

ArchivistaVM 2.0: In zwei Minuten zur Virtualisierung mit KVM

ArchivistaBox 64 Bit: Virtualisierung und DMS aus einem Guss

Contents

1 Ausgangslage	2	5 Eingesetzte Technologie	8
		5.1 Eigene Kernel	8
2 Ziele ArchivistaBox 64 Bit	3	5.2 Debian mit in Board	8
2.1 Einfach keine Konsole	3	5.3 Installation über RAM	8
2.2 Setup ab CD/Stick	3	5.4 Setup-Programm in Perl	8
2.3 Updates ohne Backups	3	5.5 Anpassbares Setup-Programm	8
2.4 Zwei Instanzen	3		
2.5 Cluster und mehr	4	6 Automtisierte Setups	9
2.6 Backup-Strategien	4	6.1 Beispiel avbox1	9
		6.2 Beispiel avbox2	10
3 Live-Demo	5	7 Stand der Entwicklung	11
3.1 Hardware 2009	5	7.1 Open Source != kostenfrei	11
3.2 Hardware 2010	5	7.2 Finanzierte Erweiterungen	11
3.3 Verwendete Hardware	6		
3.4 Cluster mit Crossover-Kabel	6	8 Zukunft und heute	13
3.5 Installationszeiten	6	8.1 Erweiterungen in Planung	13
4 ArchivistaVM ist eine Box	7	9 Stabilität und Download	14

© 27.11.2010 by Urs Pfister, www.archivista.ch

1 Ausgangslage

Das Projekt ArchivistaVM ist mehr durch Zufall denn geplant entstanden. Kurz nochmals die wichtigsten Punkte, weshalb wir heute ArchivistaVM 2.0 vorstellen können.

- 2001: Vmware Workstation
- 2007: Ganeti, Vortrag an linuxday.at (Michael Hanselmann)
- 2008: KVM auf ArchivistaBox (32 Bit)
- 2009: Kundenprojekt mit Proxmox, Fork ArchivistaVM
- 2010: Vortrag: Vergleich KVM, XEN, Vmware
- 2010: linuxday.at: Vorstellung ArchivistaBox 64 Bit mit ArchivistaVM

Nun hätten wir auch gleich das erste Problem. Der Titel ist im Prinzip bereits wieder überholt. So werden wir nachfolgend von der ArchivistaBox 64 Bit sprechen. ArchivistaVM ist dabei eine Komponente der ArchivistaBox 64 Bit.

2 Ziele ArchivistaBox 64 Bit

Bei der Entwicklung der neuen Lösung waren für uns die nachfolgenden Punkte zentral.

2.1 Einfach keine Konsole

Virtualisierung muss einfach sein (analog zur ArchivistaBox). Es kann nicht sein, dass die Virtualisierung nur für (hoch)spezialisierte Linux-Profis installier- und wartbar ist. Die alte Lösung, ArchivistaVM 1.3 hatte zwar einen grafischen Installer, aber die Installation dauerte relativ lange (ca. 20 bis 30 Minuten) und danach stand einzig die Konsole zur Verfügung. Ein späteres Update musste zwingend auf der Konsole (`apt-get install ..`) ausgeführt werden. Wir wollten mit der ArchivistaBox 64 Bit eine einfachere Lösung realisieren.

2.2 Setup ab CD/Stick

Einfachstes Setup ab CD/Stick. Wir denken, dass die Installation mit einer CD/DVD bzw. mit einem USB-Stick derzeit am einfachsten zu bewerkstelligen ist. Insbesondere die Installation ab USB-Stick fehlte uns. So mussten wir bei ArchivistaVM 1.x jeweils selber erst eine CD-ROM brennen, dann ein externes Laufwerk an den Rechner anhängen, ehe wir die Installation durchführen konnten.

2.3 Updates ohne Backups

Die bisherige Lösung ArchivistaVM 1.x bzw. Proxmox 1.x formatieren bei der Installation immer die gesamte Festplatte. Dies darf nicht sein. Einmal aufgespielt, sollten keine Daten 'vernichtet' werden können. Das Vorgehen, das bei der ArchivistaBox 64 Bit zum Einsatz kommt, sieht folgendermassen aus: 1. Bei einer unformatierten Festplatte wird diese komplett für die ArchivistaBox 64 Bit bereitgestellt (d.h. formatiert). Bei einem späteren Update wird nur jeweils die System-Partition formatiert und neu bespielt.

2.4 Zwei Instanzen

Bereits bei der ArchivistaBox 32 Bit hat es sich bewährt, dass wir das System immer doppelt verfügbar halten. Eine Instanz wird dazu verwendet, das aktuelle System zu beherbergen. In die jeweils zweite Instanz wird das Update eingespielt. Danach erfolgt ein Neustart und die zuvor zweite Instanz wird zur aktuellen, während die zuvor aktuelle Instanz nunmehr inaktiv

ist. Bei Problemen mit der aktiven Installation kann zudem immer auf die vorangehende Installation zurückgegriffen werden. Dadurch ist auch sichergestellt, dass sollte das Update Fehler enthalten (z.B. korrupter Datenträger) immer eine lauffähige Instanz zur Verfügung steht.

2.5 Cluster und mehr

Die Virtualisierung mit einer Maschine ist ganz nett, aber wirklich spannend wird es erst, wenn mehrere Rechner zu einem Verbund (Cluster) zusammengehängt werden können. Dies ist selbstverständlich mit ArchivistaVM möglich, wobei im Moment (Ende November 2010) dies noch der einzige Moment ist, wo auf die Konsole zurückzugreifen ist. Auf dem ersten Rechner (Master) ist der folgende Befehl einzugeben:

```
pveca -c
```

Mit `ifconfig` sollte danach die aktuelle IP-Adresse festgestellt (gehen wir für das untenstehende Beispiel von 192.168.0.250 aus) werden. Anschliessend auf den weiteren Rechnern (Nodes) jeweils den folgenden Befehl eingeben:

```
pveca -a -h 192.168.0.250
```

Dabei muss kurz auf dem Master (in unserem Beispiel 192.168.0.250) zugegriffen werden, d.h. es wird das Passwort abgefragt.

2.6 Backup-Strategien

Sicherungen (Backups) sind unentbehrlich für einen professionellen Betrieb einer Virtualisierungsumgebung. Als Empfehlung gilt dabei, dass pro Instanz und Tag jeweils eine komplette Kopie erstellt werden sollte. Selbstverständlich erfolgen diese Backups automatisiert.

3 Live-Demo

3.1 Hardware 2009



Letztes Jahr hatte ich die folgenden Maschinen dabei. Dummerweise gab es am Morgen etliche Male Stromausfälle. Dies führte dazu, dass während der Reorganisation erneute Stromausfälle zu beklagen waren. Um ca. 9:40 hatte ich schliesslich beide VM-Server in einem Zustand, dass ein Arbeiten damit nicht mehr möglich war. Ein Neuaufsetzen der Maschinen hätte rund 30 Minuten Zeit in Anspruch genommen. In den Vortragssaal konnte ich auch noch nicht (war ja bis ca. 9:50 Uhr belegt. So kam es, dass ich den Vortrag mit dem Skript hielt.

3.2 Hardware 2010



Dieses Jahr hab ich die folgenden Maschinen mitgenommen. Einaml davon abgesehen, dass ich anstelle von 20 Kilogramm weniger als 5 Kilogramm mitschleppe, hab ich dieses Jahr bewusst

darauf verichtet, die Maschinen vorgängig aufzusetzen. Der Titel lautet 'ArchivistaVM: In 2 Minuten zum Virtualisierungsserver'. Ich möchte daher die Installation auf zwei leeren Maschinen live durchspielen.

3.3 Verwendete Hardware

Als Demo-Boxen hab ich die Modelle ArchivistaBox Tiny mitgenommen. Bestückt sind diese mit Dual-Cores (Intel), 4 GByte RAM (8 GByte wäre machbar) sowie 120 GByte Solid-State-Festplatten.

3.4 Cluster mit Crossover-Kabel

Um auch das Erstellen eines Clusters zeigen zu können, sind die beiden Maschinen mit einem Crossover-Netzwerkkabel miteinander verbunden. Damit ist auch gesagt, dass es keine Netzwerkverbindung nach aussen gibt.

3.5 Installationszeiten

Derzeit gibt es zwei Release der ArchivistaBox 64 Bit. Es gibt ein ca. 200 MByte ISO, welches die Komponenten für die Virtualisierung enthält. Weiter gibt es ein ca. 700 MByte ISO, welches zusätzlich sämtliche Komponenten für die ArchivistaBox als DMS-System enthält.

Bei der ArchivistaBox (gepackt ca. 700 MByte, ungepackt ca. 2.1 GByte) haben wir auf diesen Maschinen eine Installationszeit von weniger als 60 Sekunden. Bei der reinen Virtualisierungslösung (vergleichbar mit ArchivistaVM 1.x sowie Proxmox 1.x) haben wir eine Installationszeit von ca. 40 Sekunden.

Zum Vergleich: Ein aktuelles Ubuntu benötigt auf der gleichen Hardware bei der Installation 8 Minuten. Und dabei ist Ubuntu noch nicht einmal langsam, vielmehr ist die ArchivistaBox rasend schnell aufgesetzt.

4 ArchivistaVM ist eine Box

Als wir ArchivistaVM 2.0 realisierten, erhielten wir bei den ersten Feedbacks mehrfach die Anfrage, ob denn bei uns eine grafische Oberfläche vorhanden wäre, oder ob diese wenigstens nachgerüstet werden könne.

Daraus ist die Idee entstanden, grundsätzlich eine grafische Oberfläche samt Browser und Java-Umgebung zu erstellen. Und weil wir bei der ArchivistaBox bereits seit Jahren mit 'fluxbox' arbeiten, haben wir uns entschieden einerseits ebenfalls auf Fluxbox zu setzen und weiter auch gleich einige Module der ArchivistaBox in ArchivistaVM zu realisieren.

Mit der Zeit merkten wir, dass der Weg zur vollständigen ArchivistaBox nicht mehr weit wäre – und so haben wir uns entschieden, der ArchivistaBox 64 Bit den gesamten Code der ArchivistaVM-Lösung zu spendieren. So konnten wir früher als geplant die ArchivistaBox, unser DMS-System, von 32 auf 64 Bit portieren.

Wir mussten dabei zwar etwas mehr Code von Archivista 1.x bzw. unserem Proxmox-Fork neu schreiben, können aber so elegant unsere beiden Lösungen auf einer CD zusammenführen. Der zusätzliche Ressourcenverbrauch aufgrund der DMS-Komponenten ist marginal, auch die grafische Oberfläche benötigt kaum CPU-Kapazität.

5 Eingesetzte Technologie

5.1 Eigene Kernel

Wir verwenden selbstgebaute Kernel, im Moment ist es 2.6.35.x. Damit können wir viele neue Netzwerkkarten problemlos unterstützen, für die wir ansonsten Patches für ältere Linux-Kernel erstellen müssten.

5.2 Debian mit in Board

Auch wenn wir weder bei der Installation noch beim Kernel direkt auf Debian zurückgreifen, so kann doch `apt-get install` jederzeit aufgerufen werden.

5.3 Installation über RAM

Im Unterschied zu einer 'klassischen' Debian-Installation, werden bei der ArchivistaBox 64 Bit keine Pakete installiert. vielmehr wird eine Kopie der Master-Installation aufgespielt. Damit ist sichergestellt, dass sämtliche Installationen eines Releases 100 Prozent die gleiche Software enthalten. Verwendet wird dabei 'squashfs', welches sämtliche Prozessoren eines Systems parallel zum Aufspielen der Daten verwenden kann.

5.4 Setup-Programm in Perl

Natürlich könnten wir Bash-Skripte schreiben, aber eine Hochsprache ist letztlich doch eleganter. Für grössere Projekte (und ein Installationsprogramm ist nun einmal kein 100-Zeilen-Programm) hat sich objektorientierte Programmierung zudem sehr bewährt. Aus diesen Gründen haben wir das Setup-Programm objektorientiert in Perl geschrieben.

5.5 Anpassbares Setup-Programm

Wir benötigen das Setup-Programm an mehreren Stellen. Es findet sich auf dem Stick, es wird auf die Festplatte installiert und wir benötigen es ebenfalls, um im laufenden Betrieb die IP-Adresse zu setzen. Es kann interaktiv und auch nicht interaktiv aufgerufen werden. Weiter kann das Setup-Programm parameterisiert aufgerufen werden. Damit kann das gesamte Setup vollautomatisiert ablaufen, wobei dabei selbständig IP-Adresse, Tastatur und einiges mehr gesetzt werden können.

6 Automtisierte Setups

Anlässlich des Demos habe ich jeweils zwei Sets aufgerufen, einmal 'avbox1' und weiter 'avbox2'. Diese beiden Setups möchte ich hier detailliert vorstellen.

6.1 Beispiel avbox1

Beispiel für Anpassung der Demo-Box avbox251 (syslinux.cfg):

```
DEFAULT linux
TIMEOUT 100

PROMPT 1
DISPLAY boot.msg

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

label debug
kernel vmlinuz
APPEND initrd=initrd.img ramdisk_size=81920

label avbox1
kernel vmlinuz
APPEND initrd=initrd.img quiet ramdisk_size=81920
keyboard.de_CH lang.de auto ip.192.168.2.251
submask.255.255.255.0 gw.192.168.2.251 dns.192.168.2.251
```

Dazu einige Erläuterungen. Grundsätzlich ist auf dem USB-Stick die Datei syslinux.cfg (allenfalls auch isolinux.cfg) anzupassen. Die Einträge 'linux' und 'debug' belassen wir stehen; wir möchten die normalen Installationsarten ja nicht 'zerstören', sondern einen neuen Eintrag (avbox1) hinzufügen. Wichtig ist weiter, dass sämtliche Einträge unter 'APPEND' auf einer Zeile stehen. Nur so werden die Optionen entgegengenommen.

Die Einträge selber sollten aufgrund des Names selbsterklärend sein, vielleicht mit der Ausnahme 'auto'. Mit dieser Option wird festgelegt, dass das Setup-Programm automatisiert abläuft. Dabei wird die Festplatte komplett formatiert und das System ohne Rückfrage installiert. Sollte allerdings bereits eine ArchivistaBox-Installation vorliegen, so wird 'nur' ein Update

installiert; wir möchten ja nicht, dass die installierten Instanzen (Betriebssysteme) mutwillig zerstört werden.

Ebenfalls zu beachten gilt es, dass die Einträge zwingend mit einem Punkt zu trennen sind. Dies deshalb, weil alle Optionen auch beim Start interaktiv über die Tastatur mitgegeben werden können. Und hier hat sich der Punkt sehr bewährt, weil er relativ einfach und über alle länderspezifischen Tastaturanpassungen eingetippt werden kann.

6.2 Beispiel avbox2

Betrachten wir nun das zweite Beispiel:

```
DEFAULT linux
```

```
TIMEOUT 100
```

```
PROMPT 1
```

```
DISPLAY boot.msg
```

```
label linux
```

```
kernel vmlinuz
```

```
APPEND initrd=initrd.img quiet ramdisk_size=81920
```

```
label debug
```

```
kernel vmlinuz
```

```
APPEND initrd=initrd.img ramdisk_size=81920
```

```
label avbox2
```

```
kernel vmlinuz
```

```
APPEND initrd=initrd.img quiet ramdisk_size=81920 lang.de
```

```
auto ip.192.168.2.252 submask.255.255.255.0 gw.192.168.2.251
```

```
dns.192.168.2.251 copyiso.tinycore.iso copydef.101.con
```

```
copyimg.v101.img
```

Auf den erste Blick sehen wir kaum Unterschiede zum ersten Beispiel. Aber, es gibt da zusätzliche Einträge mit 'copyiso', 'copydef' und 'copyimg'. Damit kann eine Instanz bereits bei der Installation mit aufgespielt werden. Im verwendeten Beispiel wird das Mini-Linux 'tinycore' aufgespielt. Zu beachten ist, dass wir mit einem FAT-Datenträger arbeiten, d.h. sämtliche anzugebenden Dateinamen müssen mit 8.3-Format (kurze Dateinamen) eingegeben werden.

7 Stand der Entwicklung

Ursprünglich war für das Jahr 2010 'nur' ArchivistaVM 2.0 geplant. Bei der Entwicklung realisierten wir, dass ein GUI wünschbar ist. Letztlich ist daraus die ArchivistaBox 64 Bit entstanden. Im Moment gibt es zwei ISO-Dateien. Im engeren Sinne ArchivistaVM mit ca. 200 MByte und die ArchivistaBox mit ca. 700 MByte.

Da wir selber in den letzten Wochen fast ausschliesslich das 'grosse' Release aufspielten, sind wir im Moment allerdings nicht mehr sicher, ob das kleine Release in dieser Form 'überleben' wird. Zu sehr lernten wir den Komfort schätzen, jederzeit über ein GUI zu verfügen. Letztlich dauert das Aufspielen des 'grossen' Releases etwa 20 bis 30 Sekunden länger; wir denken, dass dies kaum eine Rolle spielt.

7.1 Open Source != kostenfrei

Ein Open Source Projekt wie ArchivistaVM bzw. die ArchivistaBox finanziert sich nicht von alleine. Natürlich könnten wir darauf hoffen bzw. warten, bis unser Projekt in der Community derart grosse Beachtung findet, um genügend Entwickler zu finden, welche uns dann unterstützen bzw. die Weiterentwicklung garantieren.

Aber, seien wir offen und ehrlich, das funktioniert nur für den allerkleinsten Teil der Open Source Projekte. Der überaus grösste Teil der quelloffenen Programme wird heute von Firmen und für Firmen entwickelt. ArchivistaVM ist ja gerade entstanden, weil wir im Rahmen eines Kundenprojektes eine quelloffene Virtualisierungslösung suchten und bei Proxmox fündig wurden.

Bevor wir den Fork starteten, hatten wir uns zusammen mit unserem Kunden die Frage gestellt, ob wir die Dienstleistungen bei dieser Firma einkaufen wollten. Wir hatten auch eine Anfrage gestartet, allerdings darauf keine Antwort erhalten. Natürlich hätten wir auch eine zweite Anfrage starten können, letztlich haben wir uns aber für den Fork entschieden, weil wir an KVM und an modernen Kernen interessiert waren, und dies scheint uns, ist nicht das Ziel, welches Proxmox verfolgt, da dort OpenVZ neben KVM eine gleichberechtigte Rolle spielt.

7.2 Finanzierte Erweiterungen

In die Entwicklung von ArchivistaVM 2.0, d.h. des aktuellen Releases sind weit über 1000 Stunden Entwicklungszeit geflossen. Gemessen an den einigen Dutzend Lösungen, die wir bei Kunden im Einsatz haben, können wir über die Wartungsverträge aber nur etw 100 Stunden pro Jahr finanzieren.

Daher haben wir unsere Kunden gefragt, welche Features sie entwickelt haben möchten und ob sie sich bei den Features beteiligen könnten. In einigen Fällen sind daraus Erweiterungen entstanden, die wir für unsere Kunden realisieren dürfen.

Zwei Beispiele möchte ich hier explizit erwähnen. Die derzeitige Version verfügt noch nicht über die Möglichkeit, gesicherte Images (OS) über ein GUI zurückzuspielen. Eine von einem Kunden finanzierte Erweiterung wird dies sicherstellen. Ein anderer Kunde wünscht eine Überwachung der Hardware und das Ansprechen von USV-Geräten. Auch diese Erweiterung wird derzeit realisiert.

In einem dritten Fall wird es darum gehen, optische Karten sauber zu testen und zu implementieren. Nicht dass wir uns falsch verstehen. Aufgrund des aktuellen Kernels (derzeit 2.6.35.9) sind bereits die Treiber von sämtlichen optischen Karten im Kernel enthalten, aber vor einem Produktiv-Betrieb müssen diese Karten bei uns oder bei einem Kunden ausführlich getestet werden, damit es später unter Last-Betrieb keine 'bösen' Überraschungen gibt.

8 Zukunft und heute

Wir sind heute (Stand Ende November 2010) an sich sehr zufrieden mit dem Stand der Entwicklung. Nachdem wir den neuen Installer realisiert haben, können wir zügig die übrigen Komponenten angehen und realisieren. Oberstes Ziel unserer Entwicklung ist, das Arbeiten mit einem Server für die Virtualisierung so einfach wie möglich zu machen. Doch dabei soll es nicht bleiben. Wir möchten das Aufsetzen und Verwalten von Clustern ebenso einfach machen. Auf der anderen Seite (und das hat eine Kundenbefragung im Herbst 2010) ebenfalls ergeben, legen unsere Kunden grossen Wert an einer Verfeinerung beim GUI. Hochverfügbarkeit wurde erst an dritter Stelle als Ziel genannt, viel wichtiger für unsere Kunden ist es, eine Oberfläche vorzufinden, um Server und Instanzen bequem verwalten zu können.

8.1 Erweiterungen in Planung

Schon heute können wir automatisiert Instanzen bei der Installation mit aufspielen. Hier sind noch viele nützliche Erweiterungen denkbar. Eine davon ist, im laufenden Betrieb, einen neuen Installationsdatenträger (mit oder ohne Instanzen) anlegen zu können. Diese Konzept wenden wir bereits bei der ArchivistaBox mit dem Publizieren an. Ein entsprechender Prototyp läuft bei uns intern und es wird nur eine Frage der Zeit sein, bis wir dieses Feature freigeben können.

Ebenfalls gewünscht wurde zuweilen ein Benutzer-Konzept für virtualisierte Instanzen und auch das automatische Spiegeln auf eine zweite Maschine wird dann und wann gewünscht. Beide Punkte können schon heute mühelos mit der Konsole eingerichtet werden. Das entsprechende Interface ist in der Planungsphase. Ganz zentral dabei scheint uns, dass das Einrichten solcher Features äussert einfach ist.

9 Stabilität und Download

Die ArchivistaBox 64 Bit läuft Ende November 2010 stabil. Wir sind derzeit bei den Massentests. Sollten dabei noch Probleme auftauchen, könnte es noch Änderungen geben. Ich gehe aber nicht davon aus, dass dem so sein wird und so dürfen wir die entsprechenden ISO-Dateien nächste Woche freigeben. Noch nicht vorhanden ist das Handbuch zur neuen ArchivistaBox 64 Bit. Ich hoffe, dass wir diesen Teil bis zu Beginn 2011 ebenfalls abschliessen können. Die ISO-Dateien können hier bezogen werden:

ArchivistaBox 64 Bit (DMS + VM mit GUI): www.archivista.ch/avbox-64bit.iso

ArchivistaVM 64 Bit (VM ohne GUI): www.archivista.ch/avvm-64bit.iso

Sollten beim Arbeiten mit der ArchivistaBox 64 Bit Fragen entstehen, so können wir unser Support-Forum sehr empfehlen. Es ist unter help.archivista.ch/forum erreichbar. Dabei ist eine Registrierung notwendig. Aufgrund der negativen SPAM-Geschichten schalten wir die Accounts von Hand frei – und daher dauert es ein zwei Tage, bis Meldungen gepostet werden können.

Im übrigen möchten wir gerne auf den kommerziellen Support verweisen. Hier können ArchivistaBox-Server jederzeit zu fixen Konditionen erworben werden. Sollte ein Kunde seinen Server mitbringen wollen, so ist selbstverständlich auch dies machbar.

Vielen Dank für die Aufmerksamkeit.