

XEN mit Suse 9.3

Mit Suse 9.3 (nicht erst seit der 9.3!) kann man relativ schnell und einfach ein virtuelles System aufsetzen und die Funktionalität von XEN testen. Bei der Installation leistet einem YAST gute Dienste. Aber danach hält sich die Unterstützung in engen Grenzen. Wie man aber trotzdem zu einem lauffähigen System kommt, will ich hier mal anhand einer Beispielininstallation zusammenfassen.

Grundsätzliches

Ich will hier nicht die komplette Funktionsweise oder den genauen Aufbau von XEN wiedergeben¹. Aber ein wenig Hintergrundwissen schadet ja nie und damit lassen sich meine Schritte hoffentlich leichter nachvollziehen.

Um XEN (bis Version 2.0 welche hier verwendet wird) benutzen zu können, müssen sowohl der Kernel des Hostsystems als auch der des Gastsystems (die virtuelle Maschine) angepasst werden². Suse hat bereits einen Kernel mit XEN vorbereitet der sowohl für das Hostsystem (auch bezeichnet als Domain 0, DOM 0 oder Xen 0) als auch für das Gastsystem (auch Domu oder Xenu) verwendet werden kann. Am einfachsten installiert man die komplette Selektion 'XEN Virtualization' und verfügt damit über den Kernel und alles weitere notwendige. Der Kernel des Gastsystems (und eine eventuelle initrd) liegen dann im /boot-Verzeichnis des Hostsystems und der Rest des Gastsystems entweder in einer eigenen Partition (oder ähnlichem wie LVM-Volume, iscsi-Device, ...) oder in einer Image-Datei im Filesystem der Domain 0. Gesteuert wird das Gastsystem mit dem Befehl `xm`. `xm` hat ein recht angenehmes Hilfesystem. Mittels `'xm help'` und `'xm help CMD'` (z.B. `'xm help balloon'`) kann man sich einen guten Überblick über die Möglichkeiten des Programms verschaffen.

Aller Anfang ist leicht

zumindest wenn man noch etwas Platz auf der Festplatte hat um ein, zwei Partitionen zu erstellen³. Ich habe für meine Installation 2 Partitionen erstellt, eine als Systempartition und eine Swappartition. Die Systempartition mountete ich unter `/mnt/xen1` (es könnten ja mehrere Maschinen werden). Dies sollte aber nicht dauerhaft geschehen bzw. muß später wieder rückgängig gemacht werden, da die virtuelle Maschine nicht von einer bereits gemounteten Partition bootet. Bei der Swappartition muß man nur aufpassen, daß sie nicht vom Hostsystem genutzt wird. Im YAST findet man unter dem Menüpunkt 'Installation into Directory for XEN' eine abgespeckte Version des von der Grundinstallation her bekannten Installers. Man stellt hier nur in den Optionen das Zielverzeichnis auf den Mountpunkt der künftigen Systempartition ein und verändert evtl. die Software-Auswahl nach seinen Wünschen (ohne die Vorgaben zu ändern werden etwa 2,4 GB Speicherplatz auf der Festplatte verbraucht). Damit können die

¹Es gibt genügend Informationen im Netz wenn man mehr darüber wissen will. Ich werde ein paar Links am Ende anhängen. Auch ein Artikel in der Ausgabe 07/05 des Linux-Magazin ist sicherlich hilfreich für den Wissensdurstigen.

²In der Version 3.0 und mit künftigen Prozessoren wird man auch unveränderte Gastsysteme verwenden können.

³Ich werde später noch zeigen wie eine Datei als Loop-Device verwendet werden kann.

Spiele beginnen und das System wird auf die Partition installiert. Leider endet hier auch die Unterstützung durch YAST.

Nun ist alles installiert, aber was jetzt?

Für jede virtuelle Maschine sollte man eine eigene Konfigurationsdatei anlegen⁴. Im Verzeichnis `/etc/xen` gibt es auch einige Beispieldateien. Ich kopiere die Datei `xmexample1` und editiere die folgenden Einträge in der dadurch entstandenen Datei `xmkiste1`.

```
kernel = "/boot/vmlinuz-xen"
disk = [ 'phy:hdb6,hda1,w', 'phy:hdb7,hda2,w' ]
```

Der Disk-Eintrag stellt dabei ein Mapping der physikalischen Partitionen auf die virtuellen Partitionen (so wie sie dann im Gastsystem sichtbar sind) dar. Damit könnte man das Gastsystem bereits starten. Aber ein paar Vorkehrungen erleichtern uns später das Leben. Dazu mounten wir die spätere Systempartition und erstellen dort eine einfache `fstab`. Es genügen erst mal folgende Zeilen:

```
/dev/hda1 / ext3 noacl 1 1
/dev/hda2 swap swap pri=42 0 0
```

Von XEN selbst wird einem (wenn man es nicht macht, dann bei jedem Booten!) folgender Schritt nahegelegt: `mv /lib/tls /lib/tls.disabled`.

Und zu guter Letzt sollte man sich nach einem beherrzten `chroot /mnt/xen1` (oder wo auch immer man die Partition gemountet hat) mit `passwd` das Passwort des Users `root` an seine Wünsche anpassen. Nun steht aber (nach dem Verlassen der `chroot`-Shell und dem ummounten der künftigen Systempartition) einem Start der virtuellen Maschine nichts mehr entgegen.⁵

Mittels `xm create name=xen1 -c xmkiste1` wird nun die neue Domain gestartet. Dabei entspricht nun der Name `xen1` nicht dem Hostnamen sondern dem Domainnamen der über `xm` (z.B. `xm list`, `xm shutdown xen1` angesprochen werden kann. Die Option `-c` bedeutet, daß über das aktuelle Terminal sofort auf die neue Domain zugegriffen werden kann. Der letzte Parameter "`xmkiste1`" ist der Name der Datei in `/etc/xen` in der wir die restlichen Einstellungen angegeben haben.

Sollte beim starten der Domain eine Fehlermeldung wie folgende kommen:

```
Error: Error creating domain: (12, 'Cannot allocate memory')
```

dann ist zu wenig Hauptspeicher für die neue Domain frei. Dies kann mit `xm balloon Domain-0 xxx` geändert werden, wobei `xxx` für die Menge des Speichers (in MB) steht, die für die Domain-0 übrigbleiben soll⁶. Jetzt sollte einem ersten Start der neuen Domain nichts mehr im Wege stehen. Nun das ist ja alles schon ganz nett, aber...

⁴Es lassen sich auch beim Starten der Gastsysteme noch Einstellungen mitgeben (`xm help create`).

⁵Das Hostsystem muß natürlich bereits mit dem XEN-Kernel gebootet worden sein und `xend` muß laufen. Zur Vereinfachung legt YAST einen eigenen Punkt im Bütmenü des Hostsystems an.

⁶also Gesamtspeicher - Speicher aller virtuellen Maschinen

wie geht es weiter

Zuerst kann man sich mal die Konfigurationsdatei (hier `xmkiste1`) mal ansehen und einige Parameter ändern. Erste Kandidaten waren hier für mich

- `name` - Hier kann man den Domainnamen festlegen.
- `memory` - Der Speicher für die virtuelle Maschine (in MB). Bei einer Veränderung muß evtl. der Speicher der Domain-0 mit angepasst werden (`xmballoon . . .`). Dies kann auch in der Konfigurationsdatei des grub erfolgen. Besonders wer dann später mit KDE arbeiten will, sollte hier mehr Speicher angeben.
- `nics` - Anzahl der Netzwerkkarten. Standardmäßig wird eine Netzwerkkarte erzeugt.
- `extra` - Zu startender Runlevel. Hier gilt es vor allem zu beachten, daß die unprivilegierten Domains keinen direkten Hardwarezugriff haben und deshalb beim Starten (je nach Runlevel) viele Fehlermeldungen erscheinen können (z.B. `hwclock`). Diese Dienste sollten dann deaktiviert werden. Braucht man nicht zu viele Dienste kann man auch gut auf Runlevel 4 bleiben und die benötigten Dienste aktivieren (`network, sshd, ...`)

Bei den Netzwerkeinstellungen (`hostname, netmask, dhcp`) hat sich für mich bewährt, diese nicht weiter über die Konfigurationsdatei vorzugeben, sondern in der dann laufenden virtuellen Maschine mit YAST genau wie bei einer realen Maschine einzustellen. Beim Testen des Netzwerks gilt es zu bedenken, daß auf dem Host (und dem Gast) die Firewall laufen kann und diese entweder kurzfristig deaktiviert oder gleich richtig eingestellt werden sollte. Betreibt man mehrere Domains und entsprechend mehrere Netzwerkkarten, kann man komplette Netzwerke nachbilden und darin testen, entwickeln,

Den Aufbau der Netzwerke (ob sie Verbindung nach aussen haben oder wie die einzelnen Domains miteinander verbunden sind) kann man sehr einfach mit `brctl` kontrollieren und ändern. Mit `brctl show` sieht man die aktuelle Konfiguration. `brctl help` zeigt wie man neue Bridges erzeugen kann und Netzwerkkarten (echte und virtuelle) daran bindet.

Grafisches

Bisher läuft alles im Text-Modus. Für viele Fälle mag das geüben, aber nicht für alle. Ein einfacher Weg mit der virtuellen Maschine grafischen Kontakt aufzunehmen ist über VNC. Da dies genauso läuft wie mit realen Maschinen hier nur eine Kurzfassung. Auf dem Gastsystem startet man den `vncserver`. Im Verzeichnis `/.vnc` des aufrufenden Users kann man in der Datei `xstartup` einstellen, welcher Windowmanager beim Einloggen gestartet wird.

Auf der Clientseite (hier das Hostsystem) gibt es verschieden Möglichkeiten um auf den Server zuzugreifen. Mögliche Abläufe (10.0.0.111 ist hier die IP des Gastsystems):

- Server (virtuelle Maschine): `vncserver -geometry 1280x1024 :1`
- Client (Möglichkeit 1): Im eigenen Fenster:
`vncviewer 10.0.0.111:1`

- Client (Möglichkeit 2): X-Server starten und Ausgabe umleiten:

```
X vt08 :1 &
export DISPLAY=:1
vncviewer -fullscreen 10.0.0.111:1
```

Man kann natürlich auch per ssh (ohne vnc) auf den Server zugreifen. Möglicher Ablauf:

- Server (virtuelle Maschine): Hier gibt es eigentlich nichts zu tun (ggf. sshd starten)
- Client:

```
X vt08 :1 &
xterm -display :1 &
Im neuen xterm auf vt08:
ssh -X 10.0.0.111
startkde &
```

Es gibt natürlich noch viele Mischformen und andere Möglichkeiten um auf die virtuelle Maschine zuzugreifen, aber mit diesen kann man schon mal ganz gut arbeiten. Weiter sollte man sich noch zum Thema Sicherheit Gedanken machen und natürlich nicht unter root arbeiten wo es nicht unbedingt erforderlich ist. Denn man kann nun auf die virtuelle Maschine wie auf jede andere zugreifen und mit ihr arbeiten. Aber vielleicht ist dieses ja Absicht (Stichwort HoneyPot). Aber dies ist ein anderes Thema und soll hier nicht weiter verfolgt werden.

Was aber, wenn keine Partition mehr frei ist?

Da dies eigentlich in der beiliegenden Doku recht gut beschrieben ist, fasse ich mich hier wieder kurz.

- Erstellen der Datei (hier im Beispiel 2GB groß) ⁷.

```
dd if=/dev/zero of=vm1disk bs=1k seek=2048k count=1
```
- Filesystem erstellen

```
mkfs -t ext3 vm1disk
```
- Filesystem mounten

```
mount -o loop vm1disk /mnt/xen2
```
- Jetzt kann wie oben angegeben weitergemacht werden und nach /mnt/xen2 installiert werden. Es muß dann ebenfalls eine fstab erstellt und per chroot-Shell das Paßwort geändert werden. In der Konfigurationsdatei in /etc/xen muß nur der Disk-Eintrag angepasst werden:

```
disk = ['file:/full/path/to/vm1disk,sda1,w']
```

Nach dem `umount /mnt/xen2` geht es ganz normal weiter.

⁷die aber nicht wirklich belegt werden, sondern sich die Datei erst bei Bedarf vergrößert (`df -h`)

Weitere Infos

- <file:///usr/share/doc/packages/xen>
- <http://www.philipp-benner.de/configs/xen-install.html>
- <http://www.pug.org/index.php/Xen-Installation>
- <http://www.cl.cam.ac.uk/Research/SRG/netos/xen/>