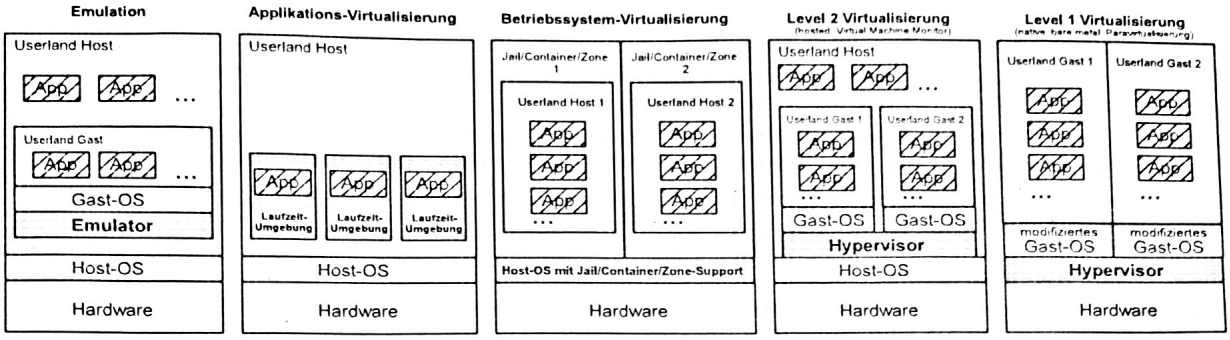
Virtualisierungskonzepte



Emulation

Bei der Emulation wird die komplette Hardware eines Rechnersystems nachgebildet, um so einem Betriebssystem, das für eine andere Hardwarearchitektur ausgelegt ist, den Betrieb zu ermöglichen. Emulatoren haben eine geringe Performance, z.B. Qemu, PearPC, DOSBox.

Applikations-Virtualisierung
bzw.
Anwendungs-Virtualisierung

Bei der Applikationsvirtualisierung werden Anwendungen lokal in einer sog. **Laufzeit-Umgebung** (=Virtuelle Maschine) ausgeführt, die nur die Komponenten bereitstellt, die die Anwendung benötigt. Diese virtuelle Umgebung wirkt dabei wie ein Puffer zwischen der Anwendung und dem Betriebssystem und verhindert Konflikte mit anderen Anwendungen oder dem Betriebssystem. Nachteil: Geringere Ausführungsgeschwindigkeit gegenüber nativer Programmausführung, z.B. Android-Apps laufen jeweils in einer eigenen VM!
Weitere Beispiele:Java VM, ThinApp, Sandboxie, App-V.

Betriebssystem-Virtualisierung

ermöglicht mehrere voneinander isolierte Ausführungsumgebungen (Container/Zones/Jails) in **einem einzelnen Betriebssystem-Kernel**.
Die Betriebssystem-Virtualisierung verfügt über die bestmögliche, fast native Performance.
Das Ausführen unterschiedlicher Kernel bzw. Betriebssysteme ist nicht möglich!
Z.B. FreeBSD Jail, Solaris Zones/Containers, Linux-VServer, OpenVZ, Virtuozzo und Docker

Level 2 Virtualisierung
(hosted,
Virtual Machine Monitor)

Hardware wird virtualisiert!

Bei der Level 2 Virtualisierung werden die Hardwareressourcen des Rechners von einem **Hypervisor** (Virtual Machine Monitor, VMM) durch Hardware-Emulation und/oder Hardware-Virtualisierung intelligent an die virtuellen Maschinen verteilt.
Jedem Gast-Betriebssystem steht so ein eigener virtuellen Rechner mit CPU, Hauptspeicher, Laufwerken, Netzwerkkarten usw. zur Verfügung.
Vorteile: Kaum Änderungen an Host- und Gast-Betriebssystemen erforderlich, Zugriff auf die wichtigsten Ressourcen wird nur durchgereicht. Dadurch fast native Verarbeitungsgeschwindigkeit der Gast-Betriebssysteme, Hohe Flexibilität. Jedes Gast-Betriebssystem hat seinen eigenen Kernel, z.B. VMware Workstation/Fusion, VirtualBox, Virtual PC für x86, Parallels Desktop/Workstation, Linux Kernel Virtual Machine (KVM).

Level 1 Virtualisierung
(native,
bare metal,
Paravirtualisierung)

Bei der Level 1 Virtualisierung wird keine Hardware virtualisiert oder emuliert, sondern die virtuell gestarteten Betriebssysteme verwenden einen Hypervisor, der eine Softwareschnittstelle (API, z.B. Virtual Machine Interface - VMI, VMBus) bereitstellt, um auf gemeinsame Ressourcen (RAM, Platte, Netz usw.) zuzugreifen.
Der **Betriebssystem-Kern muss** an diese Softwareschnittstelle **angepasst sein** um auf der VM ausgeführt werden zu können (**Paravirtualisierung**). Die Anpassung vereinfacht den Aufbau der VM und ermöglicht den darauf ausgeführten VMs eine **höhere Leistung** als beim VMM, z.B. Xen, VMware ESX, MS Hyper-V, Citrix XenServer.

Fragen zur Virtualisierung

1.) Erläutern Sie Gründe, die zum verstärkten Einsatz von Virtualisierungsprodukten führten.
2.) Was hat Green-IT mit Virtualisierung zu tun?
3.) Welche Nachteile ergeben sich bei der Virtualisierung von Rechensystemen?
4.) Welche Sicherheitsproblematik sehen Sie beim Cloud-Computing?
5.) Erläutern Sie die wesentlichsten Unterschiede zwischen Level 1 und Level 2 Virtualisierung.
6.) Was haben User-Mode und Kernel-Mode mit den CPU-Ringen zu tun?

Links

Wikipedia Artikel zur Virtualisierung	http://de.wikipedia.org/wiki/Virtualisierung_(Informatik)
Vorlesungsfolien von Christian Baun	http://baun-vorlesungen.appspot.com/BTS16/Skript/bts_SS2016_vorlesung_11_de.pdf

IT-Systeme IT12-1: Virtualisierung (nicht nur) von Betriebssystemen

Was ist Virtualisierung? nach: http://de.wikipedia.org/wiki/Virtualisierung_(Informatik) (2010)

Ein sehr offener Definitionsversuch: Virtualisierung bezeichnet Methoden, die es erlauben, Ressourcen eines Computers zusammenzufassen oder aufzuteilen.

Eine logische Schicht (=Virtualisierungssoftware) wird zwischen Anwender und Ressource eingefügt, um die physischen Gegebenheiten der Hardware zu verstecken.

Dabei wird jedem Anwender vorgemacht, dass er der alleinige Nutzer einer Ressource sei bzw. es werden mehrere, auch heterogene Hardwareressourcen zu einer homogenen Umgebung zusammengefügt.

Die für den Anwender unsichtbare bzw. transparente Verwaltung der Ressource ist dabei in der Regel die Aufgabe des Betriebssystems.

Was wird virtualisiert?

System bzw. Hardware	HW-Partitionierung IBM LPAR, Sun Logical Domains LDOMs, Hypervisor, Paravirtualisierung, Virtual Maschine Monitor (VMM), Emulation
Prozessor	Intel VT (Intel Virtualization Technology, Codename Vanderpool) AMD-V (AMD Virtualization, Codename Pacifica)
Betriebssystem	OS Container, Zoning, Jails
Anwendung	(auch Applikations-Virtualisierung genannt): Sandbox, Portable Anwendungen
Desktop	Terminaldienste
Rechenzentrum	Cloud-Computing

die nachfolgend genannten Ressourcen werden häufig auch virtualisiert:

RAM	Virtual Memory Management ab 80386 CPU, Auslagerungsdatei
Festplatte	Partition, Datei, Verzeichnis, Win7 Bibliothek
Netzwerk	VLAN, VPN, Software Defined Networking
Storage	RAID, Volume Manager, Speichervirtualisierung (SAN, NAS)

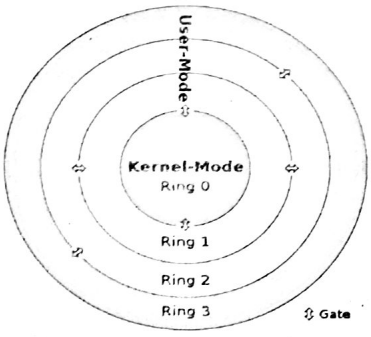
Gründe für die Virtualisierung von Rechensystemen

- Bessere Ausnutzung der Hardware (Ressourcen einsparen, Kosten senken)
- Vereinfachte Administration
- Vereinfachte Bereitstellung (z.B. von Servern)
- Maximale Flexibilität (einfaches Kopieren von virtuellen Maschinen, erstellen von Snapshots)
- Höhere Sicherheit durch Isolation vom Host-System
- Optimierung von Software-Tests und Software-Entwicklung
- Unterstützung alter Systeme und Anwendungen (*legacy*, engl. Erbe, Altlast)

Nachteile und Grenzen virtueller Maschinen

- VMs bieten eine geringere Performance als reale Maschinen
- Nicht jede reale Hardware kann virtualisiert werden (z.B. Hardwaredongles)
- Virtualisierung kann bei der Serverkonsolidierung ein *Single Point of Failure* sein (Host↓ --> viele VMs↓)
- Virtualisierung ist komplex

User-Mode und Kernel-Mode nach: http://de.wikipedia.org/wiki/Kernel-Mode



Schema der Ringe bei CPUs ab 80286

Der *Ring* bezeichnet im Umfeld der Betriebssystem-Programmierung eine Privilegierungs- bzw. Sicherheitsstufe eines Prozesses. Diese schränkt den Prozess in dem auf der CPU nutzbaren Befehlssatz und den adressierbaren Speicherbereich ein. Die Nutzung von Privilegierungsstufen ist sinnvoll, um Prozesse voneinander abzuschotten.

Die verbreiteten x86-Betriebssysteme (u.a. Linux und Windows) nutzen lediglich 2 der 4 möglichen CPU-Ringe der x86-Prozessoren. Im Ring 0 werden der Betriebssystemkern und alle Hardwaretreiber ausgeführt, während die Anwendungssoftware im unprivilegierten Ring 3, dem sog. *Userland*, arbeitet.

Durch Einführung der Prozessor-Virtualisierungstechniken Intel VT und AMD-V verwenden Virtualisierungslösungen verstärkt auch Ring 1.

Bei der **Level 1 Virtualisierung** wird der Betriebssystemkern aus Ring 0 nach Ring 1 verschoben. Die Virtualisierungsschicht (**Hypervisor**) residiert dann darunter in Ring 0 und verwaltet von dort aus einen oder mehrere in Ring 1 laufende Betriebssystemkerne.