

Virtualisierungstechnologien

Christian Baun

Karlsruher Institut für Technologie
Steinbuch Centrum für Computing
baun@kit.edu

9.7.2010

Agenda

- Virtualisierung
 - Vorteile von Virtualisierung
 - Nachteile und Grenzen von Virtualisierung
- Konzepte der Virtualisierung
 - Partitionierung
 - Hardware-Emulation
 - Applikationsvirtualisierung
 - Virtueller Maschinen-Monitor
 - Paravirtualisierung
 - Hardware-Virtualisierung
 - Betriebssystem-Virtualisierung bzw. Container bzw. Jails
 - Speichervirtualisierung
 - Netzwerkvirtualisierung

Virtualisierung – Grundlagen

- Durch Virtualisierung werden die Ressourcen eines Rechnersystems aufgeteilt und von mehreren unabhängigen Betriebssystem-Instanzen genutzt
- Virtualisierung ist stellvertretend für mehrere grundsätzlich verschiedene Konzepte und Technologien
- Jede virtuelle Maschine (VM) verhält sich wie ein vollwertiger Computer mit eigenen Komponenten, der in einer abgeschotteten Umgebung auf einer realen Maschine läuft
- In einer VM kann ein Betriebssystem mit Anwendungen genau wie auf einem realen Computer installiert werden.
 - Die Anwendungen merken nicht, dass sie sich in einer VM befinden
- Anforderungen der Betriebssystem-Instanzen werden von diesen unbemerkt von der Virtualisierungssoftware abgefangen und auf die real vorhandene oder emulierte Hardware umgesetzt

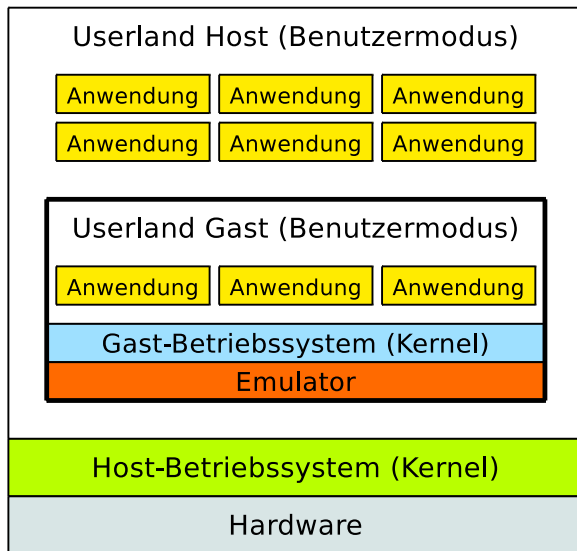
Ursprung der Virtualisierung

- Virtualisierung ist kein neues Konzept
 - Einführung bereits vor 40 Jahren bei Großrechnern
- IBM stellte in den 1960er-Jahren die Virtual Machine Facility/370, kurz VM/370 vor
 - Auf dieser Plattform wurde Mehrbenutzerbetrieb gefahren, indem mehrere Einzelbenutzerbetriebinstanzen in virtuellen Maschinen ausführt wurden
 - Jede VM stellte eine vollständige Nachbildung der darunter liegenden, physischen Hardware dar
- Quellen:
 - Creasy RJ (1981) The origin of the VM/370 time-sharing system. IBM J Res Dev 25(5):483-490
 - Amit Singh A (2005) An Introduction to Virtualization
<http://www.kernelthread.com/publications/virtualization>

Virtualisierungskonzepte

- Es existieren unterschiedliche Virtualisierungskonzepte:
 - Partitionierung
 - Hardware-Emulation
 - Applikationsvirtualisierung
 - Virtueller Maschinen-Monitor
 - Paravirtualisierung
 - Hardware-Virtualisierung
 - Betriebssystem-Virtualisierung bzw. Container bzw. Jails
 - Speichervirtualisierung
 - Netzwerkvirtualisierung
 - ...

Schema der Emulation



Auswahl an Emulatoren

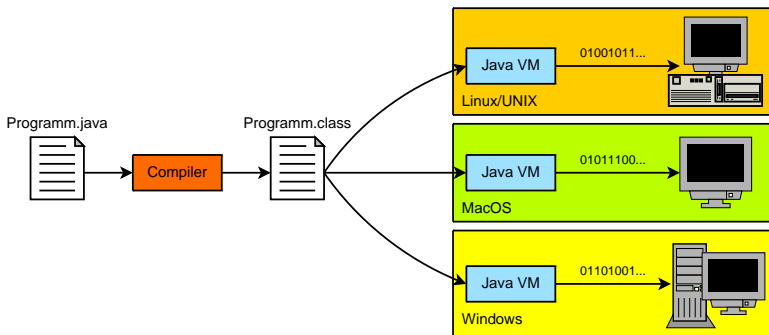
Name	Lizenz	Host	Emulierte Architektur	Gast-System
Bochs v2.3.6	LGPL	Linux, Solaris, MacOS, Windows, IRIX, BeOS	x86, AMD64	Linux, DOS, BSD, Windows, BeOS
QEMU v0.9.0	GPL	Linux, BSD, Solaris, BeOS, MacOS-X	x86, AMD64, PowerPC, ARM, MIPS, Sparc	Linux, MacOS-X, Windows, BSD
DOSBox v0.72	GPL	Linux, Windows, OS/2, BSD, BeOS, MacOS-X	x86	DOS
DOSEMU v1.4.0	GPL	Linux	x86	DOS, Windows bis 3.11
PearPC v0.4.0	GPL	Linux, MacOS-X Windows	PowerPC	Linux, MacOS-X, BSD
Baseilisk II v0.9-1	GPL	Linux, diverse UNIX, Windows NT4, BeOS, Mac OS, Amiga OS	680x0	MacOS \leq 8.1
Wabi v2.2	proprietär	Linux, Solaris	x86	Windows 3.x
MS Virtual PC v7	proprietär	MacOS-X	x86	Windows, (Linux)
M.A.M.E. v0.137	MAME-Lizenz	Linux, Windows, DOS, BeOS, BSD, OS/2	diverse Arcade	diverse Arcade
SheepShaver	GPL	Linux, MacOS-X, BSD Windows, BeOS	PowerPC, 680x0	MacOS 7.5.2 bis MacOS 9.0.4
Hercules 3.07	QPL	Linux, MacOS-X, BSD Solaris, Windows	IBM-Großrechner	IBM System/360, 370, 390

- Die Tabelle erhebt keinen Anspruch auf Vollständigkeit!

Applikationsvirtualisierung

- Bei der Applikationsvirtualisierung werden Anwendungen lokal, unter Verwendung lokaler Ressourcen in einer virtuellen Umgebung ausgeführt, die alle Komponenten bereitstellt, die die Anwendung benötigt
 - Eine VM befindet sich zwischen der auszuführenden Anwendung und dem Betriebssystem
- Populärstes Beispiel: Die Java Virtual Machine (JVM)
 - Die JVM ist der Teil der Java-Laufzeitumgebung (JRE), der für die Ausführung des Java-Bytecodes verantwortlich ist
 - Die JVM ist für die Java-Programme die Schnittstelle zum Rechnersystem und dessen Betriebssystem
- Vorteil: Plattformunabhängigkeit
- Nachteil: Geringere Ausführungsgeschwindigkeit gegenüber nativer Programmausführung

Prinzip der Java Virtual Machine (JVM)



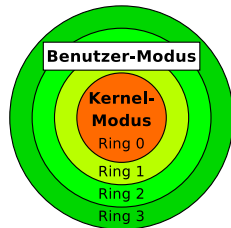
- Der Java-Compiler `javac` übersetzt den Quellcode in architekturunabhängige `.class`-Dateien, die Bytecode enthalten, der in der Java Virtual Machine lauffähig ist
- Das `java`-Programm startet eine Java-Applikation in einer Instanz der Java Virtual Machine

VMware ThinApp

- Ein weiteres Beispiel für Applikationsvirtualisierung ist VMware ThinApp
 - <http://www.vmware.com/products/thinapp/>
 - Bis 2008 unter dem Namen Thinstall bekannt
- Eine Anwendung wird in eine einzelne .exe-Datei gepackt
- Die Anwendung wird dadurch portabel und kann ohne lokale Installation verwendet werden
 - Die Anwendung kann u.a. auf einem USB-Stick ausgeführt werden
- Es erfolgen keine Einträge in der Windows Registry. Es werden auch keine Umgebungsvariablen und DLL-Dateien auf dem System erstellt
- Benutzereinstellungen und erstellte Dokumente werden in einer eigenen Sandbox gespeichert
- Nachteil: Funktioniert ausschließlich mit Microsoft Windows
 - Unter Linux ist die Nutzung mit Wine möglich

Virtualisierungsgrundlagen in der x86-Architektur (1)

- Zum Verständnis der Virtualisierung in der x86-Architektur muss das Schutzkonzept der Ringe betrachtet werden
- x86-kompatible CPUs enthalten vier Privilegienstufen zum Speicherschutz, um die Stabilität und Sicherheit zu erhöhen
- Ein Prozess kann immer nur in einem einzelnen Ring ausgeführt werden und ist nicht in der Lage, sich selbständig aus diesem zu befreien
- Die verbreiteten Betriebssysteme nutzen nur Ring 0 und 3
- Eine Ausnahme bildet OS/2
 - OS/2 nutzt Ring 2 für Anwendungen, die auf Hardware und Eingabe-/Ausgabeschnittstellen zugreifen dürfen
 - Zu dieser Gruppe gehörten bei OS/2 die Grafiktreiber



Virtualisierungsgrundlagen in der x86-Architektur (2)

- Nur Prozesse in Ring 0 haben vollen Zugriff auf die Hardware und dürfen den vollständigen Befehlsatz der CPU nutzen
 - Ring 0 wird auch als Kernel-Bereich (Kernel-Space) bezeichnet
 - In Ring 0 läuft nur der Betriebssystemkern (Kernel) und die zum Start des Betriebssystems notwendigen Hardwaretreiber
- In Ring 3 laufen die Anwendungen
 - Ring 3 wird auch als Benutzerbereich (User-Space) bezeichnet
- Ruft ein Prozess in einem weniger privilegierten Ring eine privilegierte Operation auf, erzeugt die CPU eine Ausnahme (Exception)
 - Die Exception wird im benachbarten privilegierteren Ring abgefangen und dort behandelt
 - Ausnahmen, die nicht abgefangen werden können, verursachen eine allgemeine Schutzverletzung (General Protection Fault)
 - Der aufrufende Prozess stürzt ab
 - Handelt es sich bei dem Prozess um den Kernel, stürzt das System ab

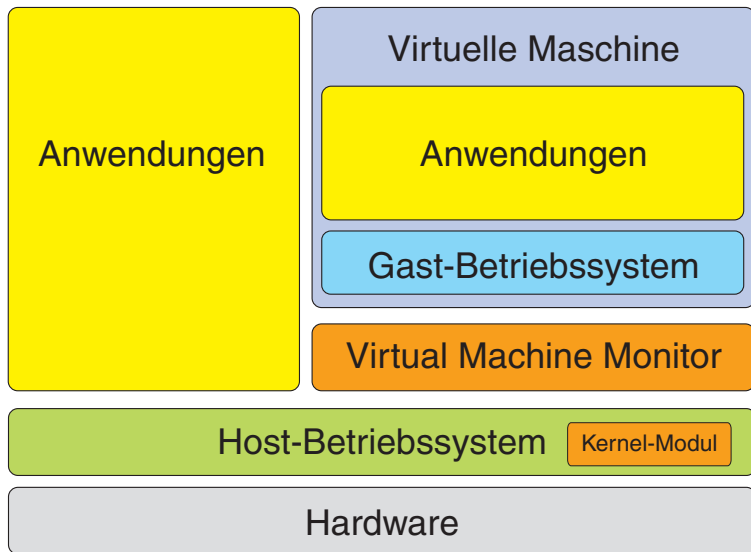
Vollständige Virtualisierung (1)

- Vollständige Virtualisierungslösungen bieten einer VM eine vollständige, virtuelle PC-Umgebung inklusive eigenem BIOS
 - Jedem Gastbetriebssystem steht ein eigener virtueller Rechner mit virtuellen Ressourcen wie CPU, Hauptspeicher, Laufwerken, Netzwerkkarten, usw. zur Verfügung
- Es kommt ein **Virtueller Maschinen-Monitor** (VMM) zum Einsatz
- Der VMM läuft *hosted* als Anwendung unter dem Host-Betriebssystem
- Der VMM verteilt die Hardwareressourcen des Rechners an die VMs
- Teilweise emuliert der VMM Hardware, die nicht für den gleichzeitigen Zugriff mehrerer Betriebssysteme ausgelegt ist
 - Ein Beispiel sind Netzwerkkarten
 - Emulation populärer Hardware vermeidet Treiberprobleme

Vollständige Virtualisierung (2)

- Vollständige Virtualisierung nutzt die Tatsache, dass x86-Systeme nur zwei von vier möglichen Privilegienstufen verwenden
 - Der VMM befindet sich in Ring 0 auf der Ebene des Betriebssystemkerns des Host-Betriebssystems und hat vollen Zugriff auf die Hardware
 - Die VMs befinden sich in einem der weniger privilegierten Ringe
 - Der VMM stellt für jede denkbare Ausnahme eine Behandlung zur Verfügung, die die privilegierten Operationen der Gastbetriebssysteme abfängt, interpretiert und ausführt
 - Der VMM stellt sicher, dass die VMs nur über den Umweg des VMM Zugriff auf die Hardware erhalten
 - Kontrollierter Zugriff auf die gemeinsam genutzten Systemressourcen ist gewährleistet

Schema des Virtual Machine Monitor



Vollständige Virtualisierung (3)

- Vorteile der Vollständigen Virtualisierung:
 - Kaum Änderungen an Host- und Gast-Betriebssystemen erforderlich
 - Zugriff auf die wichtigsten Ressourcen wird nur durchgereicht
⇒ Fast native Verarbeitungsgeschwindigkeit der Gast-Betriebssysteme
 - Jedes Gast-Betriebssystem hat seinen eigenen Kernel
⇒ Hohe Flexibilität
- Nachteile der Vollständigen Virtualisierung:
 - Wechsel zwischen den Ringen erfordern einen Kontextwechsel
⇒ Jeder Kontextwechsel verbraucht Rechenzeit
 - Fordert eine Applikation im Gast-Betriebssystem die Ausführung eines privilegierten Befehls an, liefert der VMM eine Ersatzfunktion und diese weist die Ausführung des Befehls über die Kernel-API des Host-Betriebssystems an
⇒ Geschwindigkeitseinbußen

Beispiele für Vollständige Virtualisierung

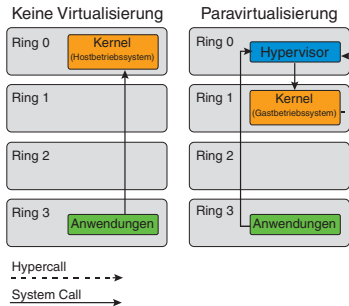
- Beispiele für Virtualisierungslösungen, die auf dem Konzept des VMM basieren, sind:
 - VMware Server, VMware Workstation und VMware Fusion
 - Microsoft Virtual PC (in der Version für x86)
 - Parallels Desktop und Parallels Workstation
 - VirtualBox
 - Kernel Virtual Machine (KVM)
 - Mac-on-Linux (MoL)

Paravirtualisierung (1)

- Bei Paravirtualisierung wird keine Hardware virtualisiert oder emuliert
 - Den Gast-Betriebssystemen steht keine emulierte Hardwareebene zur Verfügung, sondern eine API
- Virtuell gestartete Betriebssysteme verwenden eine abstrakte Verwaltungsschicht, den **Hypervisor**, um auf die physischen Ressourcen wie Speicher, Ein-/Ausgabegeräte und Netzwerkinterfaces zuzugreifen
 - Der Hypervisor ist quasi ein auf ein Minimum reduziertes **Metabetriebssystem**, das die Hardwareressourcen unter den Gastsystemen verteilt, so wie ein Betriebssystem dieses unter den laufenden Prozessen tut
 - Der Hypervisor läuft *bare metal*
 - Ein Metabetriebssystem ermöglicht den unabhängigen Betrieb unterschiedlicher Anwendungen und Betriebssysteme auf einer CPU
- Das Host-Betriebssystem (Dom0) läuft nicht im privilegierten Ring 0, sondern im weniger privilegierten Ring 1
 - Ein Host-Betriebssystem ist wegen der Gerätetreiber notwendig

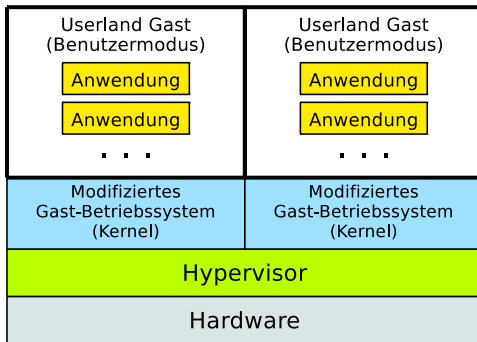
Paravirtualisierung (2)

- Weil die Betriebssysteme (Kernel) nicht mehr in Ring 0, sondern in Ring 1 laufen, können diese keine privilegierten Anweisungen ausführen
 - Als Lösung stellt der Hypervisor **Hypercalls** zur Verfügung
- Hypercalls sind vergleichbar mit Systemaufrufen (System Calls)
 - Die Interrupt-Nummern sind verschieden
 - Fordert eine Anwendung die Ausführung eines Systemaufrufs an, wird eine Ersatzfunktion im Hypervisor aufgerufen
 - Der Hypervisor weist die Ausführung des Systemaufrufs über die Kernel-API des Betriebssystems an



Paravirtualisierung (3)

- Erweiterung des Kernels um die Hypercall-Funktionalität macht eine Modifikation der Betriebssysteme notwendig
- Abfangen und Prüfen aller Systemaufrufe durch den Hypervisor führt nur zu geringen Geschwindigkeitseinbußen

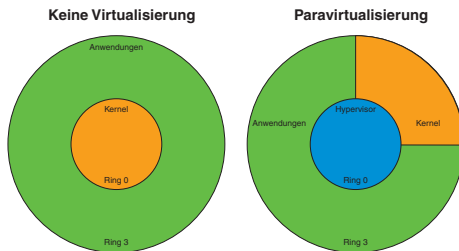


Paravirtualisierung (4)

- Beispiele: Xen, Citrix Xenserver, Virtual Iron, VMware ESX Server
- Nachteil:
 - Kernel der Gast-Betriebssysteme müssen speziell für den Betrieb in einem paravirtualisierten Kontext angepasst sein
 - Rechteinhaber proprietärer Betriebssysteme lehnen eine Anpassung aus strategischen Gründen häufig ab
⇒ Funktioniert häufig nur mit OpenSource-Betriebssystemen
- Vorteil:
 - Geschwindigkeitseinbußen, die beim VMM entstehen, werden vermieden

Problem: x86-64-Architektur (1)

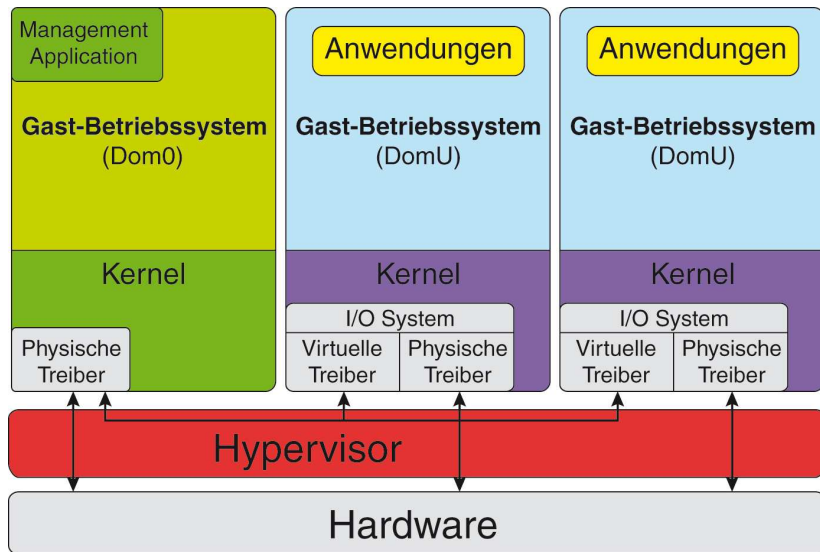
- Bei der x86-64-Architektur wurde auf die als unnötig erachteten Ringe 1 und 2 verzichtet
 - x86-64-Architekturen wie IA64 haben nur zwei Privilegienstufen zum Speicherschutz
- Der Hypervisor befindet sich wie bei der x86-32-Architektur in Ring 0
- Der Kernel mit seinem Betriebssystemkern wird bei der x86-64-Architektur in Ring 3 zu den Anwendungen verschoben



Problem: x86-64-Architektur (2)

- Um Datensicherheit zu gewährleisten, haben die unprivilegierten Domänen keinen direkten Hardwarezugriff
- Zugriff auf die physikalischen Rechnerressourcen erfolgt über virtuelle Gerätetreiber
 - I/O-Anforderungen werden aus den unprivilegierten Domänen über den Hypervisor direkt an den physischen Gerätetreiber innerhalb der privilegierten Domäne weitergeleitet
 - I/O Anforderungen werden somit gesammelt und die Hardware- und Datenzugriffe kanalisiert
 - Treiber innerhalb der unprivilegierten Domänen reichen die Dateien einfach weiter

Virtuelle Gerätetreiber bei Xen



Zusammenfassung: Voll- vs. Paravirtualisierung

- Bei Vollvirtualisierung kann man unveränderte Systeme betreiben
- Paravirtualisierung erfordert angepasste Gastsysteme
- Der VMM läuft *hosted* als Anwendung unter einem klassischen Betriebssystem wie Linux oder Windows
 - Beispiele: VMware Workstation und KVM
- Der Hypervisor läuft *bare metal* anstatt eines klassischen Betriebssystems
 - Beispiele: VMware ESX und Xen

Hardware-Virtualisierung (1)

- Aktuelle CPUs von Intel und AMD implementieren Virtualisierungserweiterungen, die man als Hardware-Virtualisierung bezeichnet
- Dank Hardware-Virtualisierung können unmodifizierte Betriebssysteme (z.B. Windows) mit Xen oder VMware ESX laufen
- Intel und AMD gehen bei der Realisierung ähnliche, aber inkompatible Wege
- AMD erweitert seit Juni 2006 seine AMD64 CPUs um den sogenannten Secure-Virtual-Machine-Befehlssatz (**SVM**)
 - Die Lösung heißt **AMD-V** und war vorher als **Pacifica** bekannt
- Die Lösung von Intel heißt **VT-x** für IA32-CPU's und **VT-i** für Itanium
 - Intels Lösung lief vormals unter dem Stichwort **Vanderpool**
- Xen unterstützt ab Version 3 Hardware-Virtualisierung
- Auch Windows Server 2008 (Hyper-V) nutzt Hardwarevirtualisierung

Hardware-Virtualisierung (2)

- Kern der Neuerung ist eine Überarbeitung der Privilegienstruktur
 - Die neuen Befehle bei AMD und Intel bieten VMs eine Erweiterung zu den bereits beschriebenen Privilegienstufen Ring 0 und Ring 3
- Die Ringstruktur wurde durch eine Erweiterung von Ring 0 um eine Ebene, die neue Hypervisor-Schicht, ergänzt
 - Die Ebene wird als Root-Betriebsmodus oder Ring -1 bezeichnet
 - Der Hypervisor bzw. VMM läuft im Root-Betriebsmodus und besitzt jederzeit die volle Kontrolle über die CPU und die Ressourcen, da damit ein höheres Privileg als Ring 0 implementiert ist
- Virtuellen Betriebssysteminstanzen steht so der gewohnte Zugriff auf die Befehle von Ring 0 zur Verfügung
- Vorteile:
 - Gastbetriebssysteme müssen nicht angepasst werden
 - Der Kernel läuft nicht wie bei der Paravirtualisierung mit den Anwendungen auf einer Privilegienstufe

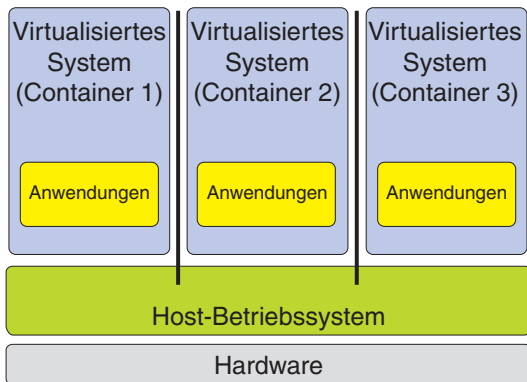
Betriebssystem-Virtualisierung / Container / Jails (1)

- Bei der Betriebssystem-Virtualisierung bzw. **Container**, spielt das Host-Betriebssystem eine entscheidende Rolle
 - Die Container werden unter BSD häufig als **Jails** bezeichnet
- Es laufen unter ein und demselben Kernel mehrere voneinander abgeschottete identische Systemumgebungen
- Es wird kein zusätzliches Betriebssystem, sondern eine isolierte Laufzeitumgebung virtuell in einem geschlossenen Container erzeugt
- Nach außen treten die virtuellen Umgebungen wie eigenständige Systeme auf
- Alle laufenden Anwendungen verwenden denselben Kernel
- Anwendungen sehen nur andere Anwendungen, mit denen sie sich in einer virtuellen Umgebung befinden
- Vorteile sind der geringe Ressourcenbedarf und die hohe Performance, da der Kernel in gewohnter Weise die Hardware des Systems verwaltet

Betriebssystem-Virtualisierung / Container / Jails (2)

- Nachteil: Alle virtuellen Umgebungen nutzen den gleichen Kernel
 - Es können nur mehrere unabhängige Instanzen eines Betriebssystems gestartet werden
 - Verschiedene Betriebssysteme können nicht gleichzeitig verwendet werden
- Diese Form der Virtualisierung wird hauptsächlich genutzt, um Anwendungen in isolierten Umgebungen zu betreiben und somit eine höhere Sicherheit zu gewährleisten
- Beispiele sind:
 - SUN Solaris
 - OpenVZ für Linux
 - Linux-VServer
 - FreeBSD Jails
 - Virtuozzo (kommerzielle Variante von OpenVZ)
 - FreeVPS

Schema der Betriebssystem-Virtualisierung



- Besonders Internet-Service-Provider, die (virtuelle) Root-Server oder Webdienste auf Mehrkernprozessorarchitekturen anbieten, nutzen diese Form der Virtualisierung
 - Wenig Performance-Verlust, hoher Grad an Sicherheit

Speichervirtualisierung

- Bei Speichervirtualisierung wird Speicher in Form virtueller Laufwerke den Benutzern zur Verfügung gestellt
 - Der logische Speicher wird vom physischen Speicher getrennt
- Die Benutzer profitieren, weil sie nicht an die physischen Grenzen realer Speichermedien gebunden sind
- Umstrukturieren oder Erweitern des physischen Speichers stört die Benutzer nicht
- Datensicherung und redundantes Vorhalten erfolgt transparent im Hintergrund
- Besserer Auslastungsgrad, da der verfügbare physische Speicher effektiver auf die vorhandenen Benutzer aufgeteilt werden kann
- Bekannte Anbieter von Lösungen zur Speichervirtualisierung sind EMC, HP, IBM, LSI und SUN

Netzwerkvirtualisierung

- Netzwerkvirtualisierung basiert auf virtuellen lokalen Netzen, sogenannten Virtual Local Area Networks (VLANs)
- Verteilt aufgestellte Geräte können durch VLANs in einem einzigen logischen Netzwerk zusammengefasst werden
 - Nützlich bei der Konzeption der IT-Infrastruktur verteilter Standorte
 - Ein VLAN trennt physikalische Netze in logische Teilnetze auf, indem es dafür sorgt, dass Datenpakete eines VLANs nicht in ein anderes VLAN weitergeleitet werden, obwohl die Teilnetze an gemeinsamen Switches angeschlossen sein können
 - Ein VLAN bildet ein nach außen abgeschirmtes Netzwerk über fremde oder nicht vertrauenswürdige Netze
 - Zu schützende Systeme können in einem eigenen Netz verborgen werden
- Aktuelle Vorgehensweise: VLAN-Markierung der Netzwerkpakete
 - Stichwort: Tagged VLAN
- Nachteil: Steigender Aufwand für die Netzwerkadministration

Gründe für Virtualisierung (1)

- Bessere Ausnutzung der Hardware
 - Serverkonsolidierung. Zusammenlegen vieler (virtueller) Server auf weniger physischen Servern
 - Kostensenkung bei Hardware, Verbrauchskosten (Strom, Kühlung), Stellplätze, Administration, usw.
 - Gartner geht davon aus, dass durch (Server-)Virtualisierung die Investitionen in neue Hard- und Software um bis zu 70% sinken können
 - Im Rechenzentrum sind Kosteneinsparungen von bis zu 50% erreichbar
- Vereinfachte Administration
 - Anzahl physischer Server wird reduziert
 - Ausgereifte Managementwerkzeuge existieren
 - VMs können im laufenden Betrieb verschoben werden (Live Migration)
 - Wartung und Technologiewechsel (der Virtualisierungsplattform) ohne Betriebsunterbrechung möglich (gilt nicht für die VMs!)
- Vereinfachte Bereitstellung
 - Neue Infrastrukturen und Server können innerhalb von Minuten manuell oder automatisch erzeugt werden

Gründe für Virtualisierung (2)

- Maximale Flexibilität
 - VMs können leicht vervielfältigt und gesichert werden
 - Snapshots vom aktuellen Zustand einer VM können erzeugt und wieder hergestellt werden
- Höhere Sicherheit
 - VMs sind gegenüber anderen VMs und dem Host-System isoliert
 - Unternehmenskritische Anwendungen können in einer VM gekapselt und so in einer sicheren Umgebung laufen
 - Ausfall einer VM tangiert die übrigen VMs und den Host nicht
- Optimierung von Software-Tests und Software-Entwicklung
 - Gleichzeitiger Betrieb mehrerer Betriebssysteme
 - Testumgebungen können schnell aufgesetzt werden
- Unterstützung alter Anwendungen
 - Legacy-Betriebssysteme oder Legacy-Anwendungen, für die keine Hardware mehr zu bekommen ist, können reanimiert werden

Nachteile und Grenzen der Virtualisierung

- Geringere Performance
 - Aktuelle Virtualisierungstechnologien sind so ausgereift, dass sich der Leistungsverlust mit 5-10% nicht sonderlich auswirkt
 - Da aktuelle Mehrkernprozessorsysteme (Intel VT/VT-x und AMD-V) mit Virtualisierung besonders effektiv genutzt werden können, spielt der Leistungsverlust eine zunehmend untergeordnete Rolle
- Nicht jede Hardware kann angesprochen oder emuliert werden
 - Kopierschutzstecker (Hardwaredongles) sind ein Problem
 - Beschleunigte Grafik kann nicht immer realisiert werden
- Beim Ausfall eines Hosts würden mehrere virtuelle Server ausfallen
 - Ausfallkonzepte und redundante Installationen sind notwendig
- Virtualisierung ist komplex
 - Zusätzliches Know-how ist notwendig

Eine kleine Kalkulation der Energiekosten

- Ein Unternehmen X hat 1000 Computerarbeitsplätze
- Szenario 1: Fat Clients (PC)
 - Elektrische Anschlussleistung pro Desktopsystem: 300 Watt
 - Elektrische Anschlussleistung pro Bildschirm: 100 Watt
- Szenario 2: Thin Clients
 - Elektrische Anschlussleistung pro Thin Client: 20 Watt
 - Elektrische Anschlussleistung pro Bildschirm: 100 Watt
 - Elektrische Anschlussleistung pro Server-Blade: 200 Watt
 - Auf ein Server-Blade passen 40 virtuelle Desktopsysteme
- Was kostet der jährliche Dauerbetrieb (24/7) bei 0,21 €/kWh?

Dieses Beispiel soll zeigen, wie einfach durch Virtualisierung Energiekosten eingespart werden können.

Szenario 1: Fat Clients (PC)

- Anschlussleistung Desktopsystem: 300 Watt
- Anschlussleistung Bildschirm: 100 Watt
- Summe der Anschlussleistung: 400 Watt
- Stromkosten (mit Schaltjahr) pro Computerarbeitsplatz

$$0,4 \text{ kW} * 24 \frac{\text{h}}{\text{Tag}} * 365,25 \frac{\text{Tag}}{\text{Jahr}} * 0,21 \frac{\text{€}}{\text{kWh}} = 736,344 \frac{\text{€}}{\text{Jahr}}$$

- Stromkosten für 1000 Computerarbeitsplätze

$$1000 * 736,344 \frac{\text{€}}{\text{Jahr}} = 736344 \frac{\text{€}}{\text{Jahr}}$$

Szenario 2: Thin Clients (1)

- Anschlussleistung Thin Client: 20 Watt
- Anschlussleistung Bildschirm: 100 Watt
- Summe der Anschlussleistung: 120 Watt
- Stromkosten (mit Schaltjahr) pro Computerarbeitsplatz (ohne Server)

$$0,12 \text{ kW} * 24 \frac{\text{h}}{\text{Tag}} * 365,25 \frac{\text{Tag}}{\text{Jahr}} * 0,21 \frac{\text{€}}{\text{kWh}} = 220,9032 \frac{\text{€}}{\text{Jahr}}$$

- Anschlussleistung Server-Blade: 200 Watt
- Stromkosten (mit Schaltjahr) pro Server-Blade

$$0,2 \text{ kW} * 24 \frac{\text{h}}{\text{Tag}} * 365,25 \frac{\text{Tag}}{\text{Jahr}} * 0,21 \frac{\text{€}}{\text{kWh}} = 368,172 \frac{\text{€}}{\text{Jahr}}$$

Szenario 2: Thin Clients (2)

- Stromkosten für 1000 Computerarbeitsplätze (ohne Server)

$$1000 * 220,9032 \frac{\text{€}}{\text{Jahr}} = 220903,2 \frac{\text{€}}{\text{Jahr}}$$

- Stromkosten für 25 Server-Blades

$$25 * 368,172 \frac{\text{€}}{\text{Jahr}} = 9204,3 \frac{\text{€}}{\text{Jahr}}$$

- Stromkosten für Computerarbeitsplätze und Server-Blades

$$220903,2 \frac{\text{€}}{\text{Jahr}} = +9204,3 \frac{\text{€}}{\text{Jahr}} = 230107,5 \frac{\text{€}}{\text{Jahr}}$$

Fazit zur Virtualisierung

- Virtualisierung bietet ein großes Einsparpotential, eröffnet aber auch neue Angriffspunkte, insbesondere auf der Ebene des Hypervisors
 - VMware vertreibt die schlanke Virtualisierungs-Lösung ESXi, bei der ein nur 32 MB großer Virtualisierungs-Kernel die Virtualisierungsfunktionen mit minimalem Betriebssystem direkt auf der Hardware realisiert
- Virtualisierung wird in den nächsten Jahren auch wegen der besseren Energieeffizienz und unkomplizierten Nutzung von Mehrkernprozessoren eine wachsende Rolle spielen
 - Laut Gartner zählt Virtualisierung zu den wichtigsten Technologien bis zum Jahr 2010
 - Desktop-Virtualisierung (inkl. Thin Clients) wird immer wichtiger
- Hardwarevirtualisierung ist durch die neuen Prozessorgenerationen stark im Kommen
- Hardware-Emulation \neq Virtualisierung
 - Ziel der Emulation ist die Nachbildung einer anderen Hardwarearchitektur

Werbung

- Erstes und einziges Buch zum Cloud Computing in deutscher Sprache
- Nur €14,95
- ISBN: 3-642-01593-X



Danke für Ihre Aufmerksamkeit!

Fragen?