

# Einführung in die Virtualisierung

Sebastian Andres <sebastian@sebastianandres.de>

14. März 2010

## Überblick

- Virtualisierungstechniken
- Virtualisierung mit xen
- Qemu
- KVM
- Benutzeroberflächen
- Netzwerk
- Links

## Virtualisierungstechniken

- Paravirtualisierung (definierte Schnittstellen, Gastssysteme müssen angepasst werden) - Xen und usermodelinux setzen teilweise diese Techniken ein. Da der Kernel von Windows nicht verändert werden kann, ist es nicht möglich, Windows mit dieser Technik als Gast-System zu virtualisieren (Unter Gast-System wird das System verstanden, welches in einer virtuellen Maschine läuft).
- Full Virtualisierung (qemu, teilweise auch vmware). Der Vorteil: die Gesamte Hardware (inkl. CPU) wird virtualisiert – dadurch lassen sich auch andere Architekturen verwenden. Nachteil: Es ist sehr langsam, da besonders die Emulation der CPU kapazitäten und Leistung bindet.
- Hardware virtualisierung (Direkte Befehle auf die CPU, braucht aber einen Prozessor, welcher dies unterstützt). Flags: vmx (Intel), svm (Amd). Auslesen kann man diese Flags in /proc/cpuinfo. KVM basiert auf dieser Technik.

## Virtualisierung mit Xen

- Braucht modifizierten Kernel für das Gast-System und für das Host-System.
- Nicht fest in den Kernel integriert. Daher sind diejenigen Kernel-Versionen, welche mit Xen (besonders auf dem Host-System) laufen meist relativ alt.
- Wird zukünftig von Distributionen nicht mehr unterstützt – die meisten Distributoren setzen mittlerweile auf KVM, das eine feste Schnittstelle zum Kernel mitbringt.

## Qemu

- Sehr langsam! (Aufgrund dessen, dass qemu nur Full-Virtualisierung verwendet.)
- Unterstützt keine Hardware-Virtualisierung.

## **KVM**

- Fester bestandteil des Kernels
- Unterstützt Hardware-Virtualisierung, und ist daher besonders schnell.

## **Benutzeroberflächen**

- Libvirt: Steuern der Maschinen über eigene Kommandozeile und XML-Files. Die Virtuellen Maschinen lassen sich somit unkompliziert über eine Shell steuern und ermöglicht es sogar, virtuelle Maschinen dynamisch über das Netzwerk zu verschieben.
- Virt-install: Erstellt Virtuelle Maschinen inkl. Konfigurationen. Die so erstellten Maschinen laufen dann ohne weitere Konfiguration mit libvirt.
- Proxmox VE: Ist eine eigene auf Debian basierende Distribution, welche dafür entwickelt wurde, virtuelle Maschinen zu erstellen und zu verwalten. Ist das Grundsystem einmal installiert (Unkompliziert, dauert etwa 20 Minuten), wird die Konfiguration des Netzwerks, die verwaltung von Speichermedien, der Virtuellen Maschinen, etc. über eine Web-Oberfläche vorgenommen.

## **Netzwerk**

- Virtuelle Netzwerkinterfaces (tun, vnet, etc.): Diese Interfaces stehen im Host-System zur verfügung. Über diese virtuellen Schnittstellen findet die Kommunikation mit dem Gast-System statt.
- User Mode Network: Mit dieser Technik stellt KVM (Defaultmäßig, wenn keine netzwerkspezifischen Optionen angegeben sind) dem Gast-System einen Internetzugang zur verfügung. Es wird eine Virtuelle Netzwerkkarte im Host-System angelegt, welche mit einem DHCP-Server, einer Firewall und NAT versehen ist. Um vom Host-System eine Netzwerkverbindung in den Gast aufzubauen, ist es notwendig, eine Portweiterleitung einzurichten! (Mehr dazu in der Manpage)
- Bridging: Hier können mehrere virtuelle Maschinen in einer Bridge gebündelt werden, um auch untereinander kontakt zu haben. Ein kleines Skript zum erstellen einer Bridge liegt diesen Folien bei (Das Skript ist nur ein Beispiel! Bitte nicht ungeprüft übernehmen!!!)

## **Links**

Dieses Dokument: <http://www.sebastianandres.de/vortraege>  
Qemu-Buch: <http://www.qemu-buch.de>  
xen-Anleitung: <http://www.pug.org/index.php/Xen-Installation>