

# ArchivistaVM - Cloud-Virtualisierung auf dem Schreibtisch

Virtualisierung muss weder komplex, teuer noch langsam sein

## Contents

<b>1</b>	<b>Einleitung</b>	<b>2</b>	<b>4</b>	<b>Budget, Summit und Universal</b>	<b>26</b>
			4.1	Installation . . . . .	26
			4.2	DRBD-Cluster . . . . .	26
			4.3	Datensicherung . . . . .	27
			4.4	Support . . . . .	27
<b>2</b>	<b>Einfach, preiswert und leistungsfähig</b>	<b>3</b>	<b>5</b>	<b>Cloud inhouse, lohnt es sich?</b>	<b>28</b>
2.1	Hardware . . . . .	3	5.1	Server und Dienste in der Wolke	29
2.2	Software . . . . .	9	5.2	Daten in der Wolke . . . . .	30
<b>3</b>	<b>ArchivistaMini</b>	<b>13</b>	5.3	Wissen in der Wolke . . . . .	31
3.1	Installation . . . . .	13	5.4	Wissen einfach inhouse halten	32
3.2	Datensicherung . . . . .	16			
3.3	Cluster . . . . .	17			
3.4	Migration von Maschinen . . .	19			
3.5	Alles in allem... . . . . .	23			

© 25.11.2016 by Archivista GmbH, Homepage: [www.archivista.ch](http://www.archivista.ch)

# 1 Einleitung

Welchen Sinn ergibt es, eine eigene Cloud 'localhost' zu betreiben? Welche Anforderungen bestehen, um ein kleines DataCenter quasi auf dem Schreibtisch zu betreiben? Welche Hardware eignet sich dafür? Welche Software kommt zum Einsatz? Mit welchem Aufwand lässt sich das eigene kleine Rechenzentrum betreiben? Wie ausfallsicher ist eine solche Lösung?

All diesen Fragen (und einigen mehr) wird in diesem Vortrag nachgegangen. Dabei wird ArchivistaVM live vorgeführt, sowohl im Setup, Betrieb und bei der Datensicherung. Ziel der Präsentation ist es, dass Interessierte ArchivistaVM jederzeit selber auf einer bestehenden oder neuen Hardware aufsetzen und betreiben können.

Dies die Vorankündigung für den diesjährigen Vortrag. Natürlich kann hier argumentiert werden, eine Cloud sei weit mehr als eine Virtualisierungslösung, die auf dem heimischen Schreibtisch stehe. Dies mag sein, aber ohne Virtualisierung ergibt der Betrieb von Server-Diensten kaum einen Sinn. Wer mag bei der heutigen Komplexität von Software alles auf einer physikalischen Hardware aufsetzen bzw. wer mag für jede neue Applikation einen neuen physikalischen Rechner beschaffen.

Und natürlich kann eingewendet werden, dass sich der Aufbau auf dem heimischen Schreibtisch in Anbetracht der Hosting-Angebote (hier stünde der Rechner in der Cloud) doch gar nicht erst lohne. Wer die Anleitung z.B. bei Hetzner, siehe [wiki.hetzner.de/index.php/Proxmox\\_VE](http://wiki.hetzner.de/index.php/Proxmox_VE) betrachtet, wird schnell erkennen, von einfach kann kaum die Rede sein. Befehle wie 'apt-get install' sind Linux-Cracks wohl vertraut, auch die Installation von Debian an sich ist nicht umwerfend schwierig, aber eben auch nicht wirklich schnell und einfach.

ArchivistaVM ist genau aus diesen Gründen entstanden. Mal eben schnell komfortabel einen Rechner aufbauen, ohne zu einer physikalischen Hardware greifen zu müssen, das wärs. Nun gibt es mittlerweile viele Technologien zur Virtualisierung. Begriffe wie Paravirtualisierung, Client- und Server-Virtualisierung, Container etc., die Stunde wäre um, ehe die Begriffe erklärt wären. Wer dazu mehr erfahren möchte, unter [archivista.ch/vmtutor.htm](http://archivista.ch/vmtutor.htm) gibt es einen verständlichen Einstieg.

Nachfolgend wird von Virtualisierung nur noch in dem Sinne die Rede sein, als dass es darum geht, eine (hoffentlich) einfache Lösung mit ArchivistaVM präsentieren zu können.

## 2 Einfach, preiswert und leistungsfähig

Der Untertitel dieses Skriptes lautet: 'Virtualisierung muss weder komplex, teuer noch langsam sein'. Abstrakt betrachtet geht es fast immer darum, etwas einfach(er), günstig(er) und leistungsfähig(er) hinzukriegen. Manchmal ist etwas günstig, dafür nicht einfach, manchmal ist etwas sogar gratis, doch die Qualität lässt zu wünschen übrig, zuweilen ist etwas teuer, dafür stimmt die Leistung und manchmal stimmt grad gar nichts. Und trotzdem 'suggeriert' der Untertitel 'einfach, preiswert und leistungsfähig' zugleich. In diesem Kapitel soll es darum gehen aufzuzeigen, unter welchen Voraussetzungen einfach, preiswert und leistungsfähig machbar ist.

### 2.1 Hardware

Bei der Wahl der Hardware empfiehlt es sich, Standard-Komponenten zu verwenden. Wer beispielsweise für die Virtualisierung nach wie vor auf teure SAS-Platten und ebensolche Hardware-RAID-Kontroller setzt, wird entsprechend tiefer in den Geldsäckel greifen müssen, als wenn ein Software-RAID mit SSD-Platten zum Einsatz kommt, um nur ein Beispiel zu erwähnen.

#### 2.1.1 Prozessoren

Desktop-AMD-Prozessoren sind preiswert und die entsprechenden Hauptplatinen können mit 'fehlertoleranten' RAM-Speicherbausteinen (ECC) bestückt werden. Die Opteron-Serie von AMD steht bei unseren Kunden im Einsatz, die 12 oder 16 Kern-Prozessoren können in Rackgehäusen zwei- oder vierfach platziert werden, die Preise mässig moderat.

Bei Intel scheiden die Desktop-Prozessoren (i3/i5/i7) eher aus, weil keine ECC-RAMs verbaut werden können. Das Mass aller Dinge im Server-Bereich bildet die XEON-Baureihe, die Preise gehoben bis 'gesalzen'. Ein guter Kompromiss stellt die Prozessoren-Reihe XEON-D dar. Einmal weil damit kompakte Lösungen wie ArchivistaVM Budget, Summit und Universal möglich sind, vorallem aber, weil bei den entsprechenden Boards immer auch gleich zwei 10-GBit-Netzwerkkarten verbaut sind.

Modellnamen der Hersteller sind zuweilen widersprüchlich. In jedem Fall lohnt sich ein Blick auf die Werte bei [www.cpubenchmark.net](http://www.cpubenchmark.net), dort können die Leistungsmerkmale der CPUs einfach miteinander verglichen werden.

<b>Intel Xeon D-1587 @ 1.70GHz</b> <a href="#">+ Compare</a>	<b>Average CPU Mark</b>
<b>Description:</b> <i>Socket: FCBGA1667</i> <b>Clockspeed:</b> 1.7 GHz <b>Turbo Speed:</b> 2.3 GHz <b>No of Cores:</b> 16 (2 logical cores per physical) <b>Max TDP:</b> 65 W	<b>13489</b>
<b>Other names:</b> Intel(R) Xeon(R) CPU D-1587 @ 1.70GHz <b>CPU First Seen on Charts:</b> Q2 2016 <b>CPUmark/\$Price:</b> NA <b>Overall Rank:</b> 68 <b>Last Price Change:</b> NA	Single Thread Rating: 1197 Samples: 1* *Margin for error: High

Auch wenn ARM-CPU's in der Lage sind, virtualisierte Instanzen auszuführen, so dürfte dem entgegenstehen, dass bei ARM-Rechnern meist wenig Hauptspeicher zur Verfügung steht. Dies müsste zwar nicht zwingend so sein und entsprechende Angebote (zuletzt auch gerade von AMD) tauchen regelmässig auch immer mal wieder auf, nur (in Massen) lieferbar sind ARM-Server bis heute nicht.

**Empfehlung:** XEON-D mit 4, 8 oder 16 CPUs, siehe bei Hauptplatinen, bei eng begrenztem Budget kann es aber nach wie vor auch eine Desktop-CPU von AMD sein.

### 2.1.2 Hauptplatinen

Bedingt durch die gewählten CPUs sind die passenden Hauptplatinen (Mainboards) einzusetzen. Vor einigen Jahre gab es fast keine leistungsfähigen mITX-Platinen, heute gibt es gerade auch für die Intel-XEON-D-Plattform eine ganze Fülle von Platinen und dies durchaus von verschiedenen Herstellern. Und auch wenn die mITX-Boards nur einen Steckplatz für eine Erweiterungskarte besitzen, so wird dieser mittlerweile nicht mehr für die 10 GBit Netzwerkkarte(n) benötigt, da bei vielen Server-Platinen zumindest zwei 10 GBit Netzwerkkarten fest verbaut sind.

**Empfehlung:** Supermicro X10SDV-8C-TLN4F (8 CPUs / 16 Threads mit XEON D-1540, 2x1 GBit, 2x10 GBit sowie max. 128 GByte RAM für ca. 1200 Franken/Euro. Sowohl Supermicro wie andere Hersteller haben die XEON-D meist mit 4, 6, 8, 12 oder 16 Cores im Angebot. Kostengünstigere Variante: AMD CPU A-Serie A10-7860K mit ASUS Mainboard A88XM-Plus

### 2.1.3 Netzwerkkarten

Die schnellste 1-GBit-Netzwerkkarte in Ehren, doch bei maximal ca. 110 MByte pro Sekunde ist der Durchsatz erschöpft. Das sogenannte 'Bonding', dass Zusammenhängen mehrerer Netzwerkkarten, ergibt wenig Sinn, eine Intel-4-Port-Netzwerkkarte kostet mehr als eine Dual-

Port-10-GBit-Karte von Mellanox. Daher gilt, 10 GBit sind heute ein absolutes Muss, um leistungsfähige redundante Lösungen realisieren zu können.

Dies umso mehr, wenn 'normale' SSD-Platten mit SATA3 mittlerweile gute 400 MByte Durchsatz bieten. Um es hier nochmals klipp und klar zu sagen: Ohne 10 GBit bei den Netzwerkkarten ergibt eine SSD-basierte Disk-Lösung keinen Sinn. Wenn 8 Instanzen 8 CPUs teilen, dann mag das ok sein, wenn diese acht Instanzen jedoch im besten Fall 110 MByte mit einer 1 GBit-Karte teilen müssen, dann ergibt dies gerade noch ca. 13.75 MByte/Sekunde Durchsatz. Selbst ein Raspberry PI bringt da mehr Leistung mit einer einigermassen schnellen SD-Karte.

Nun gibt es bei 10 GBit zwei Arten von Karten. Erstens diejenigen, die auf der Basis von Glasfasertechnologie (SSF) arbeiten und zweitens jene, die mit dem 'üblichen' RJ45-Steckanschluss arbeiten. Erstere haben/hatten den Ruf, weniger Energie zu benötigen, letztere sind dafür abwärtskompatibel zu den verbreiteten 1 GBit Karten. War früher die Hitzeentwicklung bei den RJ45-basierten Karten ein Problem, so ist dies mittlerweile nicht mehr der Fall. Insofern können heute 10 GBit Karten auf der Basis des RJ45-Steckanschlusses durchaus empfohlen werden. Zu beachten gilt es einzig, dass nicht alle RJ45-Kabel mit 10 GBit arbeiten, Cat6a ist hier das Mass aller Dinge.

10 GBit Karten, die verbreitet sind, entstammen fast alle dem Hause Intel, die Preise bewegen sich bei einigen hundert Franken/Euro. Immerhin, mit Mellanox gibt es eine preiswertere Variante, die z.B. bei [deltacomputer.de](http://deltacomputer.de) ab ca. 200 Franken/Euro (d.h. für etwa die Hälfte einer Intel-Karte) erhältlich sind.



**Empfehlung:** 2x10 GBit Karte mit RJ45 reicht für einen 3-er Cluster, siehe auch bei Hauptplatinen.

## 2.1.4 Switches

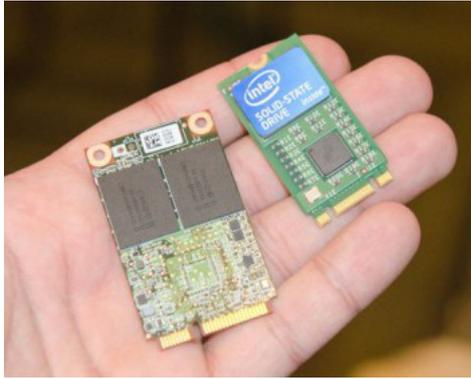
Vor einigen Jahren gab es keinen 10-GBit-Switch, der unter 10'000 Franken/Euro erhältlich gewesen wäre. Mittlerweile gibt es Modelle mit 6 oder 8 10-GBit-Anschlüssen, die für ca. 1'000 Franken/Euro zu haben sind. Nur, wer virtualisierte Instanzen gleichzeitig auf mehreren Rechnern verfügbar halten möchte, wird auch bei den Switches Redundanz benötigen. Es kann ja nicht sein, dass ein ganzer Rechnerverbund (Cluster) 'abraucht', nur weil ein 10 GBit-Switch den 'Geist' aufgibt.



Auf Switches ganz verzichtet werden kann dann, wenn die Rechner (Knoten) alle direkt miteinander verbunden werden. Auf den ersten Blick erscheinen solche Lösungen im Aufbau kompliziert(er) bzw. auch limitiert(er). Richtig ist, mit zwei 10 GBit Karten können maximal drei Rechner verbunden werden, mit vier 10 GBit Karten sind es fünf Rechner usw. Punkt-zu-Punkt-Verkabelungen haben allerdings den entscheidenden Vorteil, dass eben gar keine Switches notwendig sind. Die Verkabelung ist ferner nicht schwierig. 1 Port des 1 Rechners mit 2 Port des 2 Rechners verbinden. 1 Port des 2 Rechners mit 2 Port des 3 Rechners verbinden und 1 Port des 3 Rechners mit dem 2 Port des ersten Rechners verbinden. Die redundante Verkabelung mit 3 Rechnern und zwei Switches ist nicht einfach(er), denn neben den beiden Switches (je mit separatem Stecker für den Strom) werden anstelle von drei Kabeln deren sechs benötigt.

## 2.1.5 Festplatten

Bei den konventionellen Festplatten sind über die Jahre nur moderate Leistungszunahmen zu verzeichnen. Mehr als ca. 200 MByte dürften kaum mit einer einzelnen Platte machbar sein. Bei SSD-Platten sind es bei SATA3 bereits ca. 450 MByte. Noch mehr Durchsatz lässt sich nur mit Platten in der Bauform M2 oder PCI-basierten Festplatten erreichen. Beide Bauformen erreichen unter Umständen Geschwindigkeiten bis zu 2 GByte / Sekunde. Nur, da diese 'Dinger' intern verbaut werden, können diese im laufenden Betrieb nicht ersetzt werden. Ebenso fehlt es meist an der Anzahl Steckplätze, um einen Festplattenverbund (RAID) einzurichten.



Es mag eingewendet werden, dass bei einer Geschwindigkeit von 1 GByte / Sekunde oder noch mehr, ein Verbund an Platten (RAID) nicht mehr zeitgemäss ist. Dem ist entgegenzuhalten, dass RAID nicht nur zur Steigerung der Leistung verwendet wird, sondern vorallem auch, um die Platteninhalte redundant vorzuhalten. Insofern gilt: Hot-swap-fähige Platten ohne Redundanz ergeben keinen Sinn, ganz einfach deshalb, weil bei einem Ausfall einer Platte der gesamte Inhalt aller Platten verloren ist.

Bei Verwendung von SSD-Platten dürfte die Anzahl der Platten kleiner ausfallen, als dies beim Einsatz von konventionellen Platten der Fall ist. Über den Daumen gepeilt ersetzt eine SSD-Platte mit SATA3 drei bis vier konventionelle Platten. So ergeben Disk-Arrays mit 12, 16 oder 24 Platten mit SSD kaum mehr einen Sinn. Im Test ergab der Einsatz von sechs SSD-Platten einen Durchsatz von ca. 1.6 GByte. Rein rechnerisch müsste ein Durchsatz von ca. 2.4 GByte (6x400 MByte) resultieren, jedoch dieser Durchsatz ist von der Hauptplatine bzw. der Software (mdadm) auch zu verkraften. Ob der Einsatz von hardwarebasierten RAID-Kontrollern bzw. SAS-Platten sinnvoll ist, hängt vom Einsatzzweck ab. Üblicherweise dürfte der Preis einer SSD mit SAS12 jedoch höher liegen, als wenn zwei SAS6- alias SATA3-Platten eingesetzt werden, aktuell konnten denn auch nur wenige Modelle gefunden werden. Letztlich dürfte die höhere Verfügbarkeit der SATA3-Platten gegenüber SAS12 ein nicht zu unterschätzender Vorteil sein. Immerhin sind SAS-Systeme hot-swap-fähig einsetzbar und abwärtskompatibel zu SATA3.

## 2.1.6 Zusammenfassung

Manchmal sagen Bilder mehr als Worte. So sah ein 3-er-Cluster ArchivistaVM bisher aus:



Nun gibt es die gleiche Leistungsklasse im mITX-Format:



Die neuen Modelle ArchivistaVM erreichen plus/minus die gleiche Leistung, sind aber um Faktoren kompakter und leichter, über den Daumen gepeilt dürften die neuen Systeme fünfmal leichter sein und dreimal weniger Strom benötigen. In dieser Hinsicht passt eine solche Lösung durchaus auf den Schreibtisch.

**Hinweis:** Wer selber zwei preiswerte Rechner zu einem leistungsfähigen Cluster zusammenbauen möchte, dem kann die nachfolgende Hardware ans Herz gelegt werden: ASUS Mainboard A88XM-Plus mit AMD CPU A-Serie A10-7860K, 32 GB RAM (opt. 64 GB), Mellanox

MCX311A-XCCT 10 GBit Karten, 2xSSD 500 GB Samsung SSD 750 EVO, Preis pro Maschine mit Gehäuse: ca. 1000 Euro/Franken. Budgetbewusste verzichten auf die 10 GBit Netzwerkkarten in Verbindung mit SSD-Platten und erhalten so für ca. 600 Euro/Franken auch schon ein passables System. Wer Wert auf einen tiefen Strombedarf legt und über etwas mehr Budget verfügt, erhält mit den Mainboards auf Basis Intel XEON-D (1518 oder 1520) eine gute Wahl, eine performante 4-CPU (8 Threads) mit 35 bzw. 45 Watt Leistungsaufnahme, eine 8 Core mit 45 Watt sowie eine 16 Core CPU mit 65 Watt, solche Werte galten bislang bei AMD/Intel als unerreichbar.

## 2.2 Software

Neben der richtigen Hardware dürfte die Wahl der Software fast noch entscheidender sein. Letztlich nützt die tollste Hardware ja nur etwas, wenn auch die Software drauf läuft. Je neuer die Hardware, je aktualisierter muss in der Regel auch die Software sein, da ansonsten die Treiber für die Hardware fehlen. Eine Software kann aber auch zu 'jung' sein, der neuste Kernel, der vielleicht ein Jahr 'offiziellen' Support erhält, eignet sich nur bedingt, denn läuft das System mal stabil, so soll es nicht gleich wieder ersetzt werden müssen.

Mit Besorgnis muss zudem beobachtet werden, dass Linux (SystemD und SystemD-Udev sei Undank!) heute wesentlich 'verzahnter' ist als noch vor einigen Jahren. Nichts gegen das automatische Erkennen von Hardware durch Udev, nur nicht zwingend mit SystemD. Auch nichts gegen SystemD alleine, nur nicht zwingend mit Udev. Mittlerweile ist es faktisch so, dass diese beiden Dienste nicht mehr unabhängig laufen. ArchivistaVM verzichtet beim initialen Hochfahren auf beide Dienste, wer dazu mehr erfahren möchte:

[archivista.ch/cms/language/de/aktuell-blog/aktualisierter-unterbau](http://archivista.ch/cms/language/de/aktuell-blog/aktualisierter-unterbau)

Vereinfacht gesagt geht es darum, dass aufgrund des RAM-Modus (ArchivistaVM läuft im Hauptspeicher) einiges anders, vor allem viel schlanker, gemacht wird. Dabei ist die ArchivistaBox nicht alleine, auch Distributionen wie [alpinelinux.org](http://alpinelinux.org) oder [www.tinycorelinux.net](http://www.tinycorelinux.net) 'verzichten' gerne auf den allumfassenden autonomen Linux-Piloten. Soviel zum grundsätzlichen Konzept, nachfolgend sollen die wichtigsten Komponenten kurz vorgestellt werden.

### 2.2.1 Debian

**Vorbemerkung:** Die ArchivistaBox besteht seit dem Jahre 2005. Die ersten 5 Jahre arbeitete die ArchivistaBox mit einer selbstgebauten Linux-Distribution. Im Jahre 2010, bei der Umstellung der ArchivistaBox auf 64 Bit, reifte der Entschluss, dass eine standardkonforme Distribution die Grundlage für die ArchivistaBox bilden sollte.

Dies deshalb, weil es für die damalige Lösung (T2-Linux) nur wenig Dokumentation gab. In dieser Hinsicht hat sich der Wechsel zu Debian hin über die vergangenen Jahre alles in allem bewährt. Trotzdem entspricht die ArchivistaBox nicht zu 100 Prozent einer klassischen Debian-Installation. Hier kurz die wichtigsten Unterschiede:

- RAM-Modus: ArchivistaVM läuft komplett im Hauptspeicher (RAM-Modus), es werden ca. 300 MByte RAM benötigt. Das System wird folglich nicht installiert, sondern hochgefahren (kein Neustart nach Setup!).
- Installation: Das Einrichten des Systems erfordert keine Linux-Kenntnisse und läuft automatisiert ab.
- Kernel: Sämtliche benötigten Treiber sind in der Distribution integriert, nur so lassen sich RAM-Modus und automatische Installation überhaupt realisieren. Das komplette System ist in max. 1 Minute eingerichtet.

Ansonsten verhält sich ArchivistaVM wie eine klassische Debian-Installation.

## 2.2.2 QEMU/KVM

Qemu und KVM haben bis heute jenen Vorteil, dass die Technologie voll im Kernel integriert ist. Die Bedienung von Qemu/KVM erfolgt über die Konsole. Wer mag, kann die Gäste jederzeit direkt mit der Konsole verwalten. Zu beachten gilt es dabei einzig, dass anstelle des umständlichen Namens `qemu-system-x86_64` der Aufruf mit `kvm` erfolgt.

## 2.2.3 ArchivistaVM

Qemu/KVM enthält mittlerweile derart viele Optionen, dass das direkte Arbeiten mit Qemu/KVM wenig Spass bereiten wird. Daher gibt es für ArchivistaVM sowohl ein Web-Interface sowie eine Konsole mit dem Namen `qm`. Betreffend der Bedienung von ArchivistaVM sei auf das ausführliche Handbuch verwiesen:

[archivista.ch/cgi-bin/ridhb/search.pl?query=ArchivistaVM](http://archivista.ch/cgi-bin/ridhb/search.pl?query=ArchivistaVM)

An dieser Stelle sei hier einzig angefügt, dass ohne Angabe von Werten beim Aufruf von `qm` relativ ausführliche Hilfstexte angezeigt werden.

## 2.2.4 Cluster mit DRBD

Für einige Instanzen reicht ein Rechner für die Virtualisierung. Wenn auf einem mITX-Board 16 CPU-Kerne verfügbar sind, müssten an sich bereits ziemlich viele Instanzen notwendig sein, ehe zwingend zu einem Cluster gegriffen werden muss.

Nur, je mehr Rechner virtualisiert werden, desto gravierender ist der Ausfall eines Servers. Daher werden selbst in relativ kleinen Umgebungen vorzugsweise mehrere Rechner zu einem Verbund (Cluster) zusammengehängt. Im Zweifelsfalle lohnt sich ein 3-er-Cluster mit je 4 CPUs eher denn ein Rechner mit 8 oder 16 CPUs, weil mit einem Cluster der Ausfall eines Mitglieds verkraftet werden kann, bei einem Server müsste (für den Fall eines Ausfalles) immer gleich ein vollwertiger Ersatz bereitstehen. Das Versprechen eines Lieferanten, in 4 oder 8 Stunden vor Ort zu sein, bedeutet längst nicht, dass damit allfällige Probleme behoben sind, hier bietet sich ein Ersatz- und/oder Testrechner geradezu an, vorzugsweise als vollwertiges Mitglied des Clusters.

ArchivistaVM realisiert die Verbünde allesamt auf der Basis von DRBD. Mit der Technologie DRBD werden die Festplatteninhalte immer auf zwei verschiedenen Rechnern vorgehalten. Fällt Rechner A aus, so können die Daten der Instanzen des Rechners A immer noch auf dem Rechner B abgerufen werden. Daher ist es bei ArchivistaVM-Clustern sehr wichtig, dass der Durchsatz der Festplatten in Echtzeit auf einen zweiten Rechner übertragen werden kann. Unabhängig von ArchivistaVM lässt sich jedoch festhalten, dass bei Cluster-Lösungen die Daten immer grundsätzlich auf zwei oder mehreren Rechnern gespeichert werden sollten; nur so kann der Ausfall eines Rechners 'abgefangen' werden.

Cluster mit DRBD haben aber noch einen weiteren Vorteil. Bei der Datensicherung kann die zweite Kopie eines virtualisierten Rechners 'ausgeklint' werden. Damit kann eine vollständige Sicherung im Hintergrund erfolgen, ohne dass die Instanz während der gesamten Zeit der Sicherung ausfällt.

**DRBD-Konfiguration:** DRBD ist zwar im Kernel enthalten, jedoch bedingt ein neuer Kernel faktisch immer die dazu passende Client-Software (drbdadm). Bisher stand bei ArchivistaVM die DRBD-Version 8.3 im Einsatz, neu wird (bedingt durch den Kernel) die Version 8.9 mitgeliefert. Der Umstieg auf die aktuelle Version kann durchaus über bestehende Installationen erfolgen. Im Rahmen von ausgiebigen Speed-Tests musste jedoch festgestellt werden, dass mit Version 8.9 zwingend neue Optionen notwendig sind, um bei 10 GBit einen anständigen Durchsatz zu erreichen. Bei einem manuellen Umstieg ist die Datei **drbd.conf** bzw. sind dort die Einträge **net** und **disk** wie folgt anzupassen:

```
net {
    max-buffers 36k;
    sndbuf-size 1024k;
    rcvbuf-size 2048k;
}
disk {
    on-io-error detach;
    no-disk-flushes;
```

```
no-disk-barrier;  
c-plan-ahead 0;  
c-fill-target 24M;  
c-min-rate 80M;  
c-max-rate 720M;  
}
```

**Hinweis:** Wünschenswert wäre an sich, dass Konfigurationsdateien bei neuen Versionen kompatibel bleiben würden. Leider ist dem bei DRBD nicht so. Gerade solche 'Fallstricke' sind es, die den Einsatz einer Virtualisierungslösung extrem anspruchsvoll machen können. Wer erwartet schon beim Umstieg auf eine neuere Debian-Version, dass stunden- bis tagelange Recherchen für optimierte Config-Dateien notwendig sind? Genau darum wird bei ArchivistaVM die ganze Lösung aus einem Guss ausgeliefert, und daher macht es im Grundsatz auch keinen Sinn, immer gleich den letzten aktuellen Kernel aufzuspielen.

## 2.2.5 Alternativen

Historisch gesehen entstammt ArchivistaVM einem Fork zu Proxmox der Version 1.x. Warum es zum Fork kam, und warum ArchivistaVM kaum mehr Gemeinsamkeiten mit Proxmox aufweist, dazu hier mehr:

[archivista.ch/cms/language/de/aktuell-blog/archivistavm-2016vii](http://archivista.ch/cms/language/de/aktuell-blog/archivistavm-2016vii)

Wahrscheinlich könnten die Unterschiede heute nicht grösser sein. Proxmox bietet enorm viel Funktionalität, hat aber mittlerweile auch eine Komplexität erreicht, die beachtlich ist. Schnell mal einen Server, einen Cluster oder auch nur eine Instanz aufgesetzt, das war einmal.

Natürlich ist Proxmox nicht komplexer als dies der Fall bei den anderen 'Platzhirschen' ist, nur erscheint es zuweilen so, dass mit jedem Jahr bei den 'Leadern' die Produkte noch komplexer werden. Genau dies aber war/ist nicht das Ziel von ArchivistaVM, und daher erfolgte 2009 auch ein Fork. Dem sei hier angefügt, dass gerade Proxmox eine Leistung zur Verfügung, die es ansonsten wohl nirgends gibt, Respekt!

## 3 ArchivistaMini

ArchivistaVM gibt es in zwei Ausführungen. Einmal in der kommerziellen Version mit ArchivistaVM Budget, Summit und Universal, je sowohl mit/ohne Hardware. ArchivistaMini dagegen ist frei. Die Unterschiede in ein zwei Sätzen gesagt liegen darin, dass a) die Installation bei ArchivistaMini ohne grafisches GUI daherkommt, dass b) automatische Cluster auf zwei Knoten beschränkt sind und dass es c) für ArchivistaMini keinen Support gibt. Die ausgelieferten Sourcen und der Linux-Unterbau dagegen sind gleich, selbstverständlich auch die Config-Dateien.

Wer eine kommerzielle Version erwirbt, darf daneben beliebig viel und oft ArchivistaMini einsetzen, wer keinen Support mehr erwerben will und/oder benötigt, kann jederzeit auf ArchivistaMini zurückgreifen. Wer mit ArchivistaMini zufrieden ist, jedoch Unterstützung haben möchte, erhält diese auch, sobald er eine der Light-Version (ohne Hardware) von ArchivistaVM Budget, Summit und/oder Universal erwirbt, selbst wenn die ausgelieferte ISO-Datei gar nicht über ArchivistaMini installiert wird.

### 3.1 Installation

Entweder wird ArchivistaMini über [archivista.ch/avmini.iso](http://archivista.ch/avmini.iso) bezogen und das Setup erfolgt beim ersten Start oder es wird über [shop.archivista.ch](http://shop.archivista.ch) eine fixfertige CD für einen oder zwei Instanzen zusammengeklickt.

Bei beiden Varianten ist nach dem Download ein USB-Stick zu erstellen (es sei denn, ArchivistaMini wird virtualisiert getestet). Dafür gibt es verschiedene Tools, einige davon arbeiten besser, andere weniger gut. Informationen dazu finden sich hier:

[archivista.ch/cms/language/de/support/community/usb-stick-ab-iso-datei-windows](http://archivista.ch/cms/language/de/support/community/usb-stick-ab-iso-datei-windows)

[archivista.ch/cms/language/de/support/community/usb-stick-ab-iso-datei-linux](http://archivista.ch/cms/language/de/support/community/usb-stick-ab-iso-datei-linux)

Sobald der USB-Stick erstellt ist, kann über diesen gebootet werden. Auch dazu gibt es für ArchivistaMini eine gute Anleitung:

[archivista.ch/vmtutor.htm](http://archivista.ch/vmtutor.htm)

In einem Satz zusammengefasst: Um ArchivistaMini zu konfigurieren, wird beim ersten Bildschirm die IP-Adresse mit **ram ip.192.168.2.44** (Beispiel, mit Enter-Taste Installation starten) gesetzt, den Rest erledigt ArchivistaMini von alleine.



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

```
ram (IP address will not be assigned)
ram ip.192.168.4.200 lang.de keyboard.de (dns/gateway=192.168.4.1+german)
ram ip.192.168.1.15 submask.255.255.255.0 gw.192.168.1.1 dns.192.168.1.55
```

```
Press Return to install ArchivistaBox Environment.
ArchivistaMini (build 2016-11-16) - http://www.archivista.ch
boot: ram ip.192.168.2.44
```

**Hinweis:** Die Anleitung für Linux zum Erstellen des USB-Sticks kann auch zur Anwendung gelangen, um ArchivistaMini nach dem ersten Hochfahren in die erste Partion der Festplatte zu installieren, sodass ein Stick oder die CD zum Hochfahren danach nicht mehr notwendig ist. Bei den Modellen ArchivistaVM Budget, Summit und Universal ist dies im Standard auch bereits der Fall. Trotzdem bleibt der USB-Stick immer eine gute Wahl, weil damit jederzeit das System hochgefahren werden kann, selbst wenn die interne Platte selber defekt sein sollte. Dabei ist es möglich, ArchivistaMini ganz ohne die Platten zu booten (Option **ram ramonly**).

### 3.1.1 Passwörter und IP-Einstellungen

ArchivistaMini wird in der **Grundkonfiguration immer mit dem root-Passwort 'archivista'** ausgeliefert. Selbstverständlich können die Passwörter mit 'passwd' geändert werden. Da aber das gesamte System im Hauptspeicher (RAM) liegt, 'überleben' diese Änderungen einen Neustart nicht. Vielmehr müssen Einstellungen beim Hochfahren initialisiert werden.

ArchivistaMini sucht bei jedem Start nach dem folgenden Skript:

```
/home/data/archivista/cust/desktop/desktop.sh
```

Da die Passwörter bei Debian über die Dateien `/etc/passwd` sowie `/etc/shadow` verwaltet werden, sind diese beiden Dateien mittels dem Skript `desktop.sh` beim Hochfahren 'nachzurüsten'. Ferner können auch gleich die IP-Kenndaten gesetzt werden, sodass diese beim Hochfahren nicht mit `ram ip.192.168.0.x` manuell gesetzt werden müssen bzw. die CD über `shop.archivista.ch` mit den IP-Kenndaten erstellt muss. Dazu kann z.B. das folgende Skript erstellt werden:

```
#!/bin/sh
cp -rp /home/data/archivista/cust/desktop/passwd /etc
cp -rp /home/data/archivista/cust/desktop/shadow /etc
chown root.shadow /etc/shadow
cp -rp /home/data/archivista/cust/desktop/interfaces /etc/network
```

```
cp -rp /home/data/archivista/cust/desktop/resolv.conf /etc
/etc/init.d/networking restart
```

Es versteht sich von selbst, dass die im Skript eingetragenen Dateien wie obenstehend eingetragen vorhanden sein müssen. Wichtig für das korrekte Arbeiten von `desktop.sh` ist zudem, dass Ausführungsrechte gesetzt sind, dies wird mit `chmod a+x desktop.sh` erreicht. Nun den Server mit `shutdown now -r` neu starten.

### 3.1.2 Beispiel 'LXC'

Selbstverständlich können so auch weitere Pakete installiert werden. Am Beispiel LXC soll dies demonstriert werden. Dazu wird zunächst `apt-get update` und daran anschliessend `apt-get install lxc` eingegeben. Nach der Installation können die benötigten Pakete von `/var/cache/apt/archives/*.deb` z.B. ins Verzeichnis `debs` kopiert werden. Danach die untenstehende Zeile im Skript `desktop.sh` eintragen:

```
cd /home/data/archivista/cust/desktop/debs;dpkg -i *.deb
```

LXC wird damit beim Hochfahren automatisiert korrekt aufgebaut. Wenn nun jemand ein Web-Interface für LXC (siehe dazu auch die nachfolgenden Ausführungen) programmiert, dann wird dieses sicher Eingang in ArchivistaVM finden. Einschränkend sei hier angefügt, dass die Lösung nativ oder mit einem überschaubaren JavaScript-Toolkit daherkommen müsste. Warum ArchivistaVM hier etwas wählerisch ist, dazu hier mehr:

[archivista.ch/cms/language/de/aktuell-blog/archivistavm-2016vii](http://archivista.ch/cms/language/de/aktuell-blog/archivistavm-2016vii)

**Hinweis:** Der hauptsächliche Grund, weshalb LXC bzw. eine containerbasierte Lösung zur Virtualisierung von Maschinen bislang ArchivistaVM nicht hinzugefügt wurde, liegt darin, dass diese Technologien noch immer grossen 'Änderungen' unterworfen sind. ArchivistaVM hat nicht den Anspruch, möglichst schnell die absolut neuste Technologie zur Verfügung zu stellen, sondern eine Lösung zur Virtualisierung bereitzustellen, die über viele Jahre einfach den Dienst verrichtet. So liegen z.B. die Konfigurationsdateien der mit ArchivistaVM verwalteten Gäste seit der ersten Version im Jahre 2009 im gleichen Format vor, die verwalteten Gäste können problemlos über die ArchivistaVM-Versionen mitgetragen werden. Einfach heisst nicht, einfach mal wieder neu aufsetzen, einfach bedeutet bei ArchivistaVM einfach weiterarbeiten.

**LXC:** Für all diejenigen, die hier einwenden, mit der Version 1.x stünde eine stabile Version von LXC zur Verfügung, sei hier angefügt: Im Falle von LXC 1.0 endet der 'offizielle' Support per 1.6.2019, die Version 2.0 ist seit kurzem über die Backports von Debian Jessie vorhanden, die Version 2.0 ist allerdings nicht 100 Prozent kompatibel zur Version 1.0, siehe dazu [wiki.debian.org/LXC](http://wiki.debian.org/LXC). Mit 'LXC Web Panel' stünde an sich auch bereits eine relativ schlanke Web-Konsole zur Verfügung, allerdings lässt sich 'LXC Web Panel' aktuell nicht mit Version

1.0 LXC, und schon gar nicht mit Version 2.0 zusammen mit Debian betreiben, entsprechende Bugs sind seit dem Februar 2016 nicht gefixt, siehe dazu:

[recordnotfound.com/LXC-Web-Panel-claudyus-23169/issues](http://recordnotfound.com/LXC-Web-Panel-claudyus-23169/issues)

Ferner sei auf diesen Eintrag verwiesen:

[forum.proxmox.com/threads/moving-to-lxc-is-a-mistake.25603](http://forum.proxmox.com/threads/moving-to-lxc-is-a-mistake.25603)

Zwar erfordert LXC keinen gepatchten Kernel mehr, 100 Prozent kompatibel ist die Version 2.x jedoch nicht. LXC 1.0 wird faktisch nicht mehr weiterentwickelt, die Version 2.x jedoch erst seit kurzem verfügbar. Kann es Sinn und Zweck einer Lösung zur Virtualisierung sein, alle paar Jahre Inkompatibilitäten zu kreieren, die faktisch für den Kunden unlösbar sind? Native Virtualisierung mit Qemu/KVM bietet den Vorteil an, dass aufgesetzte Maschinen über Versionen lauffähig gehalten werden können, selbst beim Umstieg auf eine andere Technologie (z.B. XEN) können Gäste migriert werden. Dies ist weder bei LXC, bei Docker noch bei OpenVZ so der Fall. Insofern erscheinem dem Autoren die paar Franken/Euro pro CPU vertretbar, auch wenn hier (bzw. im Falle von Intel) faktisch monopolartige Preise zu zahlen sind.

## 3.2 Datensicherung

Einer der initialen Gründe, warum ArchivistaVM entstanden ist, liegt darin, dass passende Sicherungskonzepte für die Instanzen damals nicht gefunden werden konnten. Natürlich ist es lukrativ, die Maschinen kurz anzuhalten, um dann im laufenden Betrieb ein sogenanntes Abbild (Snapshot) zu erstellen. Wenn ein Hersteller aber schreibt: 'This mode provides the lowest operation downtime, at the cost of a small inconstancy risk', dann ist das für ArchivistaVM ein No-Go, zumindest im Web-Interface. Selbstverständlich wird niemand daran gehindert, die entsprechenden Befehle in QEMU aufzurufen, aber ins Backup-Konzept des Web-Interfaces gehört das Feature nicht.

Der einfachste Weg zu einem sicheren Backup führt nicht darum herum, die Maschinen runterzufahren, die Dateien zu sichern, und danach die Gäste wieder zu starten. Mit schnellen SSD-Platten können Gäste mit mehreren hundert GByte in ca. 10 bis 20 Minuten gesichert, selbst ein TByte benötigt plus/minus nicht mehr als eine Stunde.

**Hinweis:** Deutlich kürzere Sicherungen sind möglich, wenn ein DRBD-Cluster mit ArchivistaVM betrieben wird.

### 3.2.1 USB3 statt Cloud

Mit der Sicherung alleine ist der Job nicht erledigt. Ein gutes Sicherungskonzept umfasst mehrere Versionen (Tage). Je zeitkritischer die Informationen, je häufiger sind zudem die

Datenträger zu tauschen. Natürlich kann ein netzwerkfähiges Gerät (NAS) für die Sicherung verwendet werden, doch gilt es dabei zu beachten, dass eine gute Sicherung letzten Endes nicht online, sondern offline an einem anderen Ort (Safe) verfügbar gehalten werden muss.

Die Idee, virtualisierte Gäste irgendwo in der Cloud zu sichern, ist dabei keine so gute Idee, und dies selbst dann nicht, wenn die Daten verschlüsselt werden. Ein privater Schlüssel (Private Key) ist schnell kopiert, und bis z.B. 200 GByte aus der Cloud den Weg zurück auf die heimische Platte finden, dies könnte eine Weile dauern. Sichern und Zurückspielen mit USB3 ergibt einen Durchsatz von guten 200 MByte / Sekunde, womit die stündliche Leistung immerhin weit jenseits von 500 GByte liegen wird. Wichtig bei Sicherungen ist zudem, dass immer schön mit **cp** gearbeitet wird. Die Tools **scp** und **rsync** eignen sich nicht, weil damit die bei der Virtualisierung üblichen Festplattendateien (RAW-Format) nicht effizient verarbeitet werden können, dazu später mehr beim Migrieren von Gästen im Cluster-Betrieb. An dieser Stelle nur soviel: Das Sichern einer Instanz mit 40 GByte (maximale Grösse 140 GByte) dauert mit **cp** ganze 195 Sekunden, mit **rsync** dagegen 615 Sekunden; dies entspricht mehr als der dreifachen Sicherungszeit.

### 3.3 Cluster

Der einfache Aufbau eines Clusters wurde bereits anlässlich des linuxday.at im Jahre 2012 gezeigt, siehe dazu:

[www.archivista.ch/de/media/archivistavm-linuxday2012.pdf](http://www.archivista.ch/de/media/archivistavm-linuxday2012.pdf)

Im Grundsatz wird der Cluster über [shop.archivista.ch](http://shop.archivista.ch) zusammengestellt, indem das kostenfreie ArchivistaMini zweifach in den Warenkorb gelegt wird. Technologisch betrachtet wird dabei die Start-Datei **isolinux.cfg** angepasst. Nun sind die Download-Zahlen seit der Veröffentlichung von [archivista.ch/avmini.iso](http://archivista.ch/avmini.iso) wohl nicht zuletzt deshalb massiv angestiegen, weil keine Registrierung mehr notwendig ist.

**Hinweis:** Interessierte haben nachgefragt, warum es denn kein Formular ohne Registrierung gibt. Beim Erstellen einer angepassten CD werden auf dem Server (aktuell für 24 Stunden) ca. 90 MByte benötigt. Dieser Platz wäre schnell verbraucht, wenn ein 'Troll' mal eben einige Tausend CDs erstellen würde. Im aktuellen Regime gibt es pro Stunde eine neue CD, das ist für 'Trolle' nicht wirklich spannend.

Damit Cluster selber aufgebaut werden können, wird aufgezeigt, was beim Erstellen der automatisierten ISO-Datei abläuft. Angepasst werden **isolinux.cfg** und **syslinux.cfg** (Dateien sind identisch). Anbei ein Beispiel der Datei **isolinux.cfg**, um einem Cluster mit den IP-Adressen 192.168.0.31 und 192.168.0.32 aufzubauen:

```
DISPLAY boot.msg
```

```

label memtest
kernel memtest
label ram
kernel vmlinuz
APPEND initrd=initrd.img quiet pci=nocrs
    mac.0a:0b:0c:0d:0e:0f.eth0.192.168.0.31
    submask.255.255.255.0 gw.192.168.0.2 dns.192.168.0.2 lang.en
    mac.0a:0b:0c:0d:0e:8f.eth0.192.168.0.32 postcopy.postproc.sh

```

Zunächst werden anhand der Mac-Adressen die gewünschten IP-Kenndaten gesetzt. Beachtung verdient `postcopy.postproc.sh`. Mit `postcopy.sh` kann bei der Installation ein Skript von der CD in den Hauptspeicher kopiert werden. `postproc.sh` ist jenes Skript, dass daran anschliessend automatisch beim Hochfahren der Maschine abgearbeitet wird.

Die Datei `postproc.sh` hat den folgenden Inhalt:

```

#!/bin/bash
if [ ! -e /etc/drbd.conf ]; then
    /usr/bin/perl /home/cvs/archivista/jobs/cluster123.pl 192.168.0.31,32
    if [ ! -d "/var/lib/vz/template/iso" ]; then
        mkdir "/var/lib/vz/template/iso"
    fi
    if [ ! -d "/var/lib/vz/images" ]; then
        mkdir "/var/lib/vz/images"
    fi
fi

```

Die entscheidende Zeile zum Aufbau des Clusters lautet:

```

/usr/bin/perl /home/cvs/archivista/jobs/cluster123.pl 192.168.0.31,32

```

Wer möchte, kann somit den Cluster auch nach dem Aufbau von zwei einzelnen ArchivistaMini-Installationen mit diesem Skript einrichten. Damit dies klappt, ist das Skript zeitnah (max. 2 Minuten) auf beiden Rechnern zu starten. Den Rest erledigt ArchivistaVM automatisch. Damit ein Cluster 'klappt', müssen auf den Rechnern zwei identische Festplatten vorhanden sein. Bevor nach dem Aufbau des Clusters ein Neustart erfolgt, ist daran zu denken, dass die Dateien `drbd.conf` sowie `drbdcheck.conf` zu sichern sind, damit diese Einstellungen beim Hochfahren mit dem Desktop-Skript aufgespielt werden können. Ebenso notwendig ist das Sichern von `/etc/fstab` sowie das Starten des Clusters mit `/etc/init.d/drbdadm start`.

**Hinweis:** Der DRBD-Cluster in ArchivistaMini arbeitet mit einer Netzwerkkarte und jeweils zwei Festplatten. Das Einrichten weitergehender Cluster ist deutlich komplexer, ist aber z.B. hier beschrieben:

**Hinweis:** Um mit ArchivistaMini mit einer Netzwerkkarte einen guten Durchsatz zu erhalten, dürfte sich der Einsatz einer 10 GBit Karte lohnen.

## 3.4 Migration von Maschinen

Bislang wurden die Maschinen mit **rsync** über die öffentliche IP-Adresse von einem Knoten auf den anderen Rechner übertragen. Mit den immer grösser werdenden RAW-Dateien reifte der Entschluss, ob es nicht sinnvoller wäre, für die Migration auf die internen 10 GBit Karten zu setzen. Allerdings zeigte sich dabei, dass mit **rsync** nur ca. die doppelte Geschwindigkeit erreicht werden konnte, von den 10 GBit Karten her betrachtet hätte deutlich mehr Zuwachs an Durchsatz resultieren müssen (Faktor 6 bis ca. 9).

Kurz und gut, es musste ein neues Programm für die Cluster-Migration her. Dieses arbeitet direkt mit den DRBD-Laufwerken sowie (allenfalls, ab zwei Knoten) mit NFS-Freigaben. Übrig geblieben vom bisherigen Programm zum Migration von Gästen ist der Name, der Rest wurde neu entwickelt, die Parameter-Übergabe vereinfacht. Die Migration von Gästen kann entweder über das Web-Interface oder auch über die Konsole erfolgen, wobei das Web-Interface immer auf das Konsolen-Programm **qmigrate** zurückgreift. Dies war auch bisher bereits der Fall, doch war **qmigrate** abhängig von Server-Diensten, neu ist dies nicht mehr der Fall; **qmigrate** erledigt den gesamten Job in Eigenregie. Für eine Migration eines Gastes sind drei Parameter anzugeben: 1. Gast, 2. Ausgangsrechner und 3. Zielrechner. Beispiel:

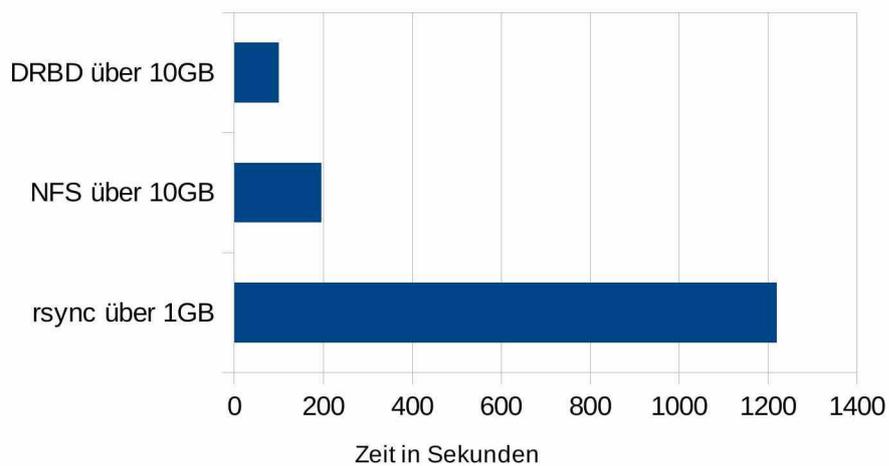
```
qmigrate 101 1 2
```

Mit obigem Befehl wird der Gast mit der ID 101 vom Rechner 1 auf den Rechner 2 übertragen. Dazu einige Messungen aus der Praxis.

### 3.4.1 Datei (15G) + Sparse-Datei (30G belegt/130G maximal)

Ergebnisse: rsync über 1GB: 1220 Sekunden, NFS über 10GB: 195 Sekunden, DRBD über 10GB: 100 Sekunden

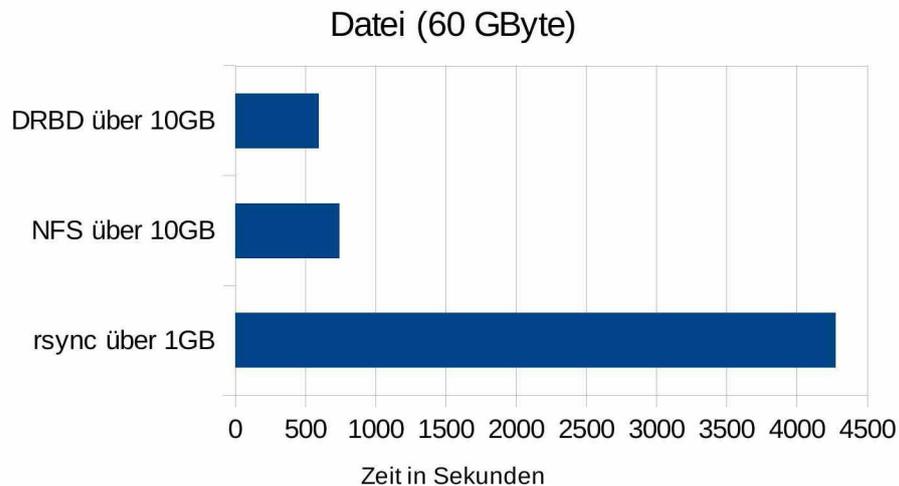
### Datei (15G) + Sparse-Datei (30G belegt/130G maximal)



Geschwindigkeitsvorteil 6.25 bis 12.2

### 3.4.2 Datei (60G)

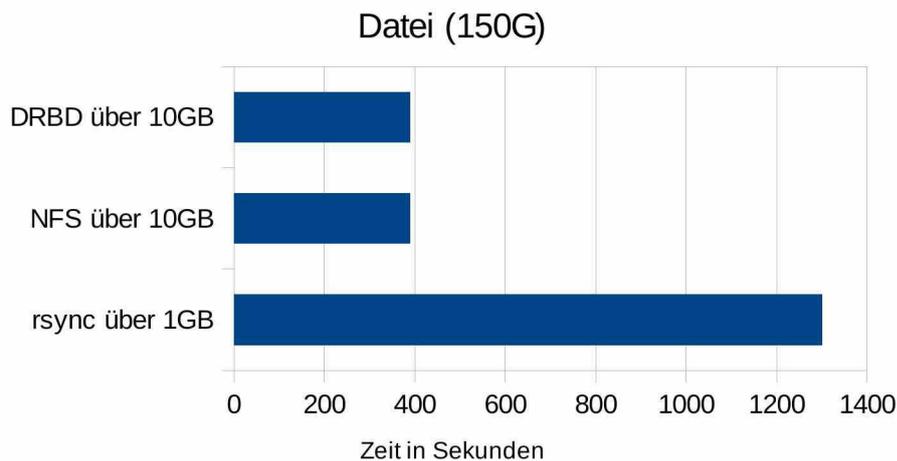
Ergebnisse: rsync über 1GB: 520 Sekunden, NFS über 10GB: 150 Sekunden, DRBD über 10GB: 150 Sekunden



Geschwindigkeitsvorteil 3.4

### 3.4.3 Datei (150G)

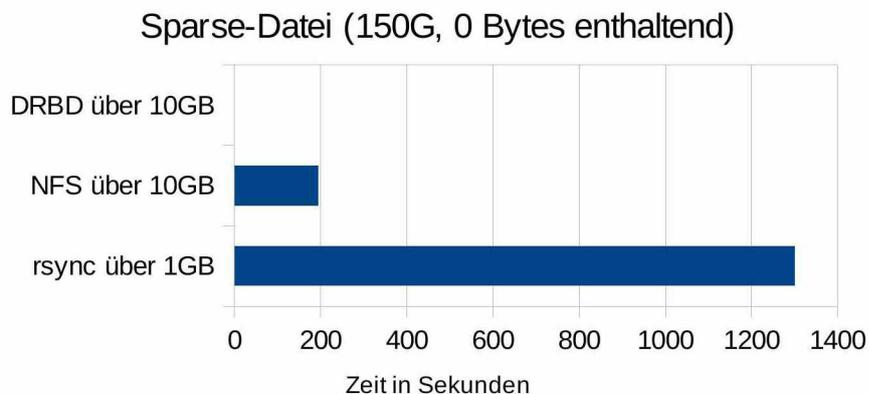
Ergebnisse: rsync über 1GB: 1300 Sekunden, NFS über 10GB: 390 Sekunden, DRBD über 10GB: 390 Sekunden



Geschwindigkeitsvorteil 3.33, kein Unterschied NFS/DRBD

### 3.4.4 Sparse-Datei (150G, 0 Bytes enthaltend)

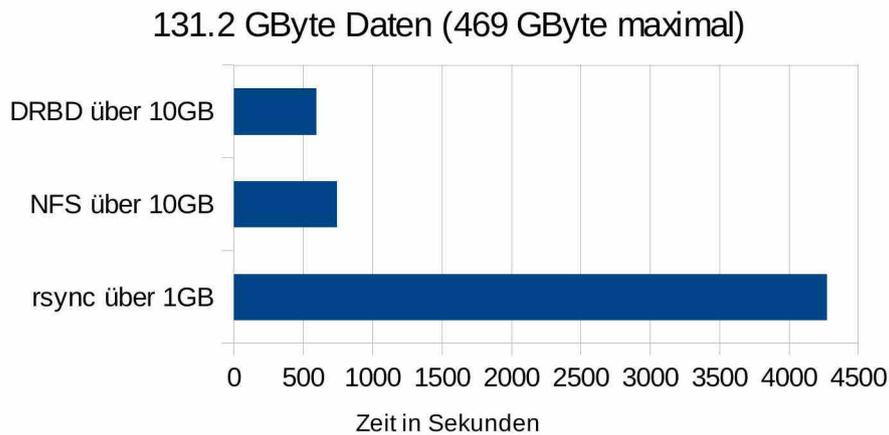
Ergebnisse: rsync über 10G: 1300 Sekunden, NFS über 10GB: 195 Sekunden, DRBD über 10GB: 0 Sekunden



Geschwindigkeitsvorteil 6.8 bis unlimitiert

### 3.4.5 Gesamtmessung über 9 Images (8 Instanzen)

131.2 GByte Daten (27 + 11 + 3.9 + 5.1 + 2.2 + 14 + 13 + 29 + 26), wobei sämtliche Abbilder maximal 469 GByte (80 + 11 + 3.9 + 5.1 + 32 + 120 + 13 + 120 + 84) einnehmen können. Folgende Ergebnisse konnten hier gemessen werden: rsync über 1GB: 4275 Sekunden, NFS über 10GB: 740 Sekunden, DRBD über 10GB: 595 Sekunden



Geschwindigkeitsvorteil 5.77 bis 7.18

### 3.4.6 Bewertung der Ergebnisse

Der Einsatz von DRBD und NFS bringen gegenüber **rsync** eklatante Geschwindigkeitsgewinne. Das Migrieren von 8 Rechnern mit 9 Images ist im Durchschnitt um mehr als den Faktor sechs schneller als bisher. Aufgrund der Messungen der beiden Dateien mit 60 und 150 GByte kann zudem festgestellt werden, dass das Kopieren grosser Dateien kein Problem darstellt. Zum Trost für **rsync** bleibt, dass auch **rsync** im Unterschied zu früher mit derart grossen Dateien umzugehen weiss und immerhin die Option **--sparse** für das Verschieben von RAW-Dateien mitbringt, wenn auch sehr ineffizient. Eine Datei mit 0 Bytes, die für 150 GByte ausgelegt ist, wird mit **cp** in einem Bruchteil einer Sekunde abgearbeitet, **rsync** benötigt dafür weit über 20 Minuten. NFS-Freigaben sind beim Kopieren mit **cp** extrem performant. Allerdings werden die leeren Blöcke leider nicht erkannt, daher die z.T. schlechteren Resultate bei NFS.

Und wer jetzt zum Taschenrechner (mit **python** geht es ganz galant) greift, und sich die Frage stellt, warum eine 150 GByte Datei noch immer sechs Minuten benötigt, wo doch vom Durchsatz her ca. 390 GByte (1 GByte / Sekunde) resultieren müssten, dem sei gesagt, dass die Messungen mit einem Produktiv-Cluster ArchivistaVM erfolgten, d.h. sämtliche Daten wurden beim Schreiben dreifach gespeichert, zunächst im Rahmen von RAID10 auf dem Zielrechner und anschliessend noch mit DRBD auf einen weiteren Rechner. Schneller geht es fast immer, fragt sich einfach wie sicher.

**Wichtig:** **qmigrate** steht ab Version 2016/XI zur Verfügung, bei vorherigen Versionen ist das Verhalten wie obenstehend mit **rsync** zu beobachten, sprich die Migration von Instanzen läuft gemächlich ab.

## 3.5 Alles in allem...

Dadurch, dass ArchivistaMini komplett im RAM arbeitet, mit 90 MByte erfrischend klein ist, sich daneben aber konform zu Debian Jessie verhält bzw. auch ein Debian Jessie ist, bietet ArchivistaMini gute Voraussetzungen, um eine Cloud auf dem heimischen Schreibtisch zu betreiben.

Die Konfiguration mit (Beispiel!) `ram ip.192.168.2.15` dürfte unschlagbar einfach sein, das anschließende Sichern der Einstellungen mit `desktop.sh` erfordert weniger fundierte Skills, sondern ein systematisches Herangehen an die Sache. Als Pluspunkt resultieren Lösungen, die mit wenig Aufwand automatisierbar sind. Sei dies über die Dateien `syslinux.cfg` oder `isolinux.cfg` der CD bzw. des Sticks oder über das Skript `desktop.sh`. Und wem dies auch zu komplex ist, für den steht `shop.archivista.ch` zur Verfügung, dort lassen sich die Maschinen zusammenklicken (Stichwort 2-er-Cluster).

1 x ArchivistaBox Mini

Please fill in IP specifications (all fields are mandatory):

Mac address:	<input type="text" value="0a:0b:0c:0d:0e:0f"/>
IP address:	<input type="text" value="192.168.0.250"/>
Submask:	<input type="text" value="255.255.255.0"/>
Gateway:	<input type="text" value="192.168.0.2"/>
DNS server:	<input type="text" value="192.168.0.2"/>

**Hinweis:** Um einen Cluster zu erstellen, muss ArchivistaMini zweimal in den Warenkorb gelegt werden, dann kann die zweite Mac- und IP-Adresse erfasst werden.

Natürlich ist es bei Open Source Projekten so eine Sache; Entwicklung und Dokumentation sind nicht kostenfrei zu haben, auch wenn sie kostenfrei angeboten werden. Nach mehr als sieben Jahren mit ArchivistaVM darf hier festgehalten werden: Was 2009 fast beiläufig mit einem Fork und einer CD von 700 MByte startete, hat sich in diesen Jahren mit einer ISO von 90 MByte, dem RAM-Modus, dem automatisierten Cluster-Betrieb mit DRBD, Backup- und User-Konzept, dem Verzicht auf Technologien, die über Jahre zu ernsthaften 'Sorgenkindern' heranwachsen könnten, nicht so schlecht geschlagen. Wem der Komfort von ArchivistaMini nicht reicht, möge einen Blick auf ArchivistaVM Budget, Summit und Universal werfen.

ArchivistaMini wird laufend weiterentwickelt, Beiträge sind jederzeit willkommen. Der Fokus wird aber auch in Zukunft dort bleiben, wo 2009 gestartet wurde: Einfach, preiswert und leistungsfähig. Bis 2020 ist Debian als Grundlage für ArchivistaMini gesetzt, was danach folgt, hängt davon ab, ob es dannzumal noch möglich sein wird, eine `initrd`-Datei ohne `SystemD` zu 'backen'. Wünschen würde sich der Autor für die Zukunft viele tollen ARM-Rechner

mit mehr RAM, mit 2 GByte macht ArchivistaVM unter ARM nur bedingt Spass, immerhin ArchivistaVM läuft schon heute problemlos unter ARM.

The screenshot shows the ArchivistaVM web interface. At the top, it says 'You are logged in as 'root' (Superuser)' and 'www.archivista.ch'. The main content area is titled 'ArchivistaVM' and contains a 'Local System Status' section. The status is 'Online' and shows the following details:

Local System Status ('avbox23')	Online
Uptime	22:13:30 up 00:20, load average: 0.09, 0.09, 0.13
CPU(s)	4 x ARMv7 Processor rev 5 (v7l)
CPU Utilization	11.46%
IO Delays	0.01%
Physical Memory (927MB/107MB)	107MB
Swap Space (99MB/0KB)	0KB
HD Space root (1.70GB/1.26GB)	78.89%
HD Space data (1.70GB/1.26GB)	78.89%
Kernel Version	Linux 3.18.9-v7+ #3 SMP PREEMPT Fri Mar 20 11:49:28 CET 2015
Options	

Wünschen würde sich der Autor an dieser Stelle ein Qemu mit echter Unterstützung über alle CPUs. Wenn ARM-Rechner vier, acht oder noch mehr Cores haben, dann sollten diese doch auch zur Anwendung kommen können. Aufgrund dessen, dass mit ARM und Qemu bis und mit Windows2000 bzw. ein betagteres x86-Linux mit einer CPU laufen, so wäre mit acht Kernen sicherlich noch viel mehr machbar.

### 3.5.1 Tipps und Tricks

Manchmal soll ein Gast gestartet werden, der sehr alt ist bzw. auf der aktuellen Hardware gar nicht laufen würde. Hier können bei den Optionen der Instanzen die Parameter 'KVM deaktivieren' sowie 'ACPI deaktivieren' gesetzt werden, ebenso kann der gewünschte Prozessor festgelegt werden (die untenstehenden Einstellungen passen auf Windows9x):

The screenshot shows the 'ArchivistaVM' configuration page. The 'Options' tab is selected, showing the following settings:

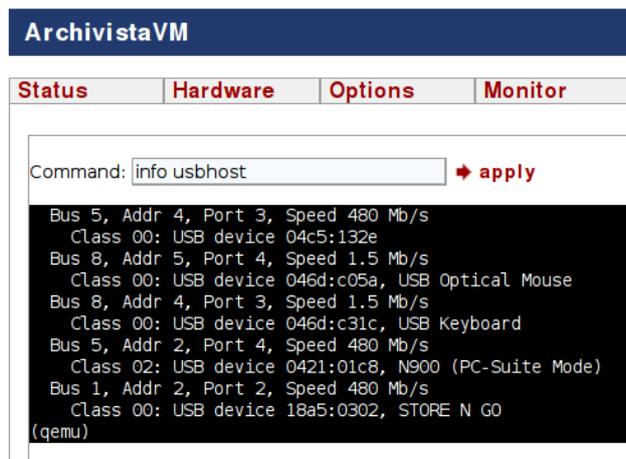
- Erstes Boot-Laufwerk: HDD IDE 0:0 (0.9764742851)
- Zweites Boot-Laufwerk: Floppy
- Drittes Boot-Laufwerk: CD-ROM
- CPU-Einheiten: 1000
- Echtzeituhr: default
- Grafikkarte: Cirrus Logic GD5446
- KVM deaktivieren:
- ACPI deaktivieren:
- CPU nach Start stoppen:
- Startdatum: [empty]
- Optionen: -cpu pentium2

At the bottom, there is a red arrow icon and the text 'speichern'.

**Hinweis:** Im Feld 'Optionen' können ganz allgemein beliebige Werte für KVM/Qemu eingetragen werden.

Dies ist z.B. dann notwendig, wenn innerhalb der virtualisierten Maschine wiederum KVM verfügbar sein soll. Die Optionen für Intel-CPU's lauten `-cpu core2duo,+vmx`, bei AMD sind es die Parameter `-cpu phenom,+svm`. Ebenfalls bei den Optionen können USB-Geräte hinzugefügt werden. Um z.B. einen Scanner durchzureichen, wird die Hersteller- und Geräte-Nummer benötigt.

Eine Liste der Geräte kann über den internen Monitor des Web-Interfaces (Reiter 'Monitor' bei den Instanzen) sowie der Eingabe `info usbhost` erfragt werden:



**ArchivistaVM**

**Status** **Hardware** **Options** **Monitor**

Command:  **→ apply**

```
Bus 5, Addr 4, Port 3, Speed 480 Mb/s
  Class 00: USB device 04c5:132e
Bus 8, Addr 5, Port 4, Speed 1.5 Mb/s
  Class 00: USB device 046d:c05a, USB Optical Mouse
Bus 8, Addr 4, Port 3, Speed 1.5 Mb/s
  Class 00: USB device 046d:c31c, USB Keyboard
Bus 5, Addr 2, Port 4, Speed 480 Mb/s
  Class 02: USB device 0421:01c8, N900 (PC-Suite Mode)
Bus 1, Addr 2, Port 2, Speed 480 Mb/s
  Class 00: USB device 18a5:0302, STORE N GO
(qemu)
```

**Hinweis** Um das korrekte Gerät zu finden, `info usbhost` einmal vor dem Einstecken des Gerätes aufrufen und danach nochmals, der neu hinzugekommene Eintrag entspricht dem eingesteckten Gerät. Alternativ kann auch über die root-Konsole mit `lsusb` nach Geräten gesucht werden.

Unser Fujitsu-Scanner hat die Geräte-Nummer `04c5:132e`. Das Gerät kann nun entweder direkt in der Konsole mit `usb_add host:04c5:132e` im laufenden Betrieb durchgereicht werden. Soll das Gerät mit dem Start eingebunden werden, kann es bei den Optionen (wie obenstehend CPU-Typ) mit `-usbdevice host:04c5:132e` fix einem Gast zugewiesen werden. Dabei gilt es zu beachten, dass ein Gerät nur einem Gast zugewiesen werden kann.

Der in ArchivistaVM integrierte Monitor zu Qemu/KVM arbeitet textbasiert. Dies hat den Vorteil, dass die Ergebnisse kopiert werden können, ein Abtippen also nicht erforderlich ist. Die Mächtigkeit des Monitors erschliesst sich einem erst, wenn mit `help` eine Gesamtübersicht abgerufen wird.

## 4 Budget, Summit und Universal

Neben der freien Version gibt es mit ArchivistaVM Budget, Summit und Universal drei Appliances, jeweils komplett montiert und frei Haus geliefert. Wer die Hardware lieber selber beschafft, kann mit der Light-Variante auch nur Support erwerben.



Bei ArchivistaVM Budget werden zwei Platten unterstützt, bei ArchivistaVM Summit sind es vier und bei ArchivistaVM Universal können zwischen sechs und 24 Festplatten eingesetzt werden. Die Eckdaten der Hardware können [shop.archivista.ch](http://shop.archivista.ch) entnommen werden. Nachfolgend wird in groben Zügen dargestellt, welchen zusätzlichen Komfort diese Versionen bieten. All die nachfolgenden Features können selbstverständlich auch mit ArchivistaMini manuell realisiert werden, **desktop.sh** ist dabei Freund und Helfer an jeder Stelle.

### 4.1 Installation

Vor der Installation werden die Mac- und IP-Adressen erfasst, ebenso wird ein allfälliges RAID festgelegt. Die Installation selber läuft komplett automatisiert ab. Bei der Installation wird gleich auch ein X-Server sowie Firefox installiert, womit direkt an der betreffenden Maschine mit grafischen GUI gearbeitet werden kann.

### 4.2 DRBD-Cluster

DRDB-Cluster-Lösungen werden immer mit Punkt-zu-Punkt 10 GBit Netzwerkkarten aufgebaut, dies im Unterschied zu ArchivistaMini, wo beim automatischen Aufbau des DRBD-

Clusters die erste Netzwerkkarte verwendet wird. Ebenfalls gibt es bei den Festplatten bis zu dreifache Redundanz (RAID10 auf dem ersten, RAID0 auf dem zweiten Rechner).

### **4.3 Datensicherung**

Bei der Datensicherung wird auf die USB-Festplatten auch ArchivistaVM mit auf die Festplatte geschrieben. Im Falle eines Totalausfalles (mehr als ein Knoten im Cluster) können die USB-Platten ArchivistaVM direkt ab der externen Festplatte hochfahren. Dieser Prozess eignet sich auch dann, wenn Sicherungen überprüft werden sollen, ohne die Instanzen zurück auf die internen Platten zu übertragen.

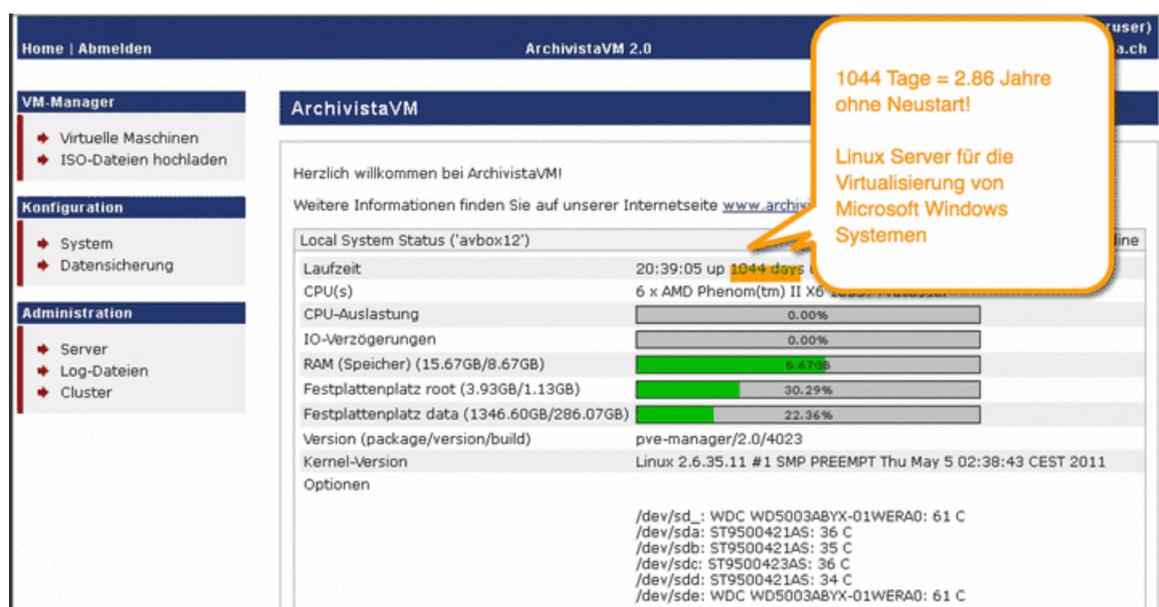
### **4.4 Support**

Im Unterschied zu ArchivistaMini, wo ein Community-Forum besteht, gibt es zu ArchivistaVM Budget, Summit und Universal bereits im Grundumfang Support und Wartung binnen acht Stunden, eine noch schnellere Reaktionszeit ist optional erhältlich. Grundsätzlich werden die Lösungen aber immer so aufgesetzt, dass ein Rechner im Cluster ausfallen kann, ohne dass die Gäste nicht abgesichert wären. Ferner wird die defekte Hardware kostenfrei ersetzt. Letzterer Punkt natürlich nur, wenn sowohl Hard- wie Software aus einer Hand geliefert wird, bei ArchivistaVM Light (nur Support für ArchivistaVM) ist der Kunde selber für seine Rechner verantwortlich.

## 5 Cloud inhouse, lohnt es sich?

Bei Server-Lösungen, gerade bei der Virtualisierung, stellt sich zunehmend nicht mehr die Sinn-, sondern die Kostenfrage. Dies ist gerade bei Server-Diensten so. Beispiel: Ein Kunde im Bereich Dokumenten-Management (DMS) hatte ArchivistaDMS virtualisiert im Betrieb. Das Archiv umfasst mittlerweile einige Millionen Seiten und einen Datenbestand mit mehreren Millionen Dokumenten (pro Jahr ca. 1 Million neue Seiten) mit über 1 TByte an Daten. Beim Aufbau der neuen Virtualisierungslösung (es musste das gleiche Produkt wie bisher sein) wurde ArchivistaDMS nicht mehr virtualisiert, das Angebot des Lieferanten für den virtualisierten Betrieb von ArchivistaDMS war derart hoch, dass locker zwei ArchivistaBoxen den Weg zum Kunden fanden. Streng genommen dürfte dies nicht passieren, eine virtuelle Lösung müsste an sich günstiger sein als Lösungen mit je eigener Hardware.

Leider hat sich besagter Kunde nicht für die Virtualisierung mit ArchivistaVM entschieden. Auch wenn er die Stabilität von ArchivistaDMS seit Jahren kannte, so konnte er sich eine Lösung ohne die preislich sehr teure Server-Hardware, die üblicherweise bei Virtualisierung offeriert und eingesetzt wird, einfach nicht vorstellen. Gerade gestandene IT-Profis sind nicht davor gefeilt, im Zweifelsfalle lieber zur teuren Edelkarosse denn zum Arbeitstier zu greifen. Wer mit einer preiswerten Lösung scheitert, hat zu hoch gepokert, wer mit einer teuren Lösung patzt, bessert nach, die Investition kann gar nicht einfach abgeschrieben werden. Immerhin, aus Kundenrückmeldungen ist bekannt, dass ArchivistaVM-Server über Jahre ohne Unterbruch im Produktivumfeld laufen, das Arbeitstier also durchaus robust arbeitet:



Nun ist es nicht so, dass kleinere bis mittlere Firmen (KMUs) im Grundsatz über Budgets für 'Edelkarossen' verfügen. Viele Unternehmen bekunden je länger je mehr Mühe, die Finanzen

für eine hochkarätige Inhouse-IT aufzubringen. Da klingt der Gang in die Wolke verlockend, nicht zuletzt auch, weil ja alle darüber reden und entsprechende Angebote mit tiefen Kosten nicht minder locken. Doch was bringen solche Angebote mit sich?

## 5.1 Server und Dienste in der Wolke

### 5.1.1 Server bei Hosting-Firmen

Anbieter von Root-Servern gibt es viele. Bei den Angeboten hat sich beim Preisgefüge erstaunlich wenig getan. Ein Root-Server mit 4-CPU's kostet auf das erste Jahr hochgerechnet um die 800 bis 1000 Euro, dies bei preisgünstigern Anbietern. Hosting-Unternehmen, die sich ausschliesslich an Firmen richten, rechnen meist nach Speicherbedarf und CPU's ab, hier können die Preise auch deutlich höher liegen. Für den Preis im ersten Jahr beim Hoster kann bereits ein eigener Server angeschafft werden, die Lebensdauer wird ohne weiteres drei bis fünf Jahre betragen. Konnte vor Jahren noch argumentiert werden, es fehle an der Leitung für die Anbindung ans Internet oder hochperformante Server seien ohne Server-Räume mit Kühlung nicht machbar, so ist dies heute nicht mehr der Fall.

Preisgünstige Internet-Anbindung gibt es längts und wer bei einer Leistungsaufnahme eines 8 oder 16-Core-Rechners bei 45 oder 65 Watt noch einen Server-Raum mit allen Schickanen betreibt, ist letztlich selber schuld. In diesem Sinne hat die Cloud auf dem heimischen Schreibtisch sehr gute Karten, dies alles war vor einigen Jahren noch undenkbar, mehr als 4 CPU's mit 95 Watt waren damals nicht machbar, performante Systeme benötigten gut und gerne 300 Watt und mehr, von den eher dürftigen bzw. teuren Leitungen einmal ganz abgesehen.

Immerhin kann bei root-Server-Angeboten heute festgestellt werden, dass fast alle Linux-Derivate zur Verfügung gestellt werden. Aber, im Unterschied zu ArchivistaVM, werden die Maschinen meist von Hand installiert, als stünde der Rechner lokal. Selten können eigene ISO-Dateien für die Installation hochgeladen werden, der Betrieb von ArchivistaMini in der Cloud beispielsweise dürfte aus diesen Gründen nicht ganz einfach sein. Dem sei angemerkt, dass ArchivistaMini auch in die erste Partition nach der Installation einer beliebigen ISO installiert werden kann (siehe dazu unter 3.1), alleine der Weg dorthin erscheint etwas gar sperrig.

Bei der Verfügbarkeit werden Angaben um die 99 Prozent gemacht, dass dies nicht wahnsinnig hoch ist, hier zur Erinnerung die 'nackten' Zahlen:

- **Verfügbarkeitsklasse 2:** 99 Prozent = 438 Min/Monat bzw. 7:18:18 Std/Monat = 87,7 Std/Jahr, d.h. 3 Tage und 15:39:36 h.
- **Verfügbarkeitsklasse 3:** 99,9 Prozent = 43:48 Min/Monat oder 8:45:58 Std/Jahr.

- **Verfügbarkeitsklasse 4:** 99,99 Prozent = 4:23 Min/Monat oder 52:36 Min/Jahr
- **Verfügbarkeitsklasse 5:** 99,999 Prozent = 26,3 Sek/Monat oder 5:16 Min/Jahr
- **Verfügbarkeitsklasse 6:** 99,9999 Prozent = 2,63 Sek/Monat oder 31,6 Sek/Jahr

Eine Verfügbarkeit jenseits von 99,9 Prozent lässt sich kaum finden. Ob dies mit ArchivistaVM erreicht werden kann, dies ist eine andere Frage. Immerhin, dank dem RAM-Modus wird ArchivistaVM nach einem Stromausfall stabil hochgefahren, weil ja eben gerade kein 'crash-gefährdetes' Root-Datei-System zur Anwendung kommt.

### 5.1.2 Cloud-Dienste einfach integrierbar?

Von ganzen Servern in der Wolke zu unterscheiden sind Dienste, die in der Wolke ausgelagert werden. Gerade webbasierte Lösungen sind verbreitet. Nur, lokal laufende Dienste lassen sich in der Regel einfach miteinander vernetzen, dies ist bei Cloud-Diensten nicht so einfach. Dazu sei als Beispiel erwähnt, dass bei einer ERP-Lösung der automatisierte Druck von Rechnungen mit Einzahlungsscheinen zum lokalen Drucker ein komplexes Unterfangen darstellen wird, das Runterladen einer PDF-Datei reicht dabei nicht, nach jeder Rechnung muss der Druckerschacht korrekt angesteuert werden, ehe der Einzahlungsschein (mit Spezialpapier) korrekt bedruckt werden kann.

Manchmal existieren für Cloud-Dienste Schnittstellen, über die automatisiert Jobs abgearbeitet werden können (API = Application Programming Interface). Ob damit z.B. die obige Aufgabenstellung einfach(er) gelöst werden kann, darf bezweifelt werden; anstelle einer Druckerkonfiguration ist zusätzlich zur Einrichtung des Druckers ein Skript zu erstellen. Dabei entscheidet der Cloud-Anbieter, wann, ob und wie dieses API lauffähig ist, die Abhängigkeit dadurch hoch.

## 5.2 Daten in der Wolke

Daten von Diensten, die in der Cloud laufen, können zwar fast immer im Sinne einer Sicherung auf den heimischen Schreibtisch übertragen werden. Trotzdem, auch hier bleibt die Abhängigkeit gross. Eine ERP-Lösung, die nur in der Cloud läuft, kann absolut betrachtet nicht mehr lokal in Betrieb genommen werden. Im besten Falle können irgendwelche Listen im Excel-Format exportiert werden, im schlechteren Fall liegen undokumentierte Daten vor, die sich nicht weiterverwenden lassen. Schnell eingeführt ist das eine, einfach konvertiert das andere. Stellvertretend für viele hier die Information betreffend den proprietären Formaten von Google: [de.wikipedia.org/wiki/Google\\_Docs,\\_Sheets,\\_Slides\\_und\\_Forms](http://de.wikipedia.org/wiki/Google_Docs,_Sheets,_Slides_und_Forms)

Zitat: 'Dazu ist anzumerken, dass die Anzeige/Konvertierung nicht immer vollständig und fehlerfrei erfolgt (meist sind es deutlich sichtbare Formatierungsfehler).' Kurz und gut, eine jede Kontrolle über die Daten ist dabei weder vorhanden noch existent.

Aber selbst wenn die Formate offengelegt sind, so übernimmt niemand eine Garantie, was mit den Daten passiert. Daten in der Wolke sind per Default online. Ein Passwort besteht aus ein paar Zeichen, nicht mehr und nicht weniger. Mittels Hash-Signaturen lässt sich wohl feststellen, ob Daten geändert wurden, nicht aber, ob sie eingesehen wurden bzw. wohin diese gelangten. Vertrauliche Informationen gehören nicht in die Wolke, genauso wie die Wolke nicht auf den heimischen Desktop gehört:

[hn.premii.com/#/article/12457067](http://hn.premii.com/#/article/12457067)

Um Missverständnissen hier vorzubeugen, damit sei nicht gesagt, dass DropBox den Mac hackt, vielmehr sei hier die Frage erlaubt, warum ein Cloud-Dienst wie DropBox überhaupt erweiterte Rechte für den lokalen Desktop benötigt. Dazu ein Beispiel: ArchivistaDMS verrichtet bei einer Fachabteilung eines Schweizer Kantons seit Jahren gute Dienste. Die Datensicherung lief jahrelang auf ein NFS-Laufwerk, bis das Laufwerk ohne Vorankündigung abgehängt wurde. Die Sicherung wurde mit hohen Kosten wieder lauffähig gemacht, einige Monate später lief sie erneut nicht mehr. Nochmals drohten hohe Kosten, der Chef der Fachabteilung folgte unserer Empfehlung, die Sicherung lokal mit Festplatten zu erledigen. Damit der Kunde die Sicherung selber überprüfen kann, wird dem Kunden ein kleines Böxli (RASPI) geliefert. Böxli ans Netz gehängt, das ganze Büro lahmgelegt. Drei Mitarbeiter der Security-IT vor Ort, ein Fremdgerät, das ginge unter keinen Umständen, das sei ein ernstes Security-Problem. Nach Stunden der Arbeit, die NFS-Sicherung läuft noch immer nicht, das Böxli mit viel Ping und Mac-Adressen-Nennung läuft, zufällig (da anwesend) der Diskussion des Chefs gelauscht: 'Ich schick die Datei mit DropBox'. Hoffen wir mal, dass die Security da mithält. Abends im Geschäftsbericht der kantonseigenen IT-Firma gelesen: 775 Server bei ca. 450 Mitarbeitern bei einem Kanton mit 90'000 Einwohner/innen.

## 5.3 Wissen in der Wolke

Wissen ist ein Gut, das langsam erarbeitet werden will, das es zu pflegen gilt. Natürlich kann die interne IT mit dem Gang in die Wolke allfällig wegrationalisiert werden. Nur ist dieses Wissen danach für alle Zeit verloren. Sollte später wieder eine interne IT notwendig sein, dürfte die Wolke gar zu Hagel und Donner mit hohen Kosten verkommen.

## 5.4 Wissen einfach inhouse halten

Die Kompetenz einer Firma, die IT inhouse zu führen, mag dann und wann mit Frustrationen verbunden sein, weil Komplexität und Kosten zum 'Schlaghammer' ausarten. Open Source und Linux bieten hier die Möglichkeit, dank Offenheit und guter Verfügbarkeit eine Autonomie zu erhalten, die mit proprietären Lösungen so nicht machbar ist.

Was bei und mit Linux mitunter fehlt, ist die Einsicht, technologisch einfache Lösungen zu implementieren. Nur weil es rein technisch machbar ist, ergibt es noch lange keinen Sinn, Software bis zur Unkenntlichkeit zu patchen. Software darf auch einfach sein, immer mehr Optionen bedeuten nicht zwangsweise mehr Komfort. Einfach heisst irgendwo auch verzichten, Mass zu halten, um sich den Kernaufgaben widmen zu können. Genau darum gibt es ArchivistaVM, Virtualisierung ja, Komplexität und hohe Kosten nein.



Kontakt: Urs Pfister, Archivista GmbH, Stegstr. 14, CH-8132 Egg,  
Mail: [webmaster@archivista.ch](mailto:webmaster@archivista.ch), [www.archivista.ch](http://www.archivista.ch)