

# Virtualisierung – ein Überblick

Frank Hofmann

Potsdam

18. April 2007

## Fragen, die sich jeder stellt

- Virtualisierung – was ist das eigentlich?
- Brauche ich das?
- Was kann ich denn damit machen?
- Worauf muss ich denn achten?

## Ziele

- Begriffsklärung
- Technische Abläufe
- Überblick über existierende Softwarelösungen
- Risiken und Sicherheit

- 1 Virtualisierung – ein Einstieg
- 2 Softwarelösungen
- 3 Anwendung und Nutzung
- 4 Referenzen

# Virtualisierung – eine Begriffsklärung (1)

„Virtualisierung ist eine Herangehensweise in der IT, die Ressourcen so zusammenfaßt und verteilt, dass ihre Auslastung optimiert wird und automatisch Anforderungen zur Verfügung steht.“

# Virtualisierung – eine Begriffsklärung (2)

- physikalische Hardwareressourcen eines Computersystems werden auf eine oder mehrere virtuelle Umgebungen verteilt
- Abstraktionsebene zwischen Hardware und Betriebssystem, um mehrere voneinander getrennte Betriebssystemumgebungen auf einer Hardware parallel zu betreiben
- Ziele:
  - ① vorhandene Ressourcen optimal nutzen
  - ② reale Umgebung zur Verfügung stellen

# Geschichte und Entwicklung (1)

Mitte der 60er Jahre



VM/370

- mehreren Benutzern eigenständige Umgebungen bereitstellen
- heißt heute: z/VM, IBM zSeries

Hardwarevirtualisierung auf Großrechnern durch Logical Partition (LPAR)

- pro zSeries-Rechner 30 LPAR möglich
- beliebig viele virtuelle Maschinen pro z/VM

# Geschichte und Entwicklung (2)

heute:

Rechenleistung wächst, wird aber nicht ausgenutzt

- Zusammenfassung von Diensten pro Virtueller Maschine
- Sicherheit und Beeinträchtigung (isolierte Umgebung)
- Hochverfügbarkeit („Schnappschüsse“)
- Verringerung der Komplexität, Administrierbarkeit



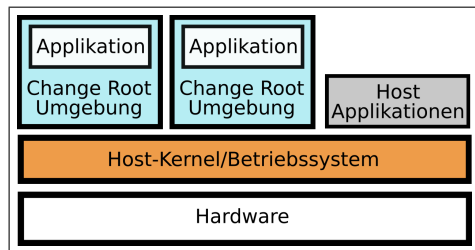
# Aufbau eines Virtualisierungssystems

- Host
  - Verwaltungsinstanz für die physikalische Hardware
  - stellt Ressourcen zur Verfügung
- Gäste
  - virtuelle Maschine
  - Laufzeitumgebung, in der separate Prozesse laufen

# Virtualisierungsansätze

- Kapselung von Prozessen
- Virtualisierung auf Betriebssystemebene
- Virtualisierung durch Emulation
- Paravirtualisierung
- Hardwarevirtualisierung

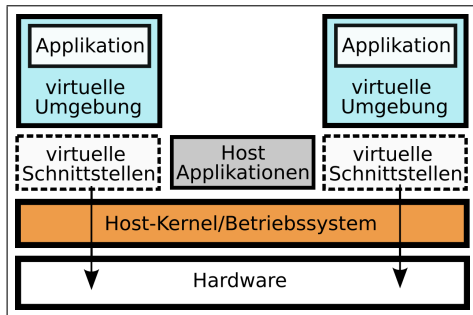
# Kapselung von Prozessen



Change-Root-Umgebung

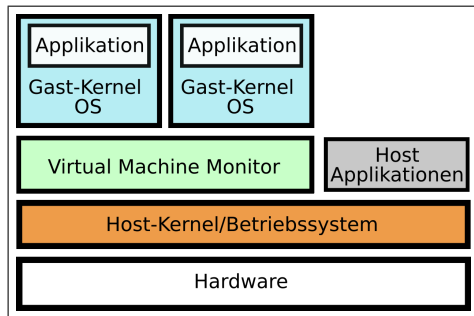
- Unterverzeichnis wird zum root-Verzeichnis für einen Prozess
- Prozess und Kindprozess werden nur in diesem Verzeichnisweig ausgeführt („einsperren“)
- aber: keine Lastverteilung möglich

# Virtualisierung auf Betriebssystemebene



- eigene Umgebung
- Kernel verteilt die Last
- Programme: Open VZ, Linux-VServer

# Virtualisierung durch Emulation



Programme: VMWare, Microsoft Virtual PC, Microsoft Virtual Server

- Teile eines Systems oder auch das gesamte System wird nachgebildet
- darunterliegende Hardware des Hostsystems muß diese Komponenten nicht besitzen
- Zwischenschicht vermittelt zwischen VM und Hostsystem

# Paravirtualisierung

- ähnlich einer Emulation
- keine Emulation der Host-Hardware
- Hardware entsprechend dem Hostsystem, nur ohne direkten Zugriff
- nur anpassbare Systeme können emuliert werden
- Projekte: XEN, User Mode Linux

# Hardwarevirtualisierung (1)

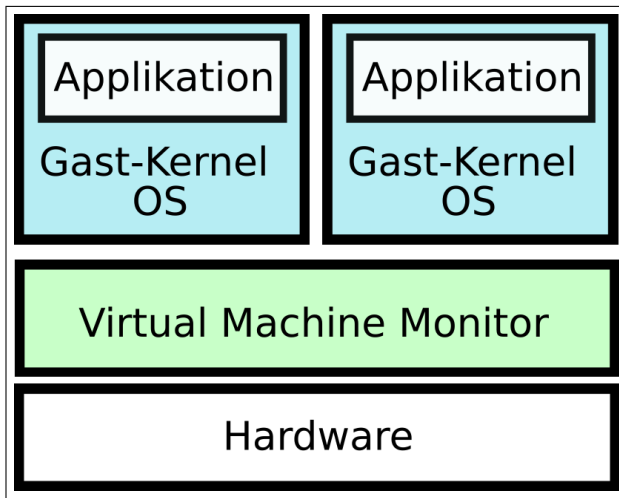


- Intel  
Intel Virtualization Technology (Intel VT)  
Modus: Virtual Machine Execution
- AMD  
Secure Virtual Machine (SVM)

Ziel:

- Management der Gäste
- mehrere Gäste teilen sich eine Plattform

# Hardwarevirtualisierung (2)





# VM Ware Server, VM Ware Player

<http://www.vmware.com/de/>



- kommerzielles Produkt (VM Ware Server), kostenlose Version mit eingeschränktem Funktionsumfang (VM Ware Player)
- Host: Windows, Linux
- Gast: Windows, Linux, \*BSD
- simuliert einen Standard-PC

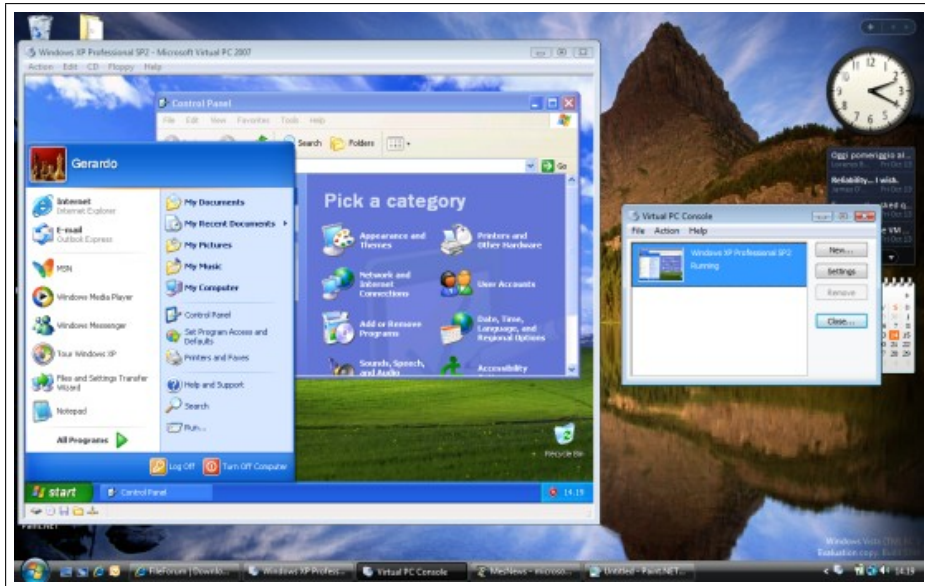
# Microsoft Virtual PC (1)

<http://www.microsoft.com/germany/mac/virtualpc>



- kostenlos (Windows) oder kostenpflichtig (Mac)
- Betriebssystem: Microsoft Windows, Apple Mac OS X
- simuliert u.a.:
  - Standard-PC mit dem Host-Prozessor (Mac: Pentium II)
  - bis zu drei Festplatten
  - ein CD- oder DVD-Laufwerk
  - RAM mit einstellbarer Größe
- keine Unterstützung für PCI-Geräte
- USB: nur Macintosh-Version

# Microsoft Virtual PC (2)



# Qemu

<http://fabrice.bellard.free.fr/qemu/>

<http://www.qemu.org>



- freier Emulator
- Unterstützung für x86, SPARC, ARM, PPC, MIPS u.a.
- Geschwindigkeit: angenehm flott

# Qemu (2)



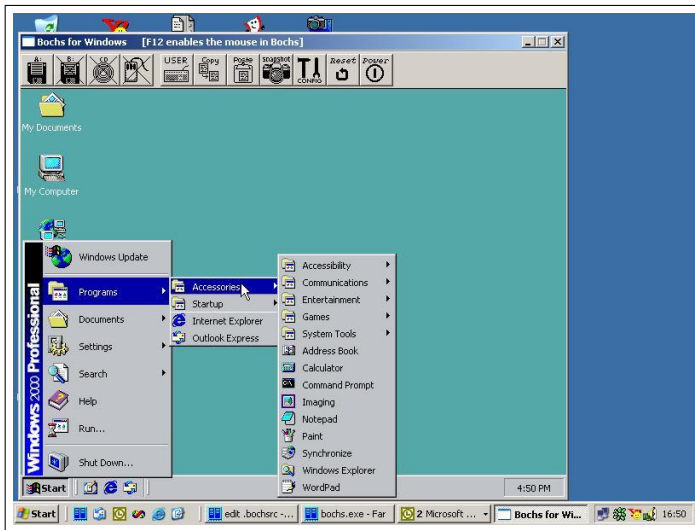
# Bochs (1)

<http://bochs.sourceforge.net>



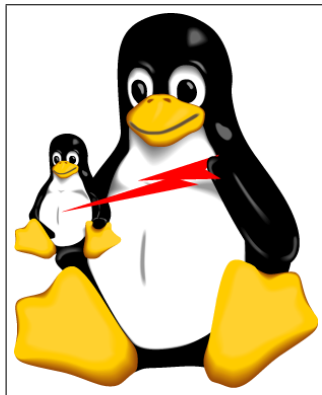
- freier x86-Emulator
- unterliegt den Bedingungen der LGPL
- Ziel der Entwicklung: vollständige PC-Kompatibilität
- verfügbar für Windows, Linux, BSD (FreeBSD, NetBSD, OpenBSD, Mac OS X), BeOS, PlayStation Portable, GP2X

## Bochs (2)



# FAU-Machine (1)

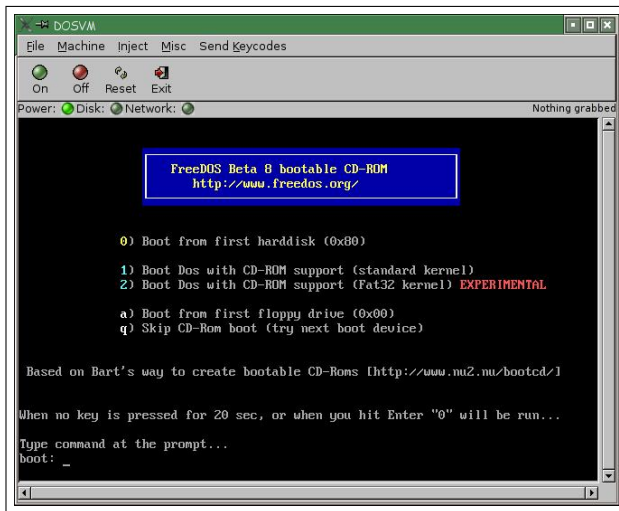
<http://www3.informatik.uni-erlangen.de/Research/FAUmachine/>



- entwickelt an der Friedrich-Alexander-Universität Erlangen-Nürnberg
- läuft als regulärer User-Prozess (nicht privilegiert)
- Möglichkeit für Automatisierungen (Linux-Installation, Tests)
- Netzwerkunterstützung, Konfiguration
- verfügbar für Linux, Portierung auf OpenBSD und Windows ist in Arbeit



# FAU-Machine (2)



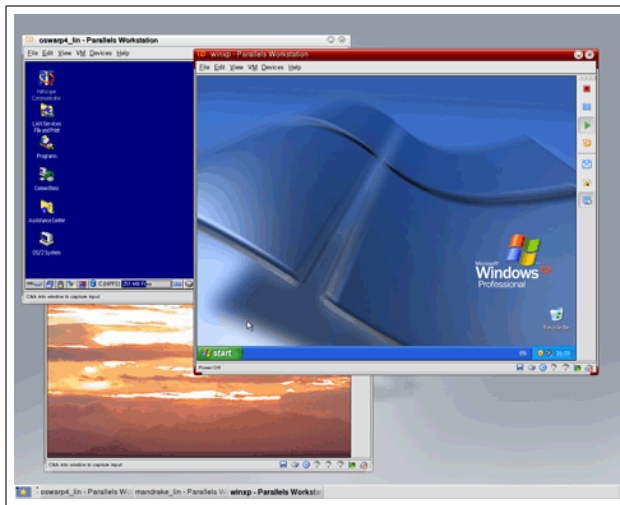
# Parallels PC (1)

<http://www.parallels.com/en/products/workstation/>



- kommerzieller Emulator für 32-Bit-Betriebssysteme für x86-Hardware, freie 15-tage-Testversion
- Host: Windows und Linux
- Gast-Systeme: u.a. Windows, Linux, FreeBSD, OS/2, Solaris/86
- Austausch von Daten zwischen Gast- und Hostsystem über Shared Folders

# Parallels PC (2)



# Virtual Box (1)

<http://www.virtualbox.org>



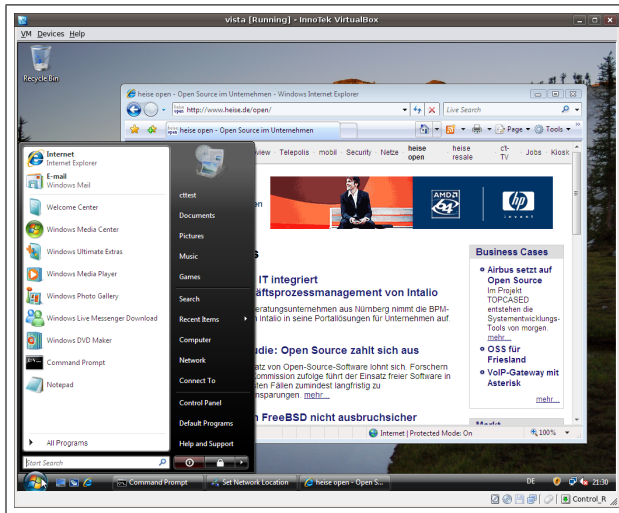
- freier Emulator (GPL) (seit Januar 2007)
- Host: Windows, Linux
- Gäste: Windows (einschließlich Vista), Linux, OS/2, diverse Unix-Varianten, Solaris/x86

Detaillierte Beschreibungen:

<http://www.pro-linux.de/berichte/virtualbox.html>

<http://www.heise.de/open/artikel/83678>

# Virtual Box (2)



# Praktische Anwendung

- Ausführung eines Betriebssystems parallel zum bisherigen System
  - Testen, Neugierde, Ausprobieren
  - Zeitersparnis (keine Zeit für Installation, Bootdauer)
  - Stabilität („Sandbox“)
- Vermeidung von Inkompatibilitäten von Programmversionen
- Installation/Deinstallation von Programmen
  - Paketabhängigkeiten
  - Programm-„reste“ (Schlüssel, Config-Files, Bibliotheken)
- Programm läuft nur auf bestimmtem Betriebssystem
- endlich mal seine Hardware ausreizen

# Empfehlungen zur Nutzung

- Systemaustattung
  - möglichst viiiiiiel RAM (Platz für ein ganzes, weiteres Betriebssystem)
  - Speicherplatz (ab 1G)
- Erwartungen anpassen
  - nicht jede Hardware wird perfekt unterstützt
  - Verzögerung durch Abstraktionsebenen
  - Gastsystem reagiert eventuell anders, da anders konfiguriert

# Links (Auswahl)

- Wikipedia – Virtualisierung  
[http://de.wikipedia.org/wiki/Virtualisierung\\_\(Informatik\)](http://de.wikipedia.org/wiki/Virtualisierung_(Informatik))
- Open VZ  
<http://www.openvz.org>



# The End

Danke für Ihre Aufmerksamkeit :-)

## Kontakt:

Frank Hofmann

Email <frank.hofmann@efho.de>

Hofmann EDV – Linux, Layout und Satz, Potsdam