Freie Universität Berlin

Proseminar Technische Informatik
A survey of virtualization technologies

Referent: Martin Weigelt

Übersicht

Freie Universität Berlin

1. Definition
2. Gründe für Virtualisierung
3. Übersicht über einige Virtualisierungsmethoden
4. Virtualisierungsmethoden kurz vorgestellt
5. Konkrete Implementierung von Xen
6. Zusammenfassung

Was ist Virtualisierung?

Freie Universität Berlin

- **Virtualisierung:**

  Technologie, die Computerressourcen kombiniert und teilt, um eine oder mehreren Betriebsumgebungen zu erzeugen, die Methoden wie Hardware und Software Partitionierung und Aggregation, teilweise oder komplette Maschinensimulation, Emulation u.a. verwenden.

  (Susanta Nanda, Tzi-cker Chiueh, “A Survey on Virtualization Technologies”)

Wozu Virtualisierung?

Freie Universität Berlin

Virtualisierung kann…

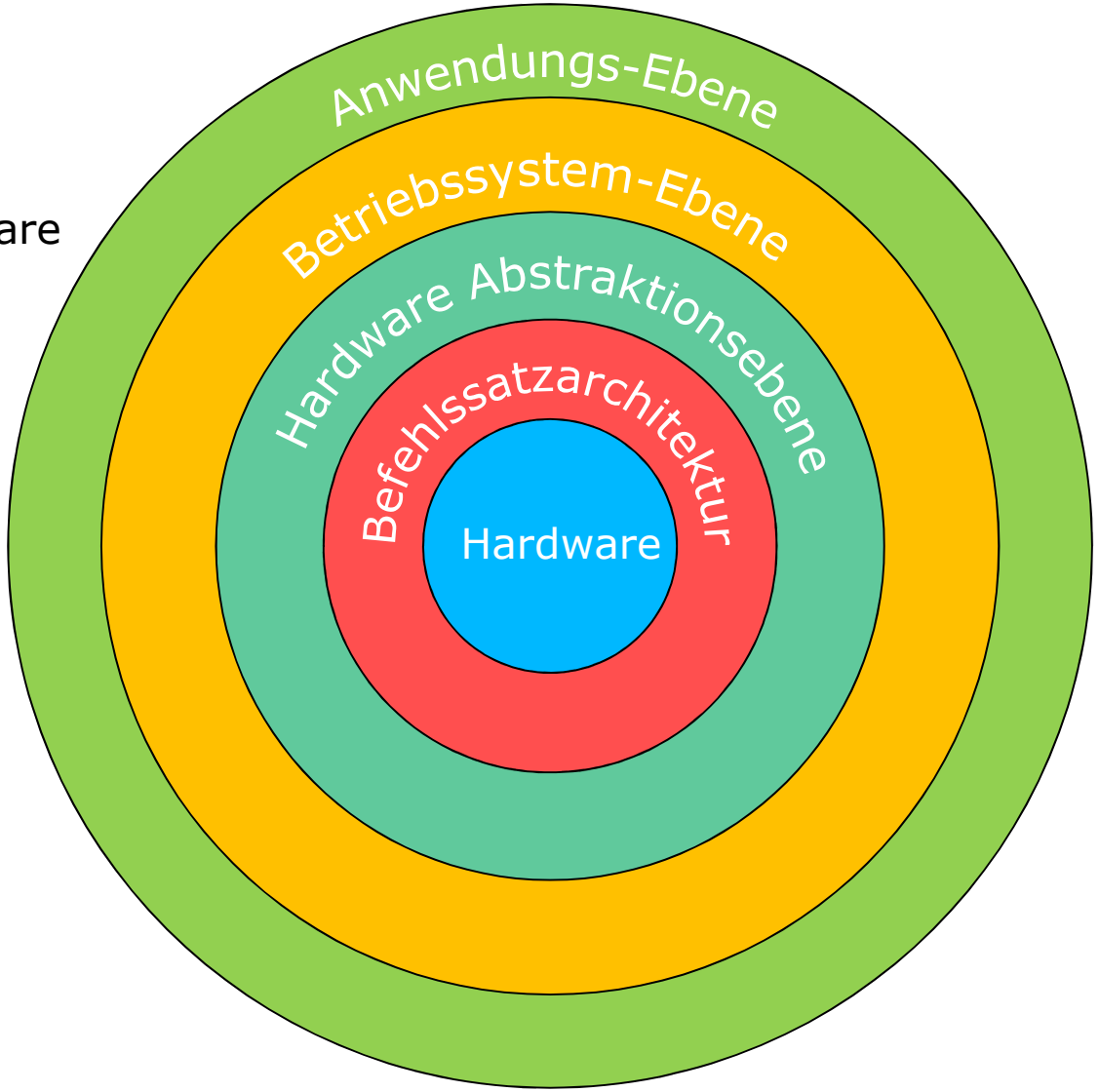
- die Auslastung von Servern verbessern.
- eine sichere und isolierte Umgebung für Programme bereitstellen.
- Hardware zur Verfügung stellen, die man nicht besitzt.
- es ermöglichen mehrere Betriebssysteme parallel zu auszuführen.
- verwendet werden um Testumgebungen für bestimmte Szenarien zu erstellen.
- Und viele weitere

Die Welt der Virtualisierung

Freie Universität Berlin

Virtualisierung auf der:

- Befehlssatzarchitektur:
  Befehle werden durch Software ersetzt
- Hardware-Abstraktionsebene:
  VMM verwaltet System-ressourcen
- Betriebssystem-Ebene:
  Eine neue virtuelle Instanz des Betriebssystems wird erzeugt
- Anwendungsebene:
  Anwendungen werden in virtuellen Kontext gesetzt.



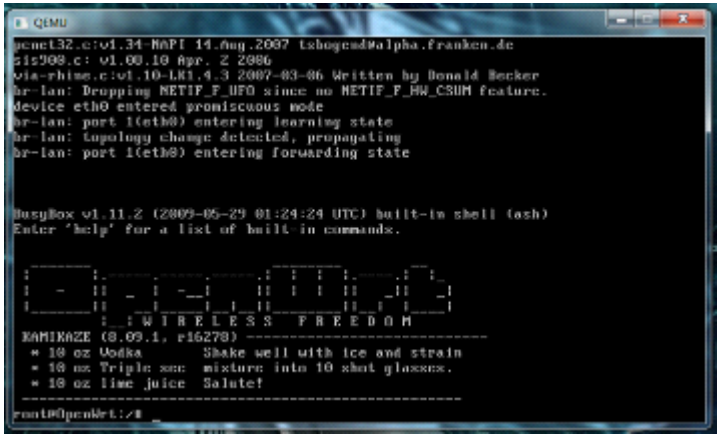
Virtualisierung auf der Befehlssatzarchitektur

Freie Universität Berlin

- Ziel:
  Software inkompatible Systeme auf nativer Hardware ausführen.

- Umsetzung:
  Hardwarebefehle des Gastsystems werden in Software umgesetzt und auf der nativen Hardware ausgeführt.

- Probleme:
  Die Performance der Gastsysteme ist viel schlechter als auf der nativen Hardware.



OpenWRT in QEMU

Virtualisierung auf der Hardware-Abstraktionsebene

Freie Universität Berlin

- Ziel:

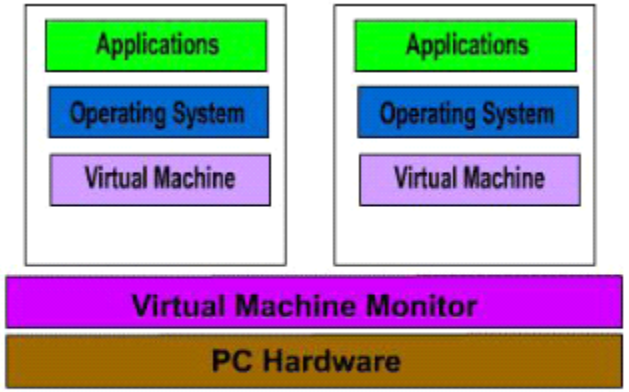
  Effiziente Virtualisierung auf einer Systemarchitektur.

- Umsetzung:

  Einführung eines „Virtual Machine Monitor"(VMM) auf der Hardwareebene zur Verwaltung und Erzeugung von Virtuellen Maschinen.

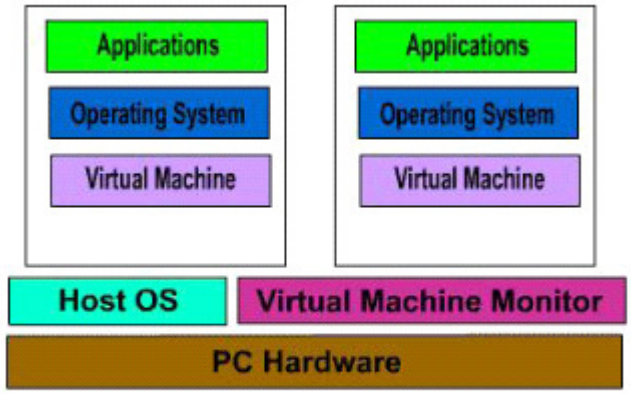
- Probleme:

  Behandlung privilegierter Befehle, Fehlerbehandlung, Optimierung der Effizienz



Stand Alone Virtual Machine

Quelle: Susanta Nanda, Tzi-cker Chiueh, "A Survey on Virtualization Technologies"



Hosted Virtual Machine

Quelle: Susanta Nanda, Tzi-cker Chiueh, "A Survey on Virtualization Technologies"

Virtualisierung auf der Betriebssystemebene

Freie Universität Berlin

- Ziel:

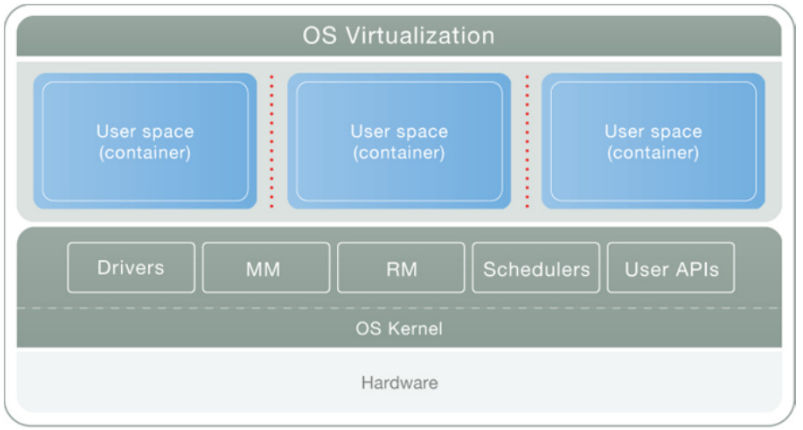
  Sichere und isolierte virtuelle Umgebung für Software unter Nutzung des vorhandenen Betriebssystems zur Virtualisierung.

- Umsetzung:

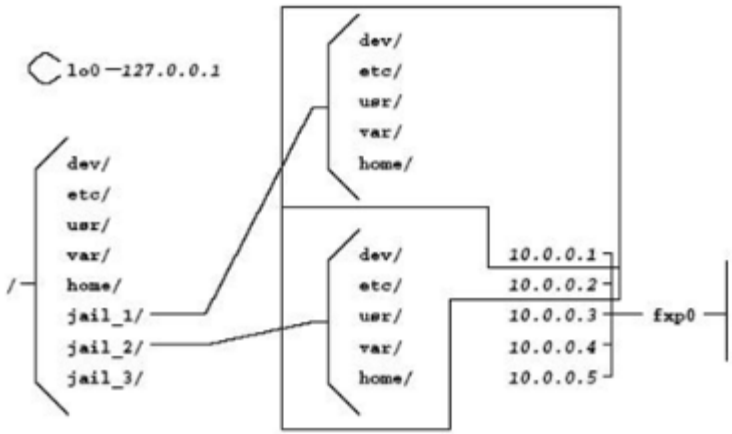
  Virtualisierung im Stack-Speicher, d.h. Virtuelle Maschinen teilen sich das Betriebssystem und die Hardware.

- Probleme:

  Beschränkt auf das vorhandene Betriebssystem.



Quelle: http://i3.parallels.com/r/upload/pvc45_virtualizing_big.png



Quelle: Susanta Nanda, Tzi-cker Chiueh, "A Survey on Virtualization Technologies"

Virtualisierung auf der Programmiersprachen-Ebene

Freie Universität Berlin

- Ziel:

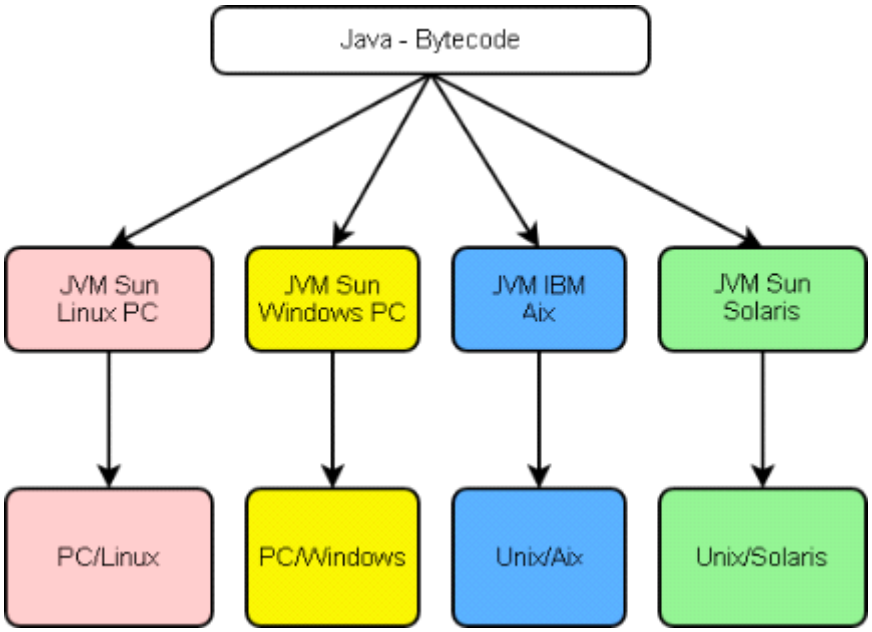
  Plattformunabhängige Programmiersprache

- Umsetzung:

  Virtuelle Maschine für Programme dieser Sprache.

- Probleme:

  Behandlung von unbefugten Speicherzugriffen



Quelle: http://upload.wikimedia.org/wikipedia/commons/0/0d/Java-jvm.png

Fallbeispiel: XEN

Freie Universität Berlin

- Xen ist eine Open-Source VMM für die X86-Architektur auf der Hardware-Abstraktionsebene

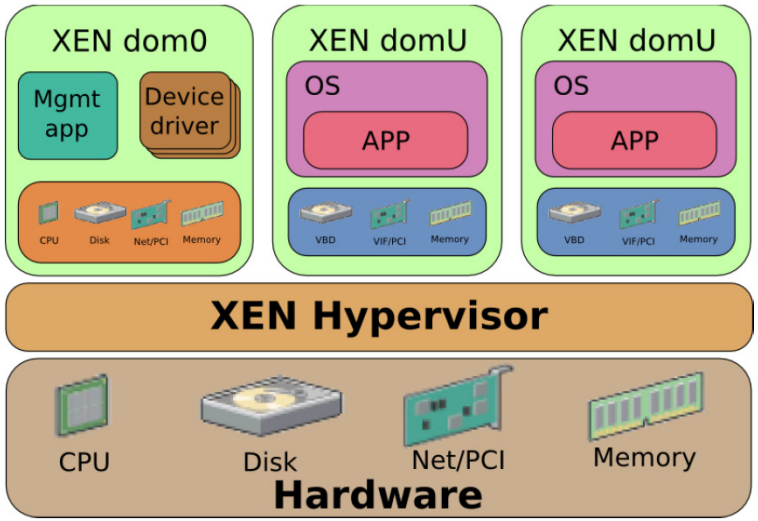
Quelle: http://www.xen.org

- Entwicklung 2001-2002 an der Universität Cambridge

- Die Zusammenarbeit des Gastsystems mit der VMM wird optimiert

Aufbau von XEN

- Die Gastsysteme müssen modifiziert werden (Paravirtualisierungskonzept)
- Gerätetreiber werden aus Sicherheitsgründen in einer extra Domain(dom0) ausgeführt
- VM ruft die API der VMM direkt auf, d.h. es findet kein Kontextwechsel statt



Quelle: http://upload.wikimedia.org/wikipedia/commons/5/5b/XEN-schema.png

Fallbeispiel: QEMU

Freie Universität Berlin

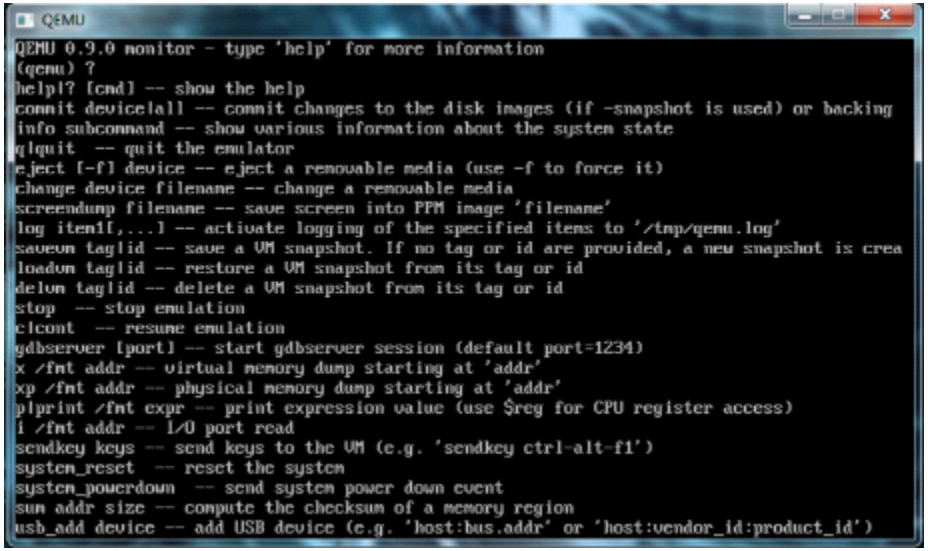
- Schneller Prozessor-Emulator mit dynamischen Übersetzungsmethoden (dynamic translation)

- 2 Betriebsarten:
  1. Linux User-Mode Emulation
  2. Komplette System Emulation

- Unterstützte Prozessor-Architekturen bei der Hardware-Emulation: Sparc32/64, MIPS, ARM, ColdFire, PowerPc, x86,…

QEMU
open source processor emulator

http://www.cagataycebi.com/free_articles/xen/img/qemu_logo.png



Qemu Monitor

QEMU

Freie Universität Berlin

- Die User-Mode Emulation kann für CPU übergreifende Ausführung von Prozessen, Kompilieren oder Debugging verwendet werden.
- Dynamic Translation: Gastsystem-Befehle werden zur Laufzeit interpretiert
- Zerlegung der Gastsystem-Befehle in Mikro-Operationen zur Steigerung der Portabilität
- Der Performanceverlust wird versucht durch Caches und durch Tricks, wie Zusammenfassung von Mikro-Operationen zu Funktionen zu minimieren.

Zusammenfassung

Freie Universität Berlin

Virtualisierung…

- optimiert die Effizienz von Systemen
- findet heute in unterschiedlichen Bereichen Anwendung z.B. Serverkonsoldierung, Sandboxing, usw.
- senkt die Hardwarekosten
- steigert die Portabilität von Software
- erhöht die Ausfallsicherheit bei Softwarefehlern oder Kompromittierungen
- macht Spaß

Vielen Dank!

Fragen?