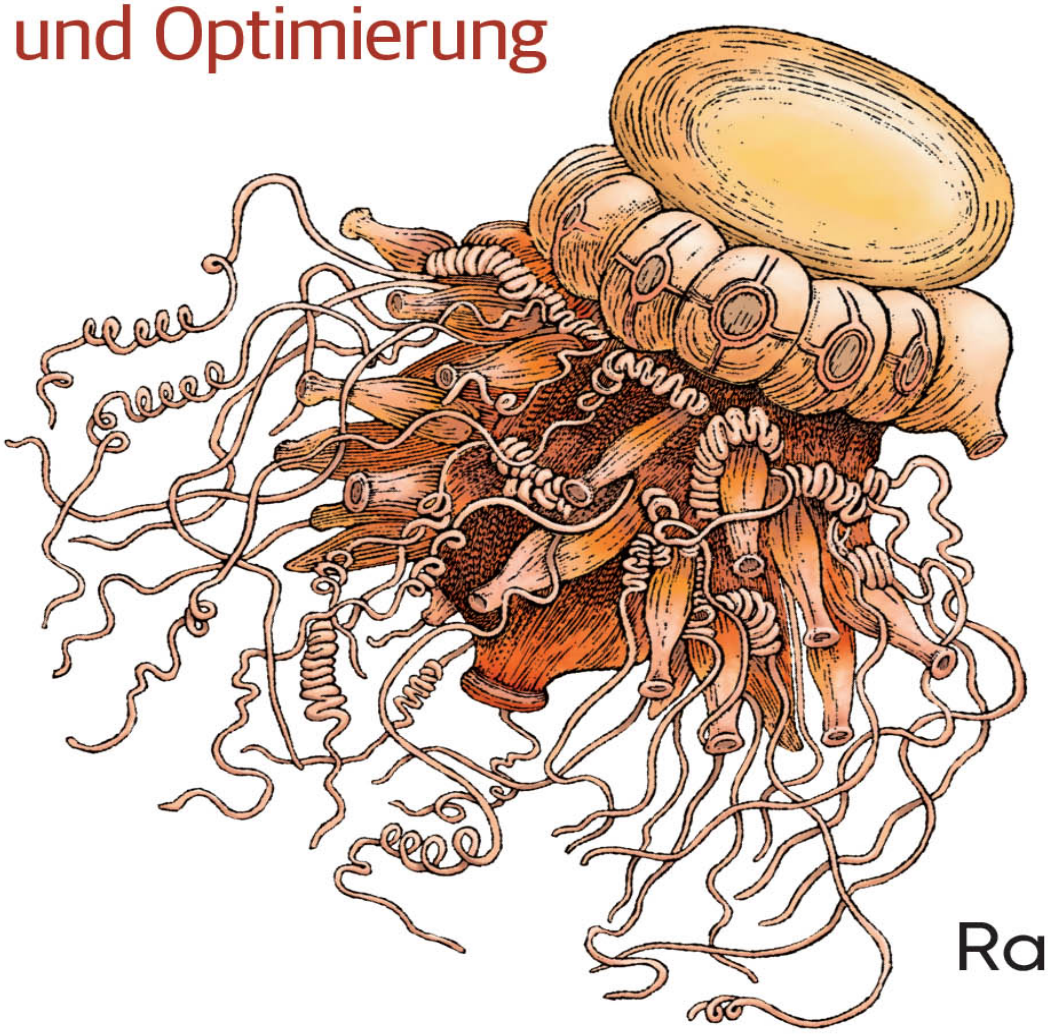
O'REILLY®

5. aktualisierte Auflage

Praxishandbuch VMware vSphere 7

Leitfaden für Installation, Konfiguration und Optimierung



Ralph Göpel

5., aktualisierte Auflage

Praxishandbuch VMware vSphere 7

Leitfaden für Installation, Konfiguration und Optimierung

Ralph Göpel

O'REILLY®

Ralph Göpel

Lektorat: Dr. Michael Barabas
Projektkoordinierung: Anja Weimer
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Gerhard Alfes, mediaService, Siegen, *www.mediaservice.tv*
Herstellung: Stefanie Weidner
Umschlaggestaltung: Karen Montgomery, Michael Oréal, *www.oreal.de*

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über *http://dnb.d-nb.de* abrufbar.

ISBN:
Print 978-3-96009-167-7
PDF 978-3-96010-483-4
ePub 978-3-96010-484-1
mobi 978-3-96010-485-8

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«. O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Hinweis:
Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.

5., aktualisierte Auflage

Wieblinger Weg 17

69123 Heidelberg

5 4 3 2 1 0

Inhalt

KAPITEL 1

Einführung

In diesem Kapitel wollen wir uns kurz mit der Thematik der Virtualisierung im Allgemeinen befassen und die wichtigsten Begrifflichkeiten klären. Anschließend geht’s dann direkt mit dem VMware-Produkt vSphere 7.0 los.

Virtualisierung im Allgemeinen

Um mal eben etwas auszuprobieren oder ein weiteres Betriebssystem zu installieren, muss man nicht unbedingt eine zusätzliche Partition zur Installation haben oder gar einen weiteren Rechner anschaffen. In den meisten Fällen reicht es, eine Software zu starten, »in« der ein zusätzlicher Computer unabhängig läuft. Eine solche Konstellation nennt man *Virtualisierung*. Den Computer, auf dem die Software installiert wird, nennt man *Host* oder *Wirt*, den Rechner, der im Fenster läuft, nennt man *virtuelle Maschine* (im Folgenden oft mit VM abgekürzt).

Natürlich erlauben viele Betriebssysteme und Anwendungen heute eine Testphase von 30 oder sogar mehr Tagen. Aber wer möchte schon gerne auf seinem

Arbeitsrechner, geschweige denn auf einem Server eine Software zu Testzwecken installieren, nur um nach ein paar Tagen festzustellen, dass sie sich nicht mehr so einfach deinstallieren lässt? In einer virtuellen Maschine kann keine Software Schaden anrichten, und mit den meisten Virtualisierungsprogrammen lässt sich vor der Installation der aktuelle Zustand einfrieren und bei Belieben anschließend wiederherstellen.

Ein Computer, auf dem beliebig viele unterschiedliche Betriebssysteme gleichzeitig laufen können, also unabhängige und verschiedene Rechner in Fenstern, ist der Traum jedes Entwicklers, Supporters oder auch versuchsfreudigen Anwenders. Programme von VMware, Microsoft, XEN, KVM oder anderen erzeugen einen kompletten virtuellen Rechner. Mit nur ein paar Mausklicks steht ein neuer Test- PC schnell zur Verfügung.

So ein »PC-Emulator« ist ein Programm, das nach dem Start virtuelle Maschinen (kurz VMs) auf einem Hostsystem, also dem realen Hardwarecomputer, zur Verfügung stellt, in denen sich weitere Betriebssysteme installieren lassen (die Gastsysteme). Eine VM verfügt wie jeder echte PC über die üblichen Hardwarekomponenten wie Prozessor, Arbeitsspeicher, Festplatte, Netzwerkkarte usw. und bringt auch ein eigenes BIOS oder EFI (Extensible Firmware Interface) mit. Die im Fenster installierten Gastbetriebssysteme bemerken üblicherweise nicht, dass die Hardware, auf der sie laufen, nur emuliert wird.

Die meisten Hersteller dieser Virtualisierungslösungen sind sich dabei sogar einig und nutzen die gleiche Hardwareausstattung für ihre virtuelle Umgebung. Da sich der i440BX-Chipsatz von Intel zu Zeiten des Pentium II als sehr stabil erwiesen hat, wird dieser meist in der VM zur Verfügung gestellt. Das wiederum hat den Vorteil, dass sowohl alte als auch neue Microsoft-Betriebssysteme und

auch so gut wie alle Linux-Distributionen in der virtuellen Maschine installiert werden können. Abhängig vom Hersteller der eingesetzten Applikation oder Hypervisoren sind auch andere Betriebssysteme verwendbar, z.B. DOS, Novell, Unix, OS/2 oder Mac OS X.

Warum virtualisieren?

Viele spezielle Anwendungen, gerade auf Serverbetriebssystemen, erfordern einen eigenständigen Rechner. Nur so verspricht der Hersteller einen reibungslosen Einsatz. Gerade bei der öffentlichen Verwaltung und in Krankenhäusern werden diese Anforderungen an den Einsatz der Software im produktiven Bereich gestellt. In der Praxis sind das dann häufig Computer, die bei 2 bis 5% CPU-Auslastung laufen! Als virtuelle Maschine könnten aber gut zehn solcher Server eine einzige Hardware benutzen, um die Systemauslastung dann bei 20 bis 50% zu halten. Die gesparten neun Computer verbrauchen somit keinen Strom und belasten die Klimaanlage nicht – in der Regel rechnet sich die Anschaffung einer Virtualisierungssoftware daher schon nach wenigen Monaten. Das ist auch im Sinne von Green IT und wohl auch die beste Möglichkeit, große Einsparungen bei der Energieversorgung, der Hardware und Wartung sowie dem Platzbedarf für neue Server zu erreichen.

Wie Sie im Laufe des Buches noch lesen werden, ist die Energieeinsparung beim Virtualisieren jedoch nur einer von mehreren wichtigen Aspekten. Da eine VM nur von wenigen Dateien repräsentiert wird, ist ein Kopieren oder *Klonen* des Systems äußerst einfach. Durch das Nachhalten mehrerer Zustände des Gastbetriebssystems (Snapshots) lässt sich ohne Probleme mal eben etwas ausprobieren – sei es eine zusätzliche Software, ein neuer Releasestand oder auch ein Servicepack für das Betriebssystem. Funktioniert

alles bestens, kann dieser Zustand getrost übernommen werden. Geht etwas daneben oder funktioniert es nicht wie gewünscht, so kann jederzeit zu einem vorherigen Stand (Snapshot) zurückgewechselt werden.

Ein weiterer Einsatzzweck für Virtualisierung: Ältere Betriebssysteme wie Windows NT laufen auf aktueller Hardware nicht mehr – meistens können sie noch nicht mal auf neuen Rechnern installiert oder portiert werden. In einer virtuellen Maschine ist das kein Problem.

Wann nicht virtualisieren?

Einzige Ausnahme bei den installierbaren Betriebssystemen sind wohl Echtzeitsysteme wie QNX, RTOS und VXworks, bei denen es auch keinen Sinn macht, diese virtuell laufen zu lassen. Zeitgenaue Abfolgen bei Betriebssystemen oder Anwendungen sollten nicht virtuell laufen – dafür ist das Laufzeitverhalten der VM nicht genau genug. Ein ESXi-Server (Host) macht keine abrupten Zeitänderungen, sondern dehnt Sekunden aus oder verkürzt sie, um die richtige Zeit irgendwann zu erreichen – auch das wäre für Echtzeitanwendungen sehr schädlich. Auch bei manchen Linux-Systemen, gerade mit spezieller Software, ist von der Virtualisierung grundsätzlich abzuraten.

Außerdem wird nur eine Standardhardware zur Verfügung gestellt, sodass spezielle Komponenten wie Steuerungssysteme oder spezielle Bus-Karten usw. leider nicht emuliert werden können. Gängige PCI- und PCIe-Adapter wie z.B. ISDN-Karten können aber – zumindest unter ESXi auf einigen Plattformen – an die VMs durchgereicht werden (PCI-Passthrough bzw. DirectPath-I/O-Konfiguration).

Was manche SysAdmins abschrecken kann: Die Administration der Systeme ist mit Virtualisierung zwar einfacher und sicherer – in gewisser Weise aber auch komplexer, weil zusätzliche Systeme (ESXi-, vCenter Server etc.) berücksichtigt werden müssen.

Die Technik virtueller Maschinen

Auf Großrechnern wie den Mainframes von IBM sind virtuelle Maschinen schon seit langer Zeit nichts Besonderes. Das Grundprinzip ist einfach: In einer Sandbox (von Betriebs- und Dateisystem unabhängige Laufzeitumgebung, in der sich gegebenenfalls gefährlicher Code ohne Gefahr für die Stabilität und Sicherheit des Systems testen lässt) wird ein Computer »emuliert«, der über alle notwendige Hardware sowie Maus und Tastatur verfügt. Innerhalb dieses geschlossenen Systems kann ein Betriebssystem gestartet werden, das auf die virtuelle Hardware zugreift, als sei es ein echter Computer.

In der Praxis gestaltet sich diese Aufgabe jedoch erheblich schwieriger. Immerhin muss eine Vielzahl benötigter Komponenten virtuell erzeugt werden. Da das Betriebssystem auf dem Hostrechner immer den exklusiven Zugriff auf die Hardware behält, kann ein virtuelles Gastbetriebssystem keinen direkten Zugriff auf die reale Hardware bekommen (mit wenigen Ausnahmen). Deshalb findet das Betriebssystem in der virtuellen Maschine auch andere Hardware vor, als tatsächlich im PC eingebaut ist.

Die wichtigsten zu emulierenden Komponenten sind:

- Prozessor

- (A)PIC ((Advanced) Programmable Interrupt Controller)
- DMA (Direct Memory Access)
- IDE- oder SATA-Controller, gegebenenfalls SCSI oder SAS
- CMOS (BIOS-Setup) oder (U)EFI ((Unified) Extensible Firmware Interface)
- Realtime-Clock (Echtzeituhr)
- PIT (Programmable Interval Timer)
- Memory und I/O-Controller
- Festplattenspeicher
- PCI- und Host-Bus sowie PCI-Bridge
- Videoadapter
- Keyboard-Controller
- Keyboard und Maus (ggf. über USB)

Neben diesen wesentlichen Komponenten wird häufig auch noch eine Reihe anderer Komponenten in der virtuellen Maschine benötigt, wie beispielsweise:

- APM oder ACPI (Power Management)
- Netzwerkkarte(n)
- CD-/DVD-Laufwerke
- Soundkarte
- COM- und LPT-Anschlüsse
- Game- bzw. Midi-Port
- USB-Anschlüsse

Zwei der wichtigsten Komponenten – CPU und Festplatte – widme ich einen eigenen Abschnitt, weil hier die meisten

Missverständnisse auftreten und sich bei Unwissenheit über bestimmte Details auch Fehler einschleichen können.

Virtueller Prozessor

Die CPU ist der Hauptbestandteil des echten und des virtuellen Rechners. Die AMD- oder Intel-Architektur hat allerdings hinsichtlich der Abbildung virtueller Maschinen gegenüber den Mainframes einige Schwächen. Letztere sind schon architektonisch auf VMs ausgelegt. Intel-Entwickler hatten früher hingegen nie einen virtuellen Rechner als primäres Ziel vorgesehen. Einen ersten Schritt machte AMD mit der »Pacifica« genannten Virtualisierungstechnik im Prozessorkern, Intel zog nach und nannte seine Virtualisierungsfunktion zunächst »Vanderpool Technology«. Heute heißt diese Funktion AMD RVI bzw. Intel VT-x und lässt sich im BIOS des Servers aktivieren.

Die AMD- und Intel-Architektur der 32- und 64-Bit-CPUs bietet vier verschiedene Privilegien an, mit denen dem Betriebssystem, Treibern und Programmen unterschiedliche Rechte zugewiesen werden können, was Sie in der Abbildung 1-1 sehen können. Normalerweise laufen OS (Operating System, Betriebssystem) und einige Treiber im sogenannten Ring 0 (Kernel Mode) und Applikationen im Ring 3 (User Mode). Der Trick beim Erzeugen eines virtuellen Systems besteht darin, es im Benutzermodus als Applikation ablaufen zu lassen.

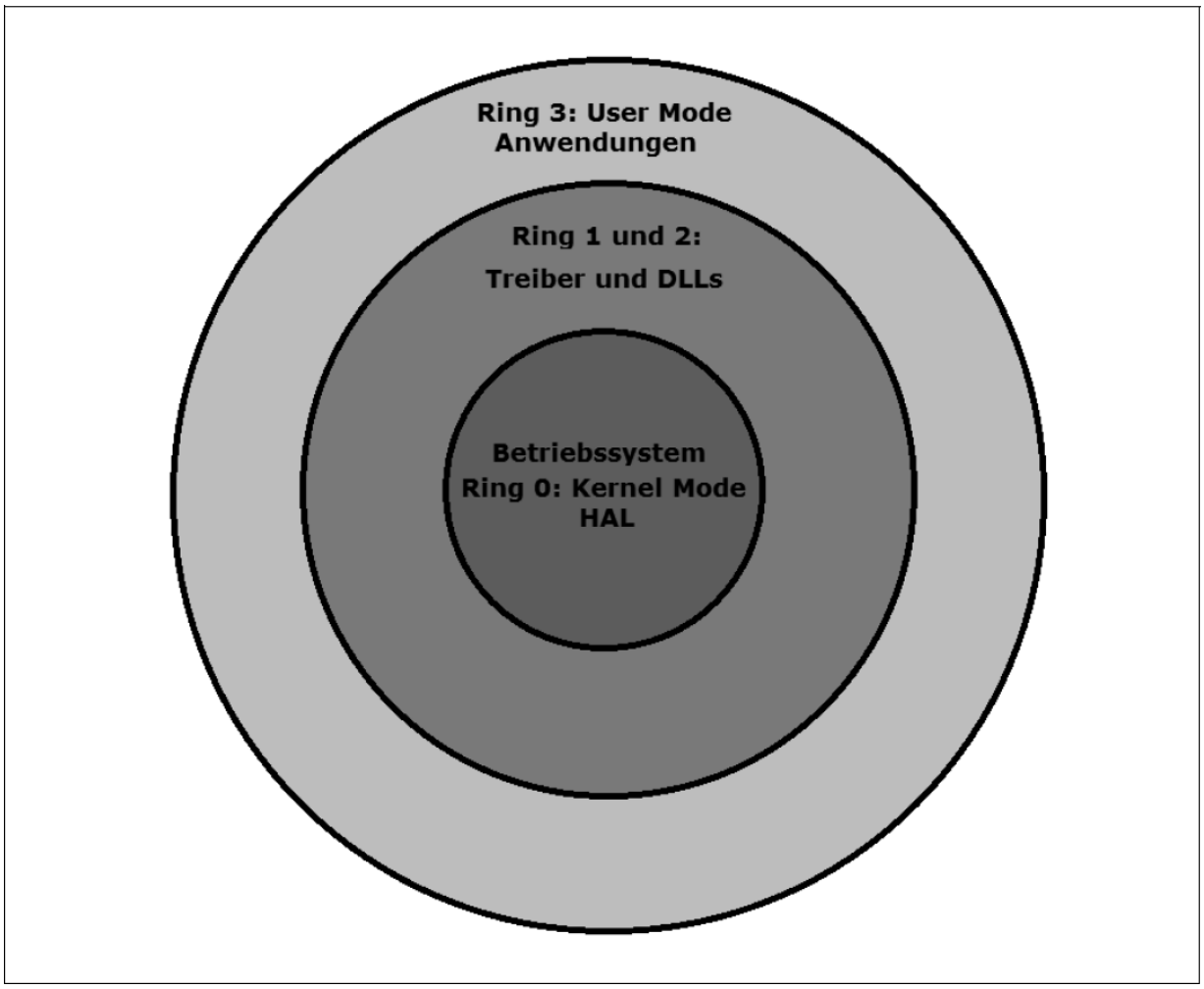


Abbildung 1-1: Die Ring-Modi eines Betriebssystems

Wenn das Betriebssystem des virtuellen Rechners nun einen Befehl ausführen will, der nur im Ring 0 gestattet ist, löst der Prozessor eine Exception (Ausnahme) aus. Routinen zur Behandlung dieser Ausnahmen können dann den privilegierten Befehl emulieren. Dabei behält das Hostsystem die volle Kontrolle über das System. Die wichtigste Voraussetzung für das Funktionieren dieses Ansatzes ist, dass der Prozessor bei jeder unberechtigt durchgeführten privilegierten Anweisung eine Exception auslöst.

Leider sind die Prozessoren bei den Ausnahmen nicht ganz so gründlich, wie man es sich wünschen könnte.

Beispielsweise werden Speicherzugriffe beim x86 über eine GDT (*Global Descriptor Table*, globale Beschreibungstabelle) abgewickelt. Diese ist eine globale Ressource und wird vom Betriebssystem verwaltet. Eigentlich müsste jeder direkte Zugriff auf diese Ressource als privilegierte Handlung angesehen werden und dürfte dementsprechend im User Mode nicht erlaubt sein. Der x86 behandelt die LGDT-Anweisung (Load Global Descriptor Table) auch als privilegiert. Allerdings führt SGDT (Store Global Descriptor Table) nicht zu einer Schutzverletzung. Allein diese Inkonsistenz macht es unmöglich, die GDT zu »virtualisieren«.

Um dennoch virtuelle Maschinen auf einer Intel-Architektur mit 32 oder 64 Bit realisieren zu können, müssen sich die Entwickler von Virtualisierungssoftware eine ganze Menge einfallen lassen. Connectix (2003 von Microsoft aufgekauft) untersuchte beispielsweise Code, der im Kernel Mode laufen soll, vor der Ausführung und änderte kritische Sequenzen um.

Damit erklärt sich auch der Eintrag im Gerätemanager der virtuellen Maschine: Er sieht den tatsächlichen Prozessor, kann aber nicht unbedingt jede Funktion von ihm nutzen – dazu später mehr.

Emulation der Festplatte

Ein weiterer wichtiger Bestandteil bei virtuellen Maschinen sind Massenspeicher. Generell funktioniert der Zugriff auf Massenspeicher ähnlich wie beim virtuellen Prozessor. Wenn das Gastsystem eine I/O-Anweisung (Input/Output, Ein-/Ausgabeanweisung) ausführt, wird eine Exception ausgelöst, da es sich hierbei um eine privilegierte Anweisung handelt. Die Virtualisierungssoftware fängt diese Exception ab und übersetzt die Anweisung in eine

Operation auf dem realen Dateisystem. Bei deren Realisierung steht der VM-Software eine ganze Reihe von Möglichkeiten zur Verfügung.

So könnte sie beispielsweise die Anweisungen direkt auf der physikalischen Festplatte ausführen. Das widerspräche allerdings dem Anspruch an eine Sandbox, denn kein Programm darin darf das Hostsystem beschädigen. Der direkte Zugriff wird zumeist nur bei CD-ROMs oder DVDs eingesetzt. Zwei Ausnahmen davon gibt es allerdings: RDM (Raw Device Mapping), also direkter Zugriff auf eine LUN (Logical Unit Number) im SAN (Storage Area Network), und Direct Passthrough, z.B. bei Netzwerkkarten. Dazu erfahren Sie später ebenfalls mehr.

Meistens wird die virtuelle Festplatte mit einer Datei auf dem Hostbetriebssystem realisiert. Die Virtualisierungssoftware verändert deren Größe teilweise dynamisch. Eine interessante Option sind dabei Festplatten, die beim Herunterfahren wieder in den Zustand von vor dem Start der virtuellen Maschine zurückversetzt werden. So kann man die virtuelle Maschine immer in einem definierten Zustand starten und kritische Tests ausführen. Beim nächsten Start befindet sich die VM wieder im Originalzustand.

Als virtuelles CD-/DVD-ROM-Laufwerk kommen meist ISO-Images zum Einsatz, die als Datei auf der lokalen Platte des Hosts, einer Freigabe im Netzwerk oder einem angeschlossenen NAS (Network Attached Storage) oder SAN (Storage Area Network) liegen. Das Betriebssystem in der VM erkennt diese als »echte« optische Medien, wobei die Zugriffe und Übertragungsraten dort erheblich höher als bei realen Laufwerken sind.

Hardwareausstattung

Welche Hardware die zu installierenden Betriebssysteme in einer VM vorfinden, hängt von mehreren Faktoren ab: Erstens von dem, was die Virtualisierungssoftware an Hardware zur Verfügung stellt, zweitens von den Komponenten des Hosts, die teilweise an die VM durchgereicht werden, und drittens von der Konfiguration durch den Anwender, der entscheiden kann, welche Hardware diese bekommt, wie viel davon dem Gast zur Verfügung steht oder was direkt durchgereicht wird. Abbildung 1-2 zeigt beispielhaft den Gerätemanager eines virtuellen Systems.

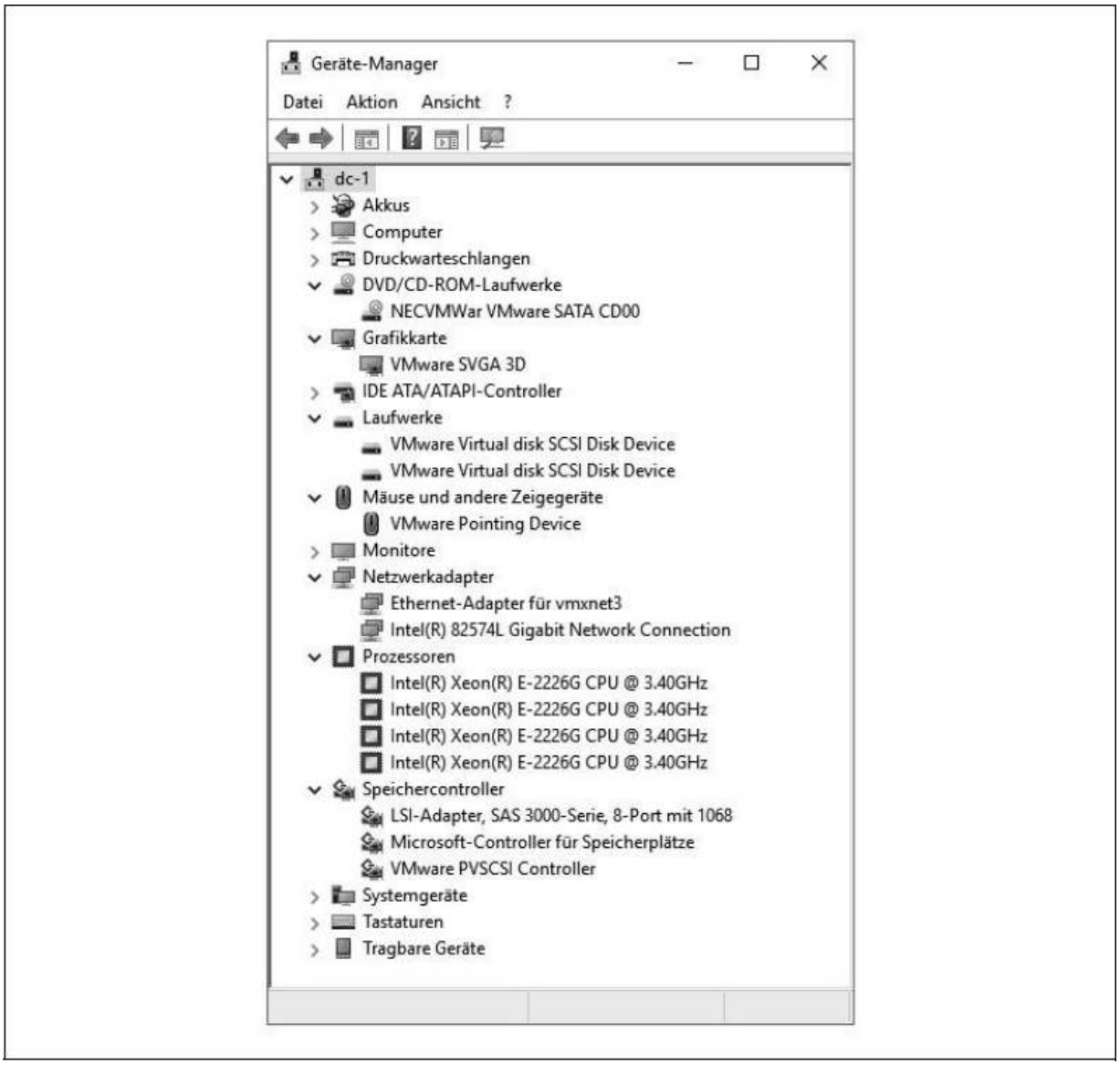


Abbildung 1-2: Gerätemanager unter Windows Server in der VM

Zu den notwendigen Komponenten der VM gehören unter anderem der Chipsatz des Motherboards (z.B. Intel 440BX), BIOS (AMI oder Phoenix) oder EFI, Festplatte sowie Grafik-, Sound- und Netzwerkkarte. Während Motherboard und Festplatte meistens mit den generischen Treibern der Betriebssysteme laufen, muss für die Grafikkarte ein angepasster Treiber installiert werden. VMware übergibt zum Beispiel der VM eine spezielle SVGA-II- oder sogar 3D-Karte, Microsoft bevorzugt hier meist eine Trio 32/64. In

den VMware Tools enthaltene Treiber sollten für den korrekten Zugriff nach der Installation des Betriebssystems noch installiert werden.

Als Netzwerkkarte wird bei VMware unter anderem eine AMD-PCnet-PCI-II- oder Intel-LAN-Pro-Karte (PCI oder PCIe), bei Microsoft eine Intel-21140-Karte emuliert, für die die meisten Linux-Derivate und moderne Windows-Versionen Treiber mitbringen. Weiterhin emuliert VMware als SCSI-Adapter BusLogic und LSI Logic, für die ältere Windows-Versionen passende Treiber haben, XP jedoch nicht. Neuere Windows-Versionen unterstützen zum Teil den einen oder anderen Adapter. Falls nicht, bietet VMware einen Treiber zum Download an. In neueren Versionen der Virtualisierungssoftware werden noch weitere Netzwerkkarten und Festplattenadapter unterstützt.

Bei seriellen und parallelen Anschlüssen, CD-, DVD- und Diskettenlaufwerken kann man in der VM wahlweise auf die Hardware des Hosts zurückgreifen oder diese Geräte vollständig emulieren. Andere Hardware wie etwa FireWire-Anschlüsse, (DSL-)Modems oder interne ISDN-Karten sucht man in der Hardwareausstattung vergeblich. VMware bietet in der neuen Version den Zugriff auf USB-2.0- und -3.1-Geräte sowie externe SCSI-Geräte wie Streamer.

Hypervisoren

Im Allgemeinen unterscheidet man zwischen zwei unterschiedlichen Ansätzen bei der Virtualisierung. Von einem Hypervisor des Typs 1 ist die Rede, wenn ohne installiertes Betriebssystem eine Virtualisierung von Gastbetriebssystemen direkt auf der Hardware möglich ist, wohingegen der Typ 2 ein vollständiges Betriebssystem voraussetzt und die Virtualisierung durch eine zusätzliche

Anwendung zur Verfügung gestellt wird. Beim ersten Typ spricht man auch häufig von *Bare Metal Hypervisor* oder auch *Paravirtualisierung*. Zum Typ 1 zählen zum Beispiel der ESXi-Server, XEN, KVM (Kernel-based Virtual Machine), dem Typ 2 ordnet man unter anderem die Workstation und Serverversionen von VMware und Microsoft zu sowie Produkte von Bochs, VirtualBox, QEMU usw.

Virtualisierung mit VMware

Das amerikanische Unternehmen VMware Inc. wurde 1998 in Palo Alto in Kalifornien gegründet, um ein Produkt zu entwickeln, das virtuelle Maschinen auf normalen Computern lauffähig machen kann. Der Mutterkonzern EMC Corporation brachte die Tochter im August 2007 an die Börse und verkaufte nur 10% der Aktien. Nach dem ersten Tag stiegen die VMware-Aktien von 29 auf 54 US-Dollar und brachten an dem einen Tag über 19 Milliarden US-Dollar Gewinn. Mittlerweile gehört EMC und damit auch die Tochterfirma VMware zu Dell.

VMware-Produkte

Als erstes Produkt brachte VMware die WORKSTATION heraus, die heute immer noch das bekannteste Produkt ist und mittlerweile in der Version 15 vorliegt. Das Produkt heißt mittlerweile »VMware Workstation Pro«. Diese Applikation wird auf einem bestehenden Betriebssystem (Windows oder Linux) installiert. Ähnlich verhielt es sich mit dem kostenlosen Nachfolger des ehemaligen GSX-Servers, dem VMware SERVER, der aber nicht weiterentwickelt wird. Sowohl Workstation als auch Server sind Anwendungen, die aufgrund des darunterliegenden

Betriebssystems und eingeschränkten Hardwareausbaus meist nur wenige VMs gleichzeitig laufen lassen können. Dabei ist es egal, ob das Betriebssystem Linux oder Windows ist und ob es sich um z.B. Windows 10 oder einen Server handelt.

Weiterhin gibt es für den nicht kommerziellen Gebrauch den kostenlosen VMware PLAYER, der eigentlich nur vorhandene virtuelle Maschinen abspielen konnte. Seit der Version 3.0 ist man aber damit ebenfalls in der Lage, neue VMs zu generieren, aktuell liegt er in der Version 15 vor und nennt sich »VMware Workstation Player«. Der Player läuft unter 32-Bit-Betriebssystemen nur allein, unter 64 Bit auch mit Server oder Workstation zusammen, wobei es dann auf die Installationsreihenfolge und die Versionen der beiden Anwendungen ankommt. Da die Konfigurationsmöglichkeiten hier eingeschränkter sind als beispielsweise bei VMware Workstation Pro, aber die Anwendung deutlich kleiner ist und weniger Ressourcen verbraucht, ist der Player durchaus dazu geeignet, als Ersatz für Workstation mit wenigen VMs eingesetzt zu werden.

VMware FUSION ist eine Virtualisierungssoftware für Apple Macintosh-Rechner, die auf einem Intel-Prozessor laufen. Damit hat man die Möglichkeit, Linux und Windows im Fenster zu installieren und laufen zu lassen. Unter Apple-Betriebssystemen ist dieses die einzige lauffähige Möglichkeit von VMware, denn weder Workstation und Server noch der Player können hier installiert werden. Fusion gibt es auch als Pro-Version, die deutlich mehr Funktionen mitbringt.

VMware ACE (Assured Computing Environment) bietet einem ACE-Manager die Möglichkeit, VMs zu erstellen, die zusätzliche Sicherheitsrichtlinien – unabhängig vom Betriebssystem – aufweisen. Zu den Einstellungen gehören

Zugriffsrechte, Systemressourcen wie Netzwerk, Drucker, Laufwerke und Ähnliches. ACE ist als Bestandteil in Workstation und Player Pro enthalten, es gibt aber auch noch weitere Komponenten (ACE Management Server), die man zusätzlich käuflich erwerben kann.

Als Desktop- und Anwendungs-Virtualisierungslösung gibt es bei VMware HORIZON (ehemals View), HORIZON CLOUD, HORIZON APPS und HORIZON FLEX jeweils in der Version 8. Die Namensgebung bei der Vorgängerversion, Virtual Desktop Infrastructure (VDI), war da schon aussagekräftiger. Der Sinn dieser Software ist es, komplette Arbeitsplatzrechner auf Hosts zur Verfügung zu stellen und nicht mehr am Arbeitsplatz. Die Technologie lässt sich mit Terminals oder Thin Clients vergleichen, die auf einen Terminalserver zugreifen – nur wird hier nicht eine Anwendung zur Verfügung gestellt, sondern ein ganzer Rechner. Bei HORIZON CLOUD (ehemals AIR DESKTOPS) liegt der Schwerpunkt auf VMware vCloud, wohingegen HORIZON FLEX eher die Unterstützung für Macs und Windows bietet.

Will man ähnlich wie bei virtuellen Maschinen nur eine Applikation zur Verfügung stellen, so bietet VMware das Produkt ThinApp. Damit ist es möglich, jegliche Version jeglicher Software als Applikation für die Benutzer zur Verfügung zu stellen. ThinApp nutzt die Thinstall-Technik, um Anwendungen in ein MSI-Paket (Microsoft Software Installation) oder eine EXE-Datei zu packen und dann auf einem Server, dem Netzwerk oder einem USB-Stick teilweise ohne Installation zur Verfügung zu stellen.

VMware vSphere with Kubernetes (auch vSphere Integrated Container, VIC) ist das neue Highlight bei der aktuellen Version. Damit können Container statt ganzer VMs zur Verfügung gestellt werden. Somit muss nicht mehr für jede Applikation eine eigene VM bereitgestellt werden,

sondern nur eine isolierte Umgebung innerhalb des Betriebssystems der VM – der Container.

Für mittelständische bis große Unternehmen war Virtual Infrastructure gedacht, das jetzt vSphere 7.0 heißt. Der große Unterschied zu den oben genannten Lösungen ist das Zusammenfassen von mehreren Servern zu einer Ressource, die Verwaltung von mehreren Virtualisierungsservern und virtuellen Maschinen von einem Rechner aus, die Ausfallsicherheit und viele weitere Funktionen, die wir in den nächsten Kapiteln ausführlich besprechen werden.

Warum VMware?

Es gibt auf dem Markt mittlerweile sehr viele kommerzielle und Open-Source-Lösungen für die Virtualisierung von Rechnern. Viele davon sind mit den oben genannten vergleichbar, mit einer Ausnahme: VMware vSphere. Viele Funktionen, die bei diesem System seit einigen Jahren verfügbar sind, werden bisher nicht von anderen Herstellern angeboten. Einen Vergleich braucht VMware bezüglich Funktionen, Sicherheit, Skalierbarkeit, Zuverlässigkeit etc. nicht zu scheuen. Auch die neue Version von Microsoft »Hyper-V« ist gegenüber vSphere in einigen Details noch eingeschränkt.

Häufig wird mir in Gesprächen gesagt, dass Hyper-V von Microsoft kostenlos ist. Der alleinstehende ESXi-Host ist ebenfalls kostenlos, erst wenn man einen vCenter Server einsetzen will und ein Datacenter braucht, muss man Lizenzen kaufen, und das ist bei dem Microsoft-Produkt genauso.

KAPITEL 2

Die Bestandteile von vSphere

In diesem Kapitel erhalten Sie einen groben Überblick über die einzelnen Bestandteile einer vSphere-7-Umgebung sowie ihrer Begrifflichkeiten und Funktionen. Außerdem werden wir die Unterschiede der verschiedenen Editionen besprechen. Die einzelnen ESXi-Einsatzszenarien werden im dritten Kapitel unter die Lupe genommen.

Komponenten der vSphere-Umgebung

»VMware vSphere 7« besteht aus mehreren Komponenten, die z.T. unabhängig voneinander sind wie z.B. ein Betriebssystem und eine Anwendung. Mit einer Anwendung alleine kann man allerdings nichts anfangen – man muss ein unterstütztes Betriebssystem haben, auf dem diese installiert werden kann. Genauso verhält es sich mit den Bestandteilen von VMware vSphere: Es ist ein erweiterbares Paket mit Einzelteilen, die in einigen Fällen nur Sinn machen, wenn man auch andere Teile hat.

Zunächst ist der ESXi-Server, auch Host genannt, zu erwähnen. Er ist ein unabhängiger Computer, auf dem die Installation der Virtualisierungsschicht, also des eigentlichen Hypervisors, durchgeführt wird. Am Host selbst kann man nur sehr bedingt eine virtuelle Maschine erstellen oder verwalten, dafür benötigt man eine weitere Anwendung, die z.B. auf einem Client installiert wird.

Das ist dann der vSphere oder HTML5-Client. In Abbildung 2-1 sehen Sie, wie ein Client-PC auf den Host zugreift.

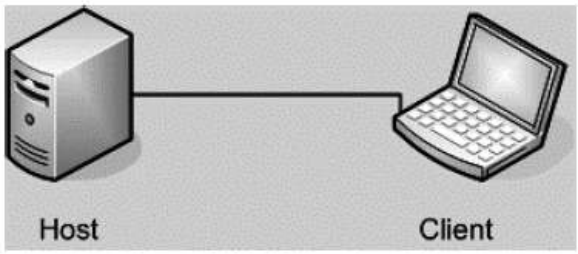


Abbildung 2-1: Hostzugriff über einen Client

Mit diesem Client kann man alle täglich anfallenden Arbeiten auf dem Host durchführen, auch die Konfiguration des Hosts, das Erstellen der virtuellen Maschinen, das Starten und Herunterfahren, Backup und Recovery usw.

Da man die Browsersitzung auch mehrfach starten kann, besteht die Möglichkeit, auch mehrere unabhängige Hosts gleichzeitig zu verwalten, wie in Abbildung 2-2 dargestellt. Das geschieht dann allerdings in verschiedenen Fenstern und es ist nicht möglich, virtuelle Maschinen von einem ESXi-Server auf den anderen zu verschieben oder Ressourcen wie Arbeitsspeicher, CPU-Leistung etc. aufzuteilen oder zu bündeln.

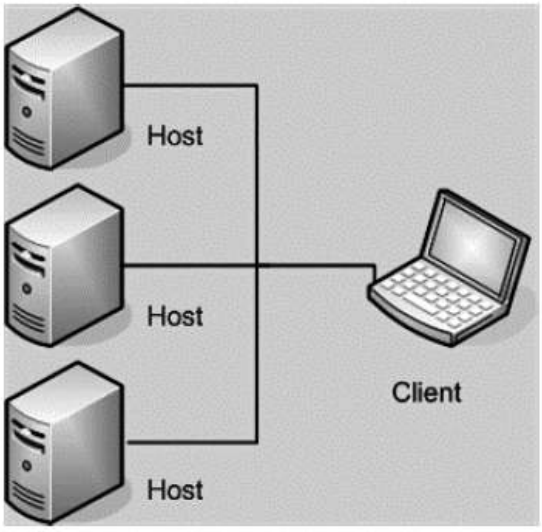


Abbildung 2-2: vSphere Client greift auf drei Hosts zu.