

Verteilte Softwaresysteme

Virtualisierungstechniken

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik
Hochschule München

Letzte Änderung: 17.12.2018 20:12

Inhaltsverzeichnis

Virtualisierung	1
Vorteile der Virtualisierung	2
Virtualisierungstechniken	2
Betriebssystemvirtualisierung	3
Allgemeiner Ansatz	3
Vollvirtualisierung	3
Containervirtualisierung	4
Paravirtualisierung	4
Softwarevirtualisierung	4
Services	4
Anwendungen	5
Hardware-Virtualisierung	5
Partitionierung	5

Virtualisierung

- Ressourcen so in einer logischen Schicht zusammenfassen, dass
 - die Auslastung optimiert wird und
 - sie Anforderungen automatisch zur Verfügung stehen

- Verknüpfen von Servern, Speichern und Netzen zu einem virtuellen Gesamtsystem
- wir unterscheiden zwischen
 - **Hardwarevirtualisierung** und
 - **Softwarevirtualisierung**

Vorteile der Virtualisierung

- **Serverkonsolidierung**
 - viele virtuelle Server auf wenigen physikalischen
 - Kostensenkung bei Hardware und für den Betrieb (Strom, Klima)
- **vereinfachte Administration**
 - Erstellung und Anpassung virtueller Server kann automatisiert werden
- **vereinfachte Bereitstellung**
 - ein neuer Server innerhalb von Minuten
- **hohe Verfügbarkeit der Systeme und Services**
 - Migration durch einfaches Clonen virtueller Maschinen
 - durch Replikation der physikalischen Ressourcen unterbrechungsfreier Betrieb auch bei Wartung
- **Service-Levels**
 - einfacher Garantien vereinbar
- **Optimierung von Software-Tests und Software-Entwicklung**
 - unterschiedliche Umgebungen sind leicht aufzusetzen
- **Unterstützung von alten Anwendungen**
 - Legacy-Anwendungen, die auf aktueller Hardware nicht mehr laufen würden
- **höhere Sicherheit**
 - hoch abgesicherte virtuelle Umgebung

Virtualisierungstechniken

1. Betriebssystemvirtualisierung

- Ausführung mehrere Betriebssysteme auf einem Rechner

2. Virtuelle Maschinen wie die JVM

- Ausführung von Programmen auf verschiedenen Betriebssystemen
- 3. **Softwarevirtualisierung**
- lokale Ausführung von Programmen ohne vorherige Installation
- 4. **Hardwarevirtualisierung**
- Verwaltung von heterogenen Hardware-Ressourcen

Betriebssystemvirtualisierung

Allgemeiner Ansatz

- auf einer Hardware mehrere Betriebssysteme
- Gast-System ist hierbei stets für die gleiche CPU-Architektur ausgelegt
 - sonst spricht man von Emulation
- stammt aus dem Umfeld von Großrechnern (seit 1960er Jahre)
- Virtueller Maschinen Monitor (VMM) oder Hypervisor stellt jedem Benutzer einen spezifischen Anteil an Ressourcen zur Verfügung
 - als Kopie der unterliegenden Hardware als Virtuelle Maschine
- erstes und bekanntestes kommerziell eingesetztes BS: VM/370 von IBM
 - darauf dann MVS (Multiple Virtual Storage) oder AIX (Unix)
 - aktuelle Version z/VM unterstützt auch Linux-Instanzen

Vollvirtualisierung

- gesamte Hardware über spezielle Treiber
- unverändertes Gastsystem kann installiert werden
- Beispiele:
 - VirtualBox oder VMware Workstation
 - * als Anwendungsprogramme auf einem Wirtsbetriebssystem
 - VMware ESX-Server mit eigenem Betriebssystemkernel
 - * läuft direkt auf der Hardware
 - **KVM** (Kernel-based Virtual Machine)
 - * basiert auf dem Linux-Kernel
 - * FK07 betreibt ein KVM-Cluster (ZPA, ob.cs.hm.edu, ...)

Containervirtualisierung

- Anwendungen wird eine komplette Laufzeitumgebung innerhalb eines geschlossenen isolierten Bereichs zur Verfügung gestellt
- gemeinsame Nutzung eines OS-Kernels
- der Hypervisor startet kein zusätzliches Betriebssystem, sondern die Container bilden die benötigten Teilfunktionalitäten ab
- Nachteil: Ausschließlich die Plattform des Wirts wird unterstützt
- Beispiele:
 - [FreeBSD Jails](#)
 - [Linux-VServer](#)
 - [Solaris Container](#)
 - [Docker](#)

Paravirtualisierung

- auf einem Basis-Betriebssystem werden zusätzliche Betriebssysteme gestartet
 - ohne HW-Virtualisierung oder -Emulation
- die virtuellen Betriebssysteme benötigen eine abstrakte Verwaltungsschicht um auf gemeinsame Ressourcen zuzugreifen
- dazu i.d.R. Anpassungen des Gastsystems notwendig
- Vorteil: virtuelle Systeme genauso performant wie die physikalischen
- Beispiel
 - Denali
 - [Xen Hypervisor](#)
 - * läuft direkt auf der Hardware
 - * von University Cambridge, heute von Citrix
 - * als Gastsysteme im Wesentlichen Linux

Softwarevirtualisierung

Services

- Serveranwendungen über ein Netzwerk so als ob sie lokal laufen würden
- auf dem Client meist Thin Clients
- Terminal-Services
 - z.B. [Citrix XenApp](#)
 - als Client z.B. Raspberry Pi mit Tastatur, Maus und Bildschirm
- auch schon “uralt”: X11

Anwendungen

- Desktop- oder Serveranwendungen werden lokal ausgeführt ohne vorher installiert worden zu sein
- **virtualisierte Anwendung**
 - virtuelle Umgebung mit allen Dateien/Komponenten zur Ausführung
- Beispiel
 - [Microsoft Application Virtualization \(App-V\)](#)
 - Software (z.B. Office) wird bei Bedarf zum Rechner des Anwenders übertragen, sobald dieser die Anwendung aufruft

Hardware-Virtualisierung

Partitionierung

- die Hardware-Virtualisierung verwaltet Ressourcen (CPU, Speicher) über die Firmware und teilt diese einer virtuellen Maschine zu
- **Partitionierung** unterteilt das Gesamtsystem dynamisch in Teilsysteme
- Beispiele
 - IBM zSerie (Mainframe)
 - IBM pSerie (Midrange)
 - ohne Neustart im laufenden Betrieb Ressourcenzuteilung nahezu beliebig veränderbar
 - auf einem zSerie problemlos mehrere hundert bis tausend Linux-Instanzen