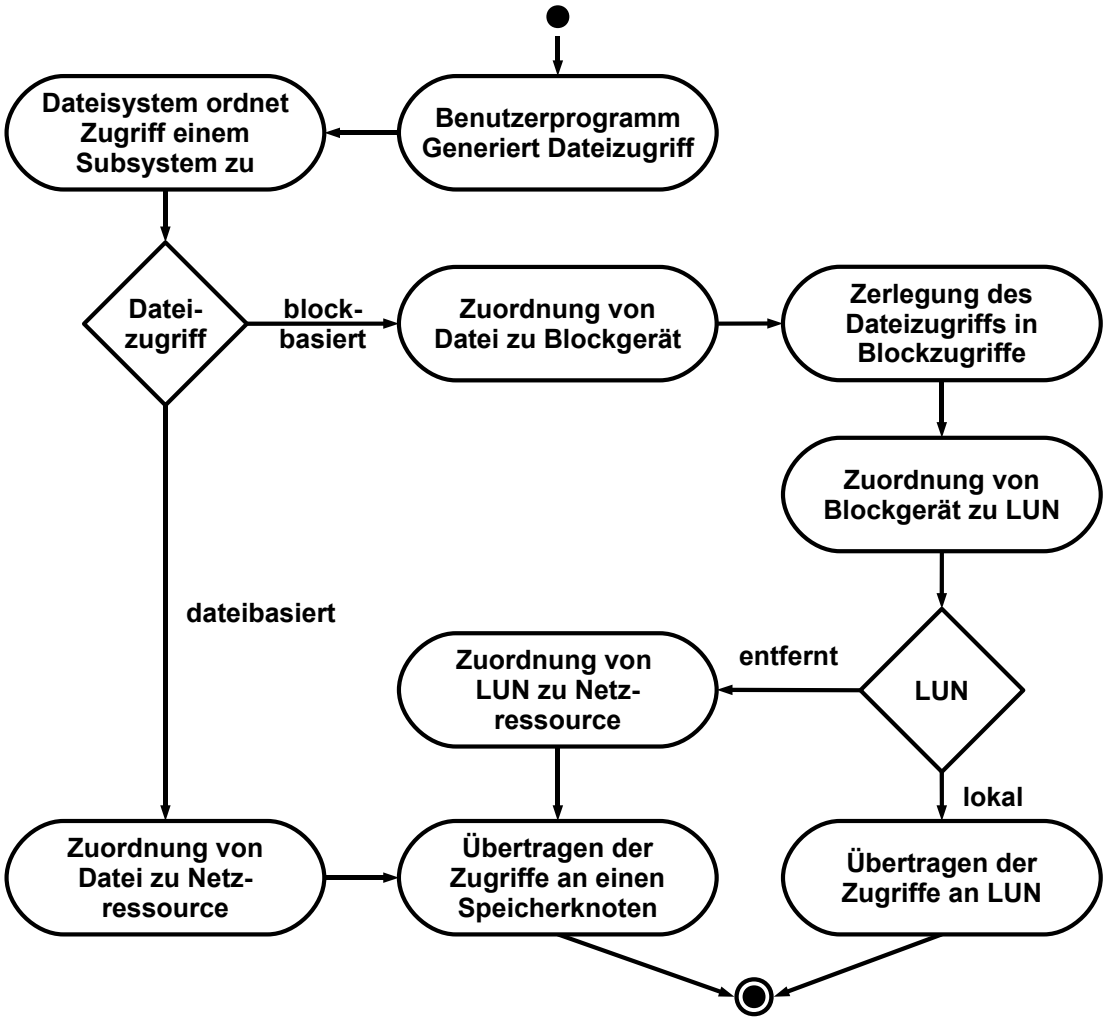


Abbildung 3.12: Datei- und blockbasierter Zugriff auf Hintergrundspeicher als Aktivitätsdiagramm

3.2.3.1 Interaktion mit Speichereinheiten

Da die Techniken, die LUNs auf Festplatten abbilden nicht Teil dieser Arbeit sind, untersucht dieses Kapitel keine Produkte, oder physischen Umsetzungen wie in die Kapitel 3.2.1 und 3.2.2. Gegenstand dieses Kapitels ist statt dessen die Interaktion von Rechenknoten mit Speichereinheiten.

Beim Zugriff einer Applikation auf Hintergrundspeicher unterscheidet man zwischen dem *dateibasierten* und dem *blockbasierten* Zugriff. Diese Unterscheidung ist wichtig, um die Funktionsweise von Techniken zur Virtualisierung von I/O-Kanälen beschreiben zu können. Der Unterschied zwischen diesen Zugriffsarten ist wie weit die Anfrage der Applikation verarbeitet wird, bevor sie an den Speicherknoten übertragen wird.

Betriebssysteme stellen ein Dateisystem zur Verfügung, um den Zugriff auf Dateien durch Applikationen zu vereinheitlichen [Tan03]. Für die Applikation ist es also unerheblich, ob

es sich um einen datei- oder blockbasierten Zugriff handelt. Greift eine Applikation auf eine Datei zu, so entscheidet das Dateisystem welches Teilsystem des Betriebssystems für die Verarbeitung der Anfrage zuständig ist und leitet die Anfrage an das entsprechende Teilsystem weiter. Ob der Zugriff datei- oder blockbasiert verläuft ist eine Eigenschaft des Teilsystems. Beispiele für blockbasierte Teilsysteme sind IDE und SCSI [Sch93], während NFS, Samba, Coda und „The Andrew File System“ Beispiele für dateibasierte Teilsysteme sind [Smi06].

Abbildung 3.12 zeigt einen Dateizugriff als Aktivitätsdiagramm am Beispiel des Betriebssystems Linux. Die Zwischenschritte des blockbasierten Dateizugriffs entsprechen den Zwischenschritten bei der Verwendung des SCSI-Subsystems des Linux Kernels ab Version 2.4. Ein dateibasierter Zugriff auf Hintergrundspeicher wird fast unverändert an den Speicherknoten übertragen. Ein blockbasierter Zugriff auf Hintergrundspeicher wird zuerst in Blockbefehle, die eine LUN ausführen kann, umgewandelt. In diesem Beispiel handelt es sich um SCSI-Befehle. Der Zugriff auf lokalen Hintergrundspeicher erfolgt immer blockweise [Tan03, Gil04, Smi06].

Abbildung 3.12 zeigt insbesondere, dass der blockbasierte Zugriff auf Hintergrundspeicher erst bei der Übertragung der Befehle zur LUN unterscheidet, ob es sich um lokalen Hintergrundspeicher (DAS, direct attached storage) oder Hintergrundspeicher auf einem Speicherknoten (NAS, network attached storage) handelt. Bei der Zuordnung von einem Blockgerät zu einer LUN (siehe Abb. 3.12) wählt der Linux Kernel den Treiber, der für die Interaktion mit der LUN verwendet werden soll aus. Der Treiber kommuniziert die Anweisungen entweder direkt über einen SCSI-HBA an die entsprechende Festplatte, oder kapselt die Befehle, so dass sie als Nutzdaten über ein Netz (das SAN) an einen Speicherknoten übertragen werden. Übernimmt der Treiber das Kapseln der SCSI-Befehle, so spricht man von einem *Software Initiator*. Ein anderer Ansatz ist, einen HBA so zu konstruieren, dass dieser die Befehle kapselt und an einen Speicherknoten überträgt. Ein solcher HBA heißt *Hardware Initiator* [Sch93]. Der wesentliche Unterschied zwischen Software und Hardware Initiatoren ist, dass bei einem Software Initiator das Betriebssystem für die Kapselung und Übertragung zuständig ist, während ein Hardware Initiator diese Aufgabe aus dem Betriebssystem herauslöst. Dieser Unterschied ist besonders interessant für Virtualisierung, da ein virtueller Server, der über einen Hardware Initiator verfügt nicht direkt mit dem SAN in Berührung kommt.

3.2.3.2 Zugang zu Speichereinheiten

Ein wesentlicher Aspekt der Virtualisierung von I/O-Kanälen ist die Isolierung von virtuellen Infrastrukturen, so dass Daten nicht von einer virtuellen Infrastruktur ungewollt in eine andere gelangen können (vgl. Kapitel 2.4). Auf Speichereinheiten bezogen bedeutet das, dass nur wenige, bestimmte Server Zugang zu einer bestimmten Speichereinheit erlangen dürfen. Betrachtet man zum Beispiel vermietete Server aus dem Szenario in Kapitel 2, so dürfen diese nicht auf Speichereinheiten anderer Kunden zugreifen können.

Den Zugang zu Speichereinheiten kann man auf zwei verschiedene Möglichkeiten kontrollieren. Eine Möglichkeit ist jeder virtuellen Infrastrukur einen virtuellen Speicherknoten zur Verfügung zu stellen, der nur erlaubte LUNs anbietet. Zugriffe auf den virtuellen Speicherknoten werden über die Segmentierung der Netze (VLANs) gesteuert. Mit dieser Methode wird nicht direkt Einfluss auf die Kommunikation zwischen Server und LUN genommen, sondern einem Server wird im Vorfeld keine Möglichkeit geboten mit fremden LUNs zu

Interagieren.

Die zweite Möglichkeit besteht darin eine zusätzliche Schicht bei der Interaktion mit Speichereinheiten einzuführen. Diese Schicht kann eine Sitzungsschicht entsprechend Schicht 5 des ISO-OSI-Referenzmodells in der Kommunikation eines Servers mit einer LUN sein. In diesem Fall kann der Zugriff auf LUNs gesteuert werden, indem sich ein Server vor dem Zugriff auf eine LUN am Speicherknoten Authentifizieren muss.

Eine andere Möglichkeit besteht darin, dass die zusätzliche Schicht zur Transformation benutzt wird. Dabei wird die Interaktion modifiziert, so dass die Anfrage eines Servers immer nur an erlaubten LUNs gerichtet ist. Diese Art der Modifikation wird auch *LUN-Masking* genannt.

3.2.4 Abstraktion und Segmentierung

Basierend auf den Erkenntnissen aus den Kapiteln 3.2.3, 3.2.2 und 3.2.1 kommt man zu dem Schluss, dass der Effekt, der durch Virtualisierung erreicht werden soll, immer entweder Abstraktion, oder Segmentierung ist. Eine weitere Folgerung ist, dass Virtualisierung wird immer dann eingesetzt, wenn vorhandene Randbedingungen eliminiert werden sollen. Bei Speichereinheiten (Kapitel 3.2.3) sind das die Kapazitätsbeschränkungen einzelner Festplatten und die benötigte Zeit, die benötigt wird um Daten zu lesen oder zu schreiben. Zwei Rechner, die teil des selben Netzes sind, können miteinander kommunizieren. VLANs helfen die Verbindungsmöglichkeiten zu beschränken, während VPNs die Aspekte von räumlicher und topologischer Trennung abstrahieren (Kapitel 3.2.2). Bei Rechenknoten wird die Beschränkung der x86-Plattform, nur einen einzigen privilegierten Betriebssystemkern betreiben zu können, eliminiert. Zusammenfassend ergibt sich folgende Definition:

> `Virtualisierung ist die Abstraktion von starren, beschränkenden Randbedingungen eines Systems zu konfigurierbaren Eigenschaften.`

3.3 Marktübersicht

Die Idee vorhandene Hardware mehrfach zu nutzen ist nicht neu. Die Mehrfachnutzung von Ethernet-Controllern ist seit Jahren gängige Praxis. Durch die Zuweisung von mehreren IP-Adressen kann ein Rechner als mehrere Endpunkte in einem IP-Netz erscheinen und durch die Benutzung von VLANs lässt sich ein einzelner Ethenernet-Controller nutzen als wären es mehrere Controller in unterschiedlichen Netzen. Der Ansatz für NPIV, der die Mehrfachnutzung eines FC-HBAs ermöglicht, ist von 2002 [T1102]. Die endgültigen Spezifikationen von SR- und MR-IOV der PCI-SIG wurden erst im September 2007 bzw. Mai 2008 freigegeben und sind die neuesten Technologien auf dem Markt [PCI07, PCI08]. Durch diesen zeitlichen Vorsprung ist Ethernet- und NPIV-fähige FC-Hardware bereits im Markt etabliert und von einer Vielzahl von Herstellern zu beziehen, während SR- und MR-IOV Hardware kaum bis gar nicht zu beziehen ist.

3.3.1 Fibre Channel und NPIV

Fibre Channel erfreut sich großer Verbreitung, da es seit geraumer Zeit mit einer Übertragungsgeschwindigkeit von vier Gigabit pro Sekunde (Gbps) verfügbar ist. Dies ist ein deutlicher Geschwindigkeitsvorteil gegenüber Ethernet, welches bis zur Einführung von 10GbE

nur 1 Gbps ermöglichte. Der größte Nachteil dieser Technologie ist der deutlich höhere Preis im Vergleich zu Ethernet. Seit der Veröffentlichung von NPIV existiert für Fibre Channel eine effiziente Methode mehrere Systeme mit unterschiedlichen Berechtigungen und Zugängen platz- und energiesparend an ein Speichernetz anzubinden. NPIV fähige Fibre Channel HBAs gibt es in vielen Varianten unter anderem vom Emulex [emu08], LSI [lsi08] und Qlogic [qlo08]. Außer NPIV fähigen HBAs benötigt man auch NPIV fähige Switche, da eine Verbindung über ein FC-Netz eine Anmeldung am Netz erfordert (siehe Abschnitt **Fibre Channel** ab Seite 39). NPIV Fibre Channel Switches sind von vielen Anbietern zu beziehen, jedoch meist als Teil einer kompletten Speicherlösung, wie z.B. bei IBM [ibm08] und Emulex. Die Switches an sich werden dabei meist zugekauft. Als echter Hersteller von NPIV fähigen Fibre Channel Switches treten nur Cisco und Brocade in Erscheinung. In der neuesten Version ist Fibre Channel bis zu 8 Gbps schnell und damit nur wenig langsamer als das neue 10 Gbps Ethernet. Das Problem der höheren Kosten im Vergleich zu Ethernet bleibt jedoch bestehen. Fibre Channel mit einer Übertragungsrate von 10 Gbps ist spezifiziert [T1107], jedoch noch nicht auf dem Markt erhältlich.

3.3.2 I/O-Hardware mit Virtualisierungsschicht

Um den Hypervisor zu entlasten, kann man I/O-Hardware mit einer Virtualisierungsschicht ausrüsten. Die Koordination des Mehrfachzugriffs verwaltet dabei die Hardware selbst und nicht mehr der Hypervisor (vgl. Kapitel 3.2.1.2). Der einzige Standard, der diese Möglichkeit beschreibt, ist SR-IOV. Da SR-IOV bereits von einem einzigen Gerät implementiert werden kann, sind die Marktbarrieren die ein SR-IOV fähiger Adapter überwinden muss relativ gering. Ein SR-IOV Adapter kann in einem herkömmlichen PCIe System benutzt werden [PCI07]. Die größte Konkurrenz zu SR-IOV Produkten sind bereits im Markt etablierte Geräte, die den selben Effekt durch andere Technologien wie zum Beispiel NPIV oder proprietäre Entwicklungen erreichen. Der Standard ist noch recht jung und so listet kaum ein Hersteller SR-IOV als unterstützte Technologie.

3.3.2.1 Intel 82895EB

Zur Zeit hat Intel genau einen 10GbE Adapter im Programm und dieser bietet auch Hardware-Unterstützung für Virtualisierung. Bevor es die PCI-SIG Standards gab, entwickelte Intel eine Technologie, die *Virtual Machine Device Queues* (VMDq) genannt wird [int08b]. Diese Technologie arbeitet direkt mit dem Hypervisor zusammen, so dass Intels Äquivalent zu virtuellen Funktionen erst im Hypervisor zu mehreren Geräten aufgefächert werden und nicht bereits im I/O-Subsystem, entsprechend dem SR-IOV Standard. Der 82895EB bietet 16 VMDqs pro Port an. Durch Intels großes Engagement bei der PCI-SIG ist jedoch davon auszugehen, dass sich diese Technologie nicht sehr von SR-IOV unterscheidet. Laut Intel sollen zukünftige Geräte explizit SR-IOV unterstützen [int08a].

3.3.2.2 Neterion X3100

Neterion ist der einzige Hersteller, der explizit SR-IOV als eingesetzte Virtualisierungstechnologie aufführt. Die X3100 Serie sind reine 10Gbps Ethernet (10GgE) Adapter die bis zu 17 virtuelle Funktionen bereit stellen. Außerdem implementiert die X3100 Serie MR-IOV in Version 0.7 [net08b].

3.3.2.3 Weitere Implementierungen

Neben diesen beiden Hauptakteuren findet man durchaus noch weitere Anbieter, die keine genaueren Angaben zu ihrer Virtualisierungstechnologie machen. Chelsio hat OEM 10GbE Adapter im Programm, die nach eigenen Angaben in Verbindung mit HP und IBM Blades verbaut werden, die wiederum mit nicht genauer spezifizierter „Virtualisierung“ werben [che08]. Auch NetEffekt bewirbt seine NE020 Adapter mit „Virtualization support“ zu dessen Fähigkeiten unter Anderem „Multiple virtual NICs“ und „Multiple PCI functions“ gehören [net08a].

3.3.3 I/O-Konsolidierung

I/O-Konsolidierung wird meist innerhalb des Chassis oder Racks betrieben. Diese Lösungen sind meist proprietär und daher entstehungsgeschichtlich nicht an den Freigabedaten von Standards fest zu machen. Grundsätzlich lassen sich die Ansätze in zwei Kategorien aufteilen. Noch nicht so weit verbreitet sind Ansätze, die ohne I/O-Server auskommen und I/O-Konsolidierung von Endpunkt zu Endpunkt umsetzen. Schon länger verfügbar sind Lösungen, die den Datenverkehr zwischen den Servern und spezialisierten I/O-Servern konsolidieren. In den I/O-Servern werden die Datenströme dann auf physisch getrennte Netze aufgeteilt.

3.3.3.1 Ansätze ohne I/O-Server

Wenn man von der Möglichkeit LAN und SAN auf IP über ein Ethernet-Netz zu betreiben, ist I/O-Konsolidierung ohne den Einsatz von I/O-Servern zur Zeit nur von Cisco verfügbar. Cisco verteibt Lösungen auf mehreren Gebieten. Mit FCoE entwickelt Cisco Produkte auf veröffentlichten Standards basierenden, während V-Frame eine proprietäre, auf InfiniBand basierende Lösung ist.

FCoE

FCoE Switche sind im Moment nur von Cisco zu beziehen, daher ist eine Entscheidung für FCoE auch eine Entscheidung für Cisco [cis08a]. Um vorhandene Speicherlösungen weiterhin verwenden zu können ist der Cisco FCoE Switch auch in der Lage die FC-Rahmen aus dem Ethernet-Rahmen zu extrahieren und nativ mit FC weiterzuleiten. Die vielen Randbedingungen von FCoE (siehe Kapitel 3.2.2.4) hemmen die schnelle Verbreitung ebenso wie der Preis für den FCoE Switch. 10GbE NICs gibt es von vielen Herstellern, u.a. Neterion, NetEffect und Intel. Entsprechende Switche (ohne FCoE) gibt es u.a. von 3Com, Force10 und Foundry Networks. Eine Alternative zu FCoE ist iSCSI, welches auf TCP aufsetzt und so nicht auf Erweiterungen des Ethernet-Standards angewiesen ist, wie FCoE.

V-Frame

Cisco's V-Frame Lösung ist die einzige InfiniBand basierte Lösung, die keinen I/O-Server verwendet. Per Treiber/Firmware wird sämtliche Kommunikation in InfiniBand Rahmen gekapselt, analog zur Kapselung von FC-Rahmen in Ethernet-Rahmen. Das Aufteilen der Datenströme auf separate Netze ist hier nicht nötig. Die Switche bieten dies dennoch, um bestehende Netze integrieren zu können [cis08b].

3.3.3.2 Ansätze mit I/O Server

Aktuell existieren Ansätze zur I/O-Konsolidierung mit I/O-Server nur basierend auf InfiniBand. Für InfiniBand Switches hat man prinzipiell die Wahl zwischen Cisco, Mellanox und Voltaire. Es gibt noch mehr Hersteller, wobei die meistens Mellanox Hardware in ihren Switchen einsetzen. Während man IB Switches untereinander mit bis zu 60 Gbps verbinden kann, so ist die Höchstgeschwindigkeit bei IB Adaptern 20 Gbps. 20 Gbps IB Adapter gibt es nur von Mellanox und Voltaire. 10 Gbps IB Adapter für I/O-Konsolidierung gibt es außer von Mellanox und Voltaire auch von Cisco.

Xsigo I/O Director

Der Xsigo I/O Director virtualisiert keine herkömmliche Hardware, so wie die mit MR-IOV verwanden Systeme, sondern ist ein komplett proprietäres Chassis in das „I/O Module“ eingesetzt werden, die entweder 1/10GbE- oder 4 Gbps FC-Schnittstellen enthalten. In den eigentlichen Servern/Blades befinden sich 10 Gbps IB Adapter, die direkt mit dem I/O Director verbunden werden. Die virtuellen HBAs und NICs werden dem Betriebssystem vom Treiber des IB Adapters bereitgestellt [xsi08].

3Leaf V-8000

3Leaf ist der einzige Anbieter, dessen Lösung bereits 20 Gbps InfiniBand benutzt. Der „V-8000 Virtual I/O Server“ ist dem I/O Director sehr ähnlich. Diese Lösung setzt einen speziellen Treiber auf den Betriebssystemen der Rechenknoten ein, der HBAs und NICs emuliert [3le08]. Dieser Treiber ist unabhängig von der eingesetzten InfiniBand-Hardware und damit weniger an den Einsatz spezieller Hardware gebunden, wie der I/O Director. Dadurch kann IB Hardware von verschiedenen Herstellern verwendet werden. Auch der I/O-Server selbst ist flexibler. Anstatt proprietärer Module wird handelsübliche PCIe Hardware verbaut. Unterstütze Geräte sind:

- Intel PRO/1000 PT dual port Gigabit Ethernet NIC
- Silicom USB PEG2 dual port Gigabit Ethernet NIC
- Emulex LPe11002 dual port 4Gb Fibre Channel HBA
- QLogic QLE2462 dual port 4 Gb Fibre Chanel HBA

VirtenSys I/O Virtualization Technology

Die Firma VirtenSys hat eine proprietäre Technologie, die wie MR-IOV funktioniert und nach eigener Aussage Kompatibel zu MR-IOV ist. VirtenSys bietet sowohl Rack-mounted I/O-Server, als als auch Blades mit dieser Technologie an. Das Besondere dabei ist, dass die PCIe Endgeräte keinerlei IOV Funktionalität benötigen, sondern komplett von der VirtenSys Technologie abstrahiert werden. Damit ist diese Technologie bereits heute bereit für den Einsatz [vir08].

NextIO I/O Gateway Virtualization

NextIO ist ein Hersteller für hoch performante PCIe Hardware (Express Connect^{TM}), die hohe Übertragungsraten und geringe Latenzzeiten versprechen. Unter der Bezeichnung „I/O Gateway Virtualization“ vertreibt NextIO seine Produkte, die ebenfalls herkömmliche PCIe

Hardware abstrahieren und in einer Multi-Root-ähnlichen Umgebung mehreren Servern zugänglich machen. Sämtliche Bemühungen von NextIO in Richtung Hardwarevirtualisierung laufen dort unter der Bezeichnung „Shared I/O“. So entwickelt NextIO nicht nur seine proprietäre Lösung weiter, sondern entwickelt auch einen dem Standard entsprechenden PCIe Switch für den Einsatz in ihren bisherigen Produkten. In diesem Zusammenhang ist NextIO Mitte Juni bereits eine Partnerschaft mit Neterion eingegangen, um möglichst schnell MR-IOV Produkte vermarkten zu können [nex08].

4 Analyse von Kombinationen

Vorausgegangenes Kapitel 3 analysiert einzelne Methoden, mit denen Virtualisierung realisiert werden kann. Bei der Analyse wird zwischen Virtualisierung in Rechenknoten, Netzen und Speicherknoten unterschieden, wobei sich die Analyse der Speicherknoten auf die Interaktion von Rechenknoten mit Speicherknoten beschränkt. Kombiniert man Virtualisierung aus allen drei Bereichen, erhält man ein System, mit dem I/O-Kanäle von Endpunkt zu Endpunkt virtualisiert werden können.

Kapitel 4 untersucht Techniken zur Virtualisierung von I/O-Kanälen. In diesem werden mögliche Kombinationen vorgestellt und deren Eigenschaften mit den Anforderungen aus Kapitel 2 verglichen. Dadurch resultiert eine Übersicht in wie weit eine Technik geeignet ist, um I/O-Kanäle zu virtualisieren.

4.1 Untersuchungsaspekte

Bevor im nächsten Kapitel auf Techniken zur Virtualisierung von I/O-Kanälen eingegangen wird, beschreibt Kapitel 4.1 die Aspekte, unter denen die Techniken betrachtet werden. Kapitel 4.1.1 erläutert warum bei der Untersuchung lediglich die Verbindung zwischen einem Server und einer LUN betrachtet wird und eine Verbindung zwischen zwei Servern nicht zusätzlich berücksichtigt werden muss. Im Anschluss daran liegt das Augenmerk in Kapitel 4.1.2 auf den virtuellen Server als Endpunkt von I/O-Kanälen und zeigt, dass die Qualität einer Technik auch an den Aufgaben, die der Hypervisor übernimmt, gemessen werden kann.

4.1.1 Betrachtung der Verbindung zwischen Server und LUN

Nach der Definition aus Kapitel 2.1 existieren I/O-Kanäle entweder zwischen einem Server und einer LUN, oder zwischen zwei Servern. Die Verbindung zwischen zwei Servern beschreibt eine allgemeine Nutzung des Netzes. Ein Server initiiert die Verbindung, die durch das Netz zu einem anderen Server führt. Der Unterschied zu einer Verbindung zwischen einem Server und einer LUN liegt darin, dass es sich beim zweiten Endpunkt um einen Speicherknoten, statt einem anderen Server handelt. Des Weiteren ist bei einer Kommunikation zwischen Server und LUN der Inhalt der Kommunikation, Zugriff auf Hintergrundspeicher, eindeutig. Im Vergleich dazu kann über den Inhalt der Kommunikation zwischen zwei Servern keine Aussage getroffen werden. Eine Kommunikation zwischen einem Server und einer LUN enthält alle bekannten Aspekte einer Kommunikation zwischen zwei Servern: ein Netz und dessen Nutzung durch Server. Im Gegensatz zu der Betrachtung der Kommunikation zwischen Server und LUN, liefert die Betrachtung der Kommunikation zwischen zwei Servern keine zusätzlichen Erkenntnisse. Aufgrund dessen wird im Folgenden dieser Arbeit lediglich die Interaktion zwischen Server und LUN betrachtet.

	Generierung der Blockzugriffe	Kapselung für Übertragung	Kommunikation
Software Initiator	OS	OS	OS
NAS	LUN	OS	OS
HW Initiator (para.)	OS	HW	HW
HW Initiator (emul.)	OS	VMM	VMM
DAS (emuliert)	OS	VMM oder HW	VMM oder HW

Abkürzungen: OS := Betriebssystem HW := Hardware

Tabelle 4.1: Verteilung der Aufgaben (horizontal) auf Teilsysteme in Abhängigkeit der Anbindungsmethode einer LUN (vertikal)

4.1.2 Aufgaben des Hypervisors als Qualitätsmetrik

Beim Einsatz von virtuellen Servern gibt es durch die Virtualisierungsschicht (siehe Kapitel 3.2.1) mehrere Möglichkeiten für einen Endpunkt eines I/O-Kanals. Ein virtueller Server hat fünf Möglichkeiten mit einer LUN zu interagieren:

1. Das Betriebssystem enthält einen Software Initiator
2. Eine LUN wird auf Dateisystemebene eingebunden (NAS)
3. Ein Hardware Initiator wird paravirtualisiert
4. Ein Hardware Initiator wird emuliert
5. Lokaler Hintergrundspeicher (DAS) wird emuliert

Diese fünf Möglichkeiten sind nach dem Aufwand, der auf den virtuellen Server entfällt, degressiv sortiert. Dieser Aufwand besteht darin, den Zugriff auf Hintergrundspeicher zu verarbeiten und an den Speicherknoten zu übertragen. Die entsprechenden Aufgaben werden vom Betriebssystem des virtuellen Servers erfüllt. Eine Übersicht über die Verteilung der Aufgaben bietet Tabelle 4.1. Der größte Aufwand entfällt auf den virtuellen Server, sofern dessen Betriebssystem einen Software Initiator enthält. In diesem Fall muss das Betriebssystem, wie in Abbildung 3.12 dargestellt, den Dateizugriff komplett implementieren. Zuerst ist es notwendig den Dateizugriff in Blockzugriffe zu zerlegen. Anschließend müssen die Blockzugriffe für die Übertragung an die LUN vorbereitet werden.Die Zugriffe müssen folglich vom Betriebssystem, entsprechend der eingesetzten Technologie, gekapselt werden. Ein Beispiel für einen solchen Zugriff stellt das Kapseln von SCSI-Befehlen in das iSCSI Protokoll (siehe 4.2.2) dar. Anschließend müssen die gekapselten Zugriffe entsprechend der spezifischen Kommunikationsprotokolle an die LUN übertragen werden. Wird die LUN als Dateisystem über einen Dienst, wie zum Beispiel NFS (siehe Kapitel 3.2.3.1), eingebunden, ist es erforderlich, dass das Betriebssystem die Dateizugriffe kapselt und überträgt. Im Gegensatz zu dem Software Initiator wird hier der Aufwand zum Zerlegen der Datei- in Blockzugriffe auf den Speicherknoten ausgelagert. Bei einem paravirtualisierten Hardware Initiator existiert (im Vergleich zu einem emulierten Hardware Initiator) ein physischer Hardware Initiator, der dem virtuellen Server zugänglich gemacht wird (vgl. Paravirtualisierung in Kapitel 3.2.1). Der virtuelle Server zerlegt die Datei- in Blockzugriffe und übergibt diese dem Hardware Initiator zur Übertragung an die LUN. Zuvor muss das Betriebssystem den Initiator konfigurieren, damit die Befehle an den richtigen Speicherknoten und folglich an die richtige LUN

übertragen werden. Bei der Emulation eines Hardware Initiators übernimmt der Hypervisor die Kapselung und Übertragung der Befehle. Im Gegensatz zu einem physischen Hardware Initiator kann der Hypervisor die Konfiguration festlegen und ist nicht auf eine Konfiguration durch das Betriebssystem angewiesen. Bei der letzten Kombinationsmöglichkeit, dem Emulieren von lokalem Hintergrundspeicher, bleibt das SAN dem virtuellen Server komplett verborgen. In diesem Fall hat der Server keine Möglichkeit die Kommunikation mit dem Speicherknoten zu beeinflussen. Da die Kommunikation mit dem Hintergrundspeicher für den virtuellen Server vollständig transparent ist, bietet dies die meisten Möglichkeiten für den Hypervisor,

Die fünf Möglichkeiten eine LUN an einen virtuellen Server anzubinden, unterscheiden sich zusätzlich darin, wieviel Kontrolle der Hypervisor über die Interaktion hat. Als die beiden Extremfälle sind hierbei der Software Initiator und der emulierte lokale Hintergrundspeicher zu nennen. Kommt ein Software Initiator im Betriebssystem des virtuellen Servers zum Einsatz, ist der Kontrollaufwand der Interaktion sehr hoch. Die Interaktion ist in diesem Fall bereits für die Übertragung gekapselt und kann daher nicht direkt vom Hypervisor kontrolliert werden. Der Hypervisor agiert lediglich als Teil der Kommunikationsschicht zwischen Server und LUN. Emuliert der Hypervisor lokalen Hintergrundspeicher, interagiert der virtuelle Server direkt mit dem Hypervisor, der die Interaktion transformiert und an die LUN überträgt (vgl. Kapitel 3.1.2). Da der Hypervisor direkt mit der LUN interagiert, verfügt er über maximale Kontrolle der Interaktion zwischen Server und LUN. Die Kontrolle des Hypervisors über die Kommunikationsmöglichkeiten der virtuellen Server erlaubt es, den Hypervisor einzusetzen, um explizit Anforderungen aus Kapitel 2 zu erfüllen.

Kapitel 3.2.1 beschreibt, dass ein virtueller Server vollständig durch Software emuliert werden kann. Dem entsprechend kann ein virtueller Server mit jeder gewünschten Eigenschaft versehen werden, indem diese in Software nachgebildet wird. Verfolgt man das Ziel eine Technik zu entwerfen, die alle Anforderungen aus Kapitel 2 erfüllt, muss der Hypervisor alle Aufgaben bewältigt, die nicht von anderen Komponenten übernommen werden.

Je mehr Aufgaben der Hypervisor erfüllen muss, desto mehr Emulation muss dieser einsetzen, Emulation aber hemmt die Leistung des Gesamtsystems (vgl. Kapitel 3.2.1). Bei den Beschreibungen der Techniken zur Virtualisierung von I/O-Kanälen werden deshalb nicht explizit die implementierten Funktionen eines Hypervisors erwähnt. Statt dessen schließt man aus den Funktionen der anderen Komponenten, welche Funktionen ein Hypervisor implmentieren muss, damit diese Technik zur Virtualisierung von I/O-Kanälen eingesetzt werden kann. Der Eignungsgrad einer Technik zur Virtualisierung von I/O-Kanälen lässt sich daran messen, wieviele Aufgaben der Hypervisor erfüllt.

4.1.3 Auswahl der Kombinationen

Abbildung 4.1 zeigt einige Möglichkeiten der Kombination von Methoden. Eine Kombination entspricht einer Schichtung von Methoden und Protokollen. Ausgehend von den Übertragungstechnologien können entlang der Pfeile mögliche Kombinationen, von innen nach außen, abgelesen werden. Nicht enthalten sind solche Kombinationen, deren Darstellung einen Pfeil nach innen benötigt. Ein Rückschritt nach innen entspricht einer Kapselung einer Verbindung in einem anderen Protokoll. Kapselung wird meist in komplexeren Infrastrukturen eingesetzt, in denen Verbindungen über die Grenzen mehrerer Netze oder Technologien hinweg aufgebaut werden. Ein Beispiel in diesem Zusammenhang ist die Kapselung von FC in eine IP-basierte Verbindung, um FC-Verbindungen über das Internet in ein anderes

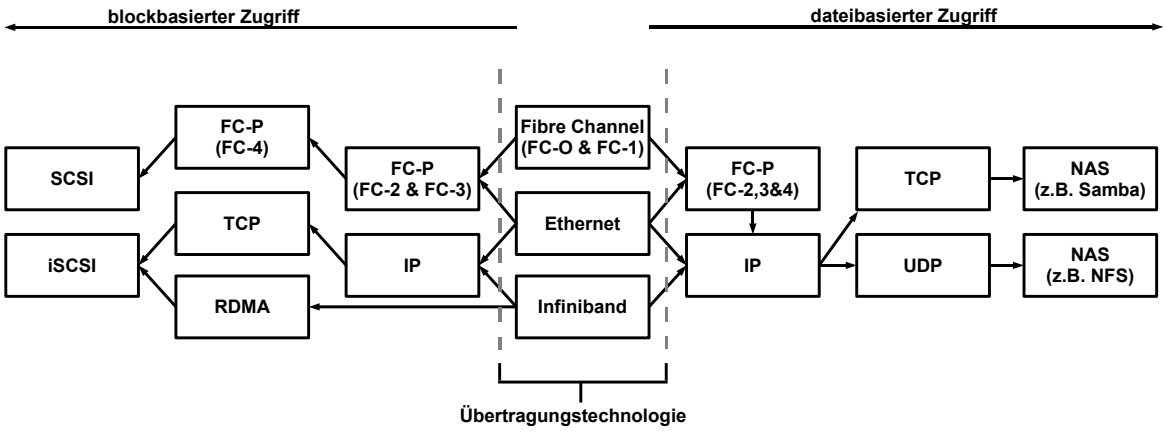


Abbildung 4.1: Kombinationsmöglichkeiten zur Interaktion mit Hintergrundspeicher

Rechenzentrum aufzubauen. Die Kapselung von Fibre Channel in TCP/IP ist in RFC 3821 spezifiziert [RRW04]. Für lokale Verbindungen, innerhalb eines Rechenzentrums, werden solche Kapselungen selten eingesetzt. Kapselung verursacht zusätzlichen Verwaltungsaufwand, der in der Regel minimiert werden sollte. Deshalb werden solche Kombinationen in dieser Arbeit nicht weiter betrachtet.

Die Abbildung unterscheidet zwischen dem blockbasierten und dateibasierten Zugriff auf Hintergrundspeicher. Diese Unterscheidung zeigt, dass die gängigsten Methoden für den dateibasierten Zugriff auf IP basieren. Eine spezielle FC-4-Implementierung für den dateibasierten Zugriff über Fibre Channel existiert nicht. Dies ist nur über die Kapselung in IP-Verbindungen nach RFC 4338 möglich [DCN06]. Der dateibasierte Zugriff über FC und FCoE ist lediglich aus Gründen der Symmetrie zum blockbasierten Zugriff eingezeichnet. Keine der beiden denkbaren Techniken wird in Rechenzentren eingesetzt.

Wegen der Schichtung der Protokolle ist es für den Zugriff unerheblich welche Technologie zugrunde liegt (vgl. Kapitel 3.1). Deshalb ist es ausreichend in Kapitel 4.2.5 auf die Eigenschaften des dateibasierten Zugriffs einzugehen. Andere Techniken müssen aufgrund dessen lediglich bis zur IP-Schicht betrachtet werden. Die folgenden Schichten verhalten sich analog zu Kapitel 4.2.5. Dies gilt auch für den blockbasierten Zugriff mit iSCSI über TCP/IP. Kapitel 3.2.3.1 listet vier Methoden zum dateibasierten Zugriff auf Hintergrundspeicher: NFS, Samba, Coda und The Andrew Filesystem. Diese Arbeit beschränkt sich auf die Betrachtung der am weitesten verbreiteten Methode NFS, die auch am LRZ eingesetzt wird. Eine Auflistung der Kombinationen, die im Rahmen dieser Arbeit untersucht werden, liefert Tabelle 4.2.

Die bisher am häufigsten eingesetzte Technik ist SCSI/FC. Diese Technik wird zu Beginn betrachtet, um eine Vergleichsbasis für andere Technologien zu schaffen. iSCSI/TCP/IP/Ethernet wird als kostengünstige Alternative zu SCSI/FC betrachtet, jedoch mit Einschränkungen hinsichtlich Sicherheit und Zuverlässigkeit. In dieser Arbeit soll auch der Unterschied der beiden Technologien in Bezug auf Virtualisierung ermittelt werden.

Die Möglichkeiten von InfiniBand werden exemplarisch jeweils für den block- und dateibasierten Zugriff dargestellt. Durch die Verwendung von IPv6 als Teil der InfiniBand Architektur und der Implementierung einer Transportschicht (vgl. Kapitel 3.2.2.3) ergeben

Kombination	datei- oder blockbasiert	Kapitel	Seite
SCSI über Fibre Channel	block	4.2.1	59
iSCSI, TCP/IP über Ethernet	block	4.2.2	64
SCSI, Fibre Channel über Ethernet (FCoE)	block	4.2.3	68
iSCSI, RDMA über InfiniBand	block	4.2.4	73
NFS, UDP/IP über Ethernet	datei	4.2.5	77
NFS, UDP/IP über InfiniBand	datei	4.2.6	81

Tabelle 4.2: Untersuchte Kombinationen

sich für InfiniBand deutlich mehr Kombinationsmöglichkeiten, als in Abbildung 4.1 aufgezeigt. Viele davon sind sich sehr ähnlich, wie zum Beispiel iSCSI/SDP/InfiniBand und iSER/InfiniBand. Nach Kapitel 3.2.2.3 entspricht SDP einer Kombination TCP/RDMA und iSER einer Kombination iSCSI/RDMA. Somit lassen sich die beiden Kombinationen iSCSI/SDP/InfiniBand und iSER/InfiniBand auch als iSCSI/TCP/RDMA/InfiniBand und iSCSI/RDMA/InfiniBand aufschreiben. Der Unterschied zwischen den beiden Kombinationen ist die Verwendung von TCP in iSCSI/SDP/InfiniBand. Kapitel 3.2.2.3 erwähnt ebenfalls, dass iSER entwickelt wurde, um TCP vollständig bei der Verwendung von iSCSI zu ersetzen. Der Unterschied zwischen den beiden Kombinationen ist damit hinfällig. Des Weiteren gibt es zahlreiche iSCSI/TCP/IP/InfiniBand Kombinationen, die sich einzig darin unterscheiden, auf welche Weise IP auf IB abgebildet wird. Da nach Möglichkeit immer das volle Potential einer Technologie ausgenutzt wird, werden diese einzelnen Ausprägungen hier nicht näher untersucht.

4.2 Techniken

In diesem Kapitel werden Techniken zur Virtualisierung von I/O-Kanälen, entsprechend der Auswahl aus Kapitel 4.1.3, vorgestellt. Mit Bezug auf die genaueren Beschreibungen aus Kapitel 3, wird für jede Technik zunächst knapp die Funktionalität beschrieben. Anschließend wird jede Kombination mit allen 22 Anforderungen aus Kapitel 2.5 verglichen und die wesentlichen Eigenschaften tabellarisch zusammengefasst .

4.2.1 SCSI über Fibre Channel

Diese Technik setzt eine spezielle Implementierung auf der Schicht FC-4 ein, um SCSI Befehle über Fibre Channel, statt über den SCSI-Bus (siehe [Sch93]) zu übertragen (vgl. Kapitel 3.2.2.1) .

Für die Virtualisierung wird NPIV eingesetzt, damit mehrere virtuelle Server einen physischen FC-HBA benutzen können. FC wird aktuell häufig als SAN Technologie eingesetzt und stellt daher die Grundlage für Vergleiche zwischen den Technologien dar. Tabelle 4.3 liefert eine Zusammenfassung über die wichtigsten Eigenschaften dieser Technik . Die darauf folgende Tabelle 4.4 liefert eine Übersicht über den Abgleich von Anforderungen mit dieser Kombination.

Abgleich von Anforderungen

Zur zentralen Verwaltung von Zonen kommt bei FC ein Management-Server zum Einsatz. Dieser ist in der Lage Zonen, die zur Umsetzung von virtuellen Infrastrukturen eingesetzt werden, eindeutig zu identifizieren. Die Anforderung nach einem eindeutigen Bezeichner für virtuelle Infrastrukturen ist in diesem Fall erfüllt (Anforderung #1).
Mit NPIV kann jeder virtuelle Server individuell am Netz angemeldet werden und erhält eine eigene N_Port ID . Der Hypervisor kennt die Zuordnung von N_Port ID zu den virtuellen Servern und kann dadurch eine Verwechselung von N_Port IDs verhindern. Die N_Port ID stellt demzufolge die geforderte Markierung des Datenstroms dar (Anforderung #2). Die Anforderung nach einer Filterung des Datenverkehrs ist hierbei erfüllt. (Anforderung #3).
Durch die Schichtung von FC können Rahmen zwischen Stern- und Ringtopologien ausgetauscht werden. Andere technologische Trennungen treten beim Einsatz von FC nicht auf. Mit FC können virtuelle Infrastrukturen technologieübergreifend angelegt werden (Anforderung #4).
Die Zusicherung von Eigenschaften einer beliebigen Verbindung wird über die FC-GS Dienste gesteuert. Diese Dienste werden auf der Schicht FC-3 bereitgestellt, während die Unterscheidung zwischen Stern- und Ringtopologie nur bis zur Schicht FC-2 reicht. Deshalb kann FC Zusicherungen über Technologiegrenzen hinweg durchsetzen (Anforderung #5).
Aktuelle Switche verfügen über Management-Zugänge, über die auch Daten hinsichtlich der Nutzung und Auslastung abgerufen werden können (Anforderung #6 und #7).
Die identifizierenden Merkmale eines FC-Endpunkts sind der WWN und die N_Port ID. Im Falle von NPIV ist der WWN eines virtuellen Servers eine Eigenschaft, die allein durch den Hypervisor verwaltet wird. Die N_Port ID wird bei jeder Netzanmeldung bezogen. Sie ist keine feste Eigenschaft des Endpunkts. Beim Einsatz von NPIV ist der WWN eines Servers beliebig durch den Hypervisor veränderbar (Anforderung #8).
In einem FC-Netz sind Datenströme entsprechend der Anforderung regulierbar. Die dazu notwendigen Eigenschaften werden in FC-LS spezifiziert (Anforderung #9).
Die virtuelle I/O-Hardware eines virtuellen Servers ist jederzeit durch den Hypervisor konfigurierbar. FC selbst sieht keine Möglichkeiten vor einen HBA vor der Netzanmeldung zu konfigurieren. Beim Einsatz von virtuellen Servern und NPIV ist dies auch nicht notwendig (Anforderung #10 und #11).
Mit den FC-GS Diensten kann jede physische Verbindung zwischen zwei Geräten individuell gesteuert werden, damit Virtuelle Infrastrukturen überlappen können (Anforderung #12).
Die Anforderung, dass die Anzahl der virtuellen Infrastrukturen nicht begrenzt sein darf, wird ebenfalls erfüllt. Solang der entsprechende FC-GS Dienst dies zulässt, können virtuelle Infrastrukturen angelegt werden, da die Zugehörigkeit zu einer Zone eine Eigenschaft einer physischen Verbindung ist (Anforderung #13).
FC HBAs können nicht redundant betrieben werden, da jeder N_Port eines HBAs eine eigene N_Port ID zugewiesen bekommt. Um Ausfallsicherheit zu gewährleisten, müssen das Netz und der Hypervisor derart konfiguriert sein, dass ein sekundärer HBA eine Verbindung zu den LUNs herstellen kann, vorausgesetzt der primäre HBA ist dazu nicht mehr in der Lage (Anforderung #14). Die Verfügbarkeit kann hierbei aufrecht erhalten werden, sofern ein einzelner HBA nicht mehr in der Lage ist, eine Verbindung zu LUNs herzustellen. Dies ist jedoch keine Eigenschaft der FC Technologie (Anforderung #15).
Verändert man die Konfiguration aktueller Hardware, treten diese Änderungen in der Regel sofort in Kraft. Konfigurationsänderungen werden über den entsprechenden FC-GS Dienst,

zum Beispiel dem Management-Server, realisiert. Auf diese Art übertragen sich Konfigurationen auf die Hardware bei der nächsten Dienstanfrage (Anforderung #16).
Die WWN ermöglicht die eindeutige Adressierung eines HBAs. Die Anforderung I/O-Hardware eindeutig adressieren zu können ist somit erfüllt (Anforderung #17).
Mit der Regulierung des Datenstroms (Anforderung #9) wird sichergestellt, dass versendete Rahmen angenommen werden können. Durch eine Checksumme wird sichergestellt, dass die Datenintegrität erhalten bleibt. Rahmen können demzufolge verlustfrei durch das Netz übertragen werden (Anforderung #18). Außerdem wird in FC auf diese Weise sichergestellt, dass die Rahmen in der richtigen Reihenfolge übertragen werden (Anforderung #20).
FC erlaubt Rahmengrößen, mit denen komplette Blöcke des Hintergrundspeichers Übertragen werden können. Eine Aufteilung von Blöcken auf mehrere Rahmen ist nicht notwendig. (Anforderung #19).
Die Wahl des Kommunikationspfades hängt von der Konfiguration im Management-Server der FC-LS Dienste ab. Dieser ermöglicht die Speicherung von Konfigurationen in Abhängigkeit des WWN oder der Port ID. Folglich kann der Kommunikationspfad auch in Abhängigkeit der virtuellen Infrastruktur, der ein Port zugehörig ist, gesteuert werden (Anforderung #21).
Allerdings erlaubt FC keine Berücksichtigung die Auslastung einer physischen Verbindung bei der Wahl des Kommunikationspfades. Die Last einer physischen Verbindung kann demnach nicht verteilt werden (Anforderung #22).

SCSI/Fibre Channel

Identifizierung & Filterung

→ Identifizierung einer Ressource durch WWN des Servers und LUN
→ Physische Verbindungen können in Zonen organisiert werden
→ Filterung nicht nötig, da zielgerichtete Weiterleitung

Flusskontrolle

→ End-to-end-credit System um Senderate der Endpunkte zu kontrollieren
→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Feste Bandbreitenzuweisung zu einer Ende-zu-Ende Verbindung (Virtual Circuit)
→ Unterteilung in Klassen (Classes of Service) bestimmt welche Methoden benutzt werden

---

Konfiguration

→ Zentrale Konfiguration über FC-GS Dienste
→ FC-GS Dienste beinhalten Namensauflösung
→ Flusskontrolle für einzelne Verbindungen möglich
→ Redundanz durch Konfiguration mehrerer Pfade und Kontrolle durch einen Endpunkt

---

Integration

→ Spezielle FC-4 Implementierung für andere Protokolle unter anderem für IPv4 und IPv6 sind vorhanden
→ Unterschiedliche FC-4 Implementierungen interagieren nicht

Tabelle 4.3: Zusammenfassung der Eigenschaften von SCSI über Fibre Channel

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit der MAC-Adresse		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang		x	
#11	Management-Zugangskontrolle		x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz		x	
#15	Aufrechterhaltung der Verfügbarkeit		x	
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung	x		
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung			x

Tabelle 4.4: Abgleich der Anforderungen mit SCSI über Fibre Channel

4.2.2 iSCSI, TCP/IP über Ethernet

Das iSCSI Protokoll ist ein Transportprotokoll, das auf TCP aufsetzt. Das iSCSI Protokoll soll dabei vollständig konform zum standardisierten SCSI-Architektur-Modell sein, um SCSI über Netze transportieren zu können [SMS+04]. Es ist möglich, iSCSI über jedes Vermittlungsprotokoll zu betreiben, das mit TCP verwendet werden kann. Die Kombination mit dem Internet Protokoll (IP) und Ethernet ist darin begründet, dass in lokalen Netzen meistens TCP/IP auf Ethernet eingesetzt wird. Durch die Wahl dieser Technik können LAN und SAN gemeinsam auf der vorhandenen Ethernet-Infrastruktur betrieben werden. Eine Zusammenfassung über die wichtigsten Eigenschaften dieser Kombination enthält Tabelle 4.5.

Die Bewertung der Anforderungserfüllung dieser Technik benötigt eine feinere „Skala“ als SCSI/FC in Kapitel 4.2.1. Dies resultiert daraus, dass ein FC Switch alle spezifizierten FC-Funktionen implementiert, wodurch ein FC-Netz sehr präzise steuerbar ist. Die Technik iSCSI/TCP/IP/Ethernet ist feiner geschichtet (vier Schichten, im Gegensatz zu zwei Schichten bei SCSI/FC). Diese feinere Schichtung muss beim Abgleich mit Anforderungen berücksichtigt werden. Nach der vereinfachten Darstellung von Schichtung in Kapitel 3.1 gilt es als erstrebenswert, TCP, IP und Ethernet streng zu trennen. Allerdings ist es zur Erfüllung von Anforderungen wichtig, die Eigenschaften von TCP, IP und Ethernet in jeder Komponente des Netzes zu kombinieren. Ein Ethernet-Switch muss demnach auch TCP- und IP-Funktionen implementieren. Solche Switche sind verfügbar, jedoch kann nicht vorausgesetzt werden, dass tatsächlich jeder Ethernet-Switch TCP- und IP-Funktionen enthält. Aus diesem Grund wird bei der Erfüllung von Anforderungen unterschieden, ob es sich um eine Eigenschaft von Ethernet oder der Protokolle TCP und IP handelt. Ethernet-Funktionen können als gegeben vorausgesetzt werden, während TCP/IP-Funktionen nicht zwangsläufig gegeben sind und die Anforderungen dadurch nicht immer erfüllt werden.

Im Gegensatz zu NPIV und FC (siehe Kapitel 4.2.1) ist keine spezielle Virtualisierungsmethode für den Einsatz mit dieser Technik bestimmt. Damit beim Einsatz dieser Technik mehrere virtuelle Server den selben HBA nutzen können, wird entweder Emulation oder ein Software Initiator eingesetzt (vgl. Kapitel 4.1.2). Der Hypervisor kann einen Hardware Initiator oder DAS emulieren. In beiden Fällen interpretiert der Hypervisor die Befehle des virtuellen Servers. Zusätzlich kann er diese kontrollieren und verändern. Wird auf dem nichtprivilegierten Betriebssystem ein Software Initiator eingesetzt, behandelt der Hypervisor iSCSI als Ethernet-Nutzdaten und kann diese nicht kontrollieren oder verändern. Tabelle 4.6 liefert eine Zusammenfassung des folgenden Abgleichs von Anforderungen mit dieser Kombination.

Abgleich von Anforderungen

Ethernet VLANs werden durch eine Rahmenmarkierung implementiert (Anforderung #2). Eine virtuelle Infrastruktur ist daher eindeutig durch die VLAN-ID identifizierbar (Anforderung #1).
Die Filterung von Datenströmen ist nach dem IEEE-Standard-802.1Q möglich [iee06]. Wenn mehrere virtuelle Server dieselbe physische Hardware benutzen, muss die Virtualisierungsschicht ankommende Rahmen an den entsprechenden virtuellen Server weiterleiten (Anforderung #3). Bei dieser Technik ist Ethernet die einzige Technologie zur Datenübertragung, wodurch die Anforderungen #4 und #5 trivialer weise erfüllt sind. Aktuelle Switche verfügen über Management-Zugänge, über die auch Daten bezüglich Nutzung und Auslastung der

physischen Ethernet-Verbindungen abgerufen werden können (Anforderung #6). Die Auslastung einer logischen Verbindung zwischen Server und LUN kann nicht gemessen werden, da ein Ethernet-Switch nicht zwischen logischen Verbindungen unterscheidet (Anforderung #7).
Die MAC-Adresse ist das identifizierende Merkmal eines Ethernet NICs. Jeder virtuelle Server besitzt eine eigene MAC-Adresse, deren Zuweisung und Kontrolle der Hypervisor übernimmt (Anforderung #8). Die Forderung eines Management-Zugangs (Anforderung #10) und die Forderung einer Zugangskontrolle für diesen (Anforderung #11) werden ebenfalls durch den Hypervisor erfüllt.
Die Ethernet Technologie (Kapitel 3.2.2.2) unterstützt nicht die aktive Regulierung von Datenströmen (Anforderung #9). Durch Glättung des Datenverkehrs (Traffic Shaping) wird die Geschwindigkeit, mit der IP-Pakete verschickt werden, kontrolliert [Tan07]. Da diese Methode kein fester Bestandteil von Ethernet ist, kann man nicht davon ausgehen, dass jeder Switch diese Methode implementiert.
Die Zugehörigkeit zu einer bestimmten virtuellen Infrastruktur eines Rahmens hängt von der VLAN-ID ab, mit der ein Rahmen markiert wird. Da ein virtueller Server bei der Generierung von Rahmen die Markierung vornimmt, können Rahmen unterschiedlich markiert werden und unterschiedlichen virtuellen Infrastrukturen gehören (Anforderung #12).
Die Anzahl der virtuellen Infrastrukturen ist durch die VLAN-ID auf 2^{12} (65536) begrenzt. In der Praxis wird diese Anzahl als hinreichend betrachtet um anzunehmen, dass die Anzahl der virtuellen Infrastrukturen unbegrenzt ist (Anforderung #13).
Durch den Einsatz von STP können Ethernet NICs redundant betrieben werden und der Komponentenausfall beeinträchtigt nicht die Verfügbarkeit eines Servers (Anforderung #14 und #15). Verändert man die Konfiguration aktueller Hardware, treten diese Änderungen in der Regel sofort in Kraft. Erfüllt wird diese Anforderung lediglich durch die eingesetzte Hardware. Es gibt keine Spezifikation bezüglich Ethernet, die ein derartiges Verhalten vorsieht oder behindert (Anforderung #16).
Die MAC-Adresse ermöglicht die eindeutige Adressierung eines NIC. Die Anforderung I/O-Hardware eindeutig adressieren zu können ist somit erfüllt (Anforderung #17).
Die Zusicherung, Daten verlustfrei zwischen zwei Kommunikationsendpunkten zu übertragen, ist eine Eigenschaft von TCP (Anforderung #18).
Die normale Größe eines Ethernet-Rahmens ist nicht hinreichend, um Dateisystemblöcke ohne Fragmentierung zu übertragen. Nur durch den Einsatz von Jumbo Frames kann diese Anforderung erfüllt werden. Diese Erweiterung von Ethernet wird schon lange durch eine Vielzahl an Hardware unterstützt. Deshalb ist diese Anforderung erfüllt (Anforderung #19).
Durch den Einsatz von Ethernet kann nicht garantiert werden, dass Datenblöcke in der richtigen Reihenfolge übertragen werden. Die Sortierung erfolgt erst am Kommunikationsendpunkt durch TCP, nicht aber innerhalb des Speichernetzes (Anforderung #20).
Der IEEE-802.1Q-Standard sieht vor, dass innerhalb eines jeden VLANs STP zum Einsatz kommt. Für jedes VLAN wird folglich ein eigener Spanning-Tree gefunden. Deshalb kann die Generierung eines Spanning-Trees durch die Zuordnung von VLAN-IDs zu Ethernet-Ports gesteuert werden. So können für unterschiedliche virtuelle Infrastrukturen unterschiedliche Kommunikationspfade festgelegt werden (Anforderung #21). Das Spanning-Tree-Protokoll deaktiviert alle Verbindungen, mit denen innerhalb eines VLANs alternative Pfade zwischen zwei bestimmten Endpunkten gebildet werden können. Deshalb gibt es beim Einsatz von STP in einem Ethernet-Netz exakt einen möglichen Kommunikationspfad zwischen zwei Endpunkten. Anforderung #22 ist demnach nicht eingehalten.

iSCSI/TCP/IP/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IQN im Sitzungsprotokollheader
→ Filterung des Ethernet-Verkehrs anhand der VLAN-ID
→ Filterung der Vermittlungs-, Transport- und Sitzungsschichten durch entsprechende Paketfilter

---

Flusskontrolle

→ Flusskontrolle in TCP bei Überlastung der Verbindung
→ TCP/IP Verkehrsglättung (Traffic Shaping) in den Endpunkten
→ Flusskontrolle in Switchen nur durch spezielle Ausrüstung

Konfiguration

→ VLAN-ID frei wählbar
→ IP-Adresse zur eindeutigen Identifizierung innerhalb eines VLANs
→ IQN besteht aus Fully Qualified Domain Name (FQDN) und LUN
→ Glättung durch Einteilung in Klassen basierend auf Attributen der Kommunikationsschichten, z.B. IQN, TCP-Port, IP-Adresse
→ Redundanz durch Spanning-Tree

Integration

→ iSCSI setzt auf TCP auf und stellt keine zusätzlichen Anforderungen an tiefere Schichten
→ Muss bei I/O-Konsolidierung mit LAN-IP-Verkehr abgestimmt werden
→ Präzise Steuerung erfordert TCP- und IP-Fähigkeiten in Ethernet Switchen

Tabelle 4.5: Zusammenfassung der Eigenschaften von iSCSI, TCP/IP über Ethernet

	Bezeichnung	Erfüllt durch			
		ETH	**TCP /IP**	**VMM**	**nicht mgl.**
#1	Eindeutiger Bezeichner	x			
#2	Datenstrommarkierung	x			
#3	Datenstromfilterung	x		x	
#4	Technologieübergreifend	x			
#5	Technologieübergreifende Zusicherungen	x			
#6	Datenstromüberwachung	x			
#7	Messbarkeit der Auslastung		x		x
#8	Veränderbarkeit der MAC			x	
#9	Flusskontrolle		x		
#10	HBA-Management-Zugang			x	
#11	Management-Zugangskontrolle			x	
#12	Überlappung	x			
#13	Unbegrenzt viele virtuelle Infrastrukturen	x			
#14	HBA Redundanz	x			
#15	Aufrechterhaltung der Verfügbarkeit	x			
#16	Sofortige Konfigurationsänderung	x			
#17	Eindeutig adressierbarer HBA	x			
#18	Verlustfreie Datenübertragung		x		
#19	Rahmengröße	x			
#20	Reihenfolge der Rahmenübertragung				x
#21	Pfadwahl nach virtueller Infrastruktur	x			
#22	Pfadwahl nach Auslastung				x

Tabelle 4.6: Abgleich der Anforderungen mit iSCSI, TCP/IP über Ethernet

4.2.3 SCSI, Fibre Channel über Ethernet (FCoE)

Wie in Kapitel 3.2.2.4 beschrieben, beinhaltet der Ansatz von FCoE, das LAN und das SAN auf einem physischen Netz zu vereinen (I/O-Konsolidierung). Durch den Einsatz von FCoE muss lediglich ein physisches Netz angeschafft und gewartet werden. Außerdem kann für LAN und SAN auf das kostengünstige Ethernet zurückgegriffen werden.

Der Rahmenaufbau von FCoE sieht vor, FC-Datenverkehr in Ethernet zu kapseln. FC-Datenverkehr kann somit schnell auf Ethernet umgesetzt werden, wodurch vorhandene FC-Speicherknoten weiterverwendet werden können. Ferner erlaubt es die Kapselung auch bestehende FC-Infrastrukturen mit einem FCoE Netz zu verbinden. Dadurch entstehen Überlappungen beim Abgleich gegen die Anforderungen aus Kapitel 2. Der Abgleich von Anforderungen gegen FC-Netze wird in Kapitel 4.2.1 vorgenommen. Dieses Kapitel beschränkt sich daher auf den Abgleich der Anforderungen in Bezug auf Ethernet. FC wird nur an den Stellen erwähnt, an denen ein Unterschied zu den Erkenntnissen aus Kapitel 4.2.1 festgestellt wird.

Die Abbildung von FC-Rahmen auf Ethernet-Rahmen übernimmt ein FCoE Switch, der sowohl über FC-Ports als auch Ethernet-Ports verfügt. Außerdem implementiert ein solcher Switch die Funktionen von FC und Ethernet um Zusicherungen für Übertragungen umsetzen zu können. Für reine Ethernet Komponenten stellt der FC-Anteil der Rahmen lediglich Nutzdaten dar und wird nicht bei der Weiterleitung berücksichtigt. Dadurch kann FCoE auch zusammen mit bestehender Ethernet Hardware betrieben werden, die nicht mit den Erweiterungen für FCoE ausgestattet ist (siehe Kapitel 3.2.2.4).

Auf Server-Seite übernimmt Software das Kapseln und Extrahieren von FC-Rahmen. Dadurch kann FCoE auf virtuellen Servern entweder mittels Software Initiator, oder als emulierter lokaler Hintergrundspeicher umgesetzt werden.

Die Virtualisierung von I/O-Kanälen wird hier auf verschiedene Weisen umgesetzt. Die Ethernet-Infrastruktur wird mit der Markierung der Ethernet-Rahmen durch VLAN-Tags in logische Infrastrukturen aufgeteilt (vgl. Kapitel 3.2.2.2). Ethernet kapselt FC-Datenverkehr und ist demnach nur für die Datenübertragung zuständig. Um I/O-Kanäle vollständig zu virtualisieren, ist es zusätzlich nötig, den Zugriff auf Speichereinheiten zu steuern. Dazu können entweder virtuelle Speichereinheiten und FC-Zonen (siehe Kapitel 3.2.2.1) eingerichtet werden, oder der Zugriff auf LUNs wird mit LUN-Masking reglementiert. LUN-Masking wird durch den Hypervisor oder den Speicherknoten realisiert (vgl. Kapitel 3.2.3.2). Eine Zusammenfassung über die wichtigsten Eigenschaften dieser Technik zeigt Tabelle 4.7. Tabelle 4.8 liefert eine Übersicht über den Abgleich von Anforderungen mit dieser Technik.

Abgleich von Anforderungen

Ethernet VLANs werden durch eine Markierung der Rahmen implementiert (Anforderung #2). Eine virtuelle Infrastruktur ist daher eindeutig durch die VLAN-ID identifizierbar. Zonen in FC werden zentral durch den Management-Server verwaltet und sind ebenfalls eindeutig identifizierbar (Anforderung #1).
Die Filterung von Datenströmen ist nach dem IEEE-Standard-802.1Q möglich [iee06]. Wenn mehrere virtulle Server dieselbe physische Hardware benutzen, muss die Virtualisierungsschicht ankommende Rahmen an den entsprechenden virtuellen Server weiterleiten (Anforderung #3).
Bei FCoE können virtuelle Infrastrukturen technologieübergreifend angelegt werden. Dabei

verfügt jeder FCoE Switch über eine Zuordnung von FC-Zonen zu VLAN-IDs und kann auf diesem Weg die Isolation des Datenverkehrs über die Grenzen einer Übertragungstechnologie hinweg realisieren (Anforderung #4).
Für die technologieübergreifende Durchsetzung der Zusicherungen, kommt das DBX Protokoll zum Einsatz. Mit diesem Protokoll kommunizieren Ethernet- und FC-Switche untereinander, um Verbindungseigenschaften zu synchronisieren (Anforderung #5).
Aktuelle Switche verfügen über Management-Zugänge, über die auch Daten hinsichtlich Nutzung und Auslastung abgerufen werden können (Anforderung #6 und #7).
Analog zur WWN in Kapitel 4.2.1 ist die MAC-Adresse das identifizierende Merkmal eines Ethernet NICs. Jeder virtuelle Server besitzt eine eigene MAC-Adresse. Die Zuweisung und Kontrolle übernimmt auch hier der Hypervisor (Anforderung #8). Analog zu Kapitel 4.2.1 werden die Forderung nach einem Management-Zugang (Anforderung #10) und die Forderung einer Zugangskontrolle für den Management-Zugang (Anforderung #11) durch den Hypervisor erfüllt.
Virtuelle Infrastrukturen können sich überlappen, indem die Switche und der Hypervisor erlauben, dass mehrere VLAN-IDs zu einem bestimmten Server weitergeleitet werden (Anforderung #12).
Die Anzahl der virtuellen Infrastrukturen ist durch die Länge der VLAN-ID im VLAN-Tag beschränkt. Für die VLAN-ID sind 12 Bit vorgesehen. Höchstens 2^{12} (4096) unterschiedliche VLANs sind möglich. Für die Anwendung im Rechenzentrum ist diese Grenze als ausreichend und die Anforderung wird als erfüllt betrachtet (Anforderung #13).
Ethernet erlaubt das redundante Betreiben von NICs (Anforderung #14). Dies geschieht entweder durch „Link-Aggregation“ oder durch Ringbildung. Bei der Link-Aggregation werden mehrere physische Verbindungen zu einer logischen Verbindung gebündelt. Fällt eine physische Verbindung aus, wird der Datenverkehr über eine andere physische Verbindung geleitet. Dabei bleibt die logische Verbindung erhalten und der Betrieb kann fortgesetzt werden [Uni09a].
Bei der Ringbildung werden Ethernet Switche derart miteinander verbunden, dass es von einem Endpunkt zu einem anderen mehrere Pfade geben kann. Durch das STP werden einzelne physische Verbindungen deaktiviert, dass Kreise unterbrochen werden. Fällt eine physische Verbindung aus, reaktiviert STP eine zuvor deaktivierte physische Verbindung, um die Verbindung wieder herzustellen [Uni09a] (Anforderung #15).
Verändert man die Konfiguration aktueller Hardware, treten diese Änderungen in der Regel sofort in Kraft. Erfüllt wird diese Anforderung lediglich durch die eingesetzte Hardware. Es gibt keine Spezifikation bezüglich Ethernet, die ein derartiges Verhalten vorsieht oder behindert (Anforderung #16).
Die MAC-Adresse ermöglicht die eindeutige Adressierung eines NIC. Die Anforderung I/O-Hardware eindeutig adressieren zu können, ist erfüllt (Anforderung #17).
Die für FCoE eingeführten Erweiterungen zum Ethernet-Standard ermöglichen die Übertragung von vollständigen Blöcken ohne Fragmentierung (Anforderung #19) und die Regulierung des Datenstroms (Anforderung #9). Durch eine Checksumme wird sichergestellt, dass die Datenintegrität erhalten bleibt. So können Rahmen verlustfrei durch das Netz übertragen werden (Anforderung #18). Durch das Vermeiden von Datenverlusten und die Beschränkung des Datenverkehrs auf exakt einen Kommunikationsweg durch STP ist zusätzlich sichergestellt, dass die Rahmen in der richtigen Reihenfolge übertragen werden (Anforderung #20).
Der IEEE-802.1Q-Standard sieht vor, dass innerhalb eines jeden VLANs STP zum Einsatz kommt. Für jedes VLAN wird demnach ein eigener Spanning-Tree gefunden. Des-

halb kann die Generierung eines Spanning-Trees durch die Zuordnung von VLAN-IDs zu Ethernet-Ports gesteuert werden. So können für unterschiedliche virtuelle Infrastrukturen unterschiedliche Kommunikationspfade festgelegt werden (Anforderung #21). Das Spanning-Tree-Protokoll deaktiviert alle Verbindungen, mit denen innerhalb eines VLANs alternative Pfade zwischen zwei bestimmten Endpunkten gebildet werden können. Deshalb gibt es beim Einsatz von STP in einem Ethernet-Netz genau einen möglichen Kommunikationspfad zwischen zwei Endpunkten. Anforderung #22 ist dadurch nicht erfüllt.

SCSI/Fibre Channel/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch FC-Header
→ Filterung des Ethernet Verkehrs anhand von VLAN-ID

Flusskontrolle

→ Zusätzliche Spezifikationen erweitern Ethernet um Flusskontrolle um das Verhalten von FC zu imitieren
→ Erweiterungen ermöglichen verlustfreie Rahmenübertragung
→ DBX Protokoll für durchgängige Flusskontrolle in Ethernet- und FC-Netzen

---

Konfiguration

→ VLAN-ID frei wählbar
→ Zuordnung von FC-Adresse zu Ethernet-Adresse
→ Eigenschaften der Flusskontrolle für jedes VLAN einzeln konfigurierbar
→ Redundanz durch Spanning-Tree

Integration

→ FC-Nutzdaten in Ethernet-Rahmen interagieren nicht mit LAN-IP-Datenverkehr
→ FC- und Ethernet-Komponenten werden getrennt voneinander Konfiguriert
→ Kombination mit vorhandener FC-Infrastruktur möglich
→ Kombination mit nicht FCoE fähiger Hardware möglich, ohne QoS

Tabelle 4.7: Zusammenfassung der Eigenschaften von FCoE

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit der MAC-Adresse		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang		x	
#11	Management-Zugangskontrolle		x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung	x		
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung			x

Tabelle 4.8: Abgleich der Anforderungen mit SCSI über FCoE

4.2.4 iSCSI, RDMA über InfiniBand

Die Kombination iSCSI via RDMA zu übertragen, ist als iSER im Anhang der Spezifikation der InfiniBand Architektur festgelegt (vgl. Kapitel 3.2.2.3). Dabei wird TCP als Transportprotokoll von RDMA ersetzt, um die Vorteile der InfiniBand Architektur, wie zum Beispiel die Implementierung einer Transportschicht auf Hardware-Ebene, besser nutzen zu können. Die wichtigsten Eigenschaften dieser Technik sind in Tabelle 4.9 zusammengefasst.

Aktuell verfügbare Infrastrukturen setzen InfiniBand als LAN-Technologie ein und nicht als System Area Network, wie in der Spezifikation beschrieben und in Abbildung 3.10 dargestellt. Ein HCA ist analog zu einem HBA, an das I/O-Subsystem (siehe Kapitel 3.2.1.2) eines physischen Servers angebunden. Die Vorteile von RDMA können auch in dieser Anordnung genutzt werden, vorausgesetzt das I/O-Subsystem unterstützt den direkten Hauptspeicherzugriff (DMA). Die heutzutage gängige Technologie für das I/O-Subsystem, PCI Express, unterstützt DMA. Tabelle 4.10 liefert eine Übersicht über den folgenden Abgleich von Anforderungen mit dieser Technik.

Abgleich von Anforderungen

Virtuelle Infrastrukturen in einem IB-Netz (Partitionen) werden durch eine Partitionskennung im BTH eindeutig identifiziert (Anforderung #1). Jeder IB-Rahmen einer RDMA-Verbindung enthält einen solchen Header und ist so immer markiert (Anforderung #2).
Partitionskennungen werden in einer *Partitionskennungstabelle* (P_Key Table) gespeichert. Ein Switch hält für jeden Port solche eine separate Tabelle vor, in der ausschließlich die Partitionskennungen geführt werden, die ein Port nutzen darf. Rahmen mit einer nicht gelisteten Partitionskennung werden verworfen. Da eine Partitionskennungstabelle mehrere Partitionskennungen enthalten kann, können sich virtuelle Infrastrukturen überlappen (Anforderung #12) und Datenströme können gefiltert werden (Anforderung #3). Bei dieser Technik ist InfiniBand die einzige Technologie zur Datenübertragung, wodurch die Anforderungen #4 und #5 trivialer weise erfüllt sind.
Zur Überwachung von Datenströmen kann ein QoS-Manager, oder der Management-Zugang von Switchen benutzt werden. Auch der parallele Einsatz von beidem ist denkbar (Anforderung #6). Die Auslastung eines Kommunikationswegs kann entweder durch einen QoS-Manager oder durch die Auswertung der von den Switchen bezogenen Informationen gemessen werden (Anforderung #7).
In einem IB-Netz wird ein Server durch eine globale IPv6-Adresse identifiziert. Zusätzlich zu der ständig verfügbaren globalen Adresse, die aus der GUID des Ports generiert wird, können einem Port weitere globale IPv6-Adressen (GID) durch den Switch zugewiesen werden. Das identifizierende Merkmal eines Servers im IB-Netz kann verändert werden (Anforderung #8).
Für die Regulierung einzelner Datenströme können auf einer physischen Verbindung bis zu 15 *Virtual Lanes* (VL) betrieben werden. Ein Switch kann die Übertragungsrate für jede VL individuell regulieren. Die Zuordnung zu einer VL erfolgt anhand des SL-Feldes im LRH (Anforderung #9).
Während die LAN-Einstellungen eines Servers durch den Switch beeinflusst werden können, ist für SAN-Einstellungen nicht explizit möglich. Im Falle von virtuellen Servern wird deshalb ein Management-Zugang durch den Hypervisor benötigt (Anforderung #10). Der Management-Zugang eines Switches kann durch Partitionierung gesichert werden (Anfor-

derung #11).
Eine Partitionskennung ist 16 Bits lang, wodurch 2^{16} verschiedene Partitionen möglich sind. Die Kennungen $0xFFFF$, $0x0000$ und $0x8000$ sind reserviert und können nicht verwendet werden. Somit bleiben $2^{16} - 3$ mögliche Partitionskennungen. Analog zu Ethernet (siehe Kapitel 4.2.2) wird dies als ausreichend betrachtet um Anforderung #13 als erfüllt anzusehen.
IB unterstützt Multipathing. HCAs können deshalb redundant betrieben werden, indem ihnen dieselbe GID zugewiesen wird (Anforderung #14). Multipathing erlaubt es, mehrere Kommunikationspfade zwischen zwei Ports zu betreiben. Welchem Pfad ein Rahmen folgt, hängt allgemein von der Konfiguration ab. Dadurch kann ein IB-Netz aufgebaut werden, in dem der Ausfall einer einzelnen Komponente keine Auswirkungen auf die Verfügbarkeit von Ressourcen und Server hat (Anforderung #15).
Die IBA sieht für die Konfiguration des Netzes vor, dass sich Konfigurationsänderungen sofort durch das Netz propagieren und umgehend in Kraft treten (Anforderung #16).
Der GUID macht einen HCA und einzelne Ports für Administratoren eindeutig adressierbar (Anforderung #17).
Jeder Switch überprüft anhand einer Checksumme, ob ein Rahmen korrekt übertragen wurde. Durch die Regulierung des Datenstroms (Anforderung #9) wird vor dem Rahmenversand sichergestellt, dass ein Rahmen auch angenommen werden kann, damit Rahmen verlustfrei durch das Netz übertragen werden (Anforderung #18).
Bei der Datenübertragung per RDMA kann ein Rahmen komplette Blöcke des Hintergrundspeichers fassen (Anforderung #19).
Durch Multipathing ist es möglich, dass Rahmen in anderer Reihenfolge am Ziel ankommen, als sie verschickt wurden. Da RDMA den direkten Zugriff auf Hauptspeicher vorsieht, beinhaltet diese Technologie nicht das Problem der beschränkten Größe von Lesepuffern. Dadurch ist Anforderung #20 auf diese Technologie nicht anwendbar.
Ein Switch oder Router kann bei der Wahl des Kommunikationspfades die Partitionskennung des BTH berücksichtigen (Anforderung #21).
Der QoS-Manager kann die Entscheidung eines Switches oder Routers bei der Pfadwahl beeinflussen. Da der QoS-Manager auch die Datenströme überwacht, kann er Switche und Router in Abhängigkeit der gemessenen Werte steuern. Somit kann Last dynamisch auf mehrere physische Verbindungen verteilt werden (Anforderung #22).

iSCSI/RDMA/InfiniBand

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IQN
→ Filterung des InfiniBand-Verkehrs anhand von Partitionskennung
→ Isolierung eines Channels durch weitere Kennungen, die beim Verbindungsaufbau vereinbart werden

Flusskontrolle

→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Höchstens 15 virtual Lanes/Service Level/Buffer pro physischer Verbindung
→ Multipathing zur Lastverteilung
→ QoS-Manager kann Pfadwahl dynamisch anpassen

Konfiguration

→ Partitionskennung frei wählbar
→ Mehrere GIDs im IPv6-Format pro Port möglich
→ GIDs werden vom Switch zugewiesen
→ Ein separater QoS-Regelsatz pro Channel möglich
→ Redundanz durch Multipathing

Integration

→ Hoher Grad an Isolation von Channels
→ Nutzung der ISO-OSI-Schichten 1-4:
- Kapselung von IP in Transportschicht (IPoIB) nach RFC 4391
- TCP-ähnliche Datenübertragung per RDMA (SDP)
- iSCSI per RDMA (iSER)

→ Nutzung der ISO-OSI-Schichten 1-3:
- IPv6 Datenverkehr kann nativ durch das Netz übertragen (RAW IPv6)

→ Nutzung der ISO-OSI-Schichten 1-2: möglich (RAW Ethertype)

Tabelle 4.9: Zusammenfassung der Eigenschaften von iSCSI, RDMA über InfiniBand

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit des GUID		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang	x	x	
#11	Management-Zugangskontrolle	x	x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung (nicht anwendbar)			
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung	x		

Tabelle 4.10: Abgleich der Anforderungen mit iSCSI, RDMA über InfiniBand

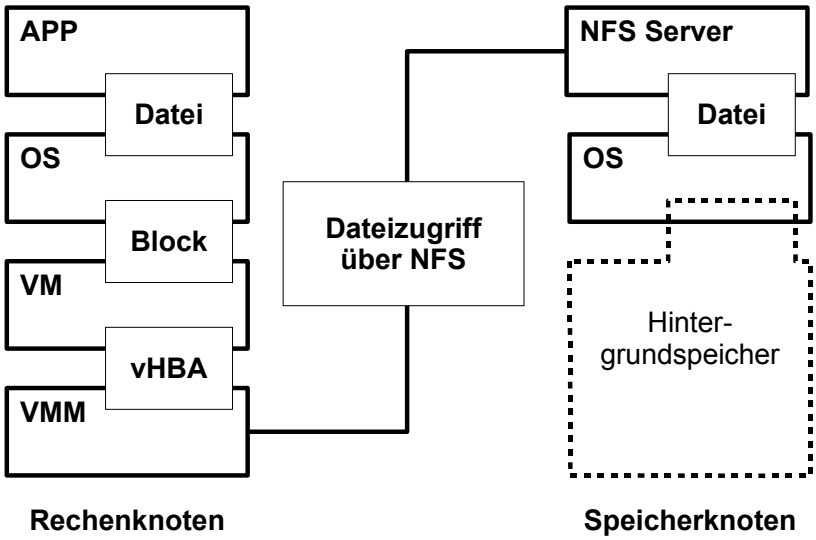


Abbildung 4.2: Dateibasierter Zugriff auf Hintergrundspeicher über NFS

4.2.5 NFS, UDP/IP über Ethernet

Mit dem NFS Protokoll wird ein Dateizugriff direkt über ein Netz an eine Speichereinheit übertragen, ohne diesen in Blockzugriffe umzuwandeln. Der Vorteil dieser Methode besteht darin, dass lediglich der angeforderte Teil einer Datei übertragen wird und nicht die Dateisystemblöcke in denen die Datei abgelegt ist. Dadurch müssen zum Einen weniger Daten übertragen werden, zum Anderen wird der Aufwand, die Daten auszulesen, an den Speicherknoten ausgelagert. Bei der Servervirtualisierung können sowohl Hypervisor als auch virtuelle Server per NFS mit Hintergrundspeicher verbunden werden.

Ein Hypervisor wird in der Regel bei der DAS-Emulation mit Speicherknoten verbunden. Dabei sind die Speichereinheiten, die den virtuellen Servern zur Verfügung gestellt werden, als Dateien auf dem Speicherknoten abgelegt. Abbildung 4.2 zeigt die Zwischenschritte beim Dateizugriff durch einen virtuellen Server. Die Kästen mit einem feinen Rahmen repräsentieren die Interaktion zwischen zwei Systemen und die Art und Weise, in der ein Dateizugriff kommuniziert wird. „Datei“ und „Block“ werden in der Abbildung als Kurzform für Datei- und Blockzugriff verwendet. „vHBA“ bezeichnet den Hardwarezugriff auf einen emulierten HBA. Bei einem Dateizugriff eines virtuellen Servers wandelt dessen Betriebssystem diesen zunächst in Blockzugriffe um und überträgt sie über den emulierten HBA an den Hypervisor. Dieser generiert aus den Blockzugriffen Dateizugriffe, die per NFS an einen Speicherknoten übertragen werden. Auf dem Speicherknoten werden diese erneut in Blockzugriffe transformiert und der eigentliche Zugriff ausgeführt.

Die alternative Anwendungmöglichkeit besteht darin, dass ein virtueller Server direkt mit dem Speicherknoten interagiert. Ein virtueller Server muss hierbei über ein minimales Betriebssystem verfügen, um über ein Netz eine NFS-Ressource einbinden zu können. Dieses wird den virtuellen Servern als DAS oder automatisch beim Bootvorgang über das LAN zugänglich gemacht. Der Hypervisor nimmt hier keine besondere Rolle ein, sondern leitet den NFS-Datenverkehr analog zu anderem Datenverkehr weiter.

Die wichtigsten Eigenschaften dieser Kombination sind in Tabelle 4.11 zusammengefasst.

Eine Übersicht über den folgenden Abgleich dieser Kombination mit den Anforderungen aus Kapitel 2.5 liefert Tabelle 4.12. Analog zu Kapitel 4.2.2 wird auch hier unterschieden, ob eine Anforderung von Ethernet oder den darauf aufbauenden Protokollstapel erfüllt wird.

Abgleich von Anforderungen

Durch die Markierung von Ethernet-Rahmen durch VLAN-IDs und den im IEEE-Standard-802.1Q definierten Funktionen sind folgende Anforderungen analog zu Kapitel 4.2.2 erfüllt: #1, #2, #3, #6, #12, #13, #14, #15, #16, #17 und #21. Bei dieser Technik ist Ethernet die einzige Technologie zur Datenübertragung, wodurch die Anforderungen #4 und #5 trivialer weise erfüllt sind. Die Auslastung einer logischen Verbindung zwischen Server und LUN kann nicht gemessen werden, da ein Ethernet-Switch nicht zwischen logischen Verbindungen unterscheidet (Anforderung #7).
Die MAC-Adresse ist das identifizierende Merkmal eines Ethernet NICs. Jeder virtuelle Server besitzt eine eigene MAC-Adresse, deren Zuweisung und Kontrolle der Hypervisor übernimmt (Anforderung #8). Die Forderung eines Management-Zugangs (Anforderung #10) und die Forderung einer Zugangskontrolle für den Management-Zugang (Anforderung #11) werden ebenfalls durch den Hypervisor erfüllt.
Die Ethernet-Technologie (Kapitel 3.2.2.2) unterstützt nicht die aktive Regulierung von Datenströmen (Anforderung #9). Durch Glättung des Datenverkehrs (Traffic Shaping) wird die Geschwindigkeit, mit der IP-Pakete verschickt werden, kontrolliert [Tan07]. Da diese Methode kein fester Bestandteil von Ethernet ist, kann man nicht davon ausgehen, dass jeder Switch diese Methode implementiert.
Diese Technik hat keine Möglichkeiten die verlustfreie Datenübertragung sicher zu stellen (Anforderung #18). Durch die Erweiterung mit Jumbo Frames können große Dateifragmente in einem einzelnen Rahmen übertragen werden, analog zu Kapitel 4.2.2. Da es im Vergleich zum blockbasierten Zugriff keine kleinste Einheit zur Datenübertragung gibt, kann die Anforderung #19 in diesem Fall nicht angewendet werden.
Durch den Einsatz von Ethernet kann nicht garantiert werden, dass Datenblöcke in der richtigen Reihenfolge übertragen werden. Auch enthält UDP keine Möglichkeit die Reihenfolge der Übertragung von Paketen festzulegen (Anforderung #20).
Das Spanning-Tree-Protokoll deaktiviert alle redundanten Verbindungen, mit denen innerhalb eines VLANs alternative Pfade zwischen zwei bestimmten Endpunkten gebildet werden können. Deshalb gibt es beim Einsatz von STP in einem Ethernet-Netz immer exakt einen möglichen Kommunikationspfad zwischen zwei Endpunkten. Anforderung #22 kann dadurch nicht erfüllt werden.

NFS/UDP/IP/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IP-Adresse und Dateisystempfad
→ Filterung des Ethernet-Verkehrs anhand der VLAN-ID
→ Filterung der Vermittlungsschichten durch entsprechenden Paketfilter

Flusskontrolle

→ keine

Konfiguration

→ VLAN-ID frei wählbar
→ IP-Adresse ist innerhalb eines VLANs eindeutig
→ NFS-Ressource wird ohne Berücksichtigung von VLANs für IP-Adressen zur Verfügung gestellt
→ Glättung durch Einteilung in Klassen, basierend auf Attributen der Kommunikationsschichten, z.B. IP-Adresse
→ Redundanz durch STP

Integration

→ Kann in jedes IP-Netz integriert werden
→ Muss mit anderem LAN-Verkehr abgestimmt werden

Tabelle 4.11: Zusammenfassung der Eigenschaften von NFS, UDP/IP über Ethernet

	Bezeichnung	Erfüllt durch			
		ETH	**UDP /IP**	**VMM**	**nicht mgl.**
#1	Eindeutiger Bezeichner	x			
#2	Datenstrommarkierung	x			
#3	Datenstromfilterung	x		x	
#4	Technologieübergreifend	x			
#5	Technologieübergreifende Zusicherungen	x			
#6	Datenstromüberwachung	x			
#7	Messbarkeit der Auslastung		x		
#8	Veränderbarkeit der MAC			x	
#9	Flusskontrolle				x
#10	HBA-Management-Zugang			x	
#11	Management-Zugangskontrolle			x	
#12	Überlappung	x			
#13	Unbegrenzt viele virtuelle Infrastrukturen	x			
#14	HBA Redundanz	x			
#15	Aufrechterhaltung der Verfügbarkeit	x			
#16	Sofortige Konfigurationsänderung	x			
#17	Eindeutig adressierbarer HBA	x			
#18	Verlustfreie Datenübertragung				x
#19	Rahmengröße (nicht anwendbar)				
#20	Reihenfolge der Rahmenübertragung				x
#21	Pfadwahl nach virtueller Infrastruktur	x			
#22	Pfadwahl nach Auslastung				x

Tabelle 4.12: Abgleich der Anforderungen mit NFS, UDP/IP über Ethernet

4.2.6 NFS, UDP/IP über InfiniBand

Diese Technik sieht vor IP-Pakete als Nutzdaten der IB-Transportschicht zu übertragen, um alle Eigenschaften von IB nutzen zu können. Dazu wird das gesamte Paket, bestehend aus NFS/UDP/IP, in einen IB-Rahmen eingebettet. Wie in Kapitel 3.2.2.3 beschrieben, implementiert IB die Schichten 1-4 des ISO-OSI-Referenzmodells.

Diese Technik nutzt IB zur Übertragung von IP-Paketen. Betrachtet man die NFS-Interaktion zweier Rechner vom Standpunkt der InfiniBand Architektur, ist der Protokollstapel NFS/UDP/IP die Applikation, die IB nutzt. Vom Standpunkt der NFS-Endpunkte aus, deckt IB die Schichten 1 und 2 des ISO-OSI-Referenzmodells ab. IB wird hier zur Sicherungsschicht abstrahiert. RFC 4391 enthält die Spezifikation des Betriebs von IP über InfiniBand (IPoIB) [CK06].

Um IB als Sicherungsschicht zu nutzen, sind viele IB-Betriebsmodi denkbar. Um alle Eigenschaften von IB verwenden zu können, wie zum Beispiel die Partitionierung der Infrastruktur, werden Funktionen aller IB-Schichten genutzt. Der gewählte Übertragungsdienst der IB-Transportschicht ist *Unreliable Datagram* (UD). Die Überlegungen, die zu dieser Wahl führten sind in RFC 4392 zusammengefasst [Kas06].

Die wichtigsten Eigenschaften dieser Technik sind in Tabelle 4.13 zusammengefasst. Eine Übersicht über den folgenden Abgleich dieser Technik mit den Anforderungen aus Kapitel 2.5 liefert Tabelle 4.14. Analog zu den Kapiteln 4.2.2 und 4.2.5 wird auch hier unterschieden, ob eine Anforderung von Ethernet oder den darauf aufbauenden Protokollstapel erfüllt wird.

Abgleich von Anforderungen

Diese Technik nutzt alle in InfiniBand implementierten Schichten, analog zu iSER/InfiniBand (siehe Kapitel 4.2.4). Da der IB-Protokollstapel sehr viele Funktionen bietet, wird ein Großteil der Anforderungen bereits durch das IB-Netz erfüllt. Deshalb ist die Erfüllung der meisten Anforderungen analog zu Kapitel 4.2.4.
IPoIB nutzt den IB-Transportdienst UD. Dieser Dienst verfügt über keine Methoden, um die verlustfreie Datenübertragung zu garantieren (Anforderung #18). Auch überprüft UD nicht die Reihenfolge, in der Rahmen empfangen werden (Anforderung #20).
Beim dateibasierten Zugriff auf Hintergrundspeicher wird der Inhalt einer Datei übertragen. Im Gegensatz zum blockbasierten Zugriff, bei dem ein Block eine bestimmte Größe aufweist, gibt es in diesem Fall keine Mengeneinheit. Da die Größe der zu übertragenden Einheiten nicht festgelegt ist, kann die Anforderung #19 hier nicht angewendet werden.

NFS/UDP/IP/InfiniBand

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IP-Adresse und Dateisystempfad
→ Filterung des InfiniBand-Verkehrs anhand von Partitionskennung
→ Filterung der Vermittlungsschichten durch entsprechenden Paketfilter

Flusskontrolle

→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Höchstens 15 virtual Lanes/Service Level/Buffer pro physischer Verbindung
→ Multipathing zur Lastverteilung
→ QoS-Manager kann Pfadwahl dynamisch anpassen

Konfiguration

→ Partitionskennung frei wählbar
→ IP-Adresse ist innerhalb einer Partition eindeutig
→ NFS-Ressource wird ohne Berücksichtigung von Partitionen für IP-Adressen zur Verfügung gestellt
→ Ein separater QoS-Regelsatz pro Channel möglich
→ Redundanz durch Multipathing

Integration

→ Hoher Grad an Isolation von Channels
→ Nutzung der ISO-OSI-Schichten 1-4:
- Kapselung von IP in Transportschicht (IPoIB) nach RFC 4391
- Durch Kapselung ist IPoIB unabhängig von anderer InfiniBand nutzung

Tabelle 4.13: Zusammenfassung der Eigenschaften von NFS, UDP/IP über InfiniBand

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x		
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit des GUID		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang	x	x	
#11	Management-Zugangskontrolle	x	x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung			x
#19	Rahmengröße (nicht anwendbar)			
#20	Reihenfolge der Rahmenübertragung			x
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung	x		

Tabelle 4.14: Abgleich der Anforderungen mit NFS, UDP/IP über InfiniBand

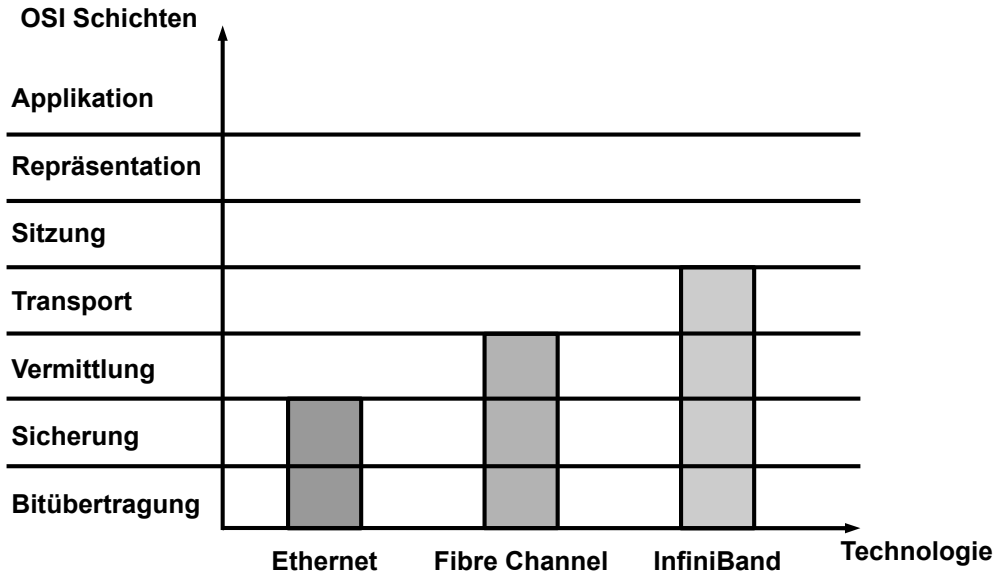


Abbildung 4.3: Abdeckung von Schichten des ISO-OSI-Referenzmodells durch Ethernet, Fibre Channel und InfiniBand

4.3 Gegenüberstellung der Kombinationen

Dieses Kapitel vergleicht die Anforderungserfüllung der unterschiedlichen Techniken aus Kapitel 4.2. Unterschiede sind insbesondere bei den Anforderungen zu erkennen, die Flusskontrolle, Zuverlässigkeit und Effizienz betreffen. Dieser Unterschied liegt darin begründet, dass Ethernet, Fibre Channel und InfiniBand unterschiedlich viele Schichten des ISO-OSI-Referenzmodells abdecken können. Je mehr Schichten eine Technologie abdeckt, desto höher ist der Funktionsumfang eines Switches, wodurch ein Datenfluss präziser gesteuert werden kann. Wie in Kapitel 3.2.2 erläutert, ist es mit Ethernet möglich, die Schichten 1 und 2 des ISO-OSI-Referenzmodells abzudecken, während Fibre Channel bis einschließlich Schicht 3 und InfiniBand sogar bis Schicht 4 reicht (vgl. Abbildung 4.3).

Nach der Gewichtung der Anforderungen in Kapitel 2.5 ist jede der untersuchten Techniken geeignet, virtuelle Infrastrukturen zu implementieren. Jede Technik ist in der Lage, ein physisches Netz in isolierte, logische Segmente zu unterteilen. Bei der Servervirtualisierung ist man immer auf Funktionen der Virtualisierungsschicht angewiesen, wenn ein HBA nicht für die parallele Nutzung durch viele virtuelle Server ausgelegt ist.

Die Datenstromüberwachung eines Switch-Ports ist im Wesentlichen eine Hardwareanforderung und kann lediglich zur Bestimmung der Auslastung einer physischen Verbindungen eingesetzt werden. Die Bestimmung der Auslastung einer logischen Verbindung erfordert eine Komponente, die zwischen logischen Verbindungen unterscheiden kann. In diesem Fall haben Fibre Channel und InfiniBand einen Vorteil gegenüber Ethernet, da diese auch höhere Schichten des ISO-OSI-Referenzmodells abdecken und die Switche zwischen Verbindungen auf diesen Schichten unterscheiden können. Da InfiniBand die meisten Schichten abdeckt (vgl. Abbildung 4.3), kann bei dieser Form die Auslastung differenzierter bestimmt werden, als bei anderen betrachteten Technologien.

Eine ähnliche Technologiereihenfolge ergibt sich auch beim Vergleich der möglichen Fluss-

kontrolle. Ethernet verfügt ohne die Erweiterungen, die für FCoE eingeführt wurden, über keine Möglichkeiten der Flusskontrolle. Durch den Einsatz von TCP kann Flusskontrolle zwischen den Endpunkten stattfinden. Ein Endpunkt kann Übertragungsprobleme erkennen und als Reaktion darauf seine Sendegeschwindigkeit anpassen (Congestion Avoidance) [Tan07]. Mit den Erweiterungen für FCoE kann Ethernet, wie TCP, Übertragungsengpässe erkennen und darauf reagieren. Zusätzlich sehen die FCoE Erweiterungen vor, dass Ethernet-Switche Engpässe kommunizieren können, so dass diese vermieden werden können, bevor es zu Kommunikationsproblemen kommt (siehe Kapitel 3.2.2.2). Fibre Channel benutzt Credit-Systeme, die eine Übertragung weiterer Rahmen nicht zulassen, sofern eine physische Verbindung maximal ausgelastet ist. Dafür werden Buffer-to-buffer-Systeme eingesetzt, mit denen die Auslastung einer physischen Leitung zwischen zwei Ports überwacht wird, sowie End-to-end-Systeme, mit denen die Übertragungsgeschwindigkeit zwischen zwei Endpunkten reguliert wird. FC ordnet jede Endpunkt-zu-Endpunkt-Verbindung einer Verkehrsklasse zu, die bestimmt, wie Credits verteilt werden. Die präziseste Flusskontrolle ist bei InfiniBand gegeben. Über eine physische Verbindung werden bis zu 15 Virtual Lanes (VL) betrieben, wobei jede VL über ein Credit-System reguliert wird. Der LRH eines IB-Rahmen enthält einen Wert für ein Service Level (vgl. Kapitel 3.2.2.3). Jeder Switch entscheidet anhand von Absender, Empfänger und Service Level über welche VL ein Rahmen weitergeleitet wird. Statt eines einzelnen Buffer-to-buffer-Systems, wie FC, stehen hier 15 solcher Systeme zur Flusskontrolle zur Verfügung.

Die verlustfreie Datenübertragung und die Übertragung von Rahmen in der richtigen Reihenfolge sind eng miteinander verknüpft, da es meistens exakt einen Kommunikationspfad zwischen zwei Endpunkten gibt. Bei genau einem Kommunikatiospfad, über den alle Rahmen verlustfrei übertragen werden, kommen die Rahmen in identischer Reihenfolge am Ziel an, in der sie gesandt wurden. Ist die Datenübertragung verlustbehaftet, können einzelne Rahmen verloren gehen. Diese müssen erneut übertragen werden. Somit hängt die Reihenfolge in der Rahmen empfangen werden davon ab, ob die Datenübertragung verlustbehaftet ist oder nicht. Diese enge Verknüpfung gilt für die Ethernet- und FC-basierten Kombinationen. Durch fortlaufende Nummerierung der Rahmen kann der Rahmenverlust erkannt und eine erneute Übertragung angefordert werden. Zusätzlich können nummerierte Rahmen umsortiert werden, damit nicht alle Rahmen erneut übertragen werden müssen. TCP kann den Datenverlust erst am Endpunkt feststellen, wohingegen FC bereits während der Übertragung von Switch zu Switch sicherstellt, dass alle Rahmen übertragen wurden. Bei einem Datenverlust in einem FC-Netz übernimmt der vorausgehende Switch die erneute Übertragung, so dass der Endpunkt, der die Rahmen ursprünglich verschickt hat, entlastet wird. InfiniBand überprüft Sequenznummern ähnlich, wie TCP, erst in den Endpunkten. In einem IB-Netz kann es mehrere Pfade zwischen zwei Punkten geben. So kann es passieren, dass Pakete auf einem wenig ausgelasteten Pfad andere Pakete auf einem mehr ausgelasteten Pfad „überholen“. Lässt man dieses Verhalten zu, kann erst der Endpunkt Rahmenverlust erkennen. Zusätzlich muss ein Endpunkt damit umgehen können, dass Rahmen in einer beliebigen Reihenfolge ankommen können.

Tabelle 4.15 fasst die tabellarischen Abgleiche der vorausgehenden Kapitel zusammen. Die Unterschiede der Technologien sind an den Anforderungen #7 und #9 sowie #18 bis #22 deutlich erkennbar.

Wie diese Gegenüberstellung veranschaulicht, zeigt sich der deutlichste Unterschied hinsichtlich der untersuchten Kombinationen bei der Flusskontrolle. Mit Flusskontrolle wird die Zuweisung von Übertragungsrate zu Verbindungen verwaltet. Je detailierter eine Technolo-

gie zwischen Verbindungen differenzieren kann, desto präziser lässt sie sich steuern und an das Nutzungsprofil des Netzes anpassen.

Kapitel	Tabelle	Bezeichnung
4.2.1	4.4	SCSI/Fibre Channel
4.2.2	4.6	iSCSI/TCP/IP/Ethernet
4.2.3	4.8	SCSI/Fibre Channel/Ethernet
4.2.4	4.10	iSCSI/RDMA/InfiniBand
4.2.5	4.12	NFS/UDP/IP/Ethernet
4.2.6	4.14	NFS/UDP/IP/InfiniBand

Legende

x := erfüllt durch Netz
o := erfüllt durch VMM
- := nicht erfüllt
n.a. := nicht anwendbar

	Bezeichnung	Kombination					
		4.2.1	4.2.2	4.2.3	4.2.4	4.2.5	4.2.6
#1	Eindeutiger Bezeichner	x	x	x	x	x	x
#2	Datenstrommarkierung	x	x	x	x	x	x
#3	Datenstromfilterung	o	o	o	o	o	x
#4	Technologieübergreifend	x	x	x	x	x	x
#5	Techn.-übergreifende Zusicherungen	x	x	x	x	x	x
#6	Datenstromüberwachung	x	x	x	x	x	x
#7	Messbarkeit der Auslastung	x	TCP	x	x	UDP	x
#8	Veränderbarkeit des GUID	o	o	o	o	o	o
#9	Flusskontrolle	x	TCP	x	x	-	x
#10	HBA-Management-Zugang	o	o	o	o	o	o
#11	Management-Zugangskontrolle	o	o	o	o	o	o
#12	Überlappung	x	x	x	x	x	x
#13	Unbegrenzt viele virt. Infrastrukturen	x	x	x	x	x	x
#14	HBA Redundanz	o	x	x	x	x	x
#15	Aufrechterhaltung der Verfügbarkeit	o	x	x	x	x	x
#16	Sofortige Konfigurationsänderung	x	x	x	x	x	x
#17	Eindeutig adressierbarer HBA	x	x	x	x	x	x
#18	Verlustfreie Datenübertragung	x	TCP	x	x	-	-
#19	Rahmengröße	x	x	x	x	n.a.	n.a.
#20	Reihenfolge der Rahmenübertragung	x	-	x	n.a.	-	-
#21	Pfadwahl nach virtueller Infrastruktur	x	x	x	x	x	x
#22	Pfadwahl nach Auslastung	-	-	-	x	-	x

Tabelle 4.15: Zusammenfassung der Abgleiche

5 Bewertung und Verbesserungspotential

Nachdem im vorausgehenden Kapitel Kombinationen vorgestellt und mit den in Kapitel 2.5 abgeleiteten Anforderungen abgeglichen wurden, zeigt dieses Kapitel Defizite und Verbesserungsmöglichkeiten von Konzepten und Technologien auf. Kapitel 5.1 nennt einige Aspekte des dateibasierten Zugriffs auf Hintergrundspeicher. Dabei wird hervorgehoben, dass die vorgestellten Kombinationen einen Vorteil bei der Integration mit anderen Techniken und anderem Datenverkehr haben. Auf der Gegenseite steht der bewusste Verzicht auf Flusskontrolle. Im Anschluss daran diskutiert Kapitel 5.2 den Einsatz von I/O-Servern zur I/O-Konsolidierung. I/O-Server haben großes Potential zur Veränderung von Management-Vorgängen und von Virtualisierungsansätzen. Management-Vorgänge werden durch I/O-Server beeinflusst, da dabei Server keinen direkten Zugang zu LAN und SAN haben. Statt dessen sind sie über ein neues Netz mit mindestens einem I/O-Server verbunden. Zum Schluss wird in Kapitel 5.3 eine Idee für Redundanz von FC-HBAs vorgestellt. Bisherige Ansätze erzeugen Redundanz, indem im Vorfeld für mehrere HBAs ähnliche Konfigurationen angelegt werden und im Fehlerfall das Betriebssystem bzw. der Hypervisor auf dem Server zwischen den HBAs umschaltet. Der eigene Ansatz basiert darauf die 1:1-Relation zwischen WWN und N_Port ID zu einer N:N-Relation zu erweitern. Mit dieser Erweiterung des Name-Servers können automatisch HBA-Konfigurationen aus einer einzigen Konfiguration generiert werden, wodurch der Konfigurationsaufwand gesenkt wird. Die Switche im Netz können die Erweiterungen nutzen, um im Fehlerfall neue Kommunikationspfade zu finden. Dadurch können FC-HBAs redundant betrieben werden und das Betriebssystem bzw. der Hypervisor entlastet werden.

5.1 Dateibasierte Anbindung von Hintergrundspeicher

Die beiden betrachteten dateibasierten Kombinationen NFS/UDP/IP/Ethernet (siehe Kapitel 4.2.5) und NFS/UDP/IP/InfiniBand (siehe Kapitel 4.2.6) stehen stellvertretend für alle Ansätze, die einen Dateizugriff an einen Speicherknoten weiterleiten, ohne diesen zuvor in Blockzugriffe umzuwandeln (vgl. Kapitel 3.2.3.1).

Das Ergebnis der Gegenüberstellung in Kapitel 4.3 ist, dass die Möglichkeiten der Flusskontrolle durch die Anzahl der implementierten Schichten bestimmt werden. Unter diesem Aspekt hat die dateibasierte Anbindung von Hintergrundspeicher den Vorteil, dass dabei allgemeine Vermittlungs- und Transportprotokolle eingesetzt werden. IP-Implementierungen sind für alle untersuchten Netztechnologien (Fibre Channel, Ethernet und InfiniBand) verfügbar. Somit kann jeder IP-basierte Ansatz technologieübergreifend eingesetzt werden. Flusskontrolle wird häufig als Funktion in Transportschichten implementiert, wie zum Beispiel in TCP [Tan07] oder InfiniBand [inf07]. Eine Transportschicht, die Flusskontrolle ermöglicht und auf IP aufsetzt, kann demnach eine Reihe von Anforderungen technologieunabhängig erfüllen.

Das NFS-Protokoll setzt auf UDP/IP auf, womit bewusst gegen den Einsatz von Flusskontrolle, außerhalb der Applikation, entschieden wurde. Die Diskussion der Gründe, die

zu dieser Entscheidung geführt haben, soll nicht Gegenstand dieser Arbeit sein. NFS unterstützt mittlerweile auch den Betrieb über TCP [Smi06] und andere Ansätze, wie zum Beispiel der blockbasierte Ansatz mit iSCSI (siehe Kapitel 4.2.2), setzen ebenfalls auf TCP/IP. Die Entwicklung von NFS über TCP [Smi06], die Entscheidung für TCP bei iSCSI (vgl. Kapitel 4.2.2) und der Einsatz des verbindungsorientierten Betriebsmodus der IB-Transportschicht für RDMA (vgl. Kapitel 3.2.2.3) sind Hinweise darauf, dass UDP bzw. verbindungslose Datagram-Protokolle häufig nicht alle gewünschten Eigenschaften besitzen. Auch gemessen an den Anforderungen, die in dieser Arbeit abgeleitet wurden, ist die Kombination NFS/TCP/IP der Kombination NFS/UDP/IP überlegen, da mit TCP zusätzlich die Anforderungen nach einer verlustfreien (Anforderung #18) und sortierten (Anforderung #20) Paketübertragung erfüllt werden.

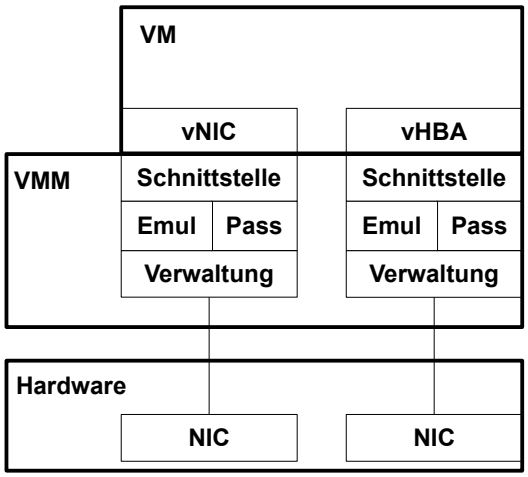
Durch den Einsatz von IP kann NFS, wie oben beschrieben, auf jeder Technologie eingesetzt werden, die im Rahmen dieser Arbeit betrachtet wird. Somit muss für NFS kein spezielles SAN zur Verfügung gestellt werden, wodurch sich NFS zur I/O-Konsolidierung eignet.

Die Stärken von NFS-basierten Ansätzen, die im Rahmen dieser Arbeit ermittelbar sind, beziehen sich auf Integration mit anderen Technologien. Diese Effekte können auch im blockbasierten Ansatz mit iSCSI genutzt werden. Die Eigenschaften, die NFS gegenüber allen blockbasierten Ansätzen besonders hervorheben, sind, dass NFS deutlich einfacher zu konfigurieren ist und dass durch den dateibasierten Zugriff eine höhere Transaktionsrate erreicht werden kann. Beide Eigenschaften sind nicht definitiv belegt, sondern lediglich häufig wiederholte Aussagen auf entsprechenden Internetplattformen, wie zum Beispiel dem Forum auf der VMware Homepage [vmw09].

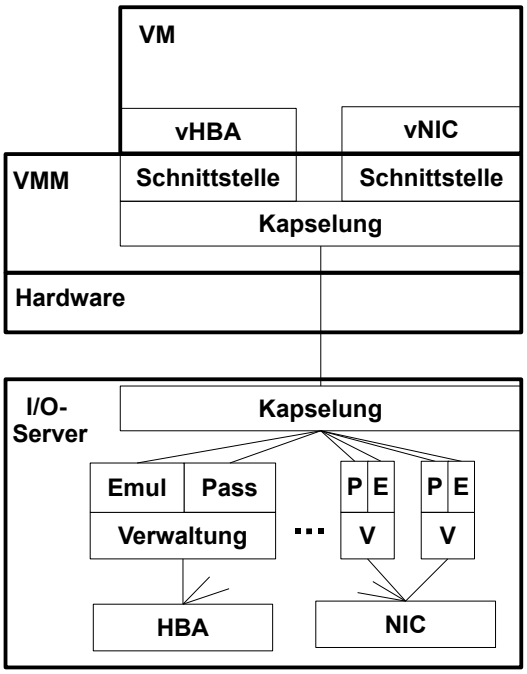
5.2 Hardware-Auslagerung in I/O-Server

Allen in Kapitel 4.2 vorgestellten Kombination gemein ist, dass für die Konfiguration des HBAs der Hypervisor zuständig ist. Wird I/O-Hardware in spezialisierte I/O-Server ausgelagert (siehe Kapitel 3.2.2.4), kann ein HBA unabhängig vom Hypervisor konfiguriert werden. Dadurch kann der Hypervisor weiter entlastet werden. Abbildung 5.1 illustriert die Entlastung eines Hypervisors, indem die Aufgaben zur Virtualisierung von Hardware vom I/O-Server übernommen werden. Die Aufgaben lassen sich unter den Sammelbegriffen Emulation, Weiterleitung (engl. Pass Thru) und Verwaltung zusammenfassen. Wie in Kapitel 3.2.1 beschrieben, versucht ein Hypervisor Aufrufe von virtuellen Servern unverändert an die physische Hardware weiterzuleiten, um den Emulationsaufwand so gering wie möglich zu halten. Zur Verwaltung von virtueller I/O-Hardware zählen Überwachung, Konfiguration und die Verteilung von Datenströmen auf die unterschiedlichen virtuellen Server. In Abbildung 5.1(a) werden diese Aufgaben durch den Hypervisor wahrgenommen. Abbildung 5.1(b) zeigt diese Aufgaben im I/O-Server. Damit die Aufrufe vom Hypervisor in den I/O-Server gelangen, müssen diese über ein Netz übertragen werden.

I/O-Server werden entweder in Verbindung mit den SR- und MR-IOV Ansätzen der PCI-SIG (siehe Kapitel 3.2.1.2) oder durch proprietäre Lösungen umgesetzt. Genaue Spezifikationen zur Funktionsweise von proprietären Entwicklungen, wie zum Beispiel dem Xsigo I/O-Director (siehe Kapitel 3.3), sind nicht einsehbar. Es ist allerdings davon auszugehen, dass die prinzipiellen Funktionen den standardisierten und veröffentlichten Ansätzen der PCI-SIG ähnlich sind (siehe Kapitel 3.2.1.2).



(a) Virtualisierung durch den Hypervisor



(b) Virtualisierung durch den I/O-Server

Abbildung 5.1: Verlagerung von Virtualisierungsaufgaben aus dem Hypervisor in den I/O-Server

Ein Vorteil einer solchen Lösung ist, dass lediglich eine I/O-Komponente pro physischem Server notwendig ist, um einem Server alle I/O-Komponenten des I/O-Servers zugänglich zu machen. Im Vergleich zu derzeit verbreiteten Lösungen, bei denen ein physischer Server jeweils eine I/O-Komponente für LAN und SAN beinhaltet (vgl. HBA und NIC in Abbildung 5.1(a)), kann die Anzahl der I/O-Hardware in einem Server und die dazugehörigen Switche und Verkabelung fast halbiert werden. Anschaffungs- und Wartungskosten können potentiell reduziert werden, da sowohl für die Herstellung als auch für den Betrieb weniger Hardware benötigt wird.

Ein weiterer Vorteil, besteht in der Möglichkeit Aufgaben zur Virtualisierung von Hardware vom Hypervisor in den I/O-Server auszulagern und dadurch den Hypervisor zu entlasten (vgl. Abbildung 5.1(b)). Statt die Aufrufe der nichtprivilegierten Betriebssysteme zu verarbeiten, kapselt der Hypervisor die Aufrufe und überträgt diese an den I/O-Server, der die Verarbeitung übernimmt. Da ein I/O-Server eine selbstständige Einheit ist, kann hier vermehrt Emulation eingesetzt werden, ohne die physischen Server zu beeinträchtigen, vorausgesetzt Volumen und Transaktionsrate werden konstant gehalten.

Der Nachteil beim Einsatz von I/O-Servern liegt darin, dass man zusätzlich zu LAN und SAN ein drittes Netz im Rechenzentrum aufbaut. Ausgehend von den aktuellen Standardtechnologien im Rechenzentrum, Ethernet für LAN und Fibre Channel für SAN, bedeutet dies die Einführung einer dritten Technologie zur Datenübertragung, InfiniBand oder MR-IOV PCIe. Der dadurch steigende Administrationsaufwand im Rechenzentrum schmälert die Vorteile, die durch den Einsatz von I/O-Servern erreicht werden.

Durch Emulation kann ein I/O-Server jeden Aspekt eines virtuellen HBAs kontrollieren. Dies ermöglicht Datenstromfilterung und beliebige Anpassungen der identifizieren Merkmale. Zusätzlich kann das Management physischer HBAs vollständig ohne die Beteiligung von physischen und virtuellen Servern durchgeführt werden. Diese Eigenschaften entsprechen den Anforderungen #3, #8, #10 und #11 aus Kapitel 2.5. Kapitel 4.2 zeigt, dass diese Anforderungen meist mit der Unterstützung des Hypervisors oder vollständig durch den Hypervisor erfüllt. Gemessen an den Aufgaben, die der Hypervisor wahrnehmen muss (vgl. Kapitel 4.1.2), bewirkt eine Auslagerung von I/O-Hardware in I/O-Server eine Verbesserung des Gesamtsystems, durch die mehr Kontrolle über die Infrastruktur bei einer Entlastung des physischen Servers erreicht wird.

5.3 Redundante Fibre Channel Pfade

Verbesserungen sind auch direkt an Technologien möglich. So sieht keine FC-Spezifikation die Möglichkeit HBAs redundant zu betreiben vor. In diesem Kapitel wird eine Idee vorgestellt, wie man FC-Dienste erweitern könnte, um die redundante Konfiguration von HBAs zu ermöglichen.

Für die Redundanz von FC-HBAs existiert das U.S. Patent 7210068 [AFA07], das eine Methode beschreibt, um dieses Problem zu lösen. Darin wird eine Methode beschrieben, bei der Multipathing eingesetzt wird, um Redundanz zu erreichen. Als wichtige Schlüsselfähigkeit gilt hier die Möglichkeit eines Servers alle Pfade, die zur selben LUN führen zu gruppieren. Aus je einer Gruppe wählt er je einen Pfad zur Kommunikation mit einer LUN aus. Die anderen, nicht gewählten Pfade werden ignoriert. Fällt der gewählte Kommunikationspfad aus, so wird aus dessen Gruppe ein anderer Pfad gewählt. Diese Methode wird auf dem Server implementiert und arbeitet ohne Unterstützung durch das Netz. Ein ähnliches Verhalten implementieren auch VMware im ESX Hypervisor [vmw09] und das Device Mapper Multipath I/O Projekt [Lor09]. Eine Spezifikation vom T11, die eine solche oder ähnliche Fähigkeit als Bestandteil von FC realisiert, existiert nicht.

Eine Möglichkeit die Redundanz von HBAs in das FC-Netz zu integrieren, wäre ein entsprechender Dienst auf der FC-3 Ebene. Dadurch erreicht man Redundanz unabhängig vom Betriebssystem des Servers und bei der Konfiguration des Netzes kann explizit Redundanz eingestellt werden, statt diesen Effekt implizit durch ähnliche Konfigurationen zu erzielen. Durch die Unterstützung eines Dienstes im FC-Netz kann so der Verwaltungsaufwand verringert werden.

Die vorhandenen FC-GS Dienste ermöglichen es bereits Konfigurationen zentral zu verwalten und an bestimmte Geräte oder Endpunkte zuzuweisen. Um durch das Netz gestütze Redundanz zu ermöglichen, muss ein *Redundanz-Server* die Aggregation von Endpunkten (N_Ports) zu einer *Redundanzeinheit* erfassen und dieser einen eigenen WWN zuweisen. HBAs werden zu einer Redundanzeinheit zusammengefasst in dem entweder der Redundanz-Server mit den entsprechenden WWNs konfiguriert wird.

Der Redundanz-Server wird dem Management-Server bei der Anmeldung eines N_Ports vorgeschaltet. Dadurch kann automatisch veranlasst werden, dass allen HBAs die Einstellungen der Redundanzeinheit zugewiesen werden. Der damit erzielte Effekt ist, dass die Konfigurationen aller N_Ports (zum Beispiel Kodierung, Übertragungsrate, Zone) der selben Redundanzeinheit einheitlich sind. Dies ist notwendig, um sicher zu stellen, dass HBAs der selben Redundanzeinheit nur solche Verbindungen aufbauen, die auch von jedem anderen

HBA dieser Einheit aufgebaut werden könnten. Dadurch kann im Falle eines Ausfalls ein anderer HBA die Verbindung übernehmen.

Damit im Falle eines Ausfalls ein anderer HBA eine Verbindung übernehmen kann, muss der Kommunikationspfad umgelenkt werden. Da jedes Gerät, insbesondere jeder Switch, selbstständig entscheidet, über welche physische Verbindung ein Rahmen weitergeleitet werden soll, kann eine Verbindung innerhalb des Netzes umgelenkt werden. Die Adressierung eines FC-Netzes ist bereits zweistufig, indem der WWN zur N_Port ID aufgelöst wird. Hier kann man Redundanz für den Ausfall einer Verbindung ermöglichen indem beim Ausfall eines HBAs dessen WWN zur N_Port ID eines intakten HBAs der selben Redundanzeinheit umgeleitet wird.

Redundanz wird schließlich dadurch erzielt, dass ein Name-Server bei der Namensauflösung die Redundanzeinheit berücksichtigt. Dazu muss er zusätzlich zur vorhandenen Auflösung WWN $\mapsto$ N_Port ID auch die Abbildungen N_Port ID $\mapsto$ WWN, WWN $\mapsto$ Redundanzeinheit und Redundanzeinheit $\mapsto$ WWN ermöglichen. Kombiniert man die Abbildungen, kann eine WWN zu jeder N_Port ID der selben Redundanzeinheit aufgelöst werden: WWN $\mapsto$ Redundanzeinheit $\mapsto$ WWN $\mapsto$ N_Port ID.

Die Abbildung N_Port ID $\mapsto$ WWN wird benötigt, damit Rahmen, die sich bereits im Netz befinden, ebenfalls an die neue N_Port ID übertragen werden können. Dies ist notwendig, um die verlustfreie Datenübertragung und die Reihenfolge der Rahmen auch im Falle eines Ausfalls garantieren zu können. Dazu verhindert zuerst der Switch, der den Ausfall feststellt, die Übertragung weiterer Rahmen, mit den vorhandenen Möglichkeiten der Flusskontrolle und benachrichtig im Anschluss daran den Name-Server. Danach kann er mit den erweiterten Funktionen des Name-Servers die neue N_Port ID bestimmen, die Rahmen neu adressieren und weiterleiten. Sind alle Rahmen weitergeleitet, signalisiert er dies dem vorherigen Switch im Kommunikationspfad und nimmt den normalen Betrieb wieder auf. Analog dazu verfährt jeder Switch, der ein entsprechendes Signal erhält. Erst der letzte Switch vor dem Endpunkt erlaubt wieder das Verschicken neuer Rahmen. So kann die Reihenfolge der Rahmen garantiert werden.

Redundanz wird in diesem Ansatz dadurch erreicht, dass die Zuordnung von WWNs zu N_Port IDs im Name-Server keine Bijektion mehr ist. Rahmen werden im Falle eines Ausfalls an einen anderen HBA adressiert. Durch das Vorschalten eines Redundanz-Servers, können Redundanzeinheiten zentral und einheitlich Konfiguriert werden, wodurch der Verwaltungsaufwand verringert wird. Indem die Switche um Funktionen zur Umadressierung von Rahmen erweitert werden, können die verlustfreie Datenübertragung und die Reihenfolge, in der Rahmen zugestellt werden weiterhin garantiert werden.

6 Zusammenfassung und Ausblick

Die Virtualisierung von I/O-Kanälen beschäftigt sich allgemein mit der Konfiguration und Überwachung von Verbindungen in Netzen. Zu den wichtigsten Aspekten zählen hierbei die Art und Weise wie Verbindungen aufgebaut werden und wohin. In dieser Arbeit konnte gezeigt werden, dass bereits heute gängige Techniken in der Lage sind einige der gewünschten Anforderungen effizient zu erfüllen. Neuere Techniken erfüllen zusätzliche Anforderungen, wodurch Rechenknoten entlastet werden können.

Eine Verkettung von Entwicklungen hat dazu geführt, dass I/O-Virtualisierung heute ein wichtiges Thema ist. Diese sind die Markteinführung von 10 Gbps Ethernet NICs, die Entwicklung von FCoE, die Veröffentlichung der PCI-SIG Standards für SR-IOV und MR-IOV und die Nutzung von InfiniBand als Netztechnologie. Im Kern dieser Entwicklung steckt die Tatsache, dass einzelne Hardware-Komponenten, die viel Übertragungsrate bieten, bezahlbar geworden sind. Produkte der unterschiedlichen Ansätze sind jetzt weit genug entwickelt, um in der Praxis eingesetzt zu werden. Da unterschiedliche Ansätze für den selben Zweck eingesetzt werden können, zum Beispiel MR-IOV und InfiniBand für Lösungen mit I/O-Servern, ist es notwendig die unterschiedlichen Techniken zu analysieren und zu vergleichen.

Diese Arbeit untersucht einige Techniken zur Virtualisierung von I/O-Kanälen, basierend auf heute gängigen Methoden. Die ermittelten Anforderungen ermöglichen eine umfassende Bewertung von Techniken zur Virtualisierung von I/O-Kanälen. Die Analyse vorhandener Techniken liefert eine Übersicht über aktuelle Möglichkeiten und schafft eine Vergleichsgrundlage für weitere Untersuchungen. Für eine Analyse von Techniken, die auf I/O-Server basieren, war es im Rahmen dieser Arbeit zu früh, da weder Hardware noch Veröffentlichungen zur Verfügung standen.

Das in dieser Arbeit betrachtete Szenario wurde derart gewählt, dass beim Management der Infrastruktur viele unterschiedliche Aspekte berücksichtigt werden müssen. Dies hat dazu geführt, dass bei der anschließenden Anwendungsfallanalyse eine Vielzahl unterschiedlicher Anforderungen an Techniken zur Virtualisierung von I/O-Kanälen aus den fünf Bereichen des Funktionsmodells der OSI Management Architektur (FCAPS) abgeleitet werden konnten. Die für die Analyse gewählten Anwendungsfälle stammen aus der Untersuchung der Dienstlebenszyklen von physischen und virtuellen Infrastrukturen. Die resultierenden Anforderungen wurden gewichtet, so dass mit deren Hilfe festgestellt werden kann, in welchem Maße eine spezielle Technik zur Virtualisierung von I/O-Kanälen geeignet ist.

Nach der Sammlung von Anforderungen wird in Kapitel 3 auf vorhandene Virtualisierungskonzepte und Produkte eingegangen. Da es sich bei I/O-Kanälen um Verbindungen handelt, müssen sowohl Endpunkte als auch Netze betrachtet werden. Daher wurde in diesem Kapitel zwischen Virtualisierung in Rechenknoten, Netzen und Speicherknoten unterschieden. Auf Speicherknoten wurde im Rahmen dieser Arbeit nicht eingegangen, da dieses Gebiet sehr weit expandiert werden kann. Statt dessen wurden die Möglichkeiten der Interaktion mit einem Speicherknoten betrachtet. Für die Virtualisierung in Rechenknoten und Netzen wird auf unterschiedliche Ansätze und Techniken eingegangen. Methoden für Virtualisierung in Rechenknoten werden in Kapitel 3.2.1 behandelt. Im Anschluss daran geht Kapitel 3.2.2 auf

die Netztechnologien ein.

Kapitel 4 untersucht Kombinationen der zuvor vorgestellten Technologien und gleicht diese gegen die ermittelten Anforderungen ab. Ein Ergebnis dieses Kapitels ist, dass die deutlichsten Unterschiede zwischen den Kombinationen im Bereich der Flusskontrolle entstehen. Demnach sollte zum Zeitpunkt der Technologieauswahl bereits abschätzbar sein in welchem Maß man Flusskontrolle einsetzen möchte. Die Gegenüberstellung der Kombinationen zeigt, dass die Möglichkeiten der Flusskontrolle besonders dadurch bestimmt werden, wie differenziert eine Technologie zwischen einzelnen Verbindungen unterscheiden kann. Für jede Kombination wurden tabellarische Übersichten, über deren wichtigste Eigenschaften und die Erfüllung von Anforderungen, erstellt.

Am Ende der Arbeit geht Kapitel 5 auf Defizite und Verbesserungsmöglichkeiten ein. Dieses Kapitel zeigt, dass es Verbesserungsansätze sowohl bei der Konzeption von Techniken zur Virtualisierung von I/O-Kanälen, als auch beim Funktionsumfang einzelnen Technologien gibt. Besonders hervorzuheben sind hier I/O-Server, die es ermöglichen Hardware aus dem physischen Server, wie er heute eingesetzt wird, herauszulösen. Dadurch kann Hardware abstrahiert und dynamisch zugewiesen werden, ähnlich zu Speichereinheiten in Speicherknoten.

Im Anschluss an diese Arbeit sind weitere Untersuchungen denkbar. Es könnten Leistungsmessungen durchgeführt werden, um Technologiekombinationen anhand von Datendurchsatz und Transaktionsrate zu bewerten. Im Zuge einer solchen Untersuchung sollte das Augenmerk auf das Nutzungsprofil des Hintergrundspeichers gelegt werden. In zahlreichen Blogs und Foren, die dateibasierten und blockbasierten Zugriff auf Hintergrundspeicher diskutieren, wird häufig argumentiert, dass der dateibasierte Zugriff bessere Ergebnisse als der blockbasierte Zugriff liefert, wenn zufällig Daten vom Hintergrundspeicher gelesen werden. Weiter wird ausgeführt, dass sich das Nutzungsprofil des Hintergrundspeichers durch einen Host sehr zufällig verhält, sofern viele virtuelle Maschinen auf diesem Host betrieben werden. Die Konsequenz daraus ist, dass in diesem Fall eine dateibasierte Anbindung von Hintergrundspeicher an den Host mehr Leistung liefert, als eine blockbasierte. Diese Folgerung widerspricht bisher veröffentlichten Messungen und bedarf daher gesonderter Betrachtung.

Ein weiterer Aspekt, der untersucht werden könnte ist die Platzierung der Virtualisierungsschicht. In dieser Arbeit wird meist davon ausgegangen, dass auf jedem Host eine Virtualisierungsschicht in Form eines Hypervisors aufgetragen wird. Kapitel 3.2.1.2 erwähnt die SR-IOV Technologie, mit der ein Teil der Virtualisierungsschicht in die Hardware integriert werden kann. Eine eingehendere Analyse an welcher Stelle eine Virtualisierungsschicht integriert werden kann und die daraus resultierenden Konsequenzen auf Management, Leistung und Sicherheit könnte detailiertere Aussagen zur Virtualisierung von I/O-Kanälen liefern, als es im Rahmen dieser Arbeit möglich war.

Wie bereits erwähnt, entstehen die größten Unterschiede, zwischen den analysierten Kombinationen, im Bereich der Flusskontrolle. Eine weiterführende Arbeit könnte auf dieser Erkenntnis aufbauen und die Bedeutung von Flusskontrolle für Rechenzentren und bestimmte Betriebszenarien untersuchen. Anforderung #22 fordert zwar die Berücksichtigung der Auslastung einer Verbindung bei der Pfadwahl, in der Praxis werden solche Funktionen nach dem Wissensstand des Autors selten genutzt. Dynamische Pfadwahl erschwert das Messen der Auslastung einer logischen Leitung, wodurch die Überprüfung von Dienstvereinbarungen erschwert wird. Weiterführende Arbeiten könnten Lösungen für dieses Problem Entwickeln oder Analysieren.

A Kombinationen

Die Tabellen, die in Kapitel 4.2 Zusammenfassungen der wichtigsten Eigenschaften der Kombinationen zeigen, werden hier erneut aufgeführt. Die Wiederholung dient der Übersicht, so dass die Eigenschaften und Abgleiche gegen die Anforderungen aus Kapitel 2.5 schnell zugänglich sind. Jede Kombination nimmt in diesem Anhang eine Doppelseite ein. Auf jeder linken Seite befindet sich die Zusammenfassung der wichtigsten Eigenschaften einer Kombination, während die rechten Seiten die entsprechenden Übersichten der Abgleiche enthalten. Die Aufzählung der Technologien erfolgt entsprechend der Reihenfolge der Tabelle auf dieser Seite.

Bezeichnung	**Zugriff**
SCSI über Fibre Channel	blockbasiert
iSCSI, TCP/IP über Ethernet	blockbasiert
SCSI, Fibre Channel über Ethernet (FCoE)	blockbasiert
iSCSI, RDMA über InfiniBand	blockbasiert
NFS, UDP/IP über Ethernet	dateibasiert
NFS, UDP/IP über InfiniBand	dateibasiert

SCSI/Fibre Channel

Identifizierung & Filterung

→ Identifizierung einer Ressource durch WWN des Servers und LUN
→ Physische Verbindungen können in Zonen organisiert werden
→ Filterung nicht nötig, da zielgerichtete Weiterleitung

Flusskontrolle

→ End-to-end-credit System um Senderate der Endpunkte zu kontrollieren
→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Feste Bandbreitenzuweisung zu einer Ende-zu-Ende Verbindung (Virtual Circuit)
→ Unterteilung in Klassen (Classes of Service) bestimmt welche Methoden benutzt werden

---

Konfiguration

→ Zentrale Konfiguration über FC-GS Dienste
→ FC-GS Dienste beinhalten Namensauflösung
→ Flusskontrolle für einzelne Verbindungen möglich
→ Redundanz durch Konfiguration mehrerer Pfade und Kontrolle durch einen Endpunkt

---

Integration

→ Spezielle FC-4 Implementierung für andere Protokolle unter anderem für IPv4 und IPv6 sind vorhanden
→ Unterschiedliche FC-4 Implementierungen interagieren nicht

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit der MAC-Adresse		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang		x	
#11	Management-Zugangskontrolle		x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz		x	
#15	Aufrechterhaltung der Verfügbarkeit		x	
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung	x		
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung			x

iSCSI/TCP/IP/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IQN im Sitzungsprotokollheader
→ Filterung des Ethernet-Verkehrs anhand der VLAN-ID
→ Filterung der Vermittlungs-, Transport- und Sitzungsschichten durch entsprechende Paketfilter

---

Flusskontrolle

→ Flusskontrolle in TCP bei Überlastung der Verbindung
→ TCP/IP Verkehrsglättung (Traffic Shaping) in den Endpunkten
→ Flusskontrolle in Switchen nur durch spezielle Ausrüstung

Konfiguration

→ VLAN-ID frei wählbar
→ IP-Adresse zur eindeutigen Identifizierung innerhalb eines VLANs
→ IQN besteht aus Fully Qualified Domain Name (FQDN) und LUN
→ Glättung durch Einteilung in Klassen basierend auf Attributen der Kommunikationsschichten, z.B. IQN, TCP-Port, IP-Adresse
→ Redundanz durch Spanning-Tree

Integration

→ iSCSI setzt auf TCP auf und stellt keine zusätzlichen Anforderungen an tiefere Schichten
→ Muss bei I/O-Konsolidierung mit LAN-IP-Verkehr abgestimmt werden
→ Präzise Steuerung erfordert TCP- und IP-Fähigkeiten in Ethernet Switchen

	Bezeichnung	Erfüllt durch			
		ETH	**TCP /IP**	**VMM**	**nicht mgl.**
#1	Eindeutiger Bezeichner	x			
#2	Datenstrommarkierung	x			
#3	Datenstromfilterung	x		x	
#4	Technologieübergreifend	x			
#5	Technologieübergreifende Zusicherungen	x			
#6	Datenstromüberwachung	x			
#7	Messbarkeit der Auslastung		x		x
#8	Veränderbarkeit der MAC			x	
#9	Flusskontrolle		x		
#10	HBA-Management-Zugang			x	
#11	Management-Zugangskontrolle			x	
#12	Überlappung	x			
#13	Unbegrenzt viele virtuelle Infrastrukturen	x			
#14	HBA Redundanz	x			
#15	Aufrechterhaltung der Verfügbarkeit	x			
#16	Sofortige Konfigurationsänderung	x			
#17	Eindeutig adressierbarer HBA	x			
#18	Verlustfreie Datenübertragung		x		
#19	Rahmengröße	x			
#20	Reihenfolge der Rahmenübertragung				x
#21	Pfadwahl nach virtueller Infrastruktur	x			
#22	Pfadwahl nach Auslastung				x

SCSI/Fibre Channel/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch FC-Header
→ Filterung des Ethernet Verkehrs anhand von VLAN-ID

Flusskontrolle

→ Zusätzliche Spezifikationen erweitern Ethernet um Flusskontrolle um das Verhalten von FC zu imitieren
→ Erweiterungen ermöglichen verlustfreie Rahmenübertragung
→ DBX Protokoll für durchgängige Flusskontrolle in Ethernet- und FC-Netzen

---

Konfiguration

→ VLAN-ID frei wählbar
→ Zuordnung von FC-Adresse zu Ethernet-Adresse
→ Eigenschaften der Flusskontrolle für jedes VLAN einzeln konfigurierbar
→ Redundanz durch Spanning-Tree

Integration

→ FC-Nutzdaten in Ethernet-Rahmen interagieren nicht mit LAN-IP-Datenverkehr
→ FC- und Ethernet-Komponenten werden getrennt voneinander Konfiguriert
→ Kombination mit vorhandener FC-Infrastruktur möglich
→ Kombination mit nicht FCoE fähiger Hardware möglich, ohne QoS

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit der MAC-Adresse		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang		x	
#11	Management-Zugangskontrolle		x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung	x		
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung			x

iSCSI/RDMA/InfiniBand

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IQN
→ Filterung des InfiniBand-Verkehrs anhand von Partitionskennung
→ Isolierung eines Channels durch weitere Kennungen, die beim Verbindungsaufbau vereinbart werden

---

Flusskontrolle

→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Höchstens 15 virtual Lanes/Service Level/Buffer pro physischer Verbindung
→ Multipathing zur Lastverteilung
→ QoS-Manager kann Pfadwahl dynamisch anpassen

Konfiguration

→ Partitionskennung frei wählbar
→ Mehrere GIDs im IPv6-Format pro Port möglich
→ GIDs werden vom Switch zugewiesen
→ Ein separater QoS-Regelsatz pro Channel möglich
→ Redundanz durch Multipathing

Integration

→ Hoher Grad an Isolation von Channels
→ Nutzung der ISO-OSI-Schichten 1-4:
- Kapselung von IP in Transportschicht (IPoIB) nach RFC 4391
- TCP-ähnliche Datenübertragung per RDMA (SDP)
- iSCSI per RDMA (iSER)

→ Nutzung der ISO-OSI-Schichten 1-3:
- IPv6 Datenverkehr kann nativ durch das Netz übertragen (RAW IPv6)

→ Nutzung der ISO-OSI-Schichten 1-2: möglich (RAW Ethertype)

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x	x	
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit des GUID		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang	x	x	
#11	Management-Zugangskontrolle	x	x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung	x		
#19	Rahmengröße	x		
#20	Reihenfolge der Rahmenübertragung (nicht anwendbar)			
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung	x		

NFS/UDP/IP/Ethernet

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IP-Adresse und Dateisystempfad
→ Filterung des Ethernet-Verkehrs anhand der VLAN-ID
→ Filterung der Vermittlungsschichten durch entsprechenden Paketfilter

---

Flusskontrolle

→ keine

Konfiguration

→ VLAN-ID frei wählbar
→ IP-Adresse ist innerhalb eines VLANs eindeutig
→ NFS-Ressource wird ohne Berücksichtigung von VLANs für IP-Adressen zur Verfügung gestellt
→ Glättung durch Einteilung in Klassen, basierend auf Attributen der Kommunikationsschichten, z.B. IP-Adresse
→ Redundanz durch STP

Integration

→ Kann in jedes IP-Netz integriert werden
→ Muss mit anderem LAN-Verkehr abgestimmt werden

	Bezeichnung	Erfüllt durch			
		ETH	**UDP /IP**	**VMM**	**nicht mgl.**
#1	Eindeutiger Bezeichner	x			
#2	Datenstrommarkierung	x			
#3	Datenstromfilterung	x		x	
#4	Technologieübergreifend	x			
#5	Technologieübergreifende Zusicherungen	x			
#6	Datenstromüberwachung	x			
#7	Messbarkeit der Auslastung		x		
#8	Veränderbarkeit der MAC			x	
#9	Flusskontrolle				x
#10	HBA-Management-Zugang			x	
#11	Management-Zugangskontrolle			x	
#12	Überlappung	x			
#13	Unbegrenzt viele virtuelle Infrastrukturen	x			
#14	HBA Redundanz	x			
#15	Aufrechterhaltung der Verfügbarkeit	x			
#16	Sofortige Konfigurationsänderung	x			
#17	Eindeutig adressierbarer HBA	x			
#18	Verlustfreie Datenübertragung				x
#19	Rahmengröße (nicht anwendbar)				
#20	Reihenfolge der Rahmenübertragung				x
#21	Pfadwahl nach virtueller Infrastruktur	x			
#22	Pfadwahl nach Auslastung				x

NFS/UDP/IP/InfiniBand

Identifizierung & Filterung

→ Identifizierung einer Ressource durch IP-Adresse und Dateisystempfad
→ Filterung des InfiniBand-Verkehrs anhand von Partitionskennung
→ Filterung der Vermittlungsschichten durch entsprechenden Paketfilter

---

Flusskontrolle

→ Buffer-to-buffer-credit System um Auslastung einer physischen Verbindung zu kontrollieren
→ Höchstens 15 virtual Lanes/Service Level/Buffer pro physischer Verbindung
→ Multipathing zur Lastverteilung
→ QoS-Manager kann Pfadwahl dynamisch anpassen

Konfiguration

→ Partitionskennung frei wählbar
→ IP-Adresse ist innerhalb einer Partition eindeutig
→ NFS-Ressource wird ohne Berücksichtigung von Partitionen für IP-Adressen zur Verfügung gestellt
→ Ein separater QoS-Regelsatz pro Channel möglich
→ Redundanz durch Multipathing

Integration

→ Hoher Grad an Isolation von Channels
→ Nutzung der ISO-OSI-Schichten 1-4:
- Kapselung von IP in Transportschicht (IPoIB) nach RFC 4391
- Durch Kapselung ist IPoIB unabhängig von anderer InfiniBand nutzung

	Bezeichnung	Erfüllt durch		nicht
		Netz	VMM	mgl.
#1	Eindeutiger Bezeichner	x		
#2	Datenstrommarkierung	x		
#3	Datenstromfilterung	x		
#4	Technologieübergreifend	x		
#5	Technologieübergreifende Zusicherungen	x		
#6	Datenstromüberwachung	x		
#7	Messbarkeit der Auslastung	x		
#8	Veränderbarkeit des GUID		x	
#9	Flusskontrolle	x		
#10	HBA-Management-Zugang	x	x	
#11	Management-Zugangskontrolle	x	x	
#12	Überlappung	x		
#13	Unbegrenzt viele virtuelle Infrastrukturen	x		
#14	HBA Redundanz	x		
#15	Aufrechterhaltung der Verfügbarkeit	x		
#16	Sofortige Konfigurationsänderung	x		
#17	Eindeutig adressierbarer HBA	x		
#18	Verlustfreie Datenübertragung			x
#19	Rahmengröße (nicht anwendbar)			
#20	Reihenfolge der Rahmenübertragung			x
#21	Pfadwahl nach virtueller Infrastruktur	x		
#22	Pfadwahl nach Auslastung	x		

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[3le08] *3Leaf Homepage*, Dezember 2008. http://www.3leafsystems.com/.

[AFA07] Anthony F. Aiello, Radek Aster: *System and method for multipath I/O support for fibre channel devices*, April 2007.

[BG08] Bernard Golden, Clark Scheffy: *Virtualization for Dummies.* Whiley, Indianapolis, 2008.

[boc09] *Bochs Projekt Homepage*, März 2009. http://bochs.sourceforge.net/.

[Bra97] Brause, Rüdiger: *Betriebssysteme: Grundlagen und Konzepte.* Springer-Verlag, Berlin, 1997.

[Brü06] Brügge, Bernd und Dutoit, Allen H.: *Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java.* Pearson Studium, München, 2006.

[Buc95] Bucher, Steven: *Understanding Fibre Channel SCSI.* 1995.

[che08] *Chelsio Homepage*, Dezember 2008. http://www.chelsio.com/.

[cis08a] *Cisco Homepage*, Dezember 2008. http://www.cisco.com/.

[cis08b] *Cisco VFrame Data Center*, Dezember 2008. http://www.cisco.com/en/US/products/ps8463/index.html.

[Cis08c] Cisco Systems: *Cisco NX-OS Unicast Routing Configuration Guide*, September 2008. http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/unicast/configuration/guide/l3_nx-os.pdf.

[CK06] Chu, J. und V. Kashyap: *Transmission of IP over InfiniBand (IPoIB).* RFC 4391 (Proposed Standard), April 2006.

[Cla00] Clark, Tom: *Zoning For Fibre Channel SANs - Technology Information*, März 2000.

[Das06] Das, Sujal: *InfiniBand: Enhancing Virtualization ROI with new data center efficiencies*, 2006. http://download3.vmware.com/vmworld/2006/tac4810.pdf.

[DCN06] DeSanti, C., C. Carlson und R. Nixon: *Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel.* RFC 4338 (Proposed Standard), Januar 2006.

[DH98] Deering, S. und R. Hinden: *Internet Protocol, Version 6 (IPv6) Specification.* RFC 2460 (Draft Standard), Dezember 1998. Updated by RFC 5095.

[DR02] Dreo Rodosek, G.: *A Framework for IT Service Management.* Habilitation, Ludwig-Maximilians-Universität München, Juni 2002.

[DVMG07] DeSanti, C., H.K. Vivek, K. McCloghrie und S. Gai: *Fibre Channel Zone Server MIB.* RFC 4936 (Proposed Standard), August 2007.

[emu08] *Emulex Homepage*, Dezember 2008. `http://www.emulex.com/`.

[Far08] Farley, Marc: *FCoE is a great dead end*, April 2008.

[fco] *Fibre Channel: Overview of the Technology.*

[Fel07] Feldman, Michael: *InfiniBand and 10GbE Head for Showdown*, Dezember 2007.

[Fer08] Ferro, Greg: *Seven Fundamental Reasons Why FCoE Will Fail in the Market*, Mai 2008.

[FS08] Fong, Liana und Malgorzata Steinder: *Duality of virtualization: simplification and complexity.* SIGOPS Oper. Syst. Rev., 42(1):96–97, 2008.

[Gai08] Gai, Silvano: *Data Center Networks and Fibre Channel over Ethernet (FCoE).* Lulu.com, 2008.

[Gil04] Gilbert, Douglas: *The Linux 2.4 SCSI subsystem HOWTO*, 2004.

[Gol07] Golde, Pierre: *NetEffect Adopts Denalis PCI Express 20 and IO Virtualization Technology Solutions.* Denali Software, Inc., November 2007.

[Gul08] Gulizia, Sheryl: *Synopsys Announces DesignWare IP for PCI Express with PCI-SIG I/O Virtualization...* Synopsys, Inc., Juni 2008.

[Hal96] Halsall, Fred: *Data Communications, Computer Networks and Open Systems.* Addison-Wesley Publishing Company, Reading, Mass., 1996.

[HAN99] Hegering, Heinz-Gerd, Sebastian Abeck und Bernhard Neumair: *Integrated Management of Networked Systems – Concepts, Architectures and their Operational Application.* Morgan Kaufmann Publishers, San Fransisco, 1999.

[HD09] Hegering, Heinz-Gerd und Vitalian Danciu: *Vorlesung: Rechnernetze I*, WS08/09.

[Hil07] Hill, Steven: *IOV: The Final Frontier of Server Virtualization.* Network Computing, Juni 2007.

[ibm08] *IBM Homepage*, Dezember 2008. `http://www.ibm.com/`.

[iee04] *IEEE Std 802.1D-2004*, Juni 2004.

[IEE05] IEEE 802.3 work group: *IEEE Std 802.3-2005*, Dezember 2005.

[iee06] *IEEE Std 802.1Q-2005*, Mai 2006.

[inf07] *InfiniBand Architecture Specification*, November 2007.

[inf09] *InfiniBand Trade Association Homepage*, April 2009. `http://www.infinibandta.org`.

[Int05] Intel Corporation: *Intel Vanderpool Technology for IA-32 Processors (VT-x)*, jan 2005.

[int08a] *Intel Homepage*, Dezember 2008. `http://www.intel.com/`.

[int08b] *Intel Virtualization Technology for Connectivity*, Mai 2008. `http://www.intel.com/network/connectivity/solutions/virtualization.htm`.

[Int08c] Intel Corporation, Cisco Systems, Nuova Systems: *DCB Capability Exchange Protocol Specification*, Januar 2008.

[Kas06] Kashyap, V.: *IP over InfiniBand (IPoIB) Architecture.* RFC 4392 (Informational), April 2006.

[Kus07] Kusnetzky, Dan: *Fault Tolerant and Fail Over is There a Difference?*, April 2007.

[Kus08] Kusnetzky, Dan: *Storage virtualization: Why aren't the big guys talking more about it?*, Februar 2008.

[LDgFK08] Lindinger, Tobias, Vitalian Danciu, Nils gentschen Felde und Ralf König: *Virtualization – Trend analysis and survey.* Technischer Bericht, Technische Universität München, August 2008.

[Lin06] Lindinger, Tobias: *Virtualisierung einer Praktikumsinfrastruktur zur Ausbildung im Bereich Sicherheit vernetzter Systeme*, Mai 2006.

[Lor09] Lorenz, Michael: *Linux-Hosts anbinden an ein Fibre-Channel-SAN.* Linux Magazin, 5, 2009.

[lsi08] *LSI Homepage*, Dezember 2008. `http://www.lsi.com/`.

[Mac07] MacKinnon, Chris A.: *Fibre Channel vs. InfiniBand vs. Ethernet.* Processor, 29(11):26, March 2007.

[Mar06] Marshall, David: *NextIO and Denali Announce First I/O Virtualization Designs for PCIe*, Juni 2006.

[McL08] McLeod, Jack: *Performance Report: Multiprotocol Performance Test of VMware ESX 3.5 on NetApp Storage Systems.* Technischer Bericht, Juni 2008.

[Mic08] Microsoft Corporation: *Storage Glossary: Basic Storage Terms*, 2008. `http://www.microsoft.com/windowsserversystem/storage/storgloss.mspx`.

[mic09] *Microsoft HyperV Homepage*, März 2009. `http://www.microsoft.com/windowsserver2008/en/us/hyperv.aspx`.

[MMT+05] Monia, C., R. Mullendore, F. Travostino, W. Jeong und M. Edwards: *iFCP - A Protocol for Internet Fibre Channel Storage Networking.* RFC 4172 (Proposed Standard), September 2005.

[net08a] *NetEffect Homepage*, Dezember 2008. `http://www.neteffect.com/`.

[net08b] *Neterion X3100 Product Brief*, 2008.

[nex08] *NextIO Homepage*, Dezember 2008. `http://www.nextio.com/`.

[PCI06] PCI-SIG: *PCI Express Base Specification Revision 2.0*, Dezember 2006.

[PCI07] PCI-SIG: *Single Root I/O Virtualization and Sharing Specification Revision 1.0*, September 2007.

[PCI08] PCI-SIG: *Multi-Root I/O Virtualization and Sharing Specification Revision 1.0*, Mai 2008.

[qem09] *QEMU Projekt Homepage*, März 2009. `http://www.nongnu.org/qemu/`.

[qlo08] *Emulex Homepage*, Dezember 2008. `http://www.qlogic.com/`.

[Ree08] Reedy, Sarah: *Ixia's 100 GbE challenge*, Juni 2008.

[RGSX06] Raj, Himanshu, Ivan Ganev, Karsten Schwan und Jimi Xenidis: *Scalable I/O Virtualization via Self-Virtualizing Devices.* Technischer Bericht, 2006.

[RRW04] Rajagopal, M., E. Rodriguez und R. Weber: *Fibre Channel Over TCP/IP (FCIP).* RFC 3821 (Proposed Standard), Juli 2004.

[Sch93] Schmidt, Friedhelm: *SCSI-Bus und IDE-Schnittstelle. Moderne Peripherie-Schnittstellen: Hardware, Protokollbeschreibung und Anwendung.* Addison-Wesley, Bonn, 1993.

[Sch08] Schaaf, Thomas: *IT-gestütztes Service-Level-Management : Anforderungen und Spezifikation einer Managementarchitektur / vorgelegt von Thomas Schaaf.* Doktorarbeit, Ludwig-Maximilians-Universität München, Dezember 2008.

[Smi06] Smith, Christopher: *Linux NFS-HOWTO*, Mai 2006.

[SMS+04] Satran, J., K. Meth, C. Sapuntzakis, M. Chadalapaka und E. Zeidner: *Internet Small Computer Systems Interface (iSCSI).* RFC 3720 (Proposed Standard), April 2004. Updated by RFCs 3980, 4850, 5048.

[SPF+06] Soltesz, Stephen, Pötzl, Herbert, Fiuczynski, Marc E., Bavier, Andy und Peterson, Larry: *Container Based OS virtualization: a scalable high performance alternatie to hypervisors*, Juli 2006.

[Ste07] Steudten, Thomas: *Fibre Channel over Ethernet: Neue SAN-Infrastruktur*, November 2007.

[T1008] T10: *Fibre Channel Protocol for SCSI, Fourth Version (FCP-4)*, September 2008.

[t11] *WWW home page for Technical Committee T11.* `http://www.t11.org`.

[T1194] T11: *Fibre Channel - Physical and Signaling Interface*, Juni 1994.

[T1195] T11: *Fibre Channel - Arbitrated Loop*, Juni 1995.

[T1196] T11: *Fibre Channel - Fabric Generic Requirements*, August 1996.

[T1102] T11: *NPIV Functionality Profile*, August 2002.

[T1106] T11: *Fibre Channel - Link Services*, Dezember 2006.

[T1107] T11: *Fibre Channel - 10 Gigabit*, Juni 2007.

[T1108a] T11: *Fibre Channel - Generic Services*, Juni 2008.

[T1108b] T11: *Fibre Channel - Security Protocols*, Oktober 2008.

[T1108c] T11: *Fibre Channel - Switched Fabric 5*, Dezember 2008.

[T1109] T11: *Fibre Channel - Framing and Signaling 3*, Februar 2009.

[Tan01] Tanenbaum, Andrew S.: *Computerarchitektur*. Pearson Studium, München, 2001.

[Tan03] Tanenbaum, Andrew S.: *Moderne Betriebssysteme*. Pearson Studium, München, 2003.

[Tan07] Tanenbaum, Andrew S.: *Computernetzwerke*. Pearson Studium, München, 2007.

[uCC04] C. Carlson, C. DeSanti und: *Transmission of IPv4 and ARP Packets over Fibre Channel*, März 2004.

[uE03] Erkens, Rainer, Tropens, Ulf und: *Speichernetze*. dpunkt.verlag GmbH, 2003.

[Uni09a] University, Industrial Ethernet: *Survey of Ethernet Redundancy Methods*. Technischer Bericht, 2009. `http://www.industrialethernetu.com/courses/303_1.htm`.

[Uni09b] University of New Hampshire InterOperability Laboratory: *UNH-IOL Fibre Channel Tutorial*, 2009.

[vir08] *VirtenSys Homepage*, Dezember 2008. `http://www.virtensys.com/`.

[vmw09] *VMware Homepage*, März 2009. `http://www.vmware.com/`.

[xen09] *Xen Projekt Homepage*, März 2009. `http://www.xen.org/`.

[xsi08] *Xsigo Homepage*, Dezember 2008. `http://www.xsigo.com/`.