

Xen

Felix Krohn

fx@kro.hn

gpg: 0x1C246E3B

UnFUG

November 2, 2006

Übersicht

- 1 Virtualisierung
 - Betriebssystemvirtualisierung
 - Systemvirtualisierung
 - Hardware-Emulation
 - Paravirtualisierung
- 2 XEN
- 3 Installation
- 4 Konfiguration
 - Beispiele
 - xend
- 5 Xen erweitern
- 6 Fragen
- 7 Links

Betriebssystemvirtualisierung: aufgebohrtes chroot

- Verschiedene Umgebungen laufen unter einem Kernel
- Trennung der Instanzen durch das Betriebssystem

Example

- Linux vServer
- *BSD jails
- (Open-)Solaris Zones
- OpenVZ, Virtuozzo

Betriebssystemvirtualisierung: aufgebohrtes chroot

- Verschiedene Umgebungen laufen unter einem Kernel
- Trennung der Instanzen durch das Betriebssystem

Example

- Linux vServer
- *BSD jails
- (Open-)Solaris Zones
- OpenVZ, Virtuozzo

Betriebssystemvirtualisierung: Vor-/Nachteile

Vorteile

- weniger Speicherbedarf, da nur ein Kernel pro Hardware
- schnell, da keine Emulation nötig
- geringer Aufwand

Nachteile

- kein eigener Kernel
- kein Zuteilen oder Einschränken von Ressourcen möglich
- keine Hardware-Abstraktion: Hardware wird direkt mit anderen Gästen geteilt
- Beispiel: CAP_NET_RAW

Betriebssystemvirtualisierung: Vor-/Nachteile

Vorteile

- weniger Speicherbedarf, da nur ein Kernel pro Hardware
- schnell, da keine Emulation nötig
- geringer Aufwand

Nachteile

- kein eigener Kernel
- kein Zuteilen oder Einschränken von Ressourcen möglich
- keine Hardware-Abstraktion: Hardware wird direkt mit anderen Gästen geteilt
- Beispiel: CAP_NET_RAW

Systemvirtualisierung

- Gastsystem wird eigene Hardware-Umgebung vorgegaukelt
- Aufrufe aus dem Gastsystem werden evtl. angepasst und an das Hostsystem weitergeleitet
- Virtualisierungssoftware bedient sich an den Ressourcen des Hostbetriebssystems

Example

- VMware: Player, Server, Workstation
- MS Virtual PC

Systemvirtualisierung

- Gastsystem wird eigene Hardware-Umgebung vorgegaukelt
- Aufrufe aus dem Gastsystem werden evtl. angepasst und an das Hostsystem weitergeleitet
- Virtualisierungssoftware bedient sich an den Ressourcen des Hostbetriebssystems

Example

- VMware: Player, Server, Workstation
- MS Virtual PC

Systemvirtualisierung: Vor-/Nachteile

Vorteile

- Gastbetriebssystem muß nicht oder nur wenig angepasst werden
- Gastsysteme können sich gegenseitig nicht "sehen"
- Erfordert keine Änderungen am Hostsystem

Nachteile

- relativ langsam, jeder Hardwarezugriff wird (tlw. mehrfach) umgewandelt und weitergegeben
- kein Zugriff auf die echte Hardware aus Gastsystemen, daher z.B. keine 3D-Beschleunigung

Systemvirtualisierung: Vor-/Nachteile

Vorteile

- Gastbetriebssystem muß nicht oder nur wenig angepasst werden
- Gastsysteme können sich gegenseitig nicht "sehen"
- Erfordert keine Änderungen am Hostsystem

Nachteile

- relativ langsam, jeder Hardwarezugriff wird (tlw. mehrfach) umgewandelt und weitergegeben
- kein Zugriff auf die echte Hardware aus Gastsystemen, daher z.B. keine 3D-Beschleunigung

Hardware-Emulation

Hardware-Emulation

- Kompletter Rechner wird emuliert

Example

- Bochs
- MS Virtual PC für PPC

Vorteile

- ermöglicht z.B. Emulation eines x86 auf einem PPC
- Erfordert keine Änderungen am Gast- oder Hostsystem

Nachteile

- extrem langsam

Hardware-Emulation

Hardware-Emulation

- Kompletter Rechner wird emuliert

Example

- Bochs
- MS Virtual PC für PPC

Vorteile

- ermöglicht z.B. Emulation eines x86 auf einem PPC
- Erfordert keine Änderungen am Gast- oder Hostsystem

Nachteile

- extrem langsam

Hardware-Emulation

Hardware-Emulation

- Kompletter Rechner wird emuliert

Example

- Bochs
- MS Virtual PC für PPC

Vorteile

- ermöglicht z.B. Emulation eines x86 auf einem PPC
- Erfordert keine Änderungen am Gast- oder Hostsystem

Nachteile

- extrem langsam

Hardware-Emulation

Hardware-Emulation

- Kompletter Rechner wird emuliert

Example

- Bochs
- MS Virtual PC für PPC

Vorteile

- ermöglicht z.B. Emulation eines x86 auf einem PPC
- Erfordert keine Änderungen am Gast- oder Hostsystem

Nachteile

- extrem langsam

Paravirtualisierung

Paravirtualisierung

- **Keine** Emulation
- stattdessen Nutzung eines "Hypervisors", der Zugriffe auf Ressourcen (Hardware) synchronisiert

Example

- Adeos
- Zukünftige Versionen von MS VPC
- Xen

Paravirtualisierung

Paravirtualisierung

- **Keine** Emulation
- stattdessen Nutzung eines "Hypervisors", der Zugriffe auf Ressourcen (Hardware) synchronisiert

Example

- Adeos
- Zukünftige Versionen von MS VPC
- Xen

Paravirtualisierung: Vor-/Nachteile

Paravirtualisierung

Vorteile

- Performance
- Hardware-Zugriffe möglich
- Immer mehr Unterstützung auf Hardware-Ebene: Vanderpool, Pacifica
- Verbreitung: SuSE, Debian, Fedora, Ubuntu, ...

Nachteile

- Wenn keine Hardware-Unterstützung: Gastssysteme müssen stark angepasst bzw. auf die Architektur portiert werden

Paravirtualisierung: Vor-/Nachteile

Paravirtualisierung

Vorteile

- Performance
- Hardware-Zugriffe möglich
- Immer mehr Unterstützung auf Hardware-Ebene: Vanderpool, Pacifica
- Verbreitung: SuSE, Debian, Fedora, Ubuntu, ...

Nachteile

- Wenn keine Hardware-Unterstützung: Gastssysteme müssen stark angepasst bzw. auf die Architektur portiert werden

Geschichte von Xen

- Entwickelt an der Universität Cambridge
- Erstes Release in 2003
- Seitdem Entwicklung durch OSS Community, kommerzieller Vertrieb durch XenSource Inc.
- 2005: Intel lanciert "Vanderpool"
- 2006: AMD zieht mit "Pacifica" nach

Features

- Open Source
- Paravirtualisierung
- Live-Migration: 60-300 ms
- unter Linux: SMP-Unterstützung
- nutzt Vanderpool und Pacifica wenn vorhanden

Aufbau

- Direkt beim Booten startet Xen
- XEN lädt und startet dann angegebene Dom0
- Über die Dom0 lässt sich Xen kontrollieren
- Dom0 beherbergt Xen-Tools, xend
- Neben der Dom0 lassen sich beliebig viele DomU starten

Binärpakete

- Von XenSource bereitgestellte Binärpakete installieren
- Alternativ: Distributionspakete verwenden
- Xen Booten
- Fertig

Dom0 aus den Quellen

- Dependencies: binutils, openssl, zlib1g-dev, grub, bridge-utils, python-krams
- runterladen, entpacken
- README lesen :-)
- `make KERNELS="linux-2.6-xen0"`
`linux-2.6-xen-config CONFIGMODE=menuconfig`
- `make dist` oder `make xen` oder `make install` oder ...
- `make tools install-tools`
- `make install-kernels` oder von Hand installieren
- `mv /lib/tls /lib/tls.disabled`
- GRUB konfigurieren
- Booten

DomU aus den Quellen

- DomU-Dateisystem vorbereiten und mounten (z.B. `debootstrap`)
- `make KERNELS="linux-2.6-xenU" linux-2.6-xen-config CONFIGMODE=menuconfig`
- Kernel irgendwo im Dom0 ablegen
- Module im DomU ablegen
- Konfiguration erstellen
- domU entsprechend konfigurieren (`fstab`, `netzwerk`, `sshd`, ...)
- domU-System unmounten
- `xm create $NAME [-c]`

Kernel: Unterschiede zwischen Dom0 und DomU

Dom0

- Alle benötigten Hardware-Treiber
- XEN --> [X] Domain0
- Je nach Netzwerk-Setup: NAT, Bridging, ...
- loop-devices

DomU

- keine konkreten Hardware-Treiber (IDE, Netzwerkkarten, ...)
- nur die benötigten Features (IPv6, iptables, ...)
- XEN --> [_] Domain0

Konfiguration

- DomU werden über Konfigurationsdateien in `/etc/xen/` eingerichtet
- Der Name der DomU muß für die Dom0 eindeutig sein

Example

```
kernel = '/boot/xendomU/vmlinuz-2.6.16.29-xenU'  
memory = 256  
name = "debian1"  
vif = [ '' ]  
disk = [ 'file:/mnt/xensys2-root.img,hda1,w',  
         'file:/mnt/xensys2-swap.img,hda2,w',  
         'phy:/dev/hdc,hdb,r' ]  
root = "/dev/hda1 ro"  
extra = "4"
```

Aufruf mit xm

- xm create – Instanz starten
- xm console – an Konsole einer DomU binden
- xm destroy – virtuellen Stecker ziehen
- xm list – momentane Domains auflisten
- xm [un]pause – Pause [beenden]
- xm save – Zustand sichern
- xm restore – Zustand wiederherstellen
- xm reboot, xm shutdown – . . .
- Als Domain ist entweder die Domainnummer oder der Name anzugeben
- Für alles weitere: xm help :-)

xm list

- r – running
- b – blocking
- p – paused
- s – shutdown
- c – crashed

–verbose

- kernel: Der Ort in der Dom0, an der zu bootende Kernel ist
- memory: soviel RAM soll diese DomU erhalten
- name: eindeutiger Name für diese DomU
- vif: Netzwerkschnittstelle, parameterlos: default-Werte verwenden
- disk: Laufwerke, die für die DomU unter dem ang. Namen erreichbar sein sollen, Berechtigung
- root: Angabe der root-partition aus Sicht der DomU. Wird von Xen als Parameter mitgegeben

Beispiel: NetBSD

Example

```
kernel = "/boot/xendomU/netbsd-XEN3_DOMU"  
memory = 64  
name = "netbsd"  
vif = [ 'mac=00:11:85:13:37:DD, bridge=xenbr0' ]  
disk = [ 'file:/mnt/netbsd-root.img,0x3,w',  
         'phy:/dev/hdc,cd0a,r' ]  
root = "/dev/wd0d"
```

Beispiel: FreeBSD Installationsimage

Example

```
kernel = "/boot/xendomU/freebsd-XENU_INSTALL"  
memory = 256  
name = "freebsd-xen-INSTALL"  
vif = [ '' ]  
disk = ['file:/mnt/freebsd/freebsd-root.img,hda1,w']  
extra = "4"  
extra = "boot_verbose"  
extra += ",boot_single"  
extra += ",vfs.root.mountfrom=ufs:/dev/md0"  
extra += ",kern.hz=100"
```

GRUB-config

Example

```
default 1
title Xen 3.0.3
root (hd0,0)
kernel /boot/xen-3.0.3-0.gz dom0_mem=256M noreboot
module /boot/vmlinuz-2.6.16.29-xen0 root=/dev/hda1 ro
max_loop=64

title normales Linux
root (hd0,5)
kernel /boot/vmlinuz-2.6.15-26-386 root=/dev/hda6 ro
quiet vga=791
```


xend-Konfiguration

- `man xend-config.sxp`
- im Prinzip 2 mögliche Netzwerkkonfigurationen:
 - 1 Bridging
 - 2 Routing
- Bei ersterem werden die virtuellen interfaces einfach an ein echtes if gebridged
- Mit zweiterem ist es möglich, ein virtuelles Netzwerk aufzubauen
- (Beispiele)

GUIs

- XenMan
- Virtual Machine Manager
- Enomalism
- Xen vServer Hoster
- weitere Hoster per google...

Fragen?

- Xen Download
- NetBSD/xen Howto
- FreeBSD/Xen Howto
- Solaris/Xen Doku
- Xen Cluster
- Intel Vanderpool
- XenWiki
- fertig bootbare Xen-Images und Konfigurationen