

Virtualisierung mit KVM, Debian und ArchivistaVM

Auswahl von Hard- und Software mit 'kleinem' Budget

Inhaltsverzeichnis

1	Einleitung	2	7	Lösung auf Stufe Mini-ATX	22
			7.1	Redundanz auf Stufe Hardware	22
2	Der erste Server	3	7.2	Diverse Versuche	22
2.1	Warum KVM? Warum Debian?	3	7.3	Umbau des Gehäuses Silverstone SUGO SG02-f	23
2.2	Das Projekt beginnt	4	7.4	Umgebauter Silverstone SUGO SG02-f im Vergleich	27
2.3	KVM und Debian = Proxmox	5			
2.4	Ein Raid-Kontroller muss her	5			
2.5	Betriebsmodi von Raid-Kontrollern	7			
3	Darum ist ArchivistaVM entstanden	9	8	Migrationen und Stolpersteine	28
3.1	Vorgeschichte	9	8.1	Zügeln alter Maschinen	28
3.2	Problematik bei Proxmox	9	8.2	Zügeln bereits virtualisierter Instanzen	28
3.3	Was ist die ArchivistaVM?	10	8.3	Geschwindigkeit	29
4	Installation von ArchivistaVM	11	8.4	Backup	29
5	Ein Betriebssystem installieren	16	8.5	Backup mit Snapshots	30
6	Erfahrungen mit den ersten ArchivistaVM-Servern	20	8.6	Rückspielen einer einzelnen Datei	30
6.1	Kostengünstige Test- bzw. Ersatzserver	20	8.7	Ein Server ist ein Server...	30
6.2	Zweite Netzwerkkarte	21	9	Das Projekt geht weiter	32
			9.1	ArchivistaVM CD frei verfügbar	32
			9.2	Mitarbeit gerne erwünscht	32

© 27.11.2009 by Urs Pfister, www.archivista.ch

1 Einleitung

Vor etwas mehr als sechs Jahren habe ich begonnen, mit VMware Workstation zu arbeiten. Die Erfahrungen dabei waren gemischter Natur. So konnte ich jeweils unsere Entwicklungsumgebung (die damals Windows benötigte) mit den verschiedensten Linux-Distributionen betreiben. Allerdings nicht ohne bei jeder neuen Distribution/Version irgendwelche Header-Dateien zusammensuchen zu müssen, um VMWare installieren zu können.

Auch die Stabilität liess dann und wann etwas zu wünschen übrig. So erlebte ich es mehr als einmal, dass eine virtuelle Festplatte in die Brüche ging. Ich sicherte mich jeweils dagegen ab, indem ich die Daten immer auch auf ein Samba-Laufwerk speicherte.

Vor etwa drei Jahren versuchte ich XEN zu installieren. Es blieb beim Versuch, was wohl eher damit zusammenhing, dass es an Zeit mangelte bzw. ich den Einstieg nicht richtig gefunden habe.

Im letzten Winter dann haben wir KVM-83 auf unsere ArchivistaBox portiert. Da die gesamte ArchivistaBox (aus verschiedenen Gründen) nach wie vor mit 32 Bit kompiliert ist, können wir auf der ArchivistaBox nur 32-Bit-Gastsysteme laufen lassen. Dafür leisteten wir uns den 'Luxus', KVM und QEMU auf der gleichen Maschine ausführen zu können. Dies aus dem einfachen Grund, dass gewisse ArchivistaBoxen Prozessoren haben, die für die Virtualisierung vorbereitet sind, andere ArchivistaBoxen dies aber nicht können.

Soweit zu meinen Vorkenntnissen. Und an sich wäre es auch dabei geblieben, wenn in diesem Frühjahr nicht ein Kunde mit der Anfrage an mich herangetreten wäre, er habe da eine Offerte für eine VMWare-Installation über sFr. 100'000 (etwa 65000 Euro) vor sich liegen. Ob wir ihn nicht begleiten könnten, er möchte zunächst einmal Ubuntu JeOS installieren. Diese Server-Variante sei ja sowohl für KVM als auch für VMWare optimiert.

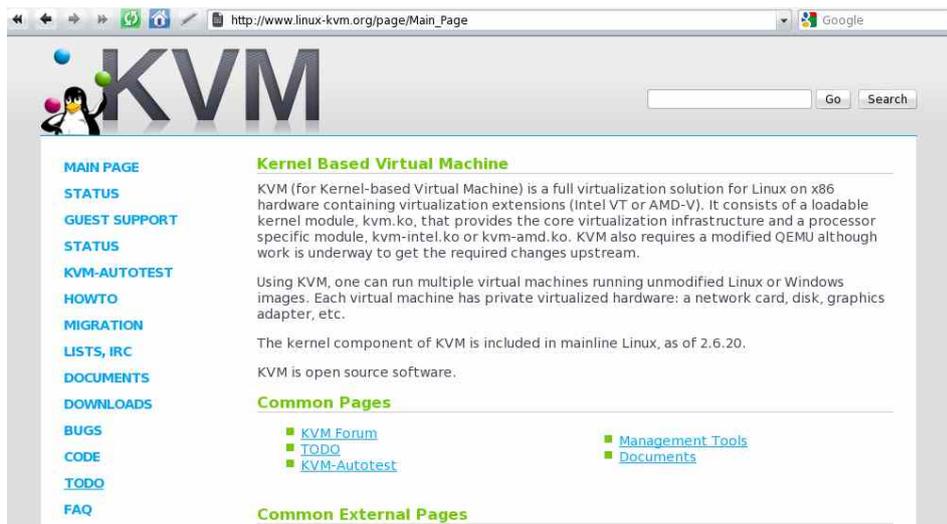
Und so kam es, dass ich 'Ja' zu einem 'Abenteuer' sagte, zu dem ich vor einem Jahr ganz bestimmt noch 'Nein' gesagt hätte.

2 Der erste Server

2.1 Warum KVM? Warum Debian?

Vielleicht sollten wir uns kurz vor Augen halten, welche Lösungen derzeit in Frage kommen können:

- VMWare: Marktleader, ClosedSource. Zwar war uns bewusst, dass es kostenlose ESX-Server-Instanzen zum Download gibt, aber ClosedSource bleibt nun mal ClosedSource. Und leider können entsprechende Systeme nur sehr beschränkt aufgesetzt werden (z.B. kein Clustering).
- XEN: OpenSource, gut verbreitet in grösseren Umgebungen, allerdings aufgrund der hohen Abhängigkeit zum Kernel 2.6.18.x für unser Projekt nicht geeignet.
- KVM: Relativ neues Produkt, als Modul ladbar (Bestandteil des Kernels), OpenSource und mittlerweile bereits gut verbreitet. Ebenfalls eine Rolle spielte, dass wir bereits Erfahrungen mit Qemu/KVM hatten und dabei gute Erfahrungen machten.
- QEmu, VirtualBox, OpenVZ und weitere Varianten kamen nicht in Frage, weil diese alle nicht die Chipsätze für die Virtualisierung verwenden und daher (gerade mit Windows-Gästen) eher einen gemächlichen Speed bieten.



Aus diesen Ausführungen lässt sich unschwer erkennen, dass wir uns für KVM entschieden haben. Betr. der Wahl des Betriebssystems, so galt es eine Wahl zwischen den Systemen T2, Ubuntu JeOS und Debian Lenny vorzunehmen.

- T2: Relativ unbekannt, bildet aber Basis für unsere ArchivistaBox und daher wäre ein gutes Mass an Skills vorhanden gewesen. Da wir allerdings den gesamten Trunk mit 64-Bit hätten bauen müssen, erschien uns der Aufwand gemessen an unseren Skills zu hoch.
- Ubuntu JeOS: Die Ubuntu-Server-Variante enttäuschte uns. Die Anleitung betreffend KVM befriedigte uns nicht wirklich, letztlich konnten wir keinen Unterschied zwischen Ubuntu- und Debian-Server erkennen.
- Debian: Wenn der Unterschied zwischen Ubuntu JeOS und Debian Lenny nicht zu sehen ist, dann doch lieber das Original.

2.2 Das Projekt beginnt

Somit war der Entschluss gefasst, wir versuchen mit Debian und KVM einen Virtualisierungsserver aufzubauen. Bevor wir loslegen konnten, benötigten wir einige Server für die Tests. Voraussetzung des Kunden war, die Server mussten bei den Festplatten hot-swap fähig sein.

Gemäss unserer Erfahrungen haben wir dem Kunden AMD-Prozessoren empfohlen, da wir mit den Intel-Prozessoren unserer ArchivistaBoxen doch einen erheblichen Verlust (ca. 30 Prozent) beim Arbeiten mit KVM feststellen mussten.

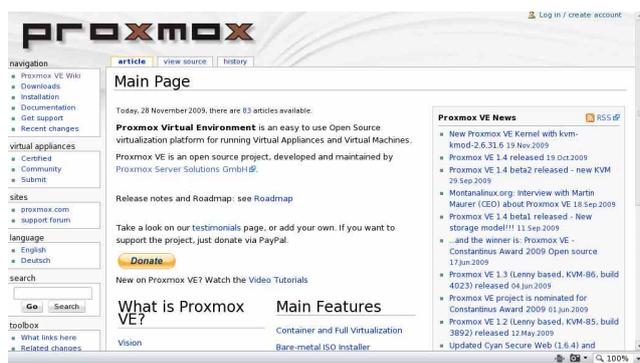


Wir haben dem Kunden zwei Vorschläge unterbreitet. Einmal eine Rackmaschine und weiter eine zusammengestellte Variante mit einem Chenbro ES34069-Gehäuse (siehe www.chenbro.com) und einem J&W MINIX 780G mini-ITX-Board (siehe www.jwele.com). Ausschlaggebend für die zweite Variante war hier, dass sowohl ich als auch der Kunde keine Rackschränke besitzen und wir es folglich schätz(t)en, es mit einer 'kleinen' Box realisieren zu können. Daneben spielte aber auch der Preis eine Rolle. Die HP-Maschine hätte ohne Festplatten soviel gekostet wie die Chenbro-Box mit vier 2.0 TByte-Festplatten, bzw. in etwa das Doppelte.

Sowohl die Boards als auch die Gehäuse sind mitunter aber nicht ganz einfach erhältlich, sodass es ca. 2 Wochen dauerte, bis alle Komponenten für drei Testmaschinen bei uns verfügbar waren.

2.3 KVM und Debian = Proxmox

Nachdem wir Debian auf die Maschinen installiert hatten, galt es für den Kunden ein Benutzerinterface auszuwählen. Weniger gefallen hat uns die Lösung, auf dem Virtualisierungsserver einen X11-Server aufzuspielen und darunter ein GUI zu betreiben. Dies wäre der Fall gewesen, wenn wir uns für Virt-Manager, siehe unter <http://www.virt-manager.org/> entschieden hätten. Ebenso sind wir nicht wirklich schlau geworden, wie wir OpenQRM (siehe www.openqrm.com) hätten installieren können. Dieser Prozess dauerte in etwa ca. 2 bis 3 Tage, ohne dass ich dem Kunden ein Resultat hätte präsentieren können. Irgendwann (es war zu nächtlicher Stunde) bin ich beim Suchen mit 'kvm gui interface' (oder so) auf Proxmox (siehe www.proxmox.com) gestossen.



Wir haben dann die Installations-CD bezogen und innerhalb einer Stunde hatten wir einen Virtualisierungsserver mit einem Web-Interface. Was uns zuvor Tage kostete, war nebenbei in extrem kurzer Zeit zu unserer vollsten Zufriedenheit erledigt. Folglich haben wir dem Kunden empfohlen, die Variante Proxmox intensiver auszuloten. Letztlich einigten wir uns mit dem Kunden zusammen auf Proxmox. Einige Tage später hat der Kunde den ersten Server entgegengenommen dürfen, zwei Server blieben bei uns.

2.4 Ein Raid-Kontrolller muss her

Der Kunde war mit dem System sehr zufrieden, allerdings wollte er nun plötzlich nicht nur Hot-Swap, sondern auch Festplattenspiegelung (RAID). Zwar hätte das J&W-Board einen internen Raid-Kontrolller gehabt, aber die internen Raid-Kontrolller geniessen plus/minus keinen guten Ruf. Und ja, das Backen entsprechender gepatchter Kernel schien uns auch nicht gerade anregend für die Sommerpause; ich wollte ja nicht unnötig die Ferien an der Ostsee gefährden.

Beim Chenbro-Gehäuse bleibt für Raid-Kontroller im Prinzip kaum Platz übrig. Und so gestaltete sich das Einbauen des Raid-Kontrollers als nicht ganz einfach. Nur mit speziellen Verbindungskabel (Riser-Cables, siehe www.mini-itx.com) ist es uns am Schluss gelungen, auch diesen Wunsch zu erfüllen. Und weil der gute Zoll sich Zeit liess, dauerte es mehr als zwei Wochen bis ich die Kontroller überhaupt einbauen konnte.



Doch damit nicht genug, die bestellten Adaptec-Kontroller 3405 (siehe www.adaptec.com) arbeiteten zwar gut und gerne, wurden im Betrieb aber sehr heiss. Ebenso störte uns die lange Bootzeit von ca. 2 Minuten. Der bestellte Highpoint-Rocket-Kontroller (siehe www.highpoint-tech.com/) erkannte dann und wann nur 3 von 4 Festplatten. Erst beim dritten Anlauf, beim 3Ware-Kontroller 3650SE (siehe www.3ware.com) sind wir überzeugt hängen geblieben. Schnelle Boot-Zeit, und ein nettes Feature im Bios, mit dem wir bei Bedarf eine kleine Startpartition sowie eine grosse Datenpartition erstellen können, haben es uns echt angetan.

2.4.1 Die 2 TB-Grenze bei Festplatten

Dazu muss gesagt werden, dass das Booten ab Festplatten, die grösser als 2 TByte sind, unter Linux nicht ganz einfach ist. Einmal müssen die Festplatten zwingend mit parted und GPT eingerichtet werden und weiter erfordert das Booten von so eingerichteten Platten entsprechende gepatchte Grub-Versionen. Weiter kann im Falle, dass eine Platte nicht mehr bootet, nur erschwert auf solche Platten zugegriffen werden. Kurz und (un)gut, im Moment wollten wir keine Boot-Partition auf einer Platte, die grösser als 2 TByte ist. Und gerade hier bietet der 3Ware-Raidcontroller 3650SE gute Dienste, weil eben eine Boot- und Daten-Partition separat erstellt werden können.

2.4.2 Batteriemodul

Bessere Raid-Kontroller haben ein weiteres nettes Feature, sie besitzen einen internen Speicher (derzeit meist zwischen 256 und 512 MByte). Als Vorteil resultiert ein sehr guter Durchsatz auf den Festplatten, als Nachteil liegen die Daten zunächst noch im RAM, auch wenn sie bereits auf der Platte sein sollten. Im Falle eines Stromausfalles bedeutet dies, dass die Daten verloren sind, obwohl das Betriebssystem glaubt, die Daten lägen bereits auf der Festplatte. Daher haben bessere/teurere Raid-Kontroller eine sogenannte Batterieeinheit, d.h. der Raid-Kontroller wird 'künstlich' mit den Daten am Leben erhalten, bis der Rechner wieder hochgefahren werden kann.



Und wenn ich an dieser Stelle einen Tipp anbringen darf: Bitte gleich eine Batterieeinheit kaufen, die nicht auf die Reid-Karte gesteckt wird. Wir müssen dann zwar irgendwo für die Batterie einen (Steck-)Platz finden, haben aber den Vorteil, dass die Batterie selber nicht derart durch den Kontroller erhitzt wird. In meinem Fall machte ich zunächst die Erfahrung, dass die aufgesteckte Batterie nach einigen Stunden überhitzte. Und das alles nur, weil im Online-Shop die beiden Produkte mit dem gleichen Bild ausgezeichnet waren - und wer kauft schon das teurere Modell bei gleichem Bild? Immerhin, mit dem zweiten Anlauf war auch diese Hürde gemeistert.

2.5 Betriebsmodi von Raid-Kontrollern

In unserem Falle, d.h. bei vier Festplatten stehen uns folgende Möglichkeiten offen:

- RAID0: Keine Redundanz, aber höherer Speed, Splitten der Daten auf mehrere Platten, nicht zu empfehlen, da beim Ausfall einer Platte sämtliche Daten verloren sind).
- RAID1: Redundanz auf zwei Platten, kein erhöhter Speed, sehr zu empfehlen, bei vier Platten können zwei unabhängige Plattenverbünde (2x2 Platten) eingerichtet werden.
- RAID10: Redundanz auf zwei Platten mit Splittung über zwei Platten. Zu empfehlen, falls Boot-Disk im Bios eingerichtet werden kann.

- RAID5: Daten werden über sämtliche Platten verteilt, wobei eine Platte ausfallen darf. Bedingt zu empfehlen, relativ lange Initialisierungszeit beim Aufbau bzw. Austauschen einer Platte und während dieser Zeit langsamer Zugriff.

Wir sehen an diesen Ausführungen, dass RAID-Kontroller eine 'gewisse Wissenschaft' für sich sind. Und selbstverständlich kostet dies alles auch noch fast mehr, als Gehäuse, Board, RAM und Prozessor zusammen. Und dennoch, der Durchsatz ist für die Virtualisierung nicht nebensächlich, weil ja die verschiedenen Instanzen eine Festplatte 'teilen' müssen.

3 Darum ist ArchivistaVM entstanden

3.1 Vorgeschichte

Ursprünglich hatten wir gedacht, mal schnell einen Virtualisierungsserver aufzubauen. Letztlich sind daraus Wochen bzw. Monate geworden. Wir haben dabei fleissig die Foren bei www.proxmox.org studiert. Ganz viele Postings gingen in die Richtung, welche Hardware läuft mit Proxmox? Wir möchten Proxmox mit einem Software-Raid betreiben, geht das? Warum läuft Kontroller XYZ nicht? Ich möchte an dieser Stelle sehr betonen, dass die Proxmox-Leute einen sehr guten Job machen.

Und dennoch frag(t)e ich mich, ob es Sinn ergibt, dass ein/e jede/r, der auf die Lösung stösst, am Ende derart viele Punkte bei der Wahl der Hardware beachten muss. Die Hardware selber kostet heute nicht mehr alle Welt, aber entweder beschäftigt sich jemand intensiv mit diesen Fragen, oder die Chance ist gross, dass die bestellte Hardware am Schluss nicht läuft. Mit den obenstehenden Ausführungen hoffe ich im übrigen etwas dazu beizutragen, dass die Wahl der Hardware leichter fällt.

3.2 Problematik bei Proxmox

Ich habe bereits erwähnt, dass wir gerade deshalb von KVM derart angetan sind, weil KVM als Kernel-Modul läuft. Damit können wir KVM mit einem jeden Kernel bzw. Linux-System laufen lassen. Proxmox selber aber bringt neben KVM auch OpenVZ (siehe www.openvz.org mit. Dies bedeutet, dass aufgrund OpenVZ eine Abhängigkeit zum Kernel 2.6.24.x besteht.

Beim Pilotprojekt haben wir dies zur Kenntnis genommen, doch sollte sich wenig später feststellen, dass eine solche Abhängigkeit nicht unproblematisch ist. Das Patchen des relativ alten Kernels ist zwar ausreichend beschrieben, allerdings bleibt es dabei, dass sollten Patches notwendig sein, dies schnell in tage- und nachtfüllende Sessions ausarten kann (und die Nächte waren im Sommer kurz). Und wo stehen wir in zwei drei Jahren, wenn wir definitiv einen neuen Kernel benötigen, weil sich die Treiber der Hardware dannzumal nicht mehr rückpatchen lassen?

Noch weniger ist es realistisch, dies einem Kunden zuzumuten. Eine Umfrage auf dem Proxmox-Forum ergab, dass eine Mehrheit lieber OpenVZ beibehalten möchte und dafür bereit ist, auf aktualisierte Kernel zu verzichten. Dies gilt es zu respektieren.

Daher haben wir uns entschlossen, die ArchivistaVM zu erstellen. Ich möchte an dieser Stelle betonen, dass wir dies ohne die Vorarbeit von Proxmox so (schnell) wohl nicht hingekriegt

hätten, aber ich bin gleichzeitig der Ansicht, dass das Festhalten an OpenVZ für eine Virtualisierungslösung, die auf KVM beruht, für mein Verständnis ganz und gar keinen Sinn ergibt.

3.3 Was ist die ArchivistaVM?



The screenshot shows a web browser window with the URL <http://www.archivista.ch/de/pages/produkte/virtualisierung-und-thinclients/installation-archivista>. The page title is "Archivista: OpenSource DMS, ERP & Virtualization Out-of-the-Box". The main heading is "Schritt für Schritt zum ArchivistaVM-Server mit der ArchivistaVM-CD". The text below the heading reads: "Bevor wir diese Schritt-für-Schritt-Anleitung durchgehen können, benötigen wir zwei Dinge. Einmal die ArchivistaVM-CD. Diese können Sie hier beziehen. Erstellen Sie davon eine bootbare CD. Weiter benötigen wir einen Rechner, auf dem wir den ArchivistaVM-Server installieren können. Dieser Rechner muss komplett für den ArchivistaVM-Server zur Verfügung stehen. Hardwaretechnisch muss es minimal ein 64-Bit-Prozessor mit minimal Dual-Core-Prozessor sein. Als Empfehlung sei ein AMD-Prozessor mit minimal 64-Bit empfohlen. Ebenfalls notwendig (für guten Speed) ist, dass der Prozessor Hyperthreading unterstützt." Below this text, it says: "Legen Sie nun die ArchivistaVM-CD ins Laufwerk ein. Stellen Sie sicher, dass das Netzwerkkabel angeschlossen ist, da ansonsten ArchivistaVM keine Netzwerkadresse herausfinden kann." To the right of the text is an image of a CD/DVD with the Archivista logo and the text "ArchivistaBox Virtualization Projectbased on Debian Lenny and KVM". A sidebar on the left contains a navigation menu with items like "Home", "Produkte", "DMS und ERP", "Virtualisierung und ThinClients", "ArchivistaVM Summit-Server", "ArchivistaVM Budget Server", "Installation ArchivistaVM", "Betriebssystem aufsetzen", "FAQs zu ArchivistaVM", "ArchivistaVM CD zum Download", "Archivista ThinClient", and "Wartung, Service".

In erster Linie ist ArchivistaVM eine Installations-CD, die im Moment noch grosse Ähnlichkeiten zu Proxmox aufweist. Die Unterschiede sind:

- OpenVZ nicht vorhanden
- Besseres Backup-Konzept (keine getarten Dateien)
- Aktuellerer Kernel (derzeit 2.6.30.6)
- Bundling mit Hardware (kommerzielle Variante)
- Virtualisierung in Virtualisierung (in Arbeit)
- Automatischer Abgleich auf mehreren Maschinen (in Arbeit)

An sich wäre geplant gewesen, die letzten beiden Punkte an dieser Stelle zu präsentieren. Beide Punkte sind allerdings noch nicht derart realisiert, dass ich sie heute zeigen möchte. Und offen gestanden sieht es im Moment nach viel Arbeit im Dezember aus, sodass es wohl 2010 wird.

4 Installation von ArchivistaVM

Bevor wir diese Schritt-für-Schritt-Anleitung durchgehen können, benötigen wir zwei Dinge. Einmal die ArchivistaVM-CD. Diese können Sie unter www.archivista.ch beziehen. Erstellen Sie davon eine bootbare CD. Weiter benötigen wir einen Rechner, auf dem wir den ArchivistaVM-Server installieren können. Dieser Rechner muss komplett für den ArchivistaVM-Server zur Verfügung stehen. Hardwaretechnisch muss es minimal ein 64-Bit-Prozessor mit minimal Dual-Core-Prozessor sein. Als Empfehlung sei ein AMD-Prozessor mit minimal 64-Bit empfohlen. Ebenfalls notwendig (für guten Speed) ist, dass der Prozessor Hyperthreading unterstützt.

Legen Sie nun die ArchivistaVM-CD ins Laufwerk ein. Stellen Sie sicher, dass das Netzwerkkabel angeschlossen ist, da ansonsten ArchivistaVM keine Netzwerkadresse herausfinden kann.



Nach einer Weile erhalten Sie den Boot-Bildschirm. Drücken Sie hier die Enter-Taste. Danach erscheinen einige Textmeldungen. Wichtig ist, dass Sie am Schluss in einen grafischen Modus gelangen und die untenstehende Maske vorfinden.



Auf der Begrüßungsmaske werden Sie damit vertraut gemacht, dass die gesamte Lösung der GPL-Lizenz untersteht. Klicken Sie auf den Button/Knopf unten rechts 'I agree'.



Falls am System mehr als eine Festplatte angeschlossen ist, können Sie hier die Festplatte für die Installation auswählen. Bei einer Festplatte klicken Sie einfach auf 'Next'. Bitte beachten Sie, dass dabei sämtliche Daten gelöscht werden. Und noch etwas, ArchivistaVM kann nur auf Festplatten installiert werden, die nicht grösser als 2 TByte gross sind. Sofern grössere Festplatten (z.B. im RAID-Verbund) verwendet werden sollen, muss für die Installation eine kleine erste Festplatte mit weniger als 2 TByte verwendet werden.



Legen Sie den Standort, die Zeitzone sowie die Tastatur fest. Anschliessend klicken Sie auf 'Next'.



Legen Sie das 'root'-Passwort fest. Damit können Sie sich später über den Web-Browser oder per Fernwartung (ssh) anmelden. Klicken Sie auf 'Next'.



Nun folgt der vielleicht schwierigste Teil der Installation. ArchivistaVM schlägt nun eine fixe IP-Adresse, die später für den Betrieb des ArchivistaVM-Servers verwendet wird, vor. Sie müssen zudem dem Rechner einen Namen geben. Damit ArchivistaVM eine Adresse finden kann, muss ein DHCP-Server vorhanden sein. Falls Sie keinen DNS-Server haben, können Sie bei Hostname z.B. auch 'archivistavm.server.local' eingeben. Klicken Sie auf 'Next'.



ArchivistaVM wird nun installiert. Dieser Vorgang dauert in der Regel einige wenige Minuten.



Sobald die Installation abgeschlossen ist, können Sie auf 'Reboot' klicken, um das System neu zu starten. Anschliessend erhalten Sie (analog zum Startvorgang) einige Textmeldungen. Am Ende sollte der Rechner neu gestartet werden und Sie sollten folgende Maske vorfinden:

```
GNU GRUB version 0.97 (636K lower / 523200K upper memory)

Archivista Virtual Environment, kernel 2.6.24-7-pve
Archivista Virtual Environment, kernel memtest86+

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

Am einfachsten lassen Sie die Pause von ca. 5 Sekunden ganz einfach verstreichen, ArchivistaVM startet dann von alleine.

```
Bringing up interface venet0: ..done
Configuring interface venet0: ..done
Starting PVE daemon: pvedaemon.
Starting PVE Tunnel Server: pvetunnel.
Starting PVE Cluster Server: pvemirror.
Starting NTP server: ntpd.
unable to load kvm module
Starting deferred execution scheduler: atd.
Starting periodic command scheduler: crond.
Starting web server: apache2.
Not starting Bacula File daemon: disabled via /etc/bacula/do_not_run

-----

Welcome to the Archivista Virtual Environment. Please use your web browser to
configure this server - connect to:

  https://192.168.0.60/

-----

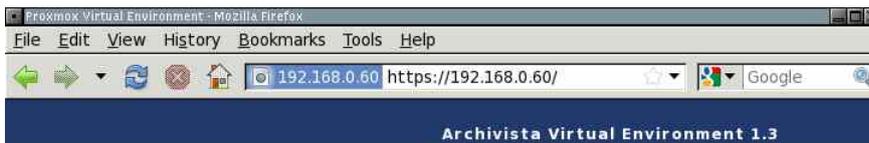
Debian GNU/Linux 5.0 virtserver tty1
virtserver login: _
```

Sofern der Startvorgang erfolgreich ist, sehen Sie am Schluss in etwa den obenstehenden Bildschirm. Sie könnten sich nun direkt auf der Konsole am ArchivistaVM-Server anmelden. Ebenfalls möglich ist das Anmelden über Fernwartung (ssh). Für das weitere Arbeiten benötigen wir die Konsole aber nicht.

5 Ein Betriebssystem installieren

Bevor Sie ein Betriebssystem virtualisiert in Betrieb nehmen können, müssen Sie sich an ArchivistaVM über einen Webbrowser anmelden. Dazu benötigen Sie den Namen oder die IP-Adresse des ArchivistaVM-Servers. Die Informationen wurden bei der Installation bestimmt.

Starten Sie nun einen Webbrowser und tippen Sie die Adresse des ArchivistaVM-Servers ein. Nach Möglichkeit sollten Sie einen Browser verwenden, der die Java-Bibliotheken enthält, da später der Bildschirm einer jeden installierten Instanz über Java-Technologie angezeigt wird.



Login

Username:

Password:

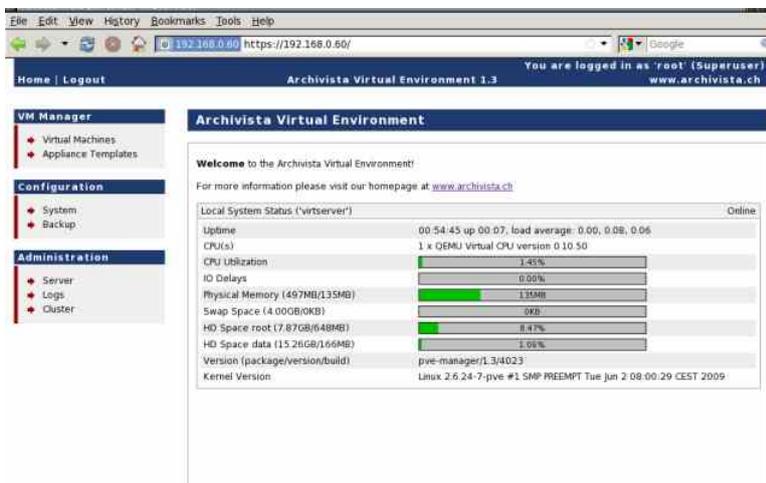
Information

Please use Microsoft Internet Explorer (> 6.0) or Firefox (> 1.5) for full functionality.

JavaScript is active: ok.



Melden Sie sich mit dem Benutzernamen 'root' und dem bei der Installation vergebenen Passwort an.



Home | Logout Archivista Virtual Environment 1.3 You are logged in as: 'root' (Superuser) www.archivista.ch

VM Manager

- Virtual Machines
- Appliance Templates

Configuration

- System
- Backup

Administration

- Server
- Logs
- Cluster

Archivista Virtual Environment

Welcome to the Archivista Virtual Environment!
For more information please visit our homepage at www.archivista.ch

Local System Status ('virtserver') Online

Uptime	00:54:45 up 00:07, load average: 0.00, 0.00, 0.06
CPU(s)	1 x QEMU Virtual CPU version 0.10.50
CPU Utilization	3.45%
IO Delays	0.00%
Physical Memory (497MB/135MB)	1.35MB
Swap Space (4.00GB/0KB)	0KB
HD Space root (7.87GB/648MB)	8.47%
HD Space data (15.26GB/166MB)	1.06%
Version (package/version/build)	pve-manager/1.3/4023
Kernel Version	Linux 2.6.24-7-pve #1 SMP PREEMPT Tue Jun 2 08:00:29 CEST 2009

Zunächst gelangen Sie in eine Übersicht des ArchivistaVM-Servers. Normalerweise werden Sie dort mehr RAM vorfinden, als dies hier ausgegeben ist. Um die Bildschirmscreens zu erstellen, haben wir einen ArchivistaVM-Server innerhalb eines ArchivistaVM-Servers installiert, daher die Bezeichnung 'QEMU'.

Klicken Sie nun auf 'Appliance Template'. Damit können Sie eine Installations-CD auf den Server hochladen. Bitte beachten Sie an dieser Stelle, dass dabei nur ISO-Dateien bis zu 2 GByte hochgeladen werden können. Grössere Dateien müssten mit WinSCP oder einem ähnlichen Programm auf den ArchivistaVM-Server übertragen werden. Dabei wäre der Pfad '/var/lib/vz/template/iso' zu verwenden.

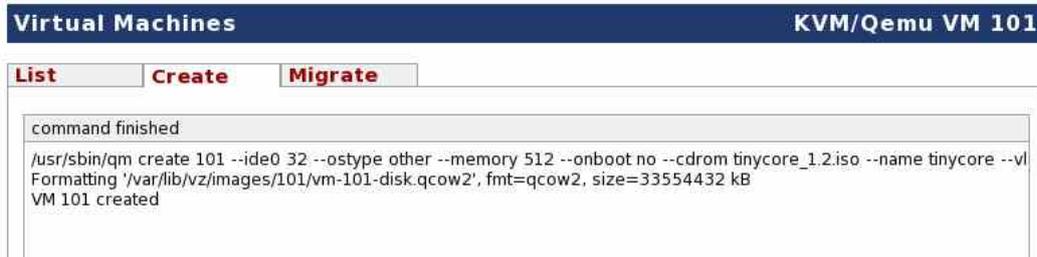
The screenshot shows the web interface for 'Archivista Virtual Environment 1.3'. At the top, it says 'You are logged in'. The main navigation menu includes 'Home | Logout', 'VM Manager', 'Configuration', and 'Administration'. The 'VM Manager' section has links for 'Virtual Machines' and 'Appliance Templates'. The 'Configuration' section has links for 'System' and 'Backup'. The 'Administration' section has links for 'Server', 'Logs', and 'Cluster'. The 'Appliance Templates' section is active, showing an 'Upload File' form with a file input field containing '/home/archivista/data/tiny/tin' and a 'Browse...' button, followed by an 'upload' button. Below this are two tables for 'OpenVZ Templates' and 'Qemu/KVM Templates and ISO images', both with columns for 'Description' and 'File name'.

Wählen Sie nun also eine ISO-Datei und klicken Sie auf 'upload'. Danach wird die Datei auf den Server geladen. Je nach Grösse der ISO-Datei bzw. Upload-Speed dauert dieser Vorgang einige Zeit.

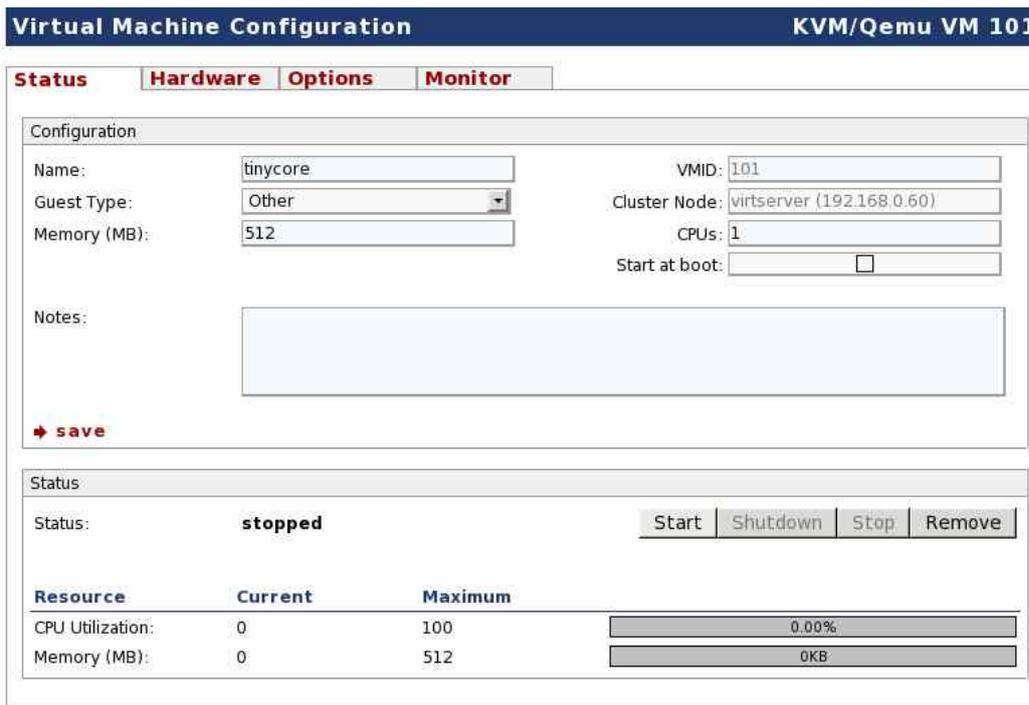
The screenshot shows the 'Virtual Machines' configuration page. It has tabs for 'List', 'Create', and 'Migrate'. An attention message states: 'Attention: This CPU does not support KVM virtual machines (no Intel VT / AMD-V support)'. The 'Configuration' section includes fields for 'Type' (Fully virtualized (KVM)), 'Installation Media' (tinycore_1.2.iso), 'Name' (tinycore), 'Disk space (GB)' (32), 'Memory (MB)' (512), 'VMID' (101), 'Cluster Node' (virtserver (192.168.0.60)), 'Start at boot' (checkbox), 'Disk type' (IDE), 'CPUs' (1), and 'Guest Type' (Other). The 'Network' section includes 'Bridge' (vmbro) and 'Network Card' (rtl8139). A 'create' button is visible at the bottom.

Nun können wir eine virtuelle Maschine installieren. Dazu wählen Sie bei 'Virtual Machines' den Reiter 'Create'. Nun müssen Sie bei 'Installation Media' entweder das CD-Rom-Laufwerk oder aber (in unserem Falle) die hochgeladene ISO-Datei bestimmen, damit die Installation

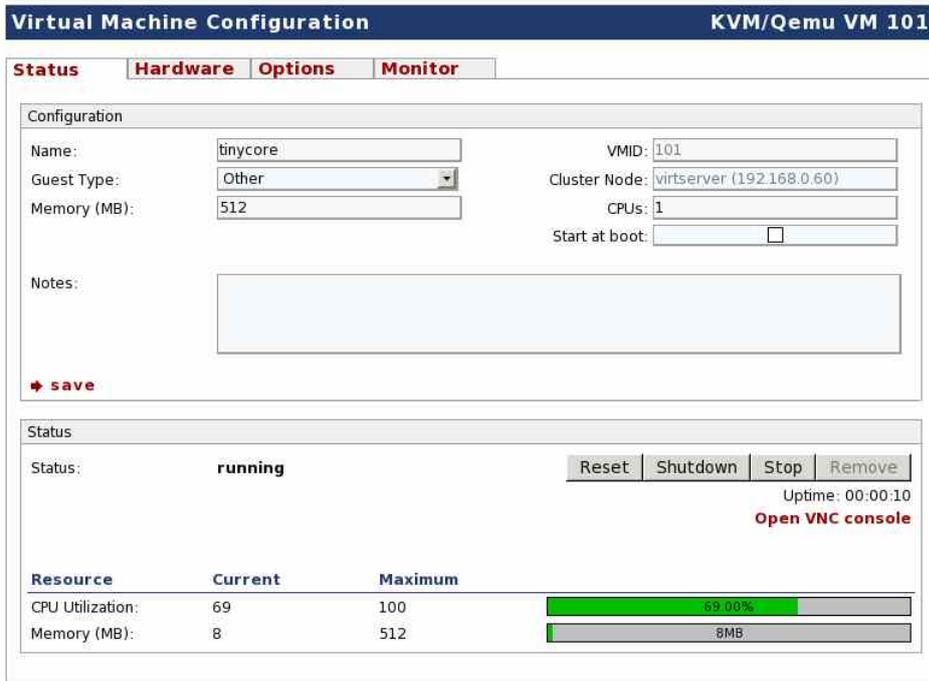
ab diesem Medium vorgenommen werden kann. Vergessen Sie auch nicht, der Instanz einen Namen zu vergeben, da ansonsten die virtuelle Maschine nicht eingerichtet werden kann.



Falls Sie alles richtig gemacht haben, erhalten Sie eine Bestätigungsmeldung mit den Einstellungen, die unsere Virtualisierungssoftware (KVM) verwendet, um die Instanz einzurichten.



Klicken Sie erneut auf 'Virtual Machines'. Sie sollten nun einen Eintrag mit dem Namen der soeben eingerichteten Instanz vorfinden. Klicken Sie auf diesen Eintrag und Sie finden (wie obenstehend) die Eckdaten der bereitgestellten Instanz vor. Nun können wir die Instanz durch Klick auf 'Start' starten.



Nach einer kurzen Zeit taucht in roter Farbe 'Open VNC console' auf. Klicken Sie auf diesen Link, so wird der Bildschirm des sich in Start befindlichen Betriebssystems (Instanz) in einem Fenster (untenstehend stellvertretend der Startvorgang von TinycoreLinux) dargestellt. Sie können nun ganz normal die Installation des Gast-Betriebssystems vornehmen.

```

Disconnect Options Clipboard Record Send Ctrl-Alt-Del Refresh
( - Tiny Core is distributed with ABSOLUTELY NO WARRANTY.
//\  Base Edition: A Busybox & Fltk minimal desktop.
v/_  http://www.tinycorelinux.com

Press <Enter> to begin or F2 and/or F3 for boot options.
boot:
Loading /boot/bzImage.....
Loading /boot/tinycore.gz.....ready.

Decompressing Linux... Parsing ELF... done.
Booting the kernel.
pci 0000:00:01.0: PIIX3: Enabling Passive Release
ide_generic: I/O resource 0x1f0-0x1f7 not free.
ide_generic: I/O resource 0x170-0x177 not free.
Booting tinycore_1.2
Running Linux Kernel 2.6.26-tinycore.
Starting udev daemon for hotplug support...Done.
Boot options checked.
Scanning hard disk partitions to create /etc/fstab
Setting Language to C Done.
Setting keymap to us Done.
Setting Timezone to UTC Done.

```

Und zum Abschluss noch dies: Sie können mit ArchivistaVM 32- und 64-Bit-Betriebssysteme installieren, selbstverständlich auch Windows-Instanzen. Sie werden erstaunt sein, wie schnell Sie mit ArchivistaVM eine Instanz installiert haben. Für eine aktuelle Ubuntu-Installation benötigen wir auf unseren ArchivistaVM-Servern i.d.R. nicht viel mehr als fünf Minuten. Eine Testinstallation von Windows7 war nach etwa 15 Minuten erledigt.

6 Erfahrungen mit den ersten ArchivistaVM-Servern

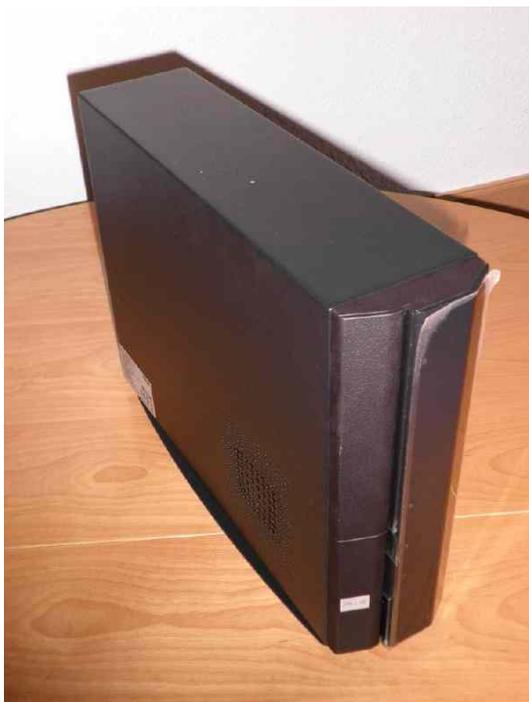
Einige Dinge haben uns bei den ersten Testservern mehrfach oder intensiv begleitet. Diese Punkte möchte ich kurz ansprechen:

6.1 Kostengünstige Test- bzw. Ersatzserver

Wie gesagt, Raid-Kontroller haben ihren Preis. Und die Ausfallsicherheit der Raid-Kontroller in Ehren, aber wenn die Maschine selber den Geist aufgibt (z.B. Netzteil), dann wäre es von Vorteil, eine zweite Maschine verfügbar zu haben. Diese sollte kostengünstig sein und soll auch für Tests verwendet werden können.

Wir haben lange nach einem Budget-Server gesucht und sind fündig geworden bei der ASUS-Pundit-Serie (siehe www.asus.com). Diese sind kostengünstig im Einkauf und stehen mittlerweile in einer zweistelligen Zahl bei Kunden im Einsatz.

Natürlich kriegen wir auf diesen Maschinen nicht ganz den Durchsatz hin, den wir mit einem schnellen RAID-Kontroller erreichen (ca. 70-100 MByte zu 120 bis 160 MByte. Allerdings relativiert sich dieser Wert im Falle eines Ausfalles einer Platte in RAID-Systemen, weil während dem Wiederherstellen des Plattenarrays die Geschwindigkeit deutlich tiefer liegt (Rückabgleichen der Inhalte benötigt Power).



6.2 Zweite Netzwerkkarte

Gute Mini-ITX-Systeme auf AMD-Basis gibt es nicht allzu viele. Und leider haben wir bis heute kein Board gefunden, das bei 2 internen GBit-Netzwerkkarten PCIe-x4 für die Raid-Karten anbietet. Es gibt zwar ein angekündigtes Board von Gigabyte, aber die Ankündigung datiert von der letzten CeBIT und gefunden haben wir dieses bisher nicht.

Obwohl wir zunächst sehr skeptisch waren, was den Durchsatz eines USB-GBit-Ethernetadapters betrifft, so erreicht ein solcher immerhin ca. 33 MByte (siehe www.linksys.com). Das ist nicht unendlich viel, aber doch immerhin oft soviel wie wir in den kleinen Hubs bei unseren Kunden zu Gesicht bekommen. Zum Vergleich: Eine gute Intel-Karte erreicht etwas über 70 MByte, also mehr als den doppelten Speed.



7 Lösung auf Stufe Mini-ATX

7.1 Redundanz auf Stufe Hardware

Wenn wir von Redundanz sprechen, dann haben wir bei der ArchivistaBox gute Erfahrungen mit einer zweiten Box gemacht. Normalerweise verwenden wir dort zwei gleiche Boxen.

Auch sind wir mit den Chenbro-Gehäusen sehr zufrieden. Allerdings schmerzt der beschränkte Ausbau bei Mini-ITX-Systemen doch etwas ein. Daher haben wir nach einer Lösung Ausschau gehalten, um in einer kompakten Box ein Mini-ATX-System mit vier externen Festplatten (Hot-Swap) unterzubringen. Das ist einfacher getan als gesagt.

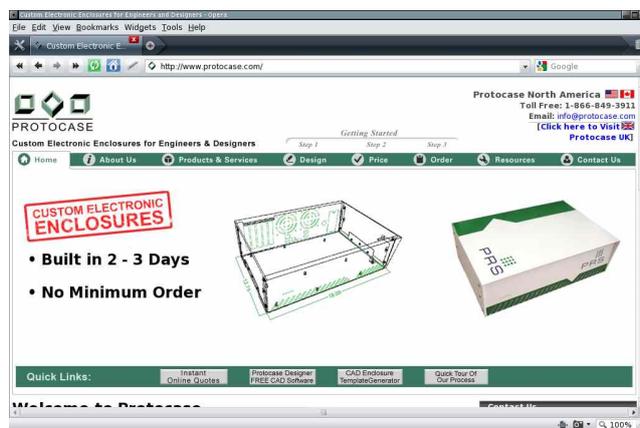
7.2 Diverse Versuche

Die kostengünstigste Lösung haben wir zunächst beim Gehäuse WingMA von Thermaltake (siehe www.thermaltake.com) gefunden. Das Gehäuse hat drei externe 5.25Einschübe.

Darin lässt sich bequem ein 4er oder 5er-Hot-Swap-Käfig (siehe z.B. www.jj-computer.com) einbauen. Allerdings befriedigte das Gehäuse von der Grösse her nicht ganz.

Ein weiterer Versuch mündete in einer Bestellung bei: <http://www.pcdesignlab.com/Product-Qv2E>. Allerdings sind die Gehäuse nicht ganz günstig (inkl. Transport ca. 250 Euro). Und leider haben die Gehäuse einen unschönen Nebeneffekt. Das Gehäuse ist etwas zu kurz geraten.

Letztlich fündig geworden sind wir bei Silverstone, siehe unter www.silverstonetek.com. Dort haben wir uns für das Gehäuse SG02-f entschieden. Allerdings muss dieses Gehäuse umgebaut werden.



Wer diesen Aufwand scheut, lässt sich das Gehäuse ganz einfach fertigen (siehe www.protocase.com). Dort kann gegen Bares ein individuelles Gehäuse bestellt werden. Das hat allerdings seinen Preis. Kosten pro Gehäuse zwischen 200 und 600 USD, Transport pro Gehäuse ca. 300 USD.

Ich hätte zwar gerne auf die Protocase-Variante zurückgegriffen, aber sie erschien mir doch etwas gar teuer. Daher habe ich mich für die Variante Sugo SG02-f entschieden.



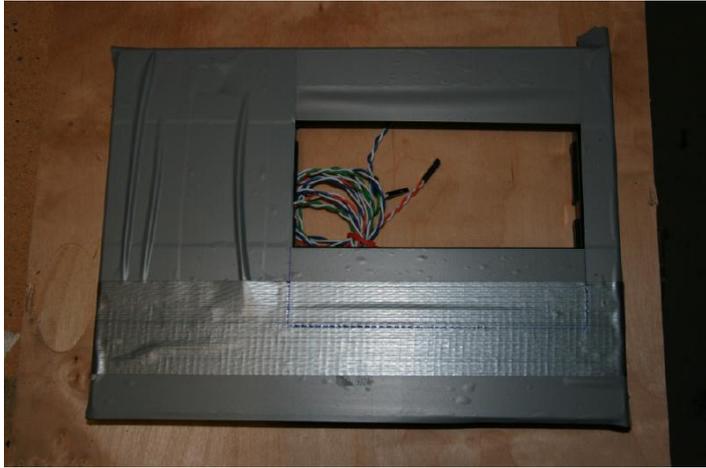
Sieht doch toll aus, oder? Nun, wie gesagt, es gibt ein kleines Problem. Das Sugo-Gehäuse hat bei der Auslieferung erst zwei externe Laufwerksschächte, das Gehäuse muss entsprechend umgebaut werden. Dies erfordert Fertigkeiten und entsprechende Maschinen, beide Voraussetzungen brachte mein Bruder idealerweise mit. Das Resultat lässt sich sehen, wobei mein Bruder ausdrücklich ausrichten lässt, falls denn jemand den Job übernehmen wolle, dann sei er gerne zu haben. Und ja, er hat sich mächtig ins Zeug gelegt. Ganze 10 Stück hat er in einer ersten Serie umgebaut, und dafür möchte ich ihm an dieser Stelle herzlichst danken.

7.3 Umbau des Gehäuses Silverstone SUGO SG02-f

Vorbemerkung: Ohne gute Werkzeuge sollte der Umbau nicht gewagt werden. Die Gefahr ist gross, dass dabei das Gehäuse ruiniert wird. Für den Umbau sollten etwa 1 bis 1.5 Stunden reichen.



Die Frontplatine muss zunächst abmontiert werden.



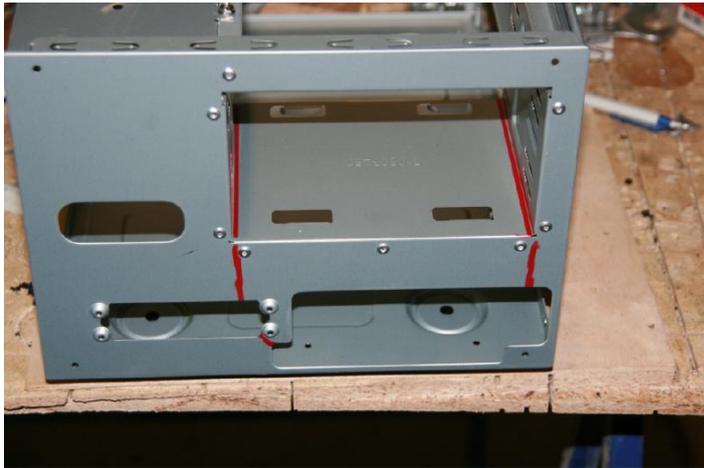
Die Vorderseite ist heikel, daher unbedingt mit Folie 'verpacken'.



Das Aussägen ist vielleicht der schwierigste Teil, da der Plastik auseinanderbrechen kann.



Nachdem wir den schwierigsten Teil geschafft haben, feilen wir die Kanten ab.



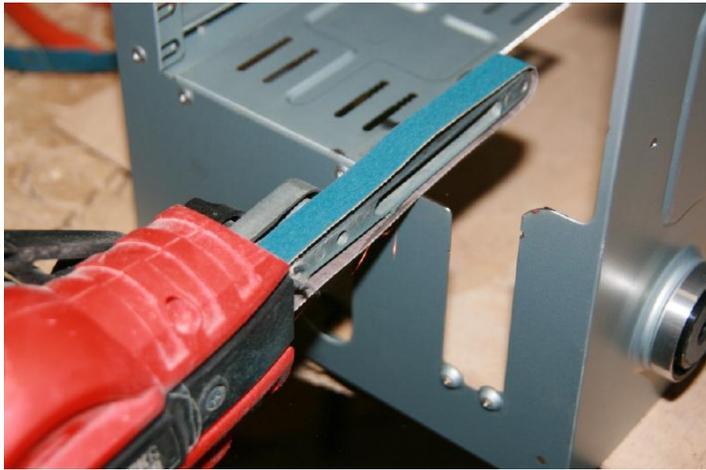
Nun markieren wir beim Gehäuse die Teile zum Ausschneiden.



Mit einem Scheibenschneider trennen wir den unteren Boden ab, damit wir den Hot-Swap-Käfig einbauen können.



Der untere Boden kann herausgenommen werden.



Auch hier gilt: Kanten schleifen, damit es später keine Schnittverletzungen gibt.



Nun benötigen wir einen handelsüblichen Hot-Swap-Käfig.



Bevor wir diesen einbauen, sollten Netzteil und Hauptplatine eingebaut werden.



Zum Schluss bauen wir den Hot-Swap-Käfig ein. Danach können wir die modifizierte Gehäuseklappe montieren.

7.4 Umgebauter Silverstone SUGO SG02-f im Vergleich



Obenstehend finden wir die verschiedenen Grössenverhältnisse unserer derzeitigen ArchivistaVM-Modelle vor. Links die günstigste Variante Budget, in der Mitte die aufwändige, aber flexibelste und relativ günstige Lösung Value und rechts die Summit-Variante, die ein Mini-ITX-Board erfordert.

Zugegeben, nun bin ich etwas gar auf Hardwarepfade gelangt. Es darf die Frage gestellt werden, was dies mit KVM und Debian zu tun hat? Sagen wir es so, es war nicht geplant, aber wer qualitativ gute Low-Cost-Lösungen aufbauen will, der wird nicht um diese Fragestellungen herum kommen. Und weil uns die Antworten dazu Wochen beschäftigt haben, sei dies auch in diesem Vortrag gewichtet. Muss ja nicht sein, dass diese Arbeit ein/e jede/r selber nochmals macht.

8 Migrationen und Stolpersteine

Nachdem wir die Hardware nun 'gebodigt' haben, möchte ich in diesem Kapitel Beispiele aus der Praxis anführen, die wir in den letzten Monaten erlebt haben.

8.1 Zügeln alter Maschinen

Wenn wir bei unseren Kunden die Server-Landschaften überblicken, dann findet sich im besten Fall irgendwo ein Ubuntu und (meist ausgelagert) ein Web-Server, der oft von externer Seite mit Linux betrieben wird. Der grosse Rest läuft unter Windows.

Anfänglich sind wir davon ausgegangen, dass nur Hilfe für das Zügeln der Maschinen benötigt würde. Es stellte sich heraus, dass selbst dies nicht ganz ohne ist, und dass es mehr Unterstützung bedarf, um gute Resultate bei den migrierten Maschinen hinzukriegen.

Zunächst zum Zügeln. Wir haben gute Erfahrungen mit Clonezilla (siehe www.clonezilla.org) gemacht. Noch besser waren die Erfahrungen, wenn wir die Maschinen direkt übers Netzwerk (mit ssh) migriert haben. Zwar war es möglich, die Maschinen auf externe USB-Platten zu speichern, allerdings passierte es mehr als einmal, dass bei diesen alten Maschinen die USB-Schnittstellen Probleme bereiteten. Dies ist beim Zügeln über die Netzwerkkarten nicht aufgetreten.

In einem Fall versagte CloneZilla. Der Kunde hatte einen Verbund von RAID-Kontrollern auf einem alten Dell-Rechner (3 Controller gebündelt). CloneZilla erkannte den RAID-Verbund nicht, d.h. CloneZilla wollte drei Festplatten zügeln, sodass in diesem Falle die Maschine komplett neu aufgebaut werden musste. Mehraufwand für diese Geschichte ca. 2 Tage.

Wichtig beim Zügeln von Windows-Maschinen ist zudem, dass vor oder nach dem Zügeln die erweiterten Gerätetreiber für die Festplatten deaktiviert werden, da Windows (ganz egal welche Version) ansonsten direkt beim Hochfahren einen Bluescreen von sich gibt. Zur Not kann nach dem Zügeln Windows mit einer Reparaturinstallation ab der Installations-CD erneut zum Leben erweckt werden.

8.2 Zügeln bereits virtualisierter Instanzen

Mittlerweile können ziemlich alle virtualisierten Festplattenformate (wie z.B. vmdk, qcow2, raw) mit KVM geöffnet werden. Sehr zu empfehlen ist zudem das Konvertieren mit `qemu-img`. Dazu ein Beispiel:

```
qemu-img convert -f vmdk win98.vmdk -O qcow2 win98.qcow2
```

Damit würde z.B. eine VMware-Festplattendatei ins Standard-Format von KVM umgewandelt.

8.3 Geschwindigkeit

Startzeiten von mehreren Minuten sind bei Servern keine Seltenheit. Nach der Migration dürfte die Startzeit kaum mehr als eine Minute betragen. Ein hohes Potential bei Windows-Instanzen haben wir über die Monate zusammengetragen und auf unserer Homepage www.archivista.ch unter 'Tipps zu Windows' (bei den Blogs) veröffentlicht.

Optimiert aufgesetzt kann ein Windows-Server 2003 in weniger als 20 Sekunden hochgefahren werden. Speedtechnisch sind keine spürbaren Unterschiede festzustellen. Am ehesten Probleme bereiten kann bei mehreren Instanzen der Durchsatz auf den Festplatten. Zwar bietet ein RAID-Kontroller einen Durchsatz von weit über 100 MByte, geteilt durch drei oder vier Instanzen sind wir allerdings 'nur' noch bei 30 bis 50 MByte.

8.4 Backup

Ein Server verschlingt schnell einige hundert GByte. Nicht zu empfehlen ist es, auf den Servern die alten Backup-Programme weiter zu verwenden. Bei mehreren Instanzen kann dies zu einer Überscheidung beim Backup führen und wenn alle Instanzen gleichzeitig wie verrückt Daten schaufeln, dann bremsen sich die Programme gegenseitig aus.

Sehr bewährt hat sich das Arbeiten mit externen eSATA-Platten. Im Moment erreichen wir einen Durchsatz von ca. 250GByte pro Stunde bzw. 1 TByte pro 4 Stunden. Dabei fahren wir den Server jeweils kontrolliert herunter und kopieren die Daten direkt auf den externen Datenträger. Generell nicht zu empfehlen ist der Einsatz von rsync bei Dateigrößen über ca. 20 GByte. Im schlechtesten Fall (d.h. bei Images über 50 GByte) benötigt rsync ein mehrfaches an Zeit als dies beim direkten Kopieren der Images ist.



Ebenfalls nicht bewährt hat sich der Einsatz des Raw-Formates, da dabei ja immer bereits beim Erstellen der virtuellen Partition der gesamte Platz reserviert wird. Folglich muss beim Backup auch die gesamte Datei geschrieben werden, selbst bei eSATA haben wir bei 600 GByte immer weit über 2 Stunden, obwohl vielleicht nur ca. 50 GByte belegt sind.

8.5 Backup mit Snapshots

KVM bietet einen sogenannten Snapshot-Modus mit. Damit kann der aktuelle Stand einer Instanz direkt in die virtuelle Festplatte gesichert werden, und dies erst noch im laufenden Betrieb. Technologisch betrachtet wird dabei der Festplatte versioniert und der Inhalt des Hauptspeichers in die Datei gesichert. Doch auch hier gilt, dass bei grossen Festplattendateien dieser Vorgang relativ lange Zeit in Anspruch nehmen kann und dass es beim Wiederherstellen zu beachten gilt, dass die Instanz danach auch zeitlich zurückgefahren wird.

Persönlich empfehlen würde ich die Snapshots daher eher vor Updates bzw. in der Entwicklung. Läuft etwas derart schief, dass die gesamte Maschine nicht mehr hochfährt, kann bequem zum letzten stabilen Zustand zurückgekehrt werden.

8.6 Rückspielen einer einzelnen Datei

Auch das ist vorgekommen. Der Kunde möchte genau eine Datei einer Sicherung zurück spielen. Beim qcow2-Format ist ein direktes Mounten der Festplatte meines Wissens nicht möglich. Es geht aber relativ einfach, indem die Datei ins raw-Format konvertiert wird.

```
qemu-img convert -f qcow2 win2003.qcow2 -O win2003.raw
```

Die Konvertierung sollte sinnvollerweise auf der Backup-Festplatte durchgeführt werden, um zu vermeiden, dass dabei gleich die 'scharfe' Festplatte mit den laufenden Instanzen vollgeschrieben wird. Danach können wir die Festplatte öffnen:

```
mount -t ntfs-3g -o loop,offset=32256 win2003.raw /mntpoint
```

Und damit dies auf der ArchivistaVM auch funktioniert, muss der ntfs-3g-Treiber natürlich auch installiert sein.

```
apt-get update
```

```
apt-get install ntfs-3g
```

8.7 Ein Server ist ein Server...

Und wenn wir gerade auf der Konsole sind. Bei einem Unterbau mit Debian können erforderliche Pakete ja jederzeit mit 'apt-get install xxx' installiert werden. Das hat sich in der Praxis als nicht unproblematisch erwiesen. Mal schnell einen Samba-Server aufgesetzt, kein Problem. Oder wie wäre es mit etwas KDE oder so? Und so kam es, dass beim Aktualisieren Maschinen ins Jenseits befördert wurden.

Dazu ein Beispiel: Die Backup-Platte wurde auf die Root-Partition gelegt, ein kurzes Skript fürs Backup der Samba-Dateien und voll war die Boot-Partition beim ersten Backup ohne

angeschlossene Festplatte. Immerhin, der Kunde konnte bei voller Root-Partition die Maschine noch auf einen zweiten Server im Live-Modus migrieren.

9 Das Projekt geht weiter

9.1 ArchivistaVM CD frei verfügbar

Ein Open Source-Projekt kann nur gedeihen, wenn ein gewisses Mass an Zeit an einem Projekt gearbeitet wird. Dank einem guten Kunden-Input konnten wir eine Lösung realisieren, die wir sonst wahrscheinlich nicht realisiert hätten. Dank mittlerweile bereits recht vielen installierten Systemen können wir an diesem Projekt auch in Zukunft weiterarbeiten.

Diese Arbeit geben wir gerne zurück. Zwar können wir die ArchivistaVM-Server nicht zum Schleuderpreis anbieten (darunter würde der Service leiden), aber ich hoffe doch, in diesem Vortrag aufgezeigt zu haben, was wo und wie bestellt und zusammengebaut werden kann.

Die ArchivistaVM-CD selber kann jederzeit über unsere Homepage www.archivista.ch bezogen werden, selbstverständlich sind dort auch die Sourcen verfügbar. Analog zur ArchivistaBox legen wir Wert darauf, ein Produkt so an die Community weiterzugeben, dass damit weder die Installation noch das Arbeiten mit einem Open Source Produkt schwieriger ist als dies bei Closed Source Produkten der Fall ist.

9.2 Mitarbeit gerne erwünscht

Und wenn jemand am Projekt mitarbeiten möchte, so ist dies gerne und jederzeit willkommen. Ich selber investiere im Schnitt ca. 1 bis 2 Abende die Woche für das Projekt, ohne dass dies durch ein Kundenprojekt abgedeckt ist. Und ich denke schon, dass eine Mitarbeit an einem Projekt, die über ein solches Mass hinausgeht, letztlich auch entlohnt werden sollte.

Mitarbeit heisst im übrigen nicht, dass Programmierkenntnisse erforderlich sind. Bessere Tests, ein gutes Handbuch, Community-Support, all diese Arbeiten kann ein/e jede/r User/in übernehmen.

Vielen Dank für die Aufmerksamkeit.