

Virtualisierung ist eine Herangehensweise in der Informationstechnologie, die Ressourcen so in einer logischen Sicht zusammenfasst, dass ihre Auslastung optimiert wird und sie Anforderungen automatisch zur Verfügung stehen. Der Lösungsansatz besteht in der Verknüpfung von Servern, Speichern und Netzen zu einem virtuellen Gesamtsystem, aus welchem die Anwendungen direkt und bedarfsgerecht ihre Ressourcen beziehen.

Man muss unterscheiden zwischen der

- **Virtualisierung von Hardware**, die sich mit der Verwaltung von Hardware-Ressourcen, beschäftigt, und die
- **Virtualisierung von Software**, die sich mit der Verwaltung von Software-Ressourcen, wie z. B. Anwendungen und Betriebssystemen beschäftigt.

Eine Virtualisierung bietet vielfältige Vorteile:

- **Serverkonsolidierung** bedeutet Zusammenlegen vieler virtueller Server auf möglichst wenigen physikalischen Servern. Zur optimalen Ausnutzung der vorhandenen Hardware-Ressourcen können auf den physikalischen Hosts virtuelle Server mit aufeinander abgestimmten Leistungsmerkmalen sowie zueinander passenden Lastprofilen zusammengelegt werden. Daraus ergibt sich nicht nur eine Kostensenkung bei der Hardware, sondern die Konsolidierung führt auch im laufenden Betrieb zu einem niedrigeren Stromverbrauch und weniger Klimatisierungsaufwand.
- **Vereinfachte Administration**, die Zahl der physischen Server verringert sich. Bei den virtuellen Servern können sehr ausgereifte Managementwerkzeuge eingesetzt werden: Die Erstellung und Anpassung der einzelnen virtuellen Server ist durch Templates oder Installationsvorlagen leicht automatisierbar.
- **Vereinfachte Bereitstellung**: Verringern der Zeit für die Bereitstellung einer neuen Infrastruktur wie z. B. eines Servers innerhalb von Minuten mit hervorragenden Automatisierungsfunktionen.

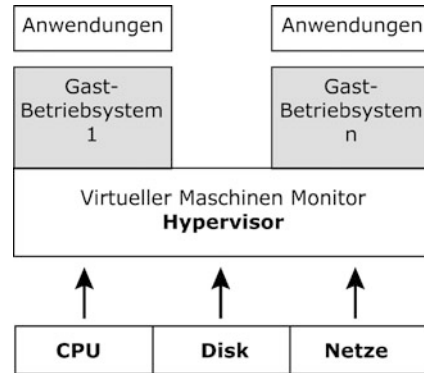
- **Hohe Verfügbarkeit der Systeme und Services** ist gewährleistet durch Migrationsfähigkeit der Systeme und Services und durch einfaches Duplizieren oder Klonen der virtuellen Maschinen. Durch Replikation der physikalischen Ressourcen ist ein praktisch unterbrechungsfreier Betrieb bei Datensicherung, Hardware-Upgrades und Ausfällen möglich.
- **Service-Levels:** Virtualisierung bietet neue Möglichkeiten zur Vereinbarung von garantierten Ressource-Zuwendungen an Applikationen. Das Lastverhalten von Anwendungen kann damit sehr genau überwacht und eingestellt werden.
- **Optimierung von Software-Tests und Software-Entwicklung:** Unterschiedliche Testumgebungen, unterschiedliche Betriebssysteme und Versionen oder unterschiedliche Setups und Umgebungen sind ohne zusätzliche Hardware schnell aufzusetzen.
- **Unterstützung von alten Anwendungen:** Migrieren von sog. Legacy-Betriebssystemen und Legacy-Anwendungen, die auf neuer aktueller Hardware nicht mehr laufen würden (z. B. Windows NT4).
- **Höhere Sicherheit:** Einrichtung einer eigens hoch abgesicherten und isolierten virtuellen Installation für Internet-Dienste oder Web-Browser. Auch unternehmenskritische Anwendungen können in einer virtuellen Maschine gekapselt werden und in einer sicheren Umgebung ablaufen.

Die verschiedenen Techniken der Virtualisierung sind:

1. **Betriebssystemvirtualisierung**, welche die Ausführung mehrerer Betriebssysteme auf einem Rechner erlaubt.
2. **Java Virtuelle Maschine (JVM)**, welche die Ausführung von Programmen auf jedem Betriebssystem ermöglicht.
3. **Softwarevirtualisierung**, welche die lokale Ausführung von Programmen ohne vorherige Installation erlaubt.
4. **Hardwarevirtualisierung**, welche die Verwaltung von heterogenen Hardware-Ressourcen in sog. Ressource-Pools ermöglicht.

9.1 Betriebssystemvirtualisierung

Die Virtualisierung von Betriebssystemen erlaubt die Ausführung mehrerer Betriebssysteminstanzen auf einem Rechner. Das Gast-System ist hierbei stets für die gleiche CPU-Architektur ausgelegt (sonst spricht man von Emulation). Das Konzept der Betriebssystemvirtualisierung ist nicht neu: Es reicht zurück in die 1960er Jahre des vergangenen Jahrhunderts und stammt aus dem Umfeld der Großrechner. Dort stellt der sog. Virtuelle Maschinen-Monitor (VMM) oder Hypervisor jedem Benutzer einen spezifischen Anteil an Ressourcen als Kopie der unterliegenden Hardware in Form einer virtuellen Maschine zur Verfügung (Abb. 9.1).

Abb. 9.1 Betriebssystemvirtualisierung

Es entsteht so die Illusion, jeder Benutzer besitze einen eigenen persönlichen Rechner mit allen Ein/Ausgabe-Operationen, Interrupts und Betriebsmodi des realen Rechners. Durch diese Technik können verschiedene Gastbetriebssysteme konkurrent auf einem einzigen physischen Großrechner laufen.

Das erste und bekannteste kommerziell eingesetzte Betriebssystem in diesem Umfeld war VM/370 von IBM, das auf Großrechnern als Trägersystem für eine Vielzahl von Betriebssystemen dienen kann wie z. B. MVS oder AIX. Auch der Betrieb von Dialogsystemen wie Conversational Monitor System (CMS) ist möglich. Die aktuelle Version z/VM unterstützt auch Linux-Instanzen.

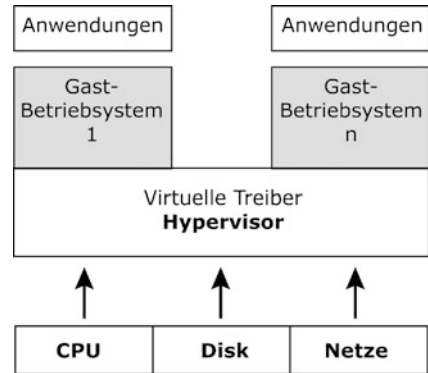
Das **Control Program (CP)** sorgt für die Partitionierung und Zuteilung von Hardware-Ressourcen wie Prozessorleistung und Datenspeicher entsprechend der Anforderungen der Anwendungen [IBM 72]. Als Folge der enormen Leistungssteigerung der Mikroprozessoren gibt es die Virtualisierung inzwischen nicht nur auf leistungsstarken Großrechnern, sondern auch im Mikroprozessor-Bereich [I 06, A 06].

Mögliche Virtualisierungskonzepte bei Betriebssystemen sind:

- Vollvirtualisierung bildet einen kompletten Rechner inklusive BIOS nach und unterstützt mehrere unterschiedliche unmodifizierte Betriebssysteme nebeneinander.
- Containervirtualisierung erlaubt die mehrfache Ausführung desselben Betriebssystems auf einem Rechner.
- Paravirtualisierung virtualisiert Teilaspekte und kann entsprechend angepasste Betriebssysteme sehr performant unterstützen.

9.1.1 Vollvirtualisierung

Bei der Vollvirtualisierung stellt das Gastbetriebssystem der virtuellen Maschine im Idealfall alle Bereiche der physischen Hardware in Form von virtueller Hardware über spezielle Gerätetreiber zur Verfügung. Dadurch kann ein unverändertes Gastbetriebssystem in einer

Abb. 9.2 Vollvirtualisierung

isolierten Umgebung ausgeführt werden. Wie Abb. 9.2 zeigt, sorgt der Hypervisor dabei für eine bedarfsgerechte Zuteilung der Ressourcen:

Als Beispiele für die Vollvirtualisierung sollen die Produkte der Firma VMware dienen [L 07].

VMware Workstation ist für den Einsatz unter einem Desktop-Betriebssystem konzipiert. Es kann damit unter den Wirtsbetriebssystemen Linux und Windows ein kompletter x86-PC bzw. x86-64-PC virtualisiert werden. Auf diesen virtuellen Systemen können die meisten x86 Betriebssysteme installiert und betrieben werden, wobei den virtuellen Maschinen Ressourcen des Hostsystems zur Verfügung stehen [VMP 07].

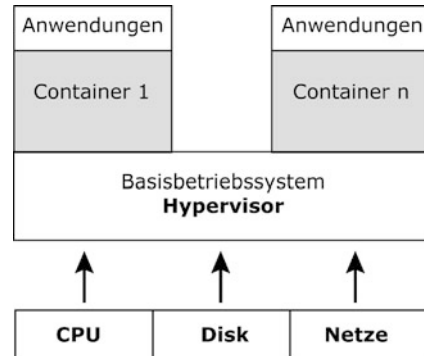
Für den Einsatz im Serverbereich wird **VMware Server** kostenfrei angeboten. Das Produkt unterscheidet sich von der Workstation-Version durch zusätzliche Leistungsmerkmale und bessere Managementfähigkeiten [VMS 07].

Der **VMware ESX Server** basiert auf einem VMware-eigenen Betriebssystemkernel und benötigt daher kein Wirtsbetriebssystem. Dadurch ist prinzipiell eine höhere Performance, Skalierbarkeit und Sicherheit möglich, da das Produkt speziell auf den Einsatzzweck als Virtualisierungsserver optimiert ist.

9.1.1.1 Hochverfügbarkeit und Lastausgleich

Die darauf aufbauende **VMware Infrastructure** erweitert den **ESX Server** um eine Hochverfügbarkeitslösung für Cluster und um automatischen dynamischen Lastausgleich (Load-Balancing). Das Verschieben von virtuellen Maschinen im laufenden Betrieb ermöglicht das zusätzliche Paket **VMotion**. Dies führt zu einer ausfallsicheren Plattform, die in ihren Managementfähigkeiten nahezu den Möglichkeiten eines Großrechners entspricht. So ist es möglich, Wartungsarbeiten wie z. B. Firmware-Upgrades oder Speichererweiterungen der Hosts ohne Betriebsunterbrechung durchzuführen [VMV 07]. Die neueste Entwicklung zielt darauf ab, den 32 MB großen Kernel des ESX-Servers direkt in das BIOS der physischen Server zu integrieren, so dass die Virtualisierungsplattform ohne weitere Softwareinstallation direkt nach dem Einschalten des Servers zur Verfügung steht.

Abb. 9.3 Containervirtualisierung



9.1.1.2 Managementfähigkeiten

Die Verwaltungssoftware *VirtualCenter* ermöglicht die zentrale Administration mehrerer VMware-Server und ESX-Server und deren virtueller Maschinen, inklusive zentraler Leistungsüberwachung und Steuerung. *VirtualCenter* besteht aus einer Serverkomponente mit Datenbank, den Software-Agenten auf jedem zu überwachenden Host, sowie einer oder mehreren Konsolen. Die Agenten überwachen den Zustand der Systeme und steuern die Ressourcenzuteilung. Die Kommunikation zwischen Konsole, Server und Agenten erfolgt über Web Services, das Informationsmodell orientiert sich dabei am weit verbreiteten Industriestandard **Common Information Model** [DMTF 07].

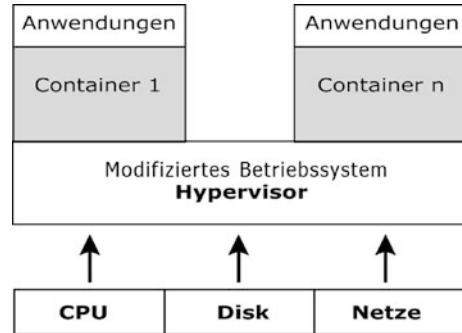
9.1.2 Containervirtualisierung

Virtualisierung durch Container stellt Anwendungen virtuell eine komplette Laufzeitumgebung innerhalb eines geschlossenen isolierten Bereichs zur Verfügung. Dadurch ist die mehrfache Nutzung eines einzelnen Betriebssystems möglich. Der Hypervisor startet auf dem Rechner kein zusätzliches, komplettes Betriebssystem, sondern die Container bilden lediglich aktuell in der Anwendung benötigte Teilfunktionalitäten des Wirtbetriebssystems ab, wie Abb. 9.3 veranschaulicht.

Der Vorteil dieses Vorgehens liegt in der guten Integration der Container an das Gastbetriebssystem. Der Nachteil dieses Konzepts besteht darin, dass die Containervirtualisierung ausschließlich die Plattform des Wirts unterstützt und man aus den Containern heraus keine Treiber laden kann. Beispiele für die Containervirtualisierung sind Produkte wie OpenSolaris [S 07], FreeBSD Jails [KW 07], Linux Vserver [Li 07] und Virtuozzo [P 08].

9.1.3 Paravirtualisierung

Die Paravirtualisierung startet auf einem Basisbetriebssystem zusätzliche Betriebssysteme virtuell neu, ohne jedoch Hardware zu virtualisieren oder zu emulieren. Die virtuell gestar-

Abb. 9.4 Paravirtualisierung

teten Betriebssysteme verwenden eine abstrakte Verwaltungsschicht, um auf gemeinsame Ressourcen wie Netzanbindung, Festplattenspeicher, Benutzereingaben zuzugreifen, ohne dass spezielle Treiber zum Einsatz kommen (Abb. 9.4). In der Regel ist dazu aber eine Anpassung des Gastsystems nötig. Ein Beispiel für die Paravirtualisierung ist der Xen Hypervisor [RM 06]. Der Vorteil der Paravirtualisierung ist die sehr effiziente Ressourcennutzung: Die virtuellen Systeme laufen nahezu ebenso performant wie die physischen Systeme.

9.2 Virtuelle Maschine

Virtuelle Maschinen zur Interpretation des von einem Compiler erzeugten Zwischencodes gibt es in zwei Ausprägungen oder Ansätzen:

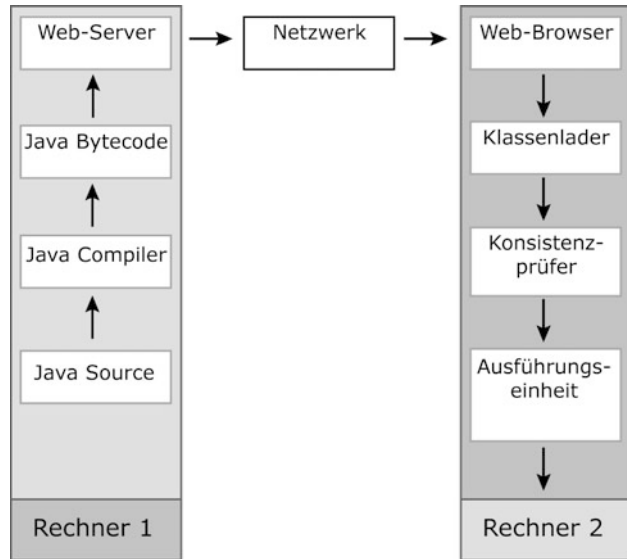
1. Java Virtuelle Maschine (JVM) und
2. das .NET-Framework mit der Laufzeitumgebung der Common Language Runtime (CLR).

9.2.1 Java Virtuelle Maschine (JVM)

Die Philosophie von Sun und Java ist, dass damit geschriebene Programme sofort auf allen Plattformen lauffähig sein sollen und auch immer dieselben Ergebnisse liefern. Die Java Virtuelle Maschine (JVM) emuliert zu diesem Zweck einen idealisierten Rechner, der auf jeder Plattform gleich funktioniert. Ein Java-Compiler übersetzt die Java-Programme zunächst in einen Bytecode, den die JVM anschließend auf den Zielplattformen interpretiert [C 06]. Die Bestandteile der JVM sind

- Klassenlader (ClassLoader),
- Konsistenzprüfer (Verifier),

Abb. 9.5 Ausführung eines Java-Applet



- Ausführungseinheit (Execution Engine) zur Interpretation des Bytecodes,
- Speicherverwaltung und automatische Speicherbereinigung (Garbage Collection).

Der Klassenlader kann Bytecode lokal oder über eine Netzverbindung laden, wobei er nach einer Konsistenzprüfung das auszuführende Programm an die Ausführungseinheit übergibt. Die Java-Laufzeitumgebung stützt sich dabei auf umfangreiche Java-Klassenbibliotheken.

Java bietet die Möglichkeit, Anwendungen zu entwickeln, die in unterschiedlichen verteilten Ausführungsumgebungen ablaufen können. Neben herkömmlichen Applikationen gibt es auch die Möglichkeit, über einen Web-Server auf einer fernen Maschine sog. *Applets* in Web-Browsern auszuführen, wie es das Abb. 9.5 veranschaulicht. Das Sicherheitskonzept von Java gewährleistet dabei, dass unbekannte Komponenten keinen Schaden anrichten können: Eine sog. *Sandbox* kapselt alle Operationen.

9.2.2 Common Language Runtime (CLR)

Die Philosophie von Microsoft mit dem .NET Framework ist, dass Programme, die in unterschiedlichen Programmiersprachen geschrieben sind, auf allen Rechnern mit der .NET-Plattform durch die *Common Language Runtime (CLR)* lauffähig sind (siehe auch Abschn. 5.11.6.1). Die CLR realisiert eine virtuelle Maschine, die eine ausreichende Abstraktion von der konkreten Basismaschine garantiert. Die virtuelle Maschine basiert wie die JVM auf einer abstrakten Stackmaschine, die nur Anweisungen einer Zwischensprache der *Common Intermediate Language (CIL)* ausführt. Die Compiler für die verschie-

denen Sprachen (C++, C#, Java, Prolog, Haskell, ...) erzeugen die notwendige Zwischensprache und unterstützen das von der .NET-Plattform definierte Typsystem. Diese Zwischensprache wird erst zur Laufzeit des Programmes kompiliert und für die konkrete Basismaschine angepasst und optimiert. Weiterhin stellt sie Mechanismen zur Speicher- und Typverwaltung sowie eine Ausnahmebehandlung bereit.

9.3 Softwarevirtualisierung

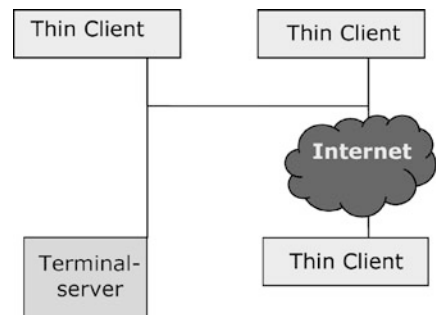
Die Softwarevirtualisierung erlaubt die virtuelle Installation von Softwareprodukten oder Betriebssystemen auf einem Rechner. Voraussetzung dafür ist, dass über ein Netzwerk Zugang zu einem Installationsserver besteht, von dem nach Bedarf Softwarekomponenten geladen werden können. Die Vorteile sind: Bessere Wartbarkeit und größere Flexibilität der Softwareinfrastruktur. Prinzipiell ermöglicht Softwarevirtualisierung in sog. **verwalteten Systemen** auch die komponentenweise und zeitabhängige Verrechnung von Softwarelizenzen.

9.3.1 Services

Die Virtualisierung von Services macht Serveranwendungen für einen Klienten transparent über ein Netzwerk, als ob sie lokal laufen würden. Hierbei kommen auf der Seite der Klienten sehr oft sog. „**Thin Clients**“ zum Einsatz, die nur ein minimales Betriebssystem zur Kommunikation und zur Ausführung der Anwendung besitzen und damit sehr preiswert in Beschaffung und Betrieb sind. Eine Studie der Fraunhofergesellschaft zeigt, dass im Vergleich zu individuell betriebenen PC-Systemen der Einsatz von Thin Clients die Unterhaltungskosten um einen Faktor 3 senken kann [K 07]. Die Klienten können dabei sowohl in einem Intranet als auch im Internet betrieben werden (Abb. 9.6). Das System bietet insbesondere Firmen Vorteile bei der sicheren Anbindung von externen Arbeitsplätzen, da ein Befall mit Schadsoftware quasi ausgeschlossen ist.

An erster Stelle sind in diesem Zusammenhang Terminal-Services zu nennen. Ein prominentes Beispiel sind die Dienste der Firma Citrix, die das Ausführen von Windows-

Abb. 9.6 Ausführung von Services auf Thin Clients



Anwendungen auf Unix-Klienten erlauben [L 05]. Die wichtigste Anwendung von Citrix heißt *Citrix Presentation Server*. Sie bietet die Möglichkeit, von einem beliebigen Computer/Device mit einem beliebigen Betriebssystem über das Internet auf eine Firmenanwendung zuzugreifen, ohne dass die eigentliche Firmensoftware auf dem verwendeten Rechner installiert sein muss; dort ist nur ein Citrix *Independent Computing Architecture (ICA)*-Client oder *Remote Desktop Protocol (RDP)*-Client installiert. Die Anwendung läuft dabei auf dem Server, und es werden nur die Grafikdaten übertragen. Da nur die Daten übertragen werden, die sich auch tatsächlich geändert haben, können die Terminaldienste auch auf langsamen Netzwerkverbindungen genutzt werden.

9.3.2 Anwendungen

Einen Schritt weiter geht die Virtualisierung von Anwendungen. Hier werden Desktop- oder Serveranwendungen lokal ausgeführt, ohne dass diese zuvor lokal installiert werden müssen. Für die *virtualisierte Anwendung* wird dazu eine virtuelle Umgebung generiert, die alle Dateien und Komponenten enthält, die das Programm zur Ausführung benötigt. Die virtuelle Umgebung wirkt dabei wie ein Puffer zwischen Anwendung und Betriebssystem, wodurch Konflikte mit anderen Anwendungen oder Betriebssystemkomponenten vermieden werden. So wird z. B. bei dem Produkt *SoftGrid* der Firma MicroSoft von einem Softwareserver bei Bedarf die Software (z. B. Office) zum Rechner des Anwenders übertragen, sobald dieser eine Anwendung aufruft, in ähnlicher Weise wie dies bei einem Online-Video der Fall ist. Kurz nach Beginn der Übertragung kann das Programm gestartet werden, wobei im Hintergrund weitere Komponenten übertragen werden. Damit dies funktioniert, muss zuvor auf dem PC des Anwenders lediglich der SoftGrid-Klient installiert werden. Der Vorteil dieses Vorgehens ist, dass bei einer größeren verteilten Installation von PCs jeder Rechner mit derselben identischen Softwareausstattung ausgeliefert werden kann. Die verschiedenen Anwendungen stehen dann über den Softwareserver nach Bedarf ohne weitere Installation wahlfrei zur Verfügung. Der Administrationsaufwand für große Installationen von PCs kann damit stark reduziert werden.

9.4 Hardware-Virtualisierung

Die Hardware-Virtualisierung verwaltet Ressourcen wie Prozessor, Hauptspeicher und Datenspeicher über die Firmware des Rechners und teilt diese einer virtuellen Maschine zu.

Die sog. *Partitionierung* unterteilt dabei ein Gesamtsystem dynamisch in Teilsysteme. Ein bekanntes Beispiel ist die Plattenpartitionierung bei der Installation von Betriebssystemen.

Die Partitionierungstechnik kommt z. B. bei IBM zum Einsatz in Großrechnern der zSerie [IBM 07] oder Midrange-Systemen der pSerie [IBM 08]. Ohne Neustart ist es mög-

lich, im laufenden Betrieb eine Ressourcenzuteilung nahezu beliebig zu verändern. Auf einem Großrechner aktueller Bauart können auf diese Weise problemlos mehrere hundert bis tausend Linux-Instanzen gleichzeitig laufen.

Bei Mikroprozessoren gibt es diese Form der Partitionierung des Gesamtsystems nicht, sondern es wird lediglich die Partitionierung der CPU unterstützt, wie bei Intel Vanderpool oder AMD Pacifica. Zur Betriebssystemvirtualisierung stehen Public Domain-Softwareprodukte wie Xen [RM 06] oder kommerzielle Produkte wie VMware [L 07] zur Verfügung.

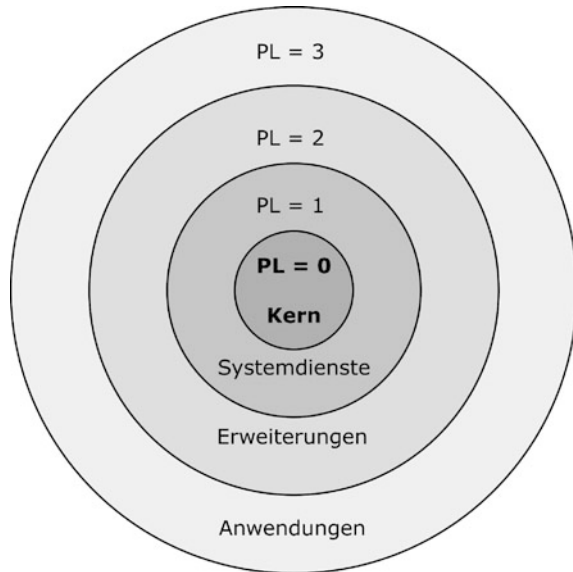
9.4.1 Prozessor

Bei aktuellen Prozessoren werden Virtualisierungsfunktionen direkt auf dem Chip unterstützt. Dabei gehen die Firmen Intel und AMD unterschiedliche Wege:

1. Die Secure *Virtual Machine Architecture Pacifica* von AMD erweitert die AMD64-CPU's um den so genannten *Secure Virtual Machine (SVM)*-Befehlssatz. Die neuen Befehle bieten virtuellen Maschinen Prozessor-Level-Support, indem sie vier verschiedene Privilegien anbieten. Normalerweise laufen wie in Abb. 9.7 gezeigt das Betriebssystem und die Treiber im so genannten Ring 0 (Kernel Mode) und Applikationen im Ring 3 (User Mode). Die neuen Befehle erlauben es, aus einem Betriebssystem heraus Virtuelle Maschinen in Ring 1, 2, oder 3 zu platzieren. Privilegierte Instruktionen in virtuellen Maschinen erzeugen Interrupts, so dass das Basissystem die volle Kontrolle über die Ressourcen behält.
2. Intels konkurrierende *Vanderpool*-Lösung mit der Bezeichnung VT-x für IA32-CPU's und VT-i für Itaniums implementieren dagegen den so genannten *Virtual Machine Extensions (VMX)*-Befehlssatz. Dabei gibt es mit „Root“ und „Non-Root“ zwei Betriebsmodi. Der Virtuelle Maschinen-Monitor (VMM) läuft im VMX Root-Modus und besitzt jederzeit die volle Kontrolle über den Prozessor und die Ressourcen, da damit prinzipiell ein höheres Privileg als Ring 0 implementiert ist. Die virtuellen Maschinen arbeiten im Non-Root-Modus, wobei auch privilegierte Befehle erlaubt sind. Es ist dabei für eine virtuelle Maschine nicht erkennbar, dass sie unter der Kontrolle eines VMM läuft.

AMDs Virtualisierungstechnologie Pacifica ist mit Vanderpool von Intel nicht kompatibel: AMD64-Prozessoren unterscheiden sich durch ihren integrierten Memory-Controller von Intels x86-CPU's. Hier hebt sich Pacifica wesentlich von Vanderpool ab, indem der Speicher-Controller ebenfalls virtualisiert wird. Eine weitere Eigenschaft von Pacifica stellt der Device Exclusion Vector (DEV) dar, der in virtuellen Maschinen Geräte behandelt, die ohne Hilfe des Prozessors direkt auf den Speicher des Systems zugreifen können.

Abb. 9.7 Privilegienstufen (PL) in den Ringen der CPU



9.4.2 Hauptspeicher

Bei der **Hauptspeichervirtualisierung** handelt es sich um eine erweiterte Art der Speicherverwaltung, die inaktive Hauptspeicherbereiche automatisch auf Festplattenspeicher auslagert, um Platz für weitere Anwendungen zu schaffen. Hierbei entsteht ein großer **linearer Adressraum**, der den real eingebauten Hauptspeicher weit übertreffen kann. Ein explizites „Swap out“ und „Swap in“ von kompletten Prozessen wird dadurch prinzipiell überflüssig, da jedem Prozess genügend Speicher zur Verfügung steht.

Der **virtuelle Speicher** bezeichnet den vom tatsächlich vorhandenen Arbeitsspeicher unabhängigen Adressraum, der einem Prozess für Daten und das Programm vom Betriebssystem zur Verfügung gestellt wird. Eine **virtuelle Adresse** beschreibt einen Ort im Speicher eines Computersystems, dessen Betriebssystem eine virtuelle Speicherverwaltung zur Adressierung verwendet. Die Gesamtheit aller virtuellen Adressen wird auch als **virtueller Adressraum** bezeichnet.

Die virtuelle Speicherverwaltung sorgt für die effiziente Nutzung vorhandenen Speichers, stellt große, linear zusammenhängende logische Speicherbereiche zur Verfügung und ermöglicht die Implementierung von Speicherschutzmechanismen. Nur die Betriebssysteme, die eine virtuelle Speicherverwaltung verwenden, können einen virtuellen Adressraum generieren und dadurch Speicherseiten, die physikalisch nicht zusammenhängend sind, für den Programmierer bzw. das Programm als logisch zusammenhängenden Speicherbereich abbilden. So stellen beispielsweise 32-Bit Betriebssysteme bis zu 4 Gigabyte für Programme und Daten zur Verfügung, auch wenn weniger physikalischer Arbeitsspeicher, z. B. nur 1 Gigabyte, eingebaut ist.

Aktuelle 64-Bit-Prozessoren erweitern den virtuellen Adressraum auf 48 Bit, so dass Prozesse bis zu 128 TeraByte ansprechen können. Neben der größeren Leistungsfähigkeit der 64-Bit-Prozessoren ist vor allem der größere Adressraum ein Grund für die Verwendung der neuen Prozessorgeneration.

9.4.3 Datenspeicher

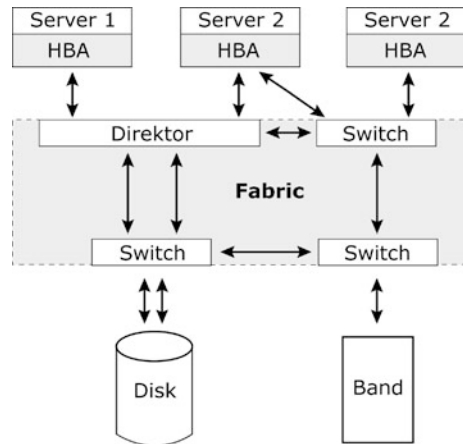
Bei der **Datenspeichervirtualisierung** soll die Rechenkapazität von der Speicherkapazität entkoppelt werden [C 05]. Man trennt die Datenspeicher von den Servern und fasst die vorhandene Speicher-Hardware auf einer neuen logischen Ebene zusammen. Auf diese Weise entsteht ein virtueller Gesamtspeicher. Der Aufbau einer logischen Sicht auf die physischen Speicher schafft die Voraussetzung, Produkte verschiedener Hersteller zentral und flexibel zu verwalten. Man bildet sog. **Speicherpools** aus verschiedenen Speichern mit unterschiedlichen Speichertechniken, auf welche die Server über ein Netzwerk zugreifen können. In diese Pools integriert man z. B. Festplattensysteme, optische Speicher und Bandarchive. Entsprechende Managementwerkzeuge unterstützen Operationen wie Kategorisieren, Kopieren oder Verschieben über die verschiedenen Speichertypen hinweg. Dadurch ergeben sich die folgenden Vorteile:

- **Datenspeicherkonsolidierung:** Die Hardware-übergreifende Zusammenfassung verschiedener physischer Speicher ermöglicht die bedarfsgerechte Skalierung und Zuweisung. Die Klienten des Systems können den virtuellen Speicherplatz auf die bekannte Art und Weise nutzen. Der als ein Gesamtbereich dargestellte Speicher kann dabei physikalisch auch geräteübergreifend verteilt sein.
- **Vereinfachte Administration:** Es wird ein flexibel skalierbarer Speicherpool geschaffen, der unabhängig von den verwendeten Technologien und Produkten zentral verwaltet werden kann.
- **Lifecycle-Management** ermöglicht die Migration von Daten zwischen teuren Direktzugriffspeichern und preiswerten Massenspeichern. Entsprechend ihrer Nachfrage wandern Datensätze automatisch zwischen den Speicherorten hin und her. Eine Beeinträchtigung des Serverbetriebs findet dabei nicht statt.
- **Überbuchen** von Speicher: Beim sog. **Over-Commitment** bekommt ein Server eine virtuelle Speicherreservierung erst beim Schreiben von Daten physisch zugeteilt. Dies verringert den Platzverschchnitt bei Filesystemen.

9.4.3.1 Speichernetze als Basis für Virtualisierung

Die bei der Umsetzung der Speicher-Virtualisierung meist eingesetzte Technik ist ein spezielles Speichernetzwerk, das Storage Area Network (SAN). Das SAN bildet ein Netzwerk zwischen Servern und Speicherressourcen, wobei es blockbasierte Daten überträgt (Abb. 9.8).

Abb. 9.8 Storage Area Network



Die SAN-Umgebung, die sog. Fabric, beinhaltet Switches oder Direktoren, welche Kanäle für den Datentransfer aufbauen. Die Server sind durch spezielle Host Bus Adapter (HBA) angebunden. Oft sind in einem SAN die Verbindungen redundant über unabhängige Pfade ausgelegt, um Ausfallsicherheit oder größeren Durchsatz zu erreichen. Da alle möglichen Datenpfade gleichzeitig sichtbar sind, aber immer nur einer aktiv sein kann, muss auf den Servern eine sog. Multi-Pathing Software zum Einsatz kommen, welche das System überwacht und im Fall eines Fehlers automatisch auf einen alternativen Pfad umschaltet.

Vom Speicherhersteller mitgelieferte Managementwerkzeuge erlauben die Partitionierung des Gesamtspeichers in sog. Logical Units (LUN). Die Zuordnung von LUNs zu Servern ist über Managementwerkzeuge wahlfrei und dynamisch möglich. Das verwendete Protokoll ist meist SCSI, so dass die zugeordneten LUNs als SCSI-Platten in den Servern sichtbar sind.

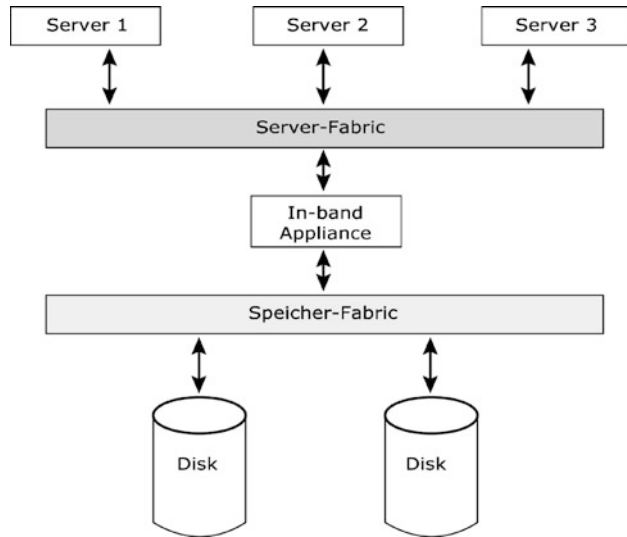
Die SAN-Technik ermöglicht auch die Verlagerung von speziellen Funktionen der Server und der Speichersysteme in das Netzwerk. Es sind dies Funktionen wie

- **Speichermanagement:** Hinzufügen und Löschen von Volumen oder Dateisystemen.
- **Instant Copy:** Direkte Kopie zwischen Platte und Band.
- **Snapshot:** Einfrieren und Abspeichern einer Konfiguration oder eines Speicherzustands zur späteren Wiederherstellung.
- **Mirroring:** Spiegelung von Plattensystemen zur Erhöhung der Ausfallsicherheit und des Durchsatzes.

Bei der praktischen Umsetzung der Speicher-Virtualisierung gibt es zwei Möglichkeiten:

1. Die kostengünstigste Lösung ist, die Virtualisierungsfunktionen direkt in die Speichergeräte oder die Switches zu integrieren. Der Hersteller liefert dabei in der Regel

Abb. 9.9 In-Band Virtualisierung



spezifische Werkzeuge zum Management der Ressourcen mit. Standardisierte Schnittstellen, die der *Storage Management Initiative – Specification (SMI-S)* entsprechen, erlauben auch die Integration von Geräten verschiedener Hersteller zu einem System. Es ist dann möglich, das Gesamtsystem von einer zentralen Konsole zu administrieren.

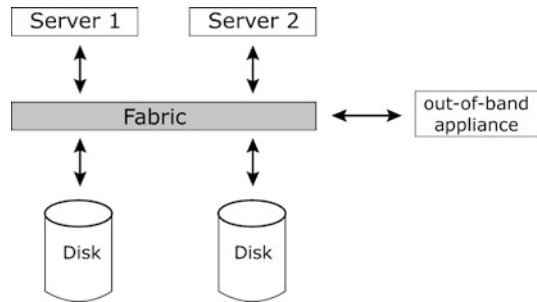
- Bei sehr hohen Anforderungen an die Verfügbarkeit und die Flexibilität der Systeme kommen vor allem sog. „Virtualisierungs-Appliances“ zum Einsatz. Es handelt sich dabei um spezielle Server, welche die Virtualisierungsfunktionen realisieren. Man unterscheidet zwischen „In-Band-Virtualisierung“ und „Out-of-Band-Virtualisierung“ [C 05].

9.4.3.2 In-Band-Virtualisierung

Bei der *In-Band-Virtualisierung* liegt das Virtualisierungssystem zwischen Server-Fabric und Speicher-Fabric (Abb. 9.9).

Der Vorteil dieses Ansatzes besteht in der direkten Beeinflussung des Datenstroms, ohne dass mit den daran angeschlossenen Servern kommuniziert werden muss. Es lassen sich so auch sehr einfach weitere Übertragungsprotokolle und Filesysteme wie z. B. Internet-Protokoll (IP), Server Message Block (SMB) oder Network Filesystem (NFS) integrieren, und das Gesamtsystem wird flexibler. Ein Nachteil ist jedoch, dass man einen Flaschenhals generiert, durch den alle Daten fließen müssen. Darüber hinaus führt ein Ausfall der Appliance zu einem Totalausfall des Systems. Diese Schwierigkeiten lassen sich jedoch durch den Aufbau von redundanten Pfaden bzw. Spiegelung des Virtualisierungsservers gut in den Griff bekommen.

Abb. 9.10 Out-of-Band-Virtualisierung



9.4.3.3 Out-of-Band-Virtualisierung

Bei der *Out-of-Band-Virtualisierung* wird die Virtualisierungs-Appliance neben dem Datenpfad platziert (Abb. 9.10).

Die Datenströme fließen dadurch ohne Umweg direkt zwischen Server und Speicher, es entsteht aber die Notwendigkeit, auf den Servern modifizierte SAN-Treiber einzuspielen, welche mit dem Virtualisierungsserver kommunizieren. Dadurch kann es vor allem in größeren Systemen zu Latenzen kommen. Ein Ausfall der Appliance lässt jedoch die Grundfunktionalität des SAN unberührt. Es lassen sich aber in diesem Fall die virtualisierten Datenblöcke nicht mehr korrekt zuordnen, und als direkte Folge sind dann auch die darauf basierenden Filesysteme nicht mehr verfügbar.

9.4.4 Netzwerke

Durch die Virtualisierung von Netzwerken ist es möglich, auf verschiedenen physikalischen Netzsegmenten logische Bereiche zu definieren. Die Abbildung eines Subnetzes mit einem gemeinsamen IP-Adressbereich über mehrere unterschiedliche physikalische Netze ist dadurch möglich.

9.4.4.1 Virtual Local Area Network

Ein *Virtual Local Area Network* (VLAN) ist ein virtuelles lokales Netz innerhalb eines physischen Netzes. Die technische Realisierung von VLANs ist im Standard **IEEE 802.1Q** definiert, welcher den Ethernet-Frame um 4 Byte erweitert (siehe Abb. 9.11).

12 Bit davon sind für die sog. VLAN-ID vorgesehen, welche ein VLAN eindeutig kennzeichnet. Ein Gerät, das zum VLAN mit der ID = 1 gehört, kann mit jedem anderen Gerät im gleichen VLAN kommunizieren, nicht jedoch mit einem Gerät mit einer anderen VLAN-ID [K 99].

Vorteile der VLAN-Technik sind

- **Transparenz:** Verteilt aufgestellte Geräte können in einem einzigen logischen Netz zusammengefasst werden. VLANs sind sehr nützlich bei der Konzeption der IT-Infrastruktur verteilter Standorte.

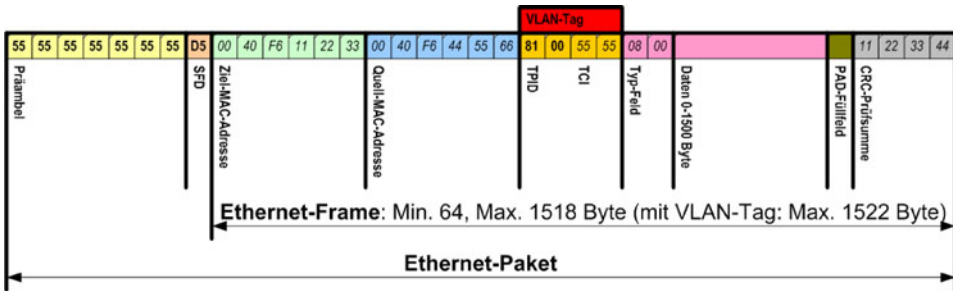


Abb. 9.11 Ethernet-Frame für VLAN

- **Sicherheit:** Bestimmte, besonders zu schützende Systeme können in einem eigenen Netz verborgen werden.

Auf der anderen Seite entsteht durch VLANs ein erhöhter Aufwand bei der Netzkonfiguration und bei der Programmierung der aktiven Netzkomponenten (Switches u. ä.).

9.4.4.2 Virtual Private Network

Mit einem *Virtual Private Network (VPN)* lassen sich über ein öffentliches Datennetz, wie etwa dem Internet, sichere private Verbindungen aufbauen. Die Verbindung der Netze wird dabei über einen sog. Tunnel zwischen VPN-Client und VPN-Server (Concentrator) realisiert, wobei die Daten meist verschlüsselt übertragen werden. Als Protokoll für die sicheren Verbindungen wird in der Regel Secure Socket Layer (SSL) eingesetzt. Die Teilnehmer müssen sich zunächst authentisieren und können sich dann z. B. aus dem Internet in ein durch eine Firewall gesichertes Firmennetz einklinken, wobei alle Ressourcen wie beispielsweise Fileserver oder Drucker transparent verfügbar sind [B 02].

Literatur

- [A 06] AMD Secure Virtualization Technique, 2006.
- [B 02] Böhmer W.: VPN Virtual Private Networks. Die reale Welt der virtuellen Netze. Carl Hanser Verlag, 2002.
- [C 05] Clark T.: Storage Virtualization. Technologies for Simplifying Data Storage and Management. Pearson Education, 2005.
- [C 06] Craig I.D.: Virtual Machines. Springer Verlag London Limited, 2006.
- [DMTF 07] Distributed Management Task Force, Inc.: Common Information Model (CIM) Standards. <http://www.dmtf.org/standards/cim/>, 2007.
- [I 06] Intel Virtualization Technology, Intel Technology Journal, Volume 10/03, 2006.
- [IBM 07] IBM: z/VM® the newest VM hypervisor based on 64-bit z/Architecture. www.vm.ibm.com, 2007.

- [IBM 08] IBM: IBM PowerVM The virtualization platform for UNIX, Linux and i5/OS customers. <http://www-03.ibm.com/systems/power/software/virtualization/index.html>, 2008.
- [IBM 72] IBM Corporation, IBM Virtual Machine Facility/370 Introduction, GC20-1800, 1972.
- [K 07] Knermann C.: PC vs. Thin Client, Fraunhofer Gesellschaft CC-ASP 2007, <http://cc-asp.fraunhofer.de/docs/PCvsTCsummary-de.pdf>, 2007.
- [K 99] Köhler R.: Auf dem Weg zu Multimedia-Netzen : VPN; VLAN-Techniken; Datenpriorisierung. FOSSIL-Verlag, Köln, 1999.
- [KW 07] Kamp P.-H., Watson R.N.M.: Jails: Confining the omnipotent root. <http://phk.freebsd.dk/pubs/sane2000-jail.pdf>, 2007.
- [L 05] Larisch D.: Citrix Presentation Server Grundlagen und Profiwissen. Carl Hanser Verlag 2005.
- [L 07] Larisch D.: Praxisbuch VMware Server. Das praxisorientierte Nachschlagewerk zu VMware Server. Carl Hanser Verlag, 2007.
- [Li 07] Linux VServer.org: Welcome to Linux-VServer.org. http://linux-vserver.org/Welcome_to_Linux-VServer.org, 2007.
- [P 08] Parallels: Parallels Virtuozzo Containers 4.0. <http://www.parallels.com/de/products/virtuozzo/?from=homepage>, 2008.
- [RM 06] Radonic A., Meyer F.: Xen3. Franzis Verlag 2006.
- [S 07] Schenker T.: Opensolaris: Hochverfügbarkeit, Virtualisierung, Container-Technologie. Diplomarbeit, Hochschule Mannheim 2007.
- [VMP 07] vmware: Expand the Power of Your PC with Virtualization. <http://www.vmware.com/products/ws/>, 2007.
- [VMS 07] vmware: Experience Server Virtualization. <http://www.vmware.com/products/server/>, 2007.
- [VMV 07] vmware: Transform IT Infrastructure with Enterprise-Class Virtualization. <http://www.vmware.com/products/vi/>, 2007.