qemu in der Praxis

qemu im alltäglichen Einsatz

von Michael Hartmann
<michael.hartmann@as-netz.de>

Allgemeines und Fakten

- Virtuelle Maschine (Emulator)
- emuliert x86, x86-64 bzw. AMD64, PowerPC und Sparc32/64 Hardware (Alpha, ARM und S390 im Teststadium)
- unterstützte Betriebssysteme: Linux, Windows, FreeBSD, NetBSD, OpenBSD und Mac OS X
- einfach zu benutzen
- nützlich (Testen/Ausprobieren von Live-CDs oder anderen Betriebssystemen)
- Open Source

Emulierte Hardware

- PCI und ISA-System
- Grafikkarte
- PS/2 Maus und Tastatur
- zwei PCI ATA-Schnittstellen (Unterstützung für vier Festplatten-Images im *VMware-*, *VirtualPC-*, *Bochs-*, *cloop-* und *dd-* Format)
- CD-ROM/DVD-Laufwerk (Image oder reales Laufwerk)
- Diskettenlaufwerk (Image oder reales Laufwerk)
- Netzwerkkarte (und ein DHCP-Server)
- Serieller Port
- Paralleler Port
- Soundkarte

Einfache Optionen

$ qemu

- -fd{a|b} Datei (Diskettenimage oder Device)
- -hd{a|b|c|d} Datei (Festplatten/CD-ROM-Image oder Device)
- -cdrom Datei (CD-ROM-Image oder Device)
- -boot {a|c|d} Bootreihenfolge festlegen
- -m *Mbits* Größe des emulierten Arbeitsspeichers
- -localtime lokale Uhrzeit benutzen (z.T. notwendig für Windows und MS-DOS)
- -full-screen im Vollbildschirm starten

Beispiele und Demo

- ReactOS (OS mit Zielsetzung völliger Kompatibilität zu Microsoft Windows NT):
  `$ qemu -cdrom ReactOS_Live.iso`
- Knoppix (bekannte Live-CD Distribution)
  `$ qemu -m 128 -hdc knoppix.iso -fullscreen`
- Windows 95
  `$ qemu -localtime -hda win95.img`
- MiniMultiboot:
  `$ qemu -localtime -cdrom multiboot.iso`
- `$ qemu -fda floppy.img -hda hda.img -boot c`

Images

- Image aus vorhandener CD erstellen:
  $ dd if=/dev/cdrom of=image.img
- Image aus vorhandener Diskette erstellen:
  $ dd if=/dev/fd0 of=image.img
- Festplatteimage zur Verwendung in qemu erstellen:
  $ qemu-img [-O cloop|vmdk|qcow|cow|raw] → name.img 5G

Forgeschrittene Optionen

- -snapshot Änderungen nicht direkt speichern
- -enable-audio Audiohardware emulieren
- -user-net Netzwerk mit DHCP aufbauen
- -n *script* Netzwerkkonfigurationsscript ausführen
- -smb *dir* Freigabe über SMB in Gast-OS
- -kernel *kernel* Pfad zum Linuxkernel
- -append '...' Appendline
- -initrd *initrd* Pfad zu Initrd

Beispiele und Demo

- Auf Festplatte installiertes Betriebssystem starten
  $ su -c 'qemu -snapshot -hda /dev/hda'
- Knoppix mit SMB-Freigabe und Netzwerk
  $ qemu -user-net -smb '/home/blackmole/'
- Eigene Linux Distribution starten
  $ qemu -hda festplatte.img -kernel /boot/vmlinuz →
  -initrd /boot/initrd -append 'root=/dev/hda'

Monitor

- Erreichbar über [Strg]+[ALT]+[2] ([1] Emulation)

• stop / c	Starten/Stoppen der Emulation
• {load|save}vm	Speichern/Laden des Zustandes
• sendkeys	Tasten an Emulation senden (z.B. [Strg]+[Alt]+[Entf])
• change	Geräte auswechseln
• commit	Änderungen speichern
• Screendump *file*	Screenshots schießen

Ausblick

- kqemu: Kernel Modul für besser Performance (um den Faktor 2, aber closed source!)
- viele weitere Optionen (man qemu)
- Unterstützung weiterer Geräte (z.B. USB)
- weitere Netzwerkfeatures (z.B. Port-Forwarding, tftp)

qemu in der Praxis

von Michael Hartmann
<michael.hartmann@as-netz.de>