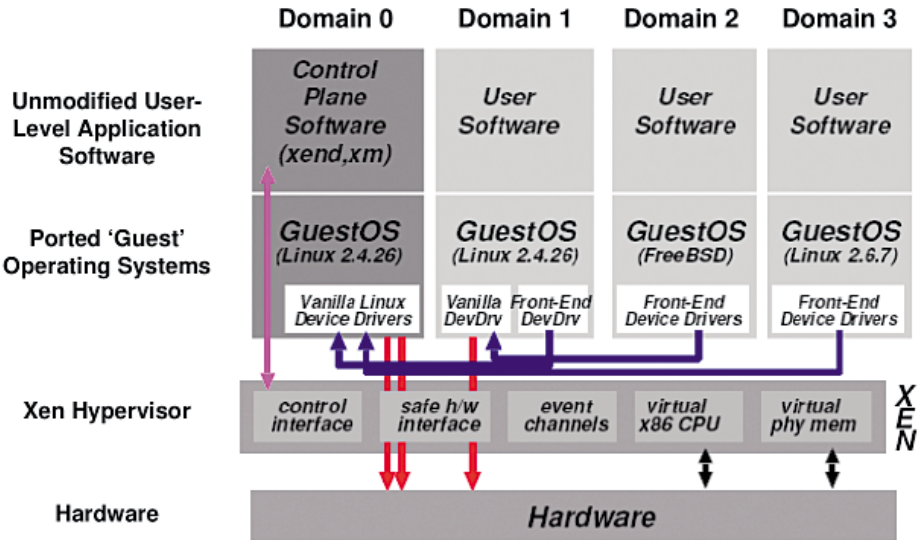
XEN: VIELE LINUX-INSTANZEN AUF EINEM RECHNER

Großrechner-Feeling

Was bis vor kurzem Großrechnern vorbehalten war, funktioniert nun auf normalen PCs: die gleichzeitige Verwendung vieler Betriebssystem-Instanzen mithilfe der freien Virtualisierungslösung Xen. Wir zeigen Konfiguration und Einsatz und sagen, für wen es sich lohnt.

VON **MATTIAS SCHLENKER**



Der Xen-Kernel ist eine dünne Software-Schicht zwischen Hardware und Gastsystemen. *Domain 0* enthält Treiber und Verwaltungswerkzeuge. (Quelle: Xensource.com)

Bei „richtigen Computern" ist es seit über dreißig Jahren gang und gäbe, dass nicht nur ein Betriebssystem die Herrschaft über die Hardware inne hat. Möglich wird diese Form der Systemvirtualisierung durch Hardware-Unterstützung und eine kompakte Software-Komponente, die als *Virtual Machine Monitor*, die Ausführung der parallel laufenden Betriebssysteme überwacht und die Ressourcen verwaltet.

Gewöhnliche PCs bieten längst genügend Rechenleistung und schnelle I/O-Schnittstellen, um sich - theoretisch - für die parallele Ausführung von Betriebssystem-Instanzen zu qualifizieren. Allerdings fehlt ihnen einerseits die Hardware-Unterstützung für „echte" Virtualisierung, andererseits war bis vor kurzem kein *Virtual Machine Monitor* erhältlich. Für gelegentliches Arbeiten oder Tests mit einer weiteren Betriebssystem-Instanz war und ist VMware ein guter Ansatz: Dank direkter Prozessorzugriffe ist diese Form der virtuellen Maschine ausreichend schnell, und durch die Emulation der übrigen PC-Hardware sind keine Anpassungen am Gastsystem notwendig. Eine im Server-Bereich beliebte Lösung sind Kernel mit einer erweiterten *chroot*-Version, bei denen der Gast-Instanz nur die eigenen Netzwerkschnittstellen und Prozesse sichtbar sind. Auch wenn diese Ansätze mit Namen wie *Virtuozzo* oder *vServer* den Eindruck erwecken, echte Virtualisierungslösungen zu sein, handelt es sich streng genommen um Pseudo-Virtualisierungslösungen.

Echte und noch dazu freie Virtualisierung bietet Xen, ein *Virtual Machine Monitor* oder *Hypervisor*, der die Tücken der PC-Hardware elegant umschifft. Ähnlich einem Micro-Kernel übernimmt Xen nur die Ressourcen-Zuteilung auf der untersten Ebene (Ring 0), verwaltet also Prozessorzugriffe und Arbeitsspeicher sowie Zugriffe auf Busse wie PCI. Der Linux-Kernel wird unter Xen als unprivilegierte Anwendung (Ring 1) gestartet. Die Anwendungsprogramme der Gastsysteme laufen schließlich in Ring 3. Da die Kernel-Anpassungen jedoch moderat ausfallen, kann der Rest des Linux-Systems ohne Modifikationen genutzt werden, und Dual-Boot zwischen Xen-Linux und einem normalen Linux stellt kein Problem dar.

Zweiklassengesellschaft

Damit der eigentliche *Hypervisor* nicht um viele Verwaltungswerkzeuge ergänzt werden musste, wird zusammen mit Xen eine privilegierte Linux-Instanz gestartet. Diese wird als *dom0* (*Domain Zero*) bezeichnet und erhält direkten Zugriff auf PCI- und USB-Geräte sowie Netzwerkschnittstellen und Festplatten. Daneben enthält *dom0* die Werkzeuge zur Initialisierung und Kontrolle weiterer Linux- oder BSD-Instanzen und stellt diesen virtuel-

Drei Linuxe gleichzeitig: openSUSE 10.1 in *Domain 0*, Kanotix im VNC-Fenster und SSH-Zugriff auf das uClibc-Mini-Linux.

le Netzwerkgeräte oder Festplatten zur Verfügung. Unprivilegierte Instanzen heißen in Xen-Speak *domU* (*Domain Unprivileged*). Ihnen stehen zunächst nur das virtuelle Netzwerk sowie virtuelle Festplatten - welche in *dom0* entweder auf reale Partitionen oder Image-Dateien abgebildet werden - zur Verfügung. Direkter Zugriff auf PCI-Karten ist in der Basiskonfiguration nicht möglich, lässt sich aber bei Bedarf freischalten. Einziges direktes Ausgabemedium ist eine mit *dom0* verdrahtete Textkonsole, die den Zugriff auch erlaubt, wenn die virtuelle Netzwerkschnittstelle noch nicht konfiguriert ist. Alle weiteren Zugriffe auf unprivilegierte Domains und aus ihnen heraus sollten über das Netzwerk erfolgen. Dank der Netzwerkstransparenz des Grafik-Subsystems X11 und des Druckersubsystems CUPS stellt dies zumindest unter Unix-artigen Systemen kein Problem dar.

Xen ganz praktisch

Das Vorhandensein von Xen 3.0 in openSUSE 10.1 und Fedora Core 5 sollte eigentlich den Einstieg beträchtlich erleichtern. Wählt man beim nächsten Reboot den Xen-Eintrag im Grub-Menü, fallen zwei Dinge auf: Die ersten Bootmeldungen entstammen nicht wie üblich Linux, sondern dem vorgelagerten Xen-Kernel. Auch ist der Framebuffer-Modus des Linux-Kernels deaktiviert. Der Grund: Nur Xen darf die BIOS-Routinen aufrufen, die zum Umschalten in den Framebuffer-Modus notwendig sind.

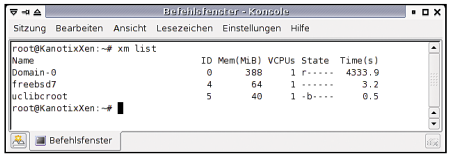
Neu in Xen 3.0 ist ACPI-Unterstützung. Wie beim normalen Linux-Kernel sollten Sie jedoch gerade bei Notebooks mit der individuellen Kombination Kernel-Hardware herausfinden, ob ACPI auf Ihren System nutzbar ist. Wenn Sie auf Nummer sicher gehen wollen, deaktivieren Sie die *powersaved*-Funktion. Vorsichtig sollten Sie auch mit den proprietären Grafikkartentreibern von Ati und Nvidia respektive deren Kernelmodulen sein. Während die Open-Source-Treiber mit Xen harmonieren, neigen die Closed-Source-Treiber gelegentlich dazu, das Xen-Linux-System abstürzen zu lassen.

Ist das Xen-System hochgefahren, steht ein erster Rundgang an. Wichtigstes Kommandozeilenwerkzeug, um Xen-Einstellungen abzufragen und zu verändern, ist das Python-Script *xm*. Dieses verbindet sich mit dem im Hintergrund laufenden Xen-Daemon *xend*. Mit dem Befehl

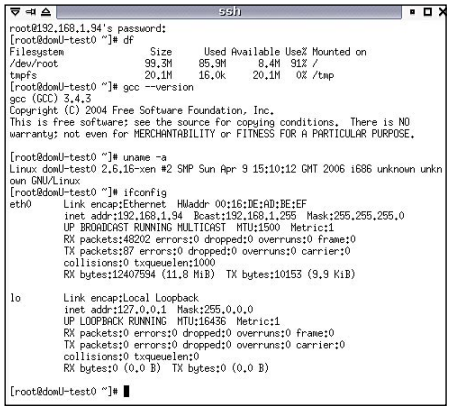
- `xm dmesg`

erhalten Sie beispielsweise die Bootmeldungen von Xen. Eine Liste aller aktuell unter Xen laufenden Betriebssystem-Instanzen zeigt

- `xm list`



Der Befehl *xm list* zeigt die aktiven Xen-Domains, hier läuft testweise ein FreeBSD als Gast.



Das *uClibc-Root-Dateisystem* hat trotz der geringen Größe alle Entwicklerwerkzeuge an Bord. Das erleichtert Experimente beispielsweise mit Apache.

Ein, zwei, viele Linuxe

Unmittelbar nach dem Start ist die Betriebssysteminstanz natürlich nur die *Domain Zero*. Um wirklich von Xen profitieren zu können, benötigen Sie noch unprivilegierte Instanzen. openSUSE 10.1 bietet hierfür bereits ein rudimentäres Framework - die einzelnen Instanzen benötigen dennoch etwas Nacharbeit. Um diese Nacharbeit abzukürzen und die unprivilegierten Instanzen kompakt zu halten, entschieden wir uns für das Root-Dateisystem des uClibc-Projektes. Sie finden es - mit leichten Modifikationen, die eine sofortige Nutzung als *Virtual Block Device* erlauben - auf der Heft-DVD.

Auf einem openSUSE-System, das mit einer Domain Zero gestartet wurde, genügt das Anlegen einer Konfigurationsdatei */etc/xen/uclibcroot.cfg* mit folgendem Inhalt:

```
kernel = „/boot/vmlinuz-2.6.16-8-xen“
memory = 32
name = „uclibcroot“
vif = [ ‘mac=00:16:de:ad:be:ef’ ]
disk = [
‘file:/tmp/uclibcroot.img,sda1,w’ ]
root = „/dev/sda1 rw“
extra = „ipaddr=192.168.1.93
gateway=192.168.1.252“
```

Der Pfad zum Dateisystem-Image ist gegebenenfalls anzupassen. Wichtig sind die Leerzeichen zwischen eckiger Klammer und einfachem Anführungszeichen. Die MAC-Adresse kann beliebig gewählt werden, sollte aber mit *00:16* anfangen - dies kennzeichnet einen nicht an Hardware-Hersteller vergebenen Bereich. Achten Sie auch darauf, dass Sie nicht zwei verschiedenen Xen-Instanzen die gleiche Hardware-Adresse zuweisen, wir zählen deshalb ab *de:ad:be:ef* hexadezimal hoch.

Die Zeile *extra* enthält die Append-Parameter des Kernels. Wir haben das uClibc-Root-Dateisystem um ein Startup-Script */etc/init.d/S40 xennet* erweitert, das die Append-Parameter ausliest und zur Netzwerkkonfigurati-

Weitere Informationen im Web

- www.xensource.com
- http://en.opensuse.org/Xen
- http://en.opensuse.org/Installing_Xen3
- www.fedoraproject.org/wiki/FedoraXenQuickstartFC5
- http://netbsd.org/Ports/xen
- www.fsmware.com
- www.opensolaris.org/os/community/xen
- http://buildroot.uclibc.org
- www.mattiasschlenker.de/06009

Feldversuch: Kanotix 2005-04 xenifiziert

Auf unserem SUSE-Testrechner befand sich noch eine Kanotix-Installation, welche wir mit möglichst wenig Aufwand als *domU* unter openSUSE 10.1 verwenden wollten. Unsere Vorgehensweise:

Zunächst erstellten wir eine Konfigurationsdatei */etc/xen/kanotix.cfg* mit dem Inhalt:

```
kernel = „/boot/vmlinuz-
➥ 2.6.16-8-xen“
ramdisk = „/boot/initrd-2.6.16-8-xen“
memory = 200
name = „kanotix“
vif = [ ‘mac=00:16:de:ad:be:f0’ ]
disk = [ ‘phy:hda5,hda5,w’,
         ‘file:/tmp/swap.img,hda3,w’
]
root = „/dev/hda5 rw“
```

Eine 300 Megabyte große Swap-Datei konfigurierten wir mit den Befehlen

```
dd if=/dev/zero bs=1048576
➥ count=300
mkswap /tmp/swap.img
```

Die übergebenen Namen der Gerätedateien sollten den Einträgen der */etc/fstab* des Gastsystemes entsprechen. Alle weiteren Dateisysteme entfernten wir aus der */etc/fstab* beziehungsweise deaktivierten sie mit dem Schlüsselwort *noauto*.

Da unprivilegierte Domains nur eine virtuelle Konsole öffnen können, müssen *tty2* bis *tty6* in der Datei */etc/inittab* des Gastes auskommentiert werden.

Ähnlich sieht es beim X-Server aus. Wir wollten zwar, dass *kdm* startet und über das Netzwerk erreichbar ist, aber in der Gastinstanz nicht versucht wird, einen lokalen X-Server zu starten. Dazu deaktivierten wir den X-Server indem wir in der Datei */etc/kdm/Xservers* alle (!) Zeilen auskommentierten. An der Datei */etc/kde3/kdm/kdmrc* nahmen wir folgende Anpassungen vor:

```
[General]
# ...
StaticServers=
# ReserveServers=:1,:2,:3
# ...
[Xdmcp]
Enable=true
```

Rechner, die Zugriff auf KDM erhalten sollen, müssen in der Datei */etc/kde3/kdm/Xaccess* eingetragen werden. KDM lässt nur Rechner zu, deren IP-Adresse zu einem Hostnamen aufgelöst werden kann. Passen Sie die Datei */etc/hosts* entsprechend an:

```
192.168.1.91 xen0.test
```

Um nicht bei jedem Neustart die Netzwerkschnittstelle *eth0* von Hand korrigieren zu müssen, trugen wir sie in der Datei */etc/network/interfaces* ein:

```
auto lo lan0 eth0
iface lo inet loopback
iface lan0 inet dhcp
iface eth0 inet static
    address 192.168.1.93
    netmask 255.255.255.0
    gateway 192.168.1.252
```

Bootet man Kanotix direkt und ohne Xen, kommt der Schnittstellenname *lan0* und DHCP-Konfiguration zum Einsatz, als unprivilegierte Xen-Domain wird *eth0* und eine statische Konfiguration verwendet.

Nach dem Start des *xenU-Kanotix* können Sie mit dem Befehl

```
X -query 192.168.1.93 :1
```

das grafische Login über das Netzwerk anfordern. Mit *Strg+Alt+F7* respektive *Strg+Alt+F8* schalten Sie zwischen den Instanzen um.

Perfekt ist diese Konfiguration noch nicht: die TLS-Bibliotheken, die Xen ausbremsen, sollten Sie verschieben. Kopieren Sie zudem den Ordner */lib/modules/linux-2.6.x-xen* der *Domain 0* in die Gastdomain, damit alle Dienste sauber gestartet werden. Bei Verwendung eines exotischen Dateisystems für die Root-Partition des Gastes kann es vorkommen, dass die *Xen-initrd* des Hosts nicht funktioniert. In diesem Fall müssen Sie entweder eine alternative *initrd* erstellen oder einen Kernel mit statisch integrierten Treibern kompilieren.

on heranzieht. Bitte werfen Sie einen Blick in unsere Beispiel-Konfigurationsdatei auf DVD, um alle Parameter in Erfahrung zu bringen.

Mit Konfigurationsdatei und Dateisystem-Image wird die neue, unprivilegierte Domain gestartet:

```
xm create /etc/xen/uclibcroot.cfg -c
```

Der Parameter *-c* weist Xen an, das aktuelle Terminal als virtuelle Konsole der neu gestarteten Instanz zu verwenden. Dieser erste Startvorgang wird recht lange dauern, da der SSH-Daemon *Dropbear* Schlüssel generiert. Sie können sowohl die Schlüsselerzeugung aber auch spätere SSH-Logins etwas beschleunigen, indem Sie die neue, unprivilegierte Domain anpingen - das sorgt für dringend benötigte Zufallereignisse, denn die sonst herangezogenen Tastatur- oder Maus-Interrupts fehlen in unprivilegierten Domains. Loggen Sie sich in dieser als *root* ein (das Kennwort ist leer) und setzen Sie das Root-Passwort mit dem Befehl *passwd*. Anschließend starten Sie die Xen-Instanz neu. Wahlweise aus der *domU* mit

```
reboot
```

oder von außerhalb über den Xen-Monitor unter Angabe der mit *xm list* ermittelten Domain-ID:

```
xm shutdown -R 1
```

Ihre uClibc-Linux-Xen-Instanz ist jetzt ganz normal per SSH erreichbar. Sollten Sie - beispielsweise wegen eines abgestürzten SSH-Daemons oder einer fehlerhaften Netzwerk-Konfiguration direkten Zugriff auf die Konsole benötigen, verwenden Sie

```
xm console 1
```

um sich mit der Konsole des Gastsystems zu verbinden. Der Befehl *xm* hält weitere Subkommandos beispielsweise für das Einfrieren von Gastsystemen oder die Migration auf einen anderen Rechner bereit, die Sie sich mit

```
xm help
```

anzeigen lassen.

Unvergleichlich

Vergleiche mit VMware oder anderen Virtualisierungslösungen, die PC-Hardware komplett emulieren, sind mit Xen 3.0 auf normaler PC-Hardware praktisch unmöglich. Aus Performance-Gesichtspunkten ist der von Xen praktizierte Ansatz, der einen modifizierten Gastkernel erfordert, sicher die beste Lösung. Sie schränkt jedoch die Auswahl der Betriebssysteme noch stark ein: als *Domain 0* ist nur Linux eine sinnvolle Wahl, für unprivilegierte Domains stehen Linux, NetBSD, FreeBSD (zum Testzeitpunkt im Alpha-Stadium) und Plan 9 zur Verfügung. Andere freie Systeme wie OpenSolaris befinden sich in der Entwicklung.

Wer ein geschlossenes System wie Windows als Gast nutzen möchte, benötigt Hardware-Unterstützung für Virtualisierung und die kostenpflichtige Enterprise-Version von Xen. Mit der breiteren Verfügbarkeit von AMD- und Intel-Prozessoren mit *Pacifica* und *Vanderpool*

Xen nachrüsten

Xen lässt sich mit wenig Aufwand auf bestehenden Linux-Distributionen nachrüsten. Die Voraussetzungen sind moderat: Python sollte in Version 2.3 vorhanden sein, Ihre Distribution sollte auf Kernel 2.6 basieren, und als Bootloader muss *Grub* vorhanden sein. Quellcodes können Sie sich aus dem Mercurial-Repository (Versionsverwaltungssystem) von Xensource beschaffen oder per HTTP herunterladen. Die Quellcodes der Version *3.0.2-Testing* vom 9. April 2006 finden Sie auch auf der Heft-DVD.

- **Entpacken Sie das Quellcode-Archiv und wechseln Sie in den entstandenen Ordner *xen-3.0-testing.hg*. Dort genügt**

```
make world
```

um, Xen nebst Xend und den benötigten Verwaltungswerkzeugen zu kompilieren. Beim *make*-Lauf werden auch zwei Kernel für privilegierte und unprivilegierte Domains kompiliert.

- **Die Installation folgt mit**

```
make install
```

Anschließend finden Sie im Verzeichnis */boot* den Kernel *vmlinuz-2.6.16-xen* sowie den Hypervisor *xen-3.0.2.gz*.

- **Tragen Sie Xen nebst Linux-Kernel in Ihre */boot/grub/menu.lst* ein:**

```
title Kanotix (Xen 3.0.2)
root  (hd0,4)
kernel /boot/xen-3.0.2.gz
module /boot/vmlinuz-2.6.16-xen
root=/dev/hda5 ro
ramdisk_size=100000 lang=de nomce
boot
```

Die Append-Parameter des Linux-Kernels haben wir vom *xenlosen* Kanotix übernommen. Auffällig ist: Xen ist jetzt der Kernel, Linux wird als Modul geladen. Falls eine Ramdisk benötigt wird, erstellen Sie diese wie gewöhnlich mit *mkinitrd* und laden sie über eine zweite *module*-Zeile.

- **Rebooten Sie den Rechner und starten Sie den Kontroll-Daemon - zunächst von Hand:**

```
/etc/init.d/xend start
```

Wenn Sie Xen dauerhaft verwenden, verlinken Sie das Startscript im Ordner Ihres Standard-Runlevels (meist */etc/rc5.d* oder */etc/rc3.d*).

Die weitere Einrichtung unprivilegierter Domains erfolgt wie im Fließtext beschrieben.

Gelegentlich fehlen dem Standard-Kernel Treiber für Dateisysteme oder Netzwerkkomponenten.

In diesem Fall können Sie den Xen-Kernel interaktiv konfigurieren und installieren

```
make linux-2.6-xen-config CONFIG-
MODE=menuconfig
make linux-2.6-xen-build
make linux-2.6-xen-install
```

Statt *CONFIGMODE=menuconfig* kann auch das etwas übersichtlichere *CONFIGMODE =xconfig* verwendet werden.

Um die Erstellung einer *initrd* zu umgehen, integrierten wir Treiber für die unmittelbar beim Boot benötigten Controller und Dateisysteme statisch an Stelle als Modul.

Zugriff?

Da aus einer unprivilegierten Linux-Instanz heraus kein X-Server gestartet werden kann, müssen andere Zugriffsmöglichkeiten auf die grafische Oberfläche der „domU"-Instanzen in Erwägung gezogen werden. Wir haben ausprobiert:

- **SSH-Zugriff mit X-Forwarding auf die unprivilegierte Instanz:**

```
ssh -X mattias@192.168.1.93
```

gefolgt vom Start des gewünschten Programms per Kommandozeile. Diese Möglichkeit funktioniert einwandfrei und erfordert nur den Start des SSH-Daemons. Ein Nachteil ist, dass bei vielen verwendeten Fenstern schnell der Überblick verloren geht, welches Fenster zu welcher Instanz gehört.

- **Start eines Display-Manager ohne lokalem X-Server: Wie im Kasten *Feldversuch mit Kanotix* beschrieben, müssen einige Konfigurationsdateien angepasst werden, bevor mit**

```
X -query 192.168.1.93 :1
```

ein Login an der unprivilegierten Instanz angefordert werden kann. Alternativ können Sie mit

```
Xvnc -geometry 800x600 -query
192.168.1.93 :1
```

einen VNC-Server in der *dom0* starten, zu dem Sie mit

```
vncviewer localhost:1
```

verbinden. Diese Lösung erspart das Umschalten zwischen virtuellen Konsolen und die Sitzung bleibt aktiv, wenn das VNC-Fenster geschlossen wird.

- **Der Display-Manager startet den VNC-Server. Tragen Sie dafür *Xvnc* mit seinen Startparametern in die Datei *Xservers* der unprivilegierten Instanz ein. Der Display-Manager ist in diesem Fall direkt per VNC verfügbar, was den Zugriff auch von Windows-Clients deutlich vereinfacht. Ein Nachteil gegenüber dem Umweg über *Xvnc -query* ist, dass nicht mehrere VNC-Instanzen gleichzeitig möglich sind.**

im Laufe des Jahres wird auch die freie Xen-Version Windows-Gäste ausführen können.

Ein tolles Feature von Xen ist die Live-Migration: Innerhalb von Sekundenbruchteilen lässt sich der Systemzustand einer Betriebssysteminstanz einfrieren und auf einen anderen Rechner transferieren. Beim Umzug bleiben sogar bestehende Datenbank- oder Netzwerkverbindungen aktiv. Xensource bietet sogar Tools für das automatische Verschieben von Instanzen auf weniger ausgelastete Rechner. Der Haken in der Praxis: Nur wenn alle Dateisysteme per NFS oder Network-Blockdevice, kurz *SAN/NAS*, eingebunden sind, klappt der Umzug. Derartige Umgebungen dürften sich allerdings nur in *Enterprise-Umgebungen* finden lassen. Auf Deutsch: schnell die Gentoo-Xen-Domain während des Rekompilierens auf das Notebook des Kumpels zu schieben, geht nicht. Immerhin ist es, möglich unprivilegierte Domains zunächst zu stoppen, das Festplatten-Image per *rsync* auf einen anderen Rechner zu kopieren und dort die Instanz neu zu starten. Da alle Schnittstellen abstrahiert werden, ist auf dem neuen Wirt keine Änderung der Konfiguration nötig, was die Downtime gering hält.

Im Test kristallisierte sich heraus, dass ein optimaler Einsatz nur dann möglich ist, wenn die „Kontrollinstanz dom0" möglichst kompakt gehalten wird und nicht selbst Dienste zur Verfügung stellt - ein minimales Debian bietet sich hierfür an. Kurzum: Wer plant, einen Server auf Xen umzustellen, sollte diesen komplett neu aufsetzen, um von Xen profitieren zu können. Hoster der gehobenen Preisklasse werden Xen verstärkt als Alternative zum pseudo-virtualisierten *Vserver* einsetzen, während die Vserver-Preiskämpfer durch den etwas höheren Arbeitsspeicher- und Festplattenbedarf gegenüber *Virtuozzo* abgeschreckt sein dürften. **:JKN**