

KVM mit ArchivistaVM in 30 Sekunden

Server-Virtualisierung für die Hosentasche direkt ab USB-Stick

Contents

1	Einleitung	2	4	Automatisierung	12
				4.1	Arbeiten mit dem Auto-Modus 12
				4.2	Telnet-Zugriff bei Problemen 13
				4.3	Anpassungen an initrd.img . . 13
				4.4	Anpassungen an system.os . . 14
2	USB-Stick erstellen	4		5	Abschliessende Bemerkungen 15
2.1	Download ISO-Datei	4		5.1	Lohnt sich das? 15
2.2	KVM-Server ab CD-Rom	4		5.2	Darum ist der KVM-Server ab USB-Stick entstanden . . 15
2.3	Einrichten USB-Stick	5		5.3	Was bringt es in der Praxis . 16
3	Arbeiten mit dem KVM-Server	7		5.4	ArchivistaDMS und mehr . . 16
3.1	Booten des Servers mit ram ramdisk	7		5.5	Vortrag am 26.11.11, linux- day.at 17
3.2	Start mit interner/externer Festplatte	10		5.6	Über den Autor und die ArchivistaBox 17
3.3	Start und Formatieren von Festplatten	10		5.7	Copyright-Notiz 18
3.4	Arbeiten mit Software-Raid	11			
3.5	Arbeiten mit ArchivistaVM	11			

© 26.10.21011 by Urs Pfister, www.archivista.ch

1 Einleitung

Dieser Artikel stellt einen portablen KVM-Server mit ArchivistaVM vor, der ab USB-Stick auf jedem Rechner in ca. 30 Sekunden gestartet und in Betrieb genommen werden kann. Dabei muss keine Software installiert werden, die Software zur Virtualisierung (sprich KVM), ein X- und Web-Server sowie das GUI (Web-Browser) werden komplett in den Hauptspeicher geladen. Die Gäste werden entweder direkt ab dem USB-Stick oder ab einer Festplatte gestartet. Nach dem Starten kann lokal (X-Server mit Web-Browser) oder von einem beliebigen Rechner aus (Web-Browser mit Java-Applets) gearbeitet werden. Zur Einleitung eine Demo, die das Resultat in ca. 30 Sekunden unter www.archivista.ch/avvm11.gif aufzeigt. Weiter seien an dieser Stelle einige Vorteile von ArchivistaVM angeführt:

1. In jedem Haushalt, in jedem Büro, fast an jeder Ecke gibt es Rechner, die eben mal stillstehen. Wohl niemand (selbstverständlich sofern zuvor gefragt wird) wird etwas dagegen haben, wenn bei einem stillstehenden Rechner ein USB-Stick eingeschoben wird, sofern die Innereien (Betriebssystem und Daten) nicht angefasst werden.
2. Weshalb einen Rechner mitschleppen, wenn das gleiche mit einem USB-Stick zu haben ist? Jeder Rechner, ein jedes Notebook bringt heute zumindest eine DualCore CPU und 2 GByte RAM mit. Mehr benötigt die nachfolgend beschriebene Lösung im Minimum nicht. Wirklich Spass macht die Lösung freilich ab einer QuadCore-CPU und 4 GByte RAM.
3. Ein virtualisiertes OS soll ab Datensicherung gestartet werden. Weshalb ein komplettes System aufspielen, die Daten zurückspielen, um dann erst den Job erledigen zu können, wenn ein VM-Server in 30 Sekunden mit einem USB-Stick und der Sicherungsplatte zu haben ist?
4. Ein KVM-Server ab USB-Stick im RAM arbeitet extrem schnell. Dadurch, dass die Software ins RAM installiert wird, dauert das Aufstarten nicht länger als wenn von Festplatte gestartet würde. Wer einmal RAM-basiert gearbeitet hat, wird keine SSD-Platte mehr kaufen wollen, um die 2 GByte Durchsatz pro Sekunde gibt es mit keiner SSD-Platte.
5. Warum KVM und ein unbekanntes Linux verwenden, es gibt doch viele Lösungen für die Virtualisierung? Weil keine andere Lösung on-the-fly in ca. 30 Sekunden in Betrieb genommen werden kann, weil kein Setup benötigt wird, weil die Sourcen offengelegt sind, ein komplettes Debian zur Verfügung steht, oder weil RAID-Systeme mit bis zu 24 Platten unterstützt werden, um nur einige Gründe zu nennen.
6. Geht die Virtualisierung in der Wolke (Cloud) nicht viel einfacher? Spätestens wenn die Wolke mal das Backup vergisst, ist die Einfachheit (inkl. Daten) weg. Ein USB-Stick kostet nicht alle Welt, das Backup erfolgt lokal und performant, das Kopieren ganzer VM-Landschaften

übers Web dauert viel länger. Zudem, mit USB-Stick und Festplatte(n) lässt sich um Faktoren schneller arbeiten, mit dem integrierten DRDB gar in Echtzeit.

7. Ist ArchivistaVM auf die Virtualisierung beschränkt? Nein, neben dem KVM-Server stehen viele (Recovery)-Tools zur Verfügung, auch eine Build-Umgebung (ein neuer Kernel ist im RAM in Rekordzeit gebaut) ist integriert. Zudem gibt es die Varianten ArchivistaDMS (Dokumenten-Management) und ArchivistaDesktop (OpenOffice, Gimp, Scribus, Kile etc.). Trotzdem wird primär ArchivistaVM vorgestellt, weil es deutlich weniger RAM als die anderen beiden Versionen benötigt.

Nachfolgend wird aufgezeigt, wie ein ArchivistaVM-Stick erstellt wird und wie mit dem System gearbeitet werden kann. Im dritten Teil werden die wichtigsten Optionen vorgestellt, ehe im letzten Teil versucht wird, ein (vorläufiges) Fazit zu ziehen. Um diesen Artikel durchzuarbeiten, wird ein USB-Stick mit minimal 8 GByte und ein Debian/Ubuntu benötigt. Alternativ könnte auch ArchivistaVM ab CD-ROM verwendet werden; der Weg über Debian/Ubuntu dürfte für Linux-Anwender/innen aber meist schneller sein, weil dabei keine CD-ROM erstellt werden muss.

2 USB-Stick erstellen

Nachfolgend wird das Erstellen eines USB-Sticks unter Debian/Ubuntu beschrieben.

2.1 Download ISO-Datei

Die ISO-Datei findet sich unter www.archivista.ch/avvm-64bit.iso. Diese Datei ist ca. 350 MByte gross.

2.2 KVM-Server ab CD-Rom

Wem das nachfolgende Vorgehen zu kompliziert ist, der kann dennoch mit ArchivistaVM arbeiten, denn der portable KVM-Server ab USB-Stick arbeitet auch ab CD-ROM. Allerdings passen weder CD-ROM noch CD-Laufwerk in die Hosentasche, sodass ich persönlich den USB-Stick immer bevorzugen würde.

Sofern mit einer CD-ROM gearbeitet wird, kann diese nach dem Brennen direkt gestartet werden. Dazu ist beim Starten wie untenstehend dargestellt **'ram'** einzugeben:



ArchivistaBox 64Bit — in 100 seconds
to a ready to use server infrastructure

```
Press Return to install ArchivistaBox Environment.  
ArchivistaVM (build 2011-04-26) - http://www.archivista.ch  
boot: ram
```

Mit Drücken der Enter-Taste wird das Setup von ArchivistaVM durchgeführt. Der Vorgang ab CD-ROM dauert etwas länger, mehr als eine Minute dürfte aber nicht notwendig sein, um ArchivistaVM zu starten. Nach dem Starten kann die CD-ROM entfernt werden. Die CD wird für das weitere Arbeiten nicht mehr benötigt.

Wichtig: Ohne die Option **'ram'** wird das Setup-Programm von ArchivistaVM gestartet, um das System auf der Festplatte einzurichten. Spätestens bei der Mitteilung, dass alle Daten zerstört werden, sollte auf **'Nein'** geklickt werden...

2.3 Einrichten USB-Stick

Nach dem Download unter Debian/Ubuntu ist eine Konsole (xterm) zu öffnen und mit **sudo su** - wird zu root gewechselt. Zunächst werden die Pakete lilo und syslinux benötigt. Diese können mit **apt-get install lilo syslinux** installiert werden. Danach kann der USB-Stick eingelegt werden. Wichtig dabei ist, die richtige Adresse des Sticks zu finden. Dazu kann **dmesg** verwendet werden.

```
archivista:~# dmesg

usb 7-3: new high speed USB device using ehci_hcd and address 15
scsi20 : usb-storage 7-3:1.0
scsi 20:0:0:0: Direct-Access    USB          Flash  DISK          1100 PQ: 0 ANSI: 0
CCS
sd 20:0:0:0: Attached scsi generic sg5 type 0
sd 20:0:0:0: [sdb] 1957888 512-byte logical blocks: (1.00 GB/956 MiB)
sd 20:0:0:0: [sdb] Write Protect is off
sd 20:0:0:0: [sdb] Mode Sense: 43 00 00 00
sd 20:0:0:0: [sdb] Assuming drive cache: write through
sd 20:0:0:0: [sdb] Assuming drive cache: write through
sdb: sdb1
sd 20:0:0:0: [sdb] Assuming drive cache: write through
sd 20:0:0:0: [sdb] Attached SCSI disk
archivista:~#
```

Auf den zuletzt angezeigten Zeilen findet sich die Adresse des Sticks. Nun kann der Stick mit **'cfdisk'** vorbereitet werden, in unserem Beispiel ist **'cfdisk /dev/sdb'** zu verwenden.

Wichtig: Die nachfolgend verwendeten Einstellungen (dev/sdb) können übernommen werden, wenn genau eine Festplatte im System vorhanden ist und der USB-Stick als zweites Gerät erscheint. Bei zwei internen Festplatten wäre /dev/sdc zu verwenden, bei drei Platten /dev/sdd und so weiter. Noch wichtiger: Wird eine falsche Gerätenummer verwendet, wird ein Datenverlust zu beklagen sein – und der USB-Stick für ArchivistaVM wird deswegen trotzdem nicht eingerichtet.

Zurück zum Stick, dieser muss bootbar sein und ist mit FAT16 (Typ 06) zu formatieren und es sind drei Partitionen einzurichten:

1. Partition Fat 16 (Typ 06) mit 1024 MByte
2. Partition Swap (Typ 82) mit 2048 MByte (empfohlen)
3. Partition ext3 (Typ 83) mit dem übrigem Platz

Die nachfolgende Abbildung verdeutlicht dies:

```

cfdisk (util-linux-ng 2.13.1.1)

Disk Drive: /dev/sdb
Size: 7818182656 bytes, 7818 MB
Heads: 241 Sectors per Track: 62 Cylinders: 1021

```

Name	Flags	Part Type	FS Type	[Label]	Size (MB)
sdb1	Boot	Primary	FAT16		1025.15
sdb2		Primary	Linux swap / Solaris		2050.29
sdb3		Primary	Linux ext3		4735.54

Nachdem der Stick initialisiert wurde, können über die Konsole die restlichen Schritte durchgeführt werden:

```

mkdosfs /dev/sdb1
lilo -M /dev/sdb
syslinux /dev/sdb1
mkdir /in
mkdir /out
mount -o loop avvm-64bit.iso /in
mount /dev/sdb1 /out
cp /in/* /out
umount /in
umount /out

```

Nun sollte der Stick booten. Damit kann der KVM-Server gestartet werden. Zunächst sollte noch eine Daten-Partition eingerichtet werden. Dies wird mit **'mkfs.ext4 /dev/sdb3'** erreicht.

Ob es sich lohnt, eine Swap-Partition auf einen USB-Stick zu legen, dies muss anhand der Qualität des Sticks entschieden werden. Bei preisgünstigeren Modellen sollte es eher nicht gemacht werden. Die Swap-Partiton wird mit **'mkswap /dev/sdb2'** eingerichtet. Mit oder ohne Swap, der USB-Stick steht nun zum Arbeiten bereit.

3 Arbeiten mit dem KVM-Server

Beim Starten ist zu beachten, dass der RAM-Modus aktiviert werden muss. Dazu ist beim ersten Start-Bildschirm minimal **'ram'** einzugeben.

Wichtig: Ich weiss, ich wiederhole mich, aber ohne die Option **'ram'** wird das Setup-Programm von ArchivistaVM gestartet, um das System auf der Festplatte einzurichten. Wer dabei unbedacht auf 'Ja' klickt, riskiert, dass das Setup-Programm die gewählte Festplatte zerstört.

3.1 Booten des Servers mit ram ramdisk

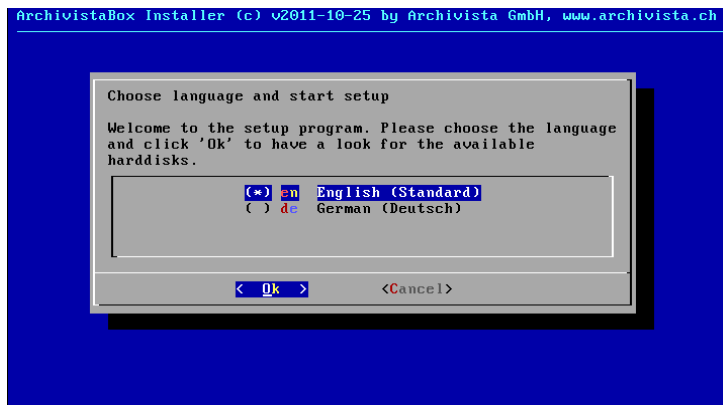
Bei diesen beiden Optionen wird der USB-Stick so gestartet, dass die virtualisierten Instanzen direkt auf der dritten Partition des USB-Sticks gespeichert werden. Das gesamte System (KVM und Images) liegt auf dem gleichen Stick, es kann hier vom KVM-Server für die Hosentasche gesprochen werden. Damit einzig mit dem USB-Stick gearbeitet werden kann, ist beim ersten Prompt wie nachfolgend dargestellt **'ram ramdisk'** einzugeben:



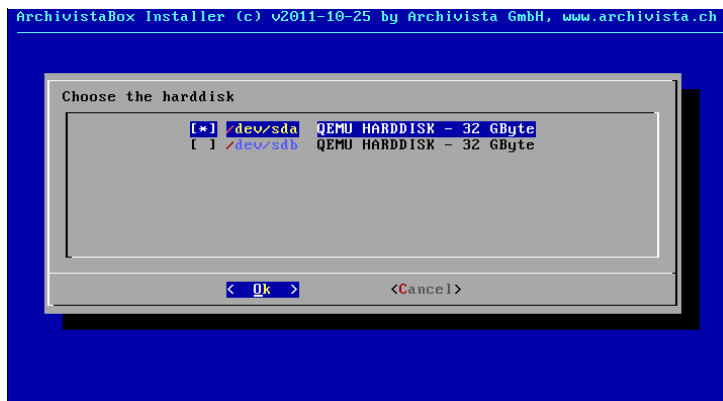
ArchivistaBox 64Bit — in 100 seconds
to a ready to use server infrastructure

```
Press Return to install ArchivistaBox Environment.  
ArchivistaVM (build 2011-04-26) - http://www.archivista.ch  
root: ram ramdisk
```

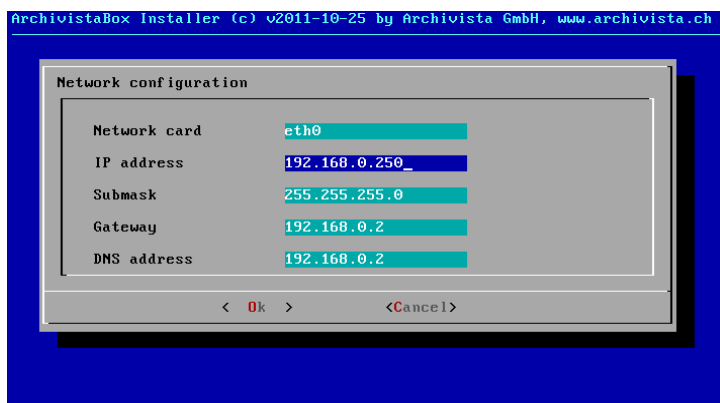
Das System wird nun eingerichtet. Nach einigen Sekunden startet das Setup-Programm. Die erste Abfrage ist zu bestätigen.



Nach zwei drei Sekunden erscheint eine Liste mit den vorhandenen Festplatten. Es kann dabei irgendeine Festplatte ausgewählt werden. Im Modus 'ram ramdisk' erfolgt kein Zugriff auf die internen Festplatten. Es wird einzig auf diese Platte zurückgegriffen, sofern entweder eine CD-ROM oder die Partitionen auf dem USB-Stick nicht eingebunden werden können (z.B. weil sie gar nicht vorhanden sind).



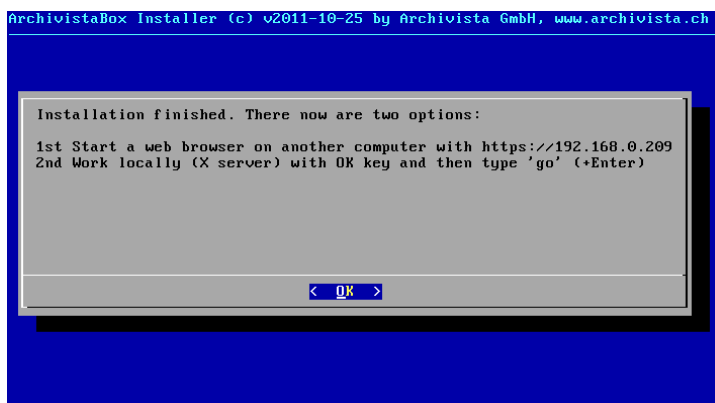
Nach ca. 10 bis 15 Sekunden erfolgt eine Abfrage für die gewünschten Netzwerk-Einstellungen. ArchivistaVM verlangt nach einer fixen IP-Adresse (ein Server für die Virtualisierung erscheint mir nicht unbedingt sehr sinnvoll ohne fixe IP-Adresse). Dabei sind die Daten analog zur untenstehenden Abbildung einzugeben:



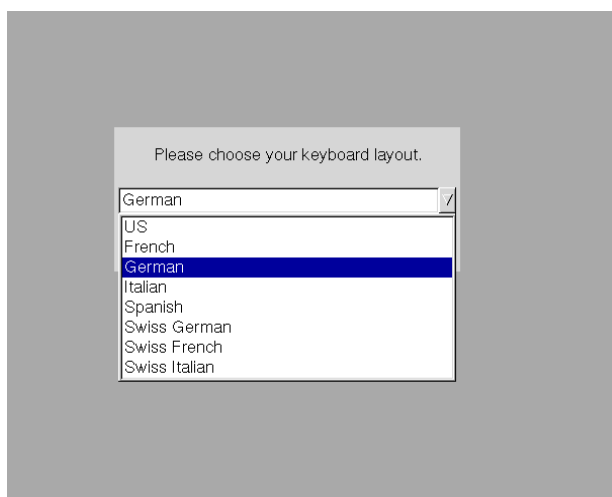
Wichtig: Sollte keine eindeutige Netzwerkadresse vorhanden sein, so kann beim Computer das Netzwerkkabel abgehängt werden. Dann kann die voreingestellte Adresse 192.168.0.250

problemlos verwendet werden. Ein Arbeiten mit ArchivistaVM ist auch ohne Netzanschluss lokal möglich.

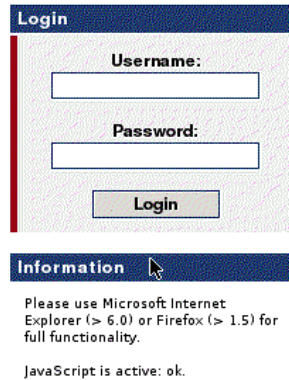
Nach weiteren ca. 5 Sekunden erscheint eine Meldung, dass das System nun zum Arbeiten zur Verfügung steht (System bitte nicht neu starten, das wäre nicht im Sinne einer RAM-basierten Installation). Es kann nun entweder mit einem externen Rechner per Web-Browser auf ArchivistaVM zugegriffen werden oder durch Eingabe von 'go' (mit Enter-Taste) ein X-Server gestartet werden.



Nach etwa 2 Sekunden kann die Tastatur festgelegt werden:



Danach wird umgehend der Browser gestartet; zum Anmelden ist beim Benutzer 'root' und beim Passwort 'archivista' einzugeben:



Das Werk ist vollendet. Unter 213.160.42.154/d2/help/node71.html findet sich eine Anleitung zum Arbeiten mit dem KVM-Server alias ArchivistaVM.

3.2 Start mit interner/externer Festplatte

Wer mit einer vorbestimmten internen/externen Festplatte arbeiten möchte, kann mit den Parametern 'ramswap./dev/xxx' und 'ramdata./dev/yyy' sowohl die gewünschte Swap- als auch Daten-Partition bestimmen. Dazu ein Beispiel:

```
ram ramswap./dev/sdc2 ramdata./dev/sdd1
```

Mit den obenstehenden Einstellungen wird die Swap-Partition auf der dritten Platte und der zweiten Partition gestartet, die Daten-Partition erfolgt ab der vierten Festplatte und der ersten Partition.

3.3 Start und Formatieren von Festplatten

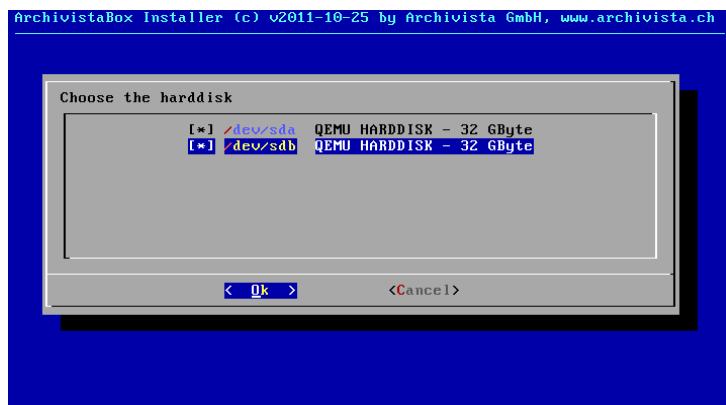
Mit der Option 'ram formathd' kann (bei leeren Festplatten) erzwungen werden, dass eine beim Start gewählte Platte formatiert und für das Arbeiten mit ArchivistaVM eingerichtet wird.

Wichtig: Mit 'ram formathd' können nur leere Festplatten eingerichtet werden. Bereits formatierte Platten werden nicht neu eingerichtet. 'ram formathd' sollte dennoch vorsichtig eingesetzt werden. Der Entscheid, ob formatiert wird oder nicht, erfolgt aufgrund der Partitionstabelle. Sollte diese defekt sein, so wird die Platte formatiert; auch wenn sich eventuell noch gewisse Daten restaurieren liessen.

3.4 Arbeiten mit Software-Raid

Virtualisierung ist nicht nur eine Frage der Prozessoren oder des Hauptspeichers, in erster Linie wichtig für die Geschwindigkeit ist der Zugriff auf die Festplatten.

Mit der Option 'ram formathd' bei leeren Festplatten kann auch ein Software-Raid eingerichtet, sofern beim Auswählen der Festplatten mehrere Festplatten ausgewählt werden:



Bei zwei Festplatten wird Raid1 eingerichtet, bei vier, sechs oder noch mehr Platten erfolgt das Einrichten mit Raid10. Beim Einrichten eines Raids erfolgt eine Bestätigungsabfrage. Soll Raid0 eingerichtet werden, muss bereits beim Aufstarten '**ram formathd raid0**' eingegeben werden; wobei ich an dieser Stelle ausdrücklich gesagt haben möchte, dass '**raid0**' auf eigene Gefahr verwendet wird.

Wichtig: Derzeit können nur von ArchivistaVM eingerichtete Software-Raids verwendet werden. Allerdings könnte ein Raid auf der Konsole nachträglich von Hand zugeschaltet werden. Die Treiber aller gängiger Hardware-Raid-Kontroller sind ebenfalls enthalten, wobei ArchivistaVM nicht mit sämtlichen Karten getestet werden konnte. Allenfalls müssten diese von Hand geladen werden, damit die Partitionen eingebunden werden können.

Alternativ zum direkten Einrichten eines Raid-Verbundes im RAM-Speicher kann das System zunächst auch komplett auf die Festplatte gespielt werden. Dies erfordert zwar einen Neustart, dafür kann ArchivistaVM danach wahlweise mit oder ohne RAM-Modus eingesetzt werden. Um ArchivistaVM auf der Festplatte einzurichten, ist beim Starten auf die Eingabe der Option '**ram**' zu verzichten. Selbstverständlich ist beim zweiten Start die Option '**ram**' einzugeben, da ansonsten das System ab der Festplatte gestartet wird.

3.5 Arbeiten mit ArchivistaVM

Dieses Skript beinhaltet keine Anleitung für das Arbeiten mit ArchivistaVM. Dazu findet sich eine Anleitung unter 213.160.42.154/d2/help/node71.html.

4 Automatisierung

Der Installer von ArchivistaVM wurde in Perl (objektorientiert) realisiert. Das Anpassen des Setup-Programmes dürfte für einigermassen kundige Perl-Programmierer/innen keine grossen Schwierigkeiten darstellen. Bevor jemand die Datei `initrd.img` und das Setup-Programm `install.pl` selber anzupassen gedenkt, sei auf die bereits vorhandenen Start-Optionen verwiesen.

4.1 Arbeiten mit dem Auto-Modus

ArchivistaVM arbeitet beim Booten mit Syslinux bzw. ab CD-ROM mit Isolinux. Dazu existieren die beiden Dateien `'syslinux.cfg'` sowie `'isolinux.cfg'` direkt im Hauptverzeichnis der ISO-Datei bzw. der ersten Partition des USB-Sticks. Betrachten wir zunächst einmal den Inhalt:

```
DEFAULT linux
TIMEOUT 100
```

```
PROMPT 1
DISPLAY boot.msg
```

```
label linux
kernel vmlinuz
APPEND initrd=initrd.img quiet ramdisk_size=81920
```

```
label ram
kernel vmlinuz
APPEND initrd=initrd.img quiet ram ramdisk_size=2097152
```

```
label box244
kernel vmlinuz
APPEND initrd=initrd.img quiet ram ramdisk_size=4194304
      keyboard.de_CH lang.de auto ip.192.168.2.244
      submask.255.255.255.0 gw.192.168.2.1 dns.192.168.2.1 ramdisk
```

Wichtig: Die Zeile APPEND hat auf einer Zeile zu stehen, diese wurde (damit die gesamte Zeile abgebildet werden kann) obenstehende zweimal umbrochen.

Einige Informationen zu dieser Datei. Zunächst wird beim Booten zehn Sekunden gewartet, bis der Start erfolgt. Dies ist etwas missverständlich, aber abgerechnet wird in Zehntel-Sekunden. Wenn Sie eine Minute warten möchten, dann wäre TIMEOUT auf 600 zu setzen.

Mit DEFAULT wird festgelegt, welche Definition ohne weitere Eingaben gestartet wird. Wenn z.B. erreicht werden möchte, dass immer im RAM-basierten Modus gestartet wird, muss bei Default **'ram'** festgelegt werden.

Spannend ist weiter die Variante 'box244'. Dabei handelt es sich um ein automatisierte Startup, welche das System automatisiert hochfährt. Zentral ist dabei 'auto'. Damit wird erreicht, dass das Setup-Programm keine Fragen mehr stellt. Und in diesem Modus (kombiniert mit 'box244') ist das System in 30 Sekunden ab einem einigermaßen schnellen USB2-Stick eingerichtet.

Es gibt weitere Parameter, diese finden sich direkt im Setup-Programm 'install.pl' unter der Funktion 'sub cmdline'.

4.2 Telnet-Zugriff bei Problemen

Am Ende des Setups wird ein SSH-Server gestartet, sodass auf ArchivistaVM problemlos auf der Konsole zugegriffen werden kann. Was aber, wenn das System vor dem Starten des SSH-Servers hängenbleibt? In einem solchen Falle kann mit **'ram'** das Setup-Programm gestartet werden. Anstatt den Anweisungen danach Folge zu leisten, wird das Programm abgebrochen. Nun kann auf der Konsole die Netzwerkkarte und ein telnet-Server eingerichtet werden. Dies geht so:

```
ifconfig eth0 192.168.0.250
route add default gw 192.168.0.2
telnetd -l /bin/login
```

Danach kann von einem entfernten Rechner aus mit telnet auf die Konsole zugegriffen werden, beim Anmelden ist das Passwort 'archivista' zu verwenden.

4.3 Anpassungen an initrd.img

In der Datei initrd.img befindet sich das gesamte Mini-Linux. Beim Hochfahren des Systems wird zunächst dieses Mini-System eingerichtet, ehe über den Stick (bzw. CD-ROM) auf 'system.os' zugegriffen wird. Die Datei 'system.os' enthält ein komplettes Debian-System bzw. ArchivistaVM. Beim Hochfahren des Servers wird in einem zweiten Schritt dieses System im RAM eingerichtet, und zwar äusserst speditiv. Pro 1 GByte an Software sind etwa 10 Sekunden Zeit notwendig. Beide Dateien können beliebig angepasst werden. initrd.img wird wie folgt ausgepackt:

```
zcat initrd.img | cpio -iv
```

Dabei wird das gesamte System im aktuellen Verzeichnis ausgepackt. Es können nun beliebige Änderungen am System vorgenommen werden. Für das Neupacken der Datei ist der folgende Einzeiler zu verwenden:

```
find . | cpio -o -H newc | gzip -c >../isolinux/initrd.img
```

4.4 Anpassungen an system.os

In der Datei system.os befindet sich das gesamte Linux-System von ArchivistaVM. Dazu ist zu sagen, dass bei ArchivistaVM nicht Debian- um Debian-Paket installiert wird (dieser Vorgang würde viel zu lange dauern), sondern dass das gesamte System mit 'unsquashfs' entpackt wird.

```
unsquashfs -dest /os /tmp/cd/system.os
```

Das Packen wird mit dem folgenden Befehl erreicht:

```
mksquashfs / /inst/system.os -noappend -e /inst /proc /sys"
```

5 Abschliessende Bemerkungen

5.1 Lohnt sich das?

Die Frage ist etwas rethorisch gemeint. Wäre ich nicht der Überzeugung, dass es sich lohnen würde, hätte ich den KVM-Server ab dem USB-Stick wohl nicht entwickelt. Allerdings gibt es dazu auch eine Story, und diese sei hier angeführt.

5.2 Darum ist der KVM-Server ab USB-Stick entstanden

Vor einigen Wochen bin ich zusammen mit meiner Familie in den Urlaub gefahren. Normalerweise habe ich für diese Fälle ein QuadCore-Notebook bei mir. Leider hatte ich das Netzteil dazu im Büro vergessen. Es galt rasch zu entscheiden: 1. Nächsten Zug nehmen (vier Stunden später) und Netzteil im Büro abholen oder 2. Erwerb eines preiswerten Notebooks am Hauptbahnhof in Zürich. Für die Familie war der Fall klar. Ich durfte dabei in 10 Minuten zwischen HP und Acer wählen, mit AMD-CPU habe er keines, meinte der Verkäufer. HP war dann doch nicht da und das Acer hatte letztlich sogar eine AMD-CPU.

Darauf installiert war ein Windows. Ich hatte nun drei Möglichkeiten: A) Das neue Notebook mit der ArchivistaBox (Stick war dabei) bespielen und darauf hoffen, dass WLAN arbeiten würde. Plan B) war, eine LiveCD am Kiosk um die Ecke zu erwerben. Mein Notebook hatte aber kein CD-ROM-Laufwerk. Folglich fiel Plan B) ins Wasser bzw. die österreichischen Alpen. blieb C) und das hiess mit Windows zu arbeiten. Noch im Zug hab ich das Notebook ausgepackt und in Betrieb genommen. Etwa 45 Minuten nach Zürich (knapp vor Sargans) konnte ich mit Windows arbeiten, bis dahin wurde es mit ca. 5 Bootvorgängen eingerichtet. Immerhin, der Akku hatte danach noch ca. 30 Prozent Kapazität.

Im Wissen, dass mein System in ca. 1 Minute (inkl. OpenOffice) aufgespielt gewesen wäre, brannte es mir schon irgendwie unter den Nägeln, Windows ins Nirwana zu befördern. Nur eben, die Geschichte mit den WLAN-Treibern, nicht unbedingt ein Glanzstück bei einem Debian-basierten System.

Was macht ein Benutzer, während er auf Bestätigungsmeldungen wartet? Er denkt daran, dass das Notebook-Leben besser sein könnte bzw. müsste. Und irgendwann war Plan D) geboren. Ich wollte mein System auf einem USB-Stick haben. Und ich wollte es ab diesem Stick zum Leben erwecken, ohne die inneren Werte des Gerätes zerstören zu müssen. Natürlich gibt es im Web Anleitungen für Live-Systeme ab USB-Sticks. Aber das ist/war nicht das, was ich

wollte. Es sollte ein System her, dass nicht nur mein System beherbergen konnte, sondern gleich alle Betriebssysteme zusammen und das schön virtualisiert.

Und damit der USB-Stick nicht schon in den Ferien schlapp macht, wollte ich mein System im RAM laufen lassen. Als Vorlage diente mir dabei TinyCore-Linux, allerdings konnte ich mich mit dem Paket-Management-System und mit 32-Bit von TinyCore-Linux nie so richtig anfreunden. Wofür gibt es 64-Bit und 8 oder noch mehr GByte RAM? Damit sei gesagt, dass ein vergessenes Netzteil und die österreichischen Alpen für die Entstehung des KVM-Servers ab USB-Stick herzuhalten haben.

5.3 Was bringt es in der Praxis

Offen gestanden ist es für ein abschliessendes Fazit zu früh. Persönlich arbeite ich nun seit einigen Wochen mit dem System; das System läuft unter einer Bedingung sehr stabil und extrem schnell. Was mit einem RAM-basierten System nicht gemacht werden sollte, ist die System-Partition mutwillig zu füllen. Ist das RAM erst mal aufgebraucht, ist die RAM-Platte nämlich weggehaucht. Das ist mir bisher zwar erst mutwillig passiert, aber ausgeschlossen kann es derzeit nicht.

Für einen produktiven Server-Betrieb (KVM-Virtualisierung) sollten zwei weitere Dinge beachtet werden. Einmal ECC-RAMs (bei AMD-Prozessoren laufen die normalen ECC-RAMs problemlos auf jedem Desktop-Board) und weiter sollte ein Notstrom-Gerät (USV) zur Verfügung stehen. Beide Punkte sind aber bei einem Server-Betrieb ohnehin zu empfehlen.

Der hier vorgestellte KVM-Server kann mit dem gleichen Speed ab dem USB-Stick installiert und gestartet werden, wie wenn das System von der Platte gestartet würde. Wenn ich daran denke, dass ich meinen KVM-Server zwischen Zürich und Sargans sicher gut und gerne über 50 Mal aufsetzen hätte können (wohl eher könnte, einmal reicht ja schliesslich), dann wundert es mich, weshalb das Einrichten eines Windows-Systems (das ja bereits auf der Festplatte vorinstalliert ist) plus/minus 45 Minuten Zeit in Anspruch nimmt. So gesehen gehört Windows definitiv in den virtualisierten Käfig, darin lässt es sich ab USB-Stick oder Platte auch schnell hochfahren, genau wie all die anderen Systeme, die ich ab und wann gerne verwende.

5.4 ArchivistaDMS und mehr

In diesem Artikel wurde ArchivistaVM vorgestellt. Ebenfalls existieren daneben ArchivistaDMS (Dokumenten-Management) sowie ArchivistaDesktop (DMS und Desktop-Applikationen). Alle drei Systeme enthalten den KVM-Server für die Virtualisierung und alle drei ISO-Dateien können auf den Stick gepackt und wie obenstehend im RAM aufgestartet werden. Bei ArchivistaDMS sind minimal 4 GByte RAM erforderlich, beim ArchivistaDesktop sind plus/minus 8

GByte für ein Arbeiten vorzuhalten. Dafür können direkt im RAM OpenOffice, Gimp, Scribus, Kile und viele weiteren Desktop-Applikationen gestartet werden.

5.5 Vortrag am 26.11.11, linuxday.at

Mehr zu ArchivistaVM gibt es am 26. November 2011 in Dornbirn am 13. LinuxDay.at. Dort zeigen wir gerne all unsere Lösungen, es können USB-Sticks mitgebracht werden, um sie mit dem KVM-Server zu bespielen, und auch einen Vortrag zum grossen Bruder, zum SwissRocket-Cluster gibt es. Der Vortrag lautet:

5.5.1 Hochverfügbare Virtualisierung mit KVM, DRBD und ArchivistaVM

Beginn: 26.11.2011 13:00, Referent: Urs Pfister, Zielgruppe: Profi, Raumnummer: Aula

Der Vortrag stellt die Open Source Lösung Archivista SwissRocket vor, mit der hochverfügbare Virtualisierung in 10 bis 15 Minuten aufgebaut werden kann. Dabei werden die Komponenten KVM, DRBD und ArchivistaVM vorgestellt. Der Vortrag ist in zwei Teile unterteilt. Zunächst wird ein Cluster aus der Sicht des Anwenders aufgebaut. Im zweiten Teil erfolgt ein Blick hinter die Kulissen der Komponenten. Gemachte Erfahrungen (positive wie negative) bei realisierten Projekten runden den Vortrag ab.

Weitere Informationen zum linuxday.at finden sich unter www.linuxday.at

5.6 Über den Autor und die ArchivistaBox

Mit 16 den ersten Computer (CPC464), mit 30 die eigene Firma, kurz darauf zu Linux hinübergewechselt. Und seit dieser Zeit gilt meine Leidenschaft Open Source und allem was dazugehört. Zur Hauptsache beschäftigt mich die Weiterentwicklung der ArchivistaBox, deren Sourcen unter der GPLv2-Lizenz stehen.

Im Jahre 2011 hat die ArchivistaBox den Swiss Open Source Award in der Kategorie Special gewonnen, siehe dazu auch www.ossaward.ch. Jury Mitglied Matthias Günther (IGE) meint zu Archivista: *Archivista war in der Schweiz ein Pionier im Bereich der Kombination Open Source und Businesslösung. Mit der zusätzlichen Bündelung mit Hardware (und dem 'Recyclen' alter Hardware) ist der Firma Archivista ein Produkt gelungen, dessen lange Lebensdauer und Erfolg exemplarisch zeigt, wie Open Source Software in geschäftskritischen Bereichen verwendet werden kann.*

5.7 Copyright-Notiz

Das Copyright liegt beim Autor. Auf Nachfrage (bzw. nach erfolgter Zustimmung durch den Autor) darf dieser Text publiziert werden, sofern der Text komplett, kostenfrei (Unkostenbeitrag ausgeschlossen) und ohne im Inhalt platzierte Werbung publiziert wird.