

Xen vs. KVM

Virtualization done right



by Commander1024

Überblick

- Virtualisierung, Xen, KVM: WTF?!?
- Zielsetzung
- Pro/Contra Xen, KVM, Alternativen
- Networking
- Administrationswerkzeuge
- Gastinstallation
- Fazit
- kurzer Hands-on

Ziele

- Konsolidierung verschiedener Systeme (Fokus auf Linux; Möglichkeit, Windows zu betreiben)
- Komplette Abschottung voneinander
- Maximale (priorisierbare) Performance
- Snapshots, Online- / Offlinemigration, Cloning
- Änderung des RAMs im Betrieb (balloning)
- Einfache Administration
- Unabhängigkeit von einer (proprietären) GUI
- ggf. Hardware einem Gast exklusiv zuteilen

Xen

Pro:

- Geschwindigkeit durch Paravirtualisierung

Contra:

- Quirks für Hardware Console nötig
- Zeitgeber unstable (time went backwards)
- Kein Bestandteil des Vanilla Kernels
- Entwickler (Citrix) nachlässig mit Patchpflege

Pro:

- Max. Kompatibilität durch Vollvirtualisierung
- Geschwindigkeit durch Teil-Paravirtualisierung
- optionale APIs zum Host verwendbar (virtio)
- Text-, Grafikausgabe oder Headlessbetrieb
- Teil jedes modernen Linuxes, Teil des Kernels
- Dank libvirt, virt-manager, virsh einfache Administration auf cmd-line und GUI

Alternativen

Kommerziell:

- Vollvirtualisierung:
VMWare, Virtuozzo,
MS Hyper-V,
Parallels, Citrix Xen
Server
- Container:
Virtuozzo

OpenSource:

- Vollvirt:
Virtualbox
- Container:
lguest, OpenVZ,
Linux-VServer

Voraussetzungen

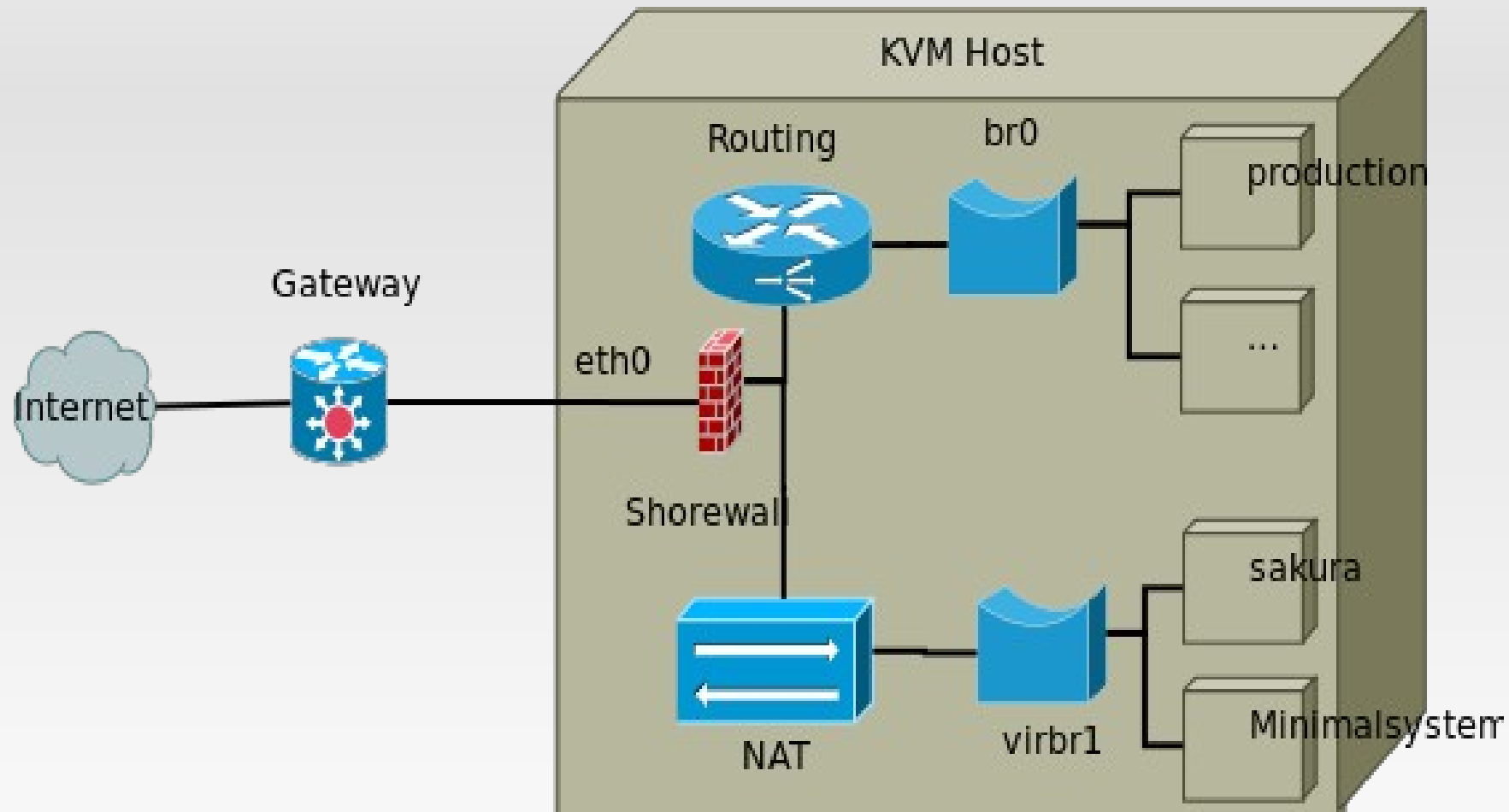
- Minimalinstallation eines Linuxes
- CPU mit Intel VT-x oder AMD Pacifica
- 1 oder mehr NICs
- Storage (lokal, remote, LVM, File-based, iscsi, fiber channel, etc.)
- kvm-(amd/intel)
- libvirt (optional: virsh, virt-top, virt-install)
- optional: GUI auf Workstation, Web-Interface

Vorbereitungen für Networking

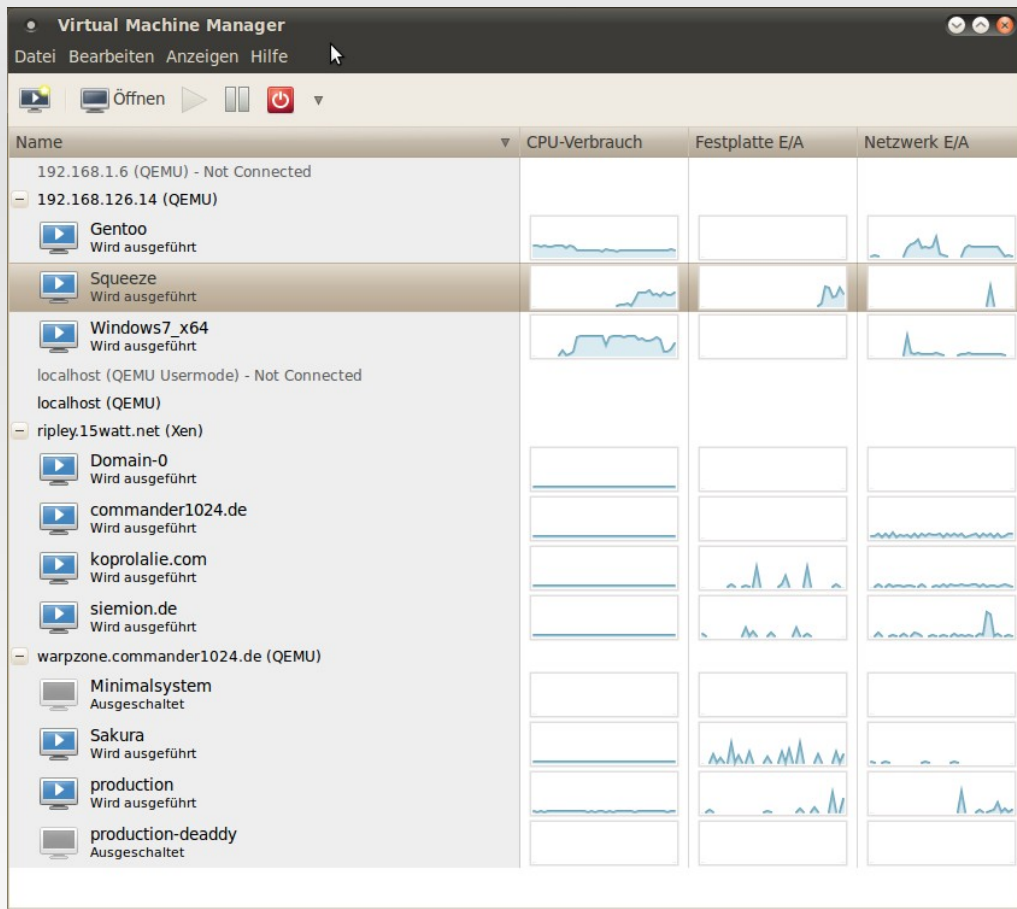
libvirt/kvm nutzt standardmäßig Bridges, um seine Gäste daran anzuhängen:

- Diese können isolated, nat'ed oder routed sein
- Man errichtet seine eigene Infrastruktur,
→ ermöglicht komplexere Setups
- Man weist dem Gast seine eigene NIC zu

One-Box complex Setup



Admin Frontends



```
virt-top 21:11:46 - x86_64 8/8CPU 1600MHz 7998MB
4 domains, 2 active, 2 running, 0 sleeping, 0 paused, 2 inactive D:0 0:0
X:0
CPU: 0,9% Mem: 1152 MB (1152 MB by guests)
```

ID	S	RDRQ	WRRQ	RXBY	TXBY	%CPU	%MEM	TIME	NAME
3	R	1	4	196	150	0,8	1,0	56:24:46	Sakura
4	R	0	0	122	122	0,0	12,0	58:03:72	production (Minimalsystem)
-	-	-	-	-	-	-	-	-	(production-deaddy)

```
warpzone:~# virsh
Welcome to virsh, the virtualization interactive terminal.
```

```
Type: 'help' for help with commands
      'quit' to quit
```

```
virsh # list
Id Name                               State
-----
3 Sakura                               running
```

Viele Frontends für libvirt:

- virt-manager (python GUI)
- virsh (cmd line)
- oVirt (web, fedora)
- Mit mehr ist zu rechnen!

Installation eines Gastes

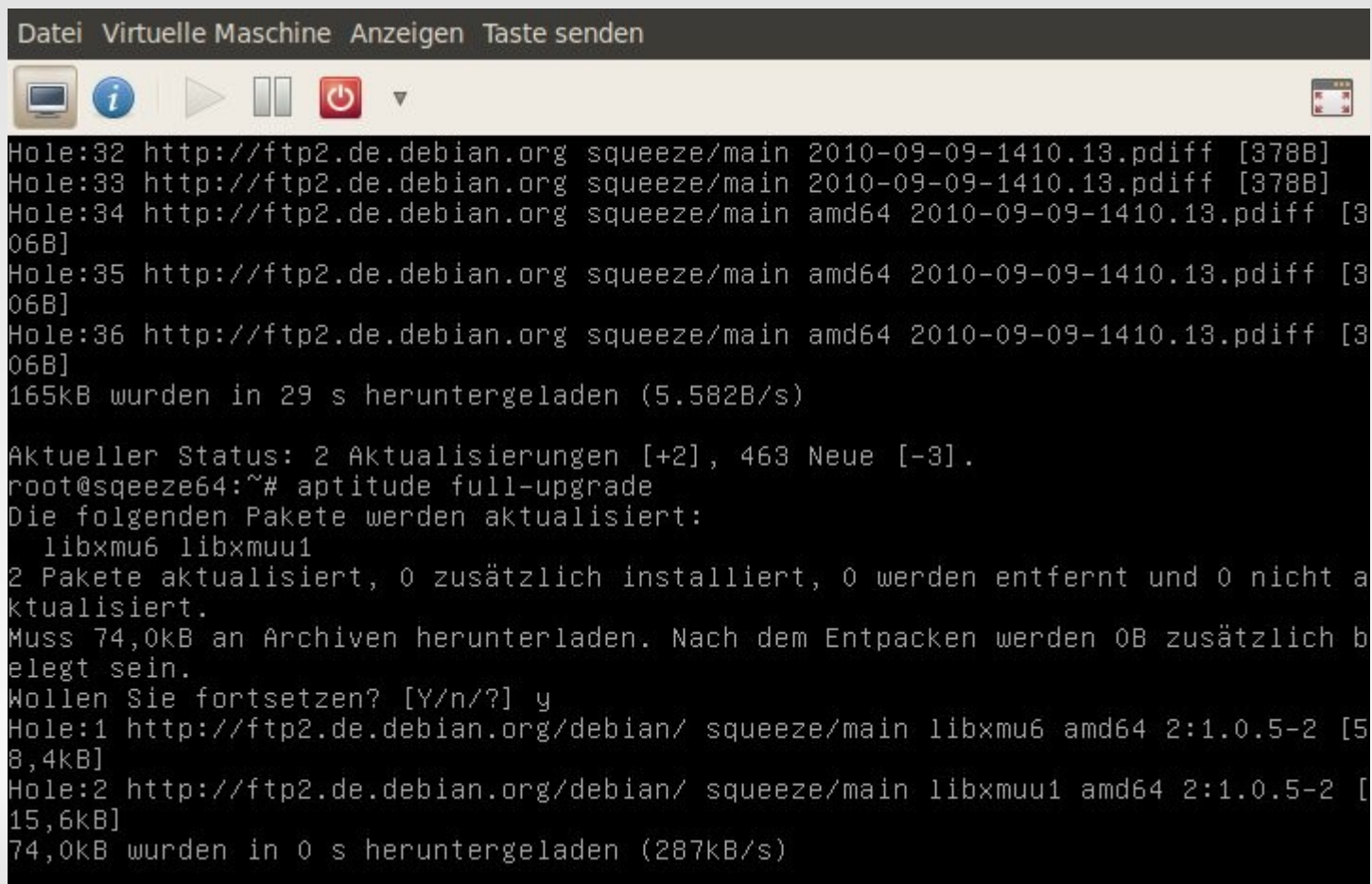
Mannigfaltige Möglichkeiten

- virt-install
- via libvirt-Frontend VM anlegen und verbinden
- direkter Aufruf von kvm

- Ausgabe via serieller Konsole (/dev/pts/X)
- Ausgabe auf VNC Server (ssh-Tunnel, VPN!)

Der erste Boot / Installation

- Mit dem ersten Boot der VM ist die Installation wie auf echter Hardware ein Kinderspiel



```
Datei Virtuelle Maschine Anzeigen Taste senden
Hole:32 http://ftp2.de.debian.org squeeze/main 2010-09-09-1410.13.pdiff [378B]
Hole:33 http://ftp2.de.debian.org squeeze/main 2010-09-09-1410.13.pdiff [378B]
Hole:34 http://ftp2.de.debian.org squeeze/main amd64 2010-09-09-1410.13.pdiff [306B]
Hole:35 http://ftp2.de.debian.org squeeze/main amd64 2010-09-09-1410.13.pdiff [306B]
Hole:36 http://ftp2.de.debian.org squeeze/main amd64 2010-09-09-1410.13.pdiff [306B]
165kB wurden in 29 s heruntergeladen (5.582B/s)

Aktueller Status: 2 Aktualisierungen [+2], 463 Neue [-3].
root@squeeze64:~# aptitude full-upgrade
Die folgenden Pakete werden aktualisiert:
  libxmu6 libxmu1
2 Pakete aktualisiert, 0 zusätzlich installiert, 0 werden entfernt und 0 nicht aktualisiert.
Muss 74,0kB an Archiven herunterladen. Nach dem Entpacken werden 0B zusätzlich belegt sein.
Wollen Sie fortsetzen? [Y/n/?] y
Hole:1 http://ftp2.de.debian.org/debian/ squeeze/main libxmu6 amd64 2:1.0.5-2 [58,4kB]
Hole:2 http://ftp2.de.debian.org/debian/ squeeze/main libxmu1 amd64 2:1.0.5-2 [15,6kB]
74,0kB wurden in 0 s heruntergeladen (287kB/s)
```

Fazit

- KVM bietet also alle Vorteile der bislang etablierten Virtualisierungssysteme:
- Einfaches Setup von Virtualbox
- Mächtiges Networking von VMWare
- Geschwindigkeit und Skalierbarkeit von Xen
- Liest / Schreibt gängige Imageformate: cloop, bochs, raw, cow, dmg, iso, qcow(2), vmdk, vpc
- On/Offline Migration, Durchreichen von Hardware sind noch experimentelle Features.

Noch Fragen? / Hands-On

Vielen Dank für die Aufmerksamkeit!