University of Applied Sciences
HOCHSCHULE EMDEN·LEER
Fachbereich Technik
Abteilung Elektrotechnik und Informatik

Bachelor-Thesis

Effiziente Speicherung virtueller Festplatten mit bestehender OpenSource-Software (Arbeitstitel)

Bastian de Groot

15. November 2010

Prüfer	Prof. Dr. Jörg Thomaschewski
Zweitprüfer	Dr. Arvid Requate

Inhaltsverzeichnis

1 Einleitung

Betriebsystemvirtualisierung ist, wenn sich mehrere virtuelle Betriebsysteminstanzen Hardwareressourcen wie CPU, RAM oder Festplatten teilen. Der Virtualisierungskern (Hypervisor) stellt den virtuellen Betriebsysteminstanzen eine in Software und Hardware realisierte Umgebung zur Verfügung, die für die darin laufenden Instanzen kaum von einer echten Hardwareumgebung unterscheidbar sind. Es gibt unterschiedliche technische Ansätze der Virtualisierung, wie Paravirtualisierung oder Vollvirtualisierung. Diese Kategorisierung bezieht sich darauf, wie der Hypervisor die vorhandene Hardware der virtuellen Instanz bereitstellt. Auf diesem Gebiet gibt es eine sehr aktive Entwicklung. [Prz] [Bau]

Wenig beachtet bei der Entwicklung von Virtualisierungssoftware ist jedoch die Speicherung von virtuellen Festplatten. In dieser Arbeit wird dieser Punkt aufgegriffen und die Möglichkeit der Optimierung mit der Copy-on-Write Strategie beleuchtet.

Copy-on-Write ist eine Optimierungsstrategie, die dazu dient unnötiges Kopieren zu vermeiden und somit die für Bereitstellung einer geklonten virtuellen Maschine benötigte Zeit zu minimieren. Copy-on-Write Images für Desktopvirtualisierung nutzen diese Strategie. Hierbei wird nicht für jeden Benutzer ein eigenes Image kopiert, sondern alle Benutzer verwenden ein Master-Image. Falls ein Benutzer Änderungen an diesem Master-Image vornimmt, werden die Änderungen separat abgespeichert.

1.1 Zieldefinition

Ziel dieser Arbeit soll es sein, Möglichkeiten zur effizienten Speicherung von virtuellen Festplatten aufzuzeigen. Da Hersteller proprietärer Software das Veröffentlichen von Performance-Tests zum Teil nicht ohne Einschränkungen genehmigen, wird hierbei ausschließlich auf bestehende Open Source Lösungen zurückgegriffen. Die freien Open Source Lösungen werden miteinander verglichen und eine effiziente Lösung herausgearbeitet. Außerdem wird betrachtet, wie die für das Copy-on-Write benötigten Master-Images im Netzwerk effizient verteilt werden können. [vmw]

1.2 Vorgehen und Kurzzusammenfassung

Zunächst werden die vorhandenen Softwarelösungen für Copy-on-Write und für die Verteilung der Master-Images erläutert. Danach werden diese anhand verschiedener anwendungsrelevanter Kriterien miteinander verglichen. Nachdem die besten Lösungen beider Kategorien gefunden wurden, werden Softwaretools erstellt, die die Nutzung der gefundenen Lösung ohne tiefgreifende Vorkenntnisse ermöglicht.

1.3 Anmerkung zu den verwendeten Literaturquellen

Diese Arbeit enthält neben den herkömmlichen Literaturquellen auch Mailinglisten- und Forenbeiträge, sowie Blogeinträge.

Bei Quellenangaben im Bereich der Open Source Software gibt es einige Punkte die zu beachten sind. Es gibt keine einheitliche Dokumentation der Software. Häufig

sind die Informationen nicht an einer zentralen Stelle vereint, sondern liegen verstreut im Internet in Foren, Blogs, Mailinglisten oder auch in Manpages und den Quelltexten selbst. Die Relevanz und die Richtigkeit einer solcher Quellen ist schwer zu bewerten, da Blogs, Mailinglisten und Foren keinen Beschränkungen unterliegen.

Die oben genannte Verstreuung birgt, neben der schwierigen Bewertbarkeit der Richtigkeit und Relevanz, ein weiteres Problem. Da sehr viele Autoren zum einem Thema etwas schreiben, werden unterschiedliche Begriffe synonym verwendet oder sind mehrdeutig.

Alle Quellen werden mit der zu Grunde liegenden Erfahrung des Autors dieser Arbeit ausgewählt und überprüft, können aber aus den oben genannten Gründen keine absolute Richtigkeit für sich beanspruchen.

2 Analyse Copy-on-Write

Für das Erstellen mehrerer gleichartiger Virtueller Maschinen benötigt man mehrere Virtuelle Festplatten. Das kann man auf herkömmliche Art und Weise lösen, in dem ein vorhandenes Festplattenimage n mal kopiert wird. Durch das häufige Kopieren entstehen allerdings große Mengen an Daten. Außerdem benötigt es viel Zeit Festplattenimages zu kopieren. Um diesen beiden Problemen entgegen zu wirken gibt es Copy-on-Write. Wie in Abbildung 2.1 zu sehen ist, wird bei Copy-on-Write nicht das gesamte Image kopiert. Es werden in dem Copy-on-Write-Image nur die Veränderungen zu dem so genannten Master- oder Quellimage gespeichert. Für die Platzersparnis werden Sparse-Dateien genutzt welche im Folgenden erklärt werden. Außerdem werden die unterschiedlichen Verfahren zur Verwendung von Copy-on-Write erläutert und analysiert.

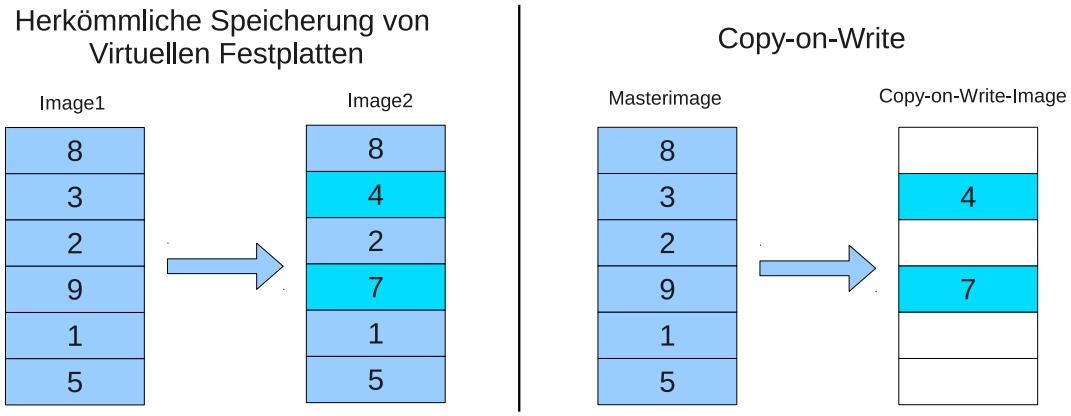

Abbildung 2.1: Copy-on-Write

2.1 Sparse-Dateien

Eine Sparse-Datei ist eine Datei, die nicht vom Anfang bis zum Ende beschrieben ist. Sie enthält also Lücken. Um Speicherplatz zu sparen, werden diese Lücken bei Sparse-Dateien nicht auf den Datenträger geschrieben. Die Abbildung 2.2 zeigt, dass der tatsächlich benutzte Speicherplatz auf der Festplatte weitaus geringer sein kann als die eigentliche Dateigröße. Eine Sparse-Datei ist kein eigenes Imageformat sondern eine Optimierungsstrategie. Sie verhilft Copy-on-Write-Images zu einer großen Platzersparnis. In Imageformaten wie qcow2 oder vhd ist die Optimierungsstrategie ein fester Bestandteil. [spa]

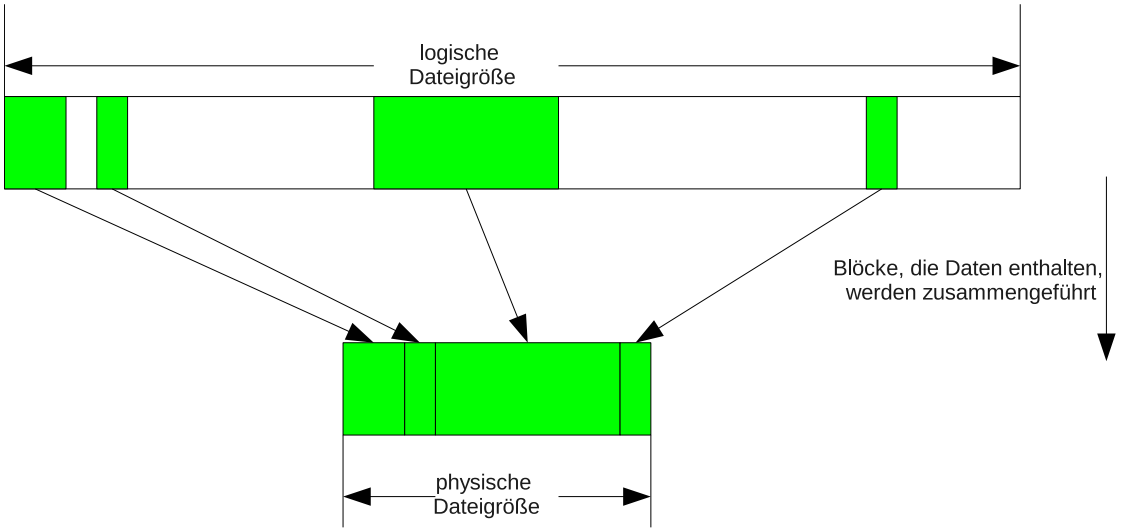

Abbildung 2.2: Sparse-Datei

2.2 qcow2

Das Imageformat qcow2 ist im Rahmen des qemu Projekts entwickelt wurde. Es ist der Nachfolger des ebenfalls aus dem qemu Projekt stammenden Formats qcow. [McL] [qem]

Vorteile

- einfache Einrichtung

Nachteile

- Kompatibilitätsprobleme mit Xen und anderen offenen Virtualisierungstechniken (z.B. VirtualBox)

2.3 vhd

Das Format vhd ist von Conectix und Microsoft entwickelt worden. Die Spezifikation des Imageformats wurde von Microsoft im Zuge des "Microsoft Open Specification Promise" freigegeben. Seit der Freigabe der Spezifikation bieten einige Open Source Virtualisierungslösungen wie qemu, Xen oder VirtualBox die Möglichkeit dieses Format zu verwenden. [mso] [vhd]

Vorteile

- einfache Einrichtung

Nachteile

- Weiterentwicklung ist fragwürdig
- Verwendung von Copy-on-Write mit KVM nicht möglich

2.4 dm-snapshots

Die dm-snapshots sind eine Funktion des Device Mappers. Device Mapper ist ein Treiber im Linux-Kernel. Er erstellt virtuelle Gerätedateien, die mit bestimmten Features wie zum Beispiel Verschlüsselung ausgestattet sind. Bei dm-snapshots wird eine solche virtuelle Gerätedatei erstellt. Sie wird aus zwei anderen Gerätedateien zu-

sammengesetzt. Die erste Gerätedatei ist der Ausgangspunkt, wenn an daran Änderungen vorgenommen werden, werden sie als Differenz in der zweiten Gerätedatei gespeichert. Dm-snapshots benötigen keine Unterstützung der Virtualisierungstechnik, da sie für diese nicht von physikalischen Festplattenpartitionen unterscheidbar sind. Dieses ist nicht nur ein Vorteil, sondern zugleich auch ein Nachteil. Es muss immer vor dem Starten einer virtuellen Maschine das Copy-on-Write-Image und das Masterimage zu einer Gerätedatei verbunden werden. [Bro] [dmk]

Vorteile

- hohes Entwicklungsstadium
- sichere Weiterentwicklung
- unabhängig von Virtualisierungstechnik

Nachteile

- Aufwendige Einrichtung
- erfordert zusätzlichen Programmstart vor dem VM-Start

2.5 LVM-Snapshots

LVM-Snapshots sind ein Teil des Logical Volume Managers. LVM ist eine Software-Schicht die über den eigentlichen Hardware-Festplatten einzuordnen ist. Es basiert auf Device Mapper. LVM ermöglicht das Anlegen von virtuelle Partitionen (logical volumes). Diese können sich über mehrere Festplatten-Partitionen erstrecken und Funktionen wie Copy-on-Write bereitstellen. [lvmc] [lvma] [lvmb]

Vorteile

- hohes Entwicklungsstadium

- sichere Weiterentwicklung
- unabhängig von Virtualisierungstechnik

Nachteile

- Aufwendige Einrichtung
- Live-Migration nicht möglich
- Nutzung von Sparse-Dateien schwer umsetzbar

2.6 Benchmarks

Ein wichtiger Punkt für die Entscheidung welche Copy-on-Write Implementierung optimal ist, ist die Lese- und Schreibgeschwindigkeit. Hierbei gibt es zwei Zugriffsarten, einmal den sequentiellen Zugriff und den wahlfreien oder auch zufälligen Zugriff. Die Testergebnisse werden in diesem Kapitel zusammenfassend aufgeführt. Die kompletten Testergebnisse befinden sich im Anhang.

2.6.1 Testbedingungen

Das Hostsystem für die Performance-Tests hat einen AMD Athlon II X2 250 Prozessor und 4 GiB RAM. Als Betriebsystem kommt ein 64 bit Debian testing zum Einsatz. Bei den KVM-Tests ist 2.6.32-5-amd64 der eingesetzte Kernel, bei Xen ist es 2.6.32-5-xen-amd64.

Während der Performance-Tests laufen neben der Virtuellen Maschine auf dem Hostsystem keine anderen aktiven Programme, die das Ergebnis verfälschen könnten. Als Referenz zu den Copy-on-Write-Techniken werden jeweils eine echte Festplattenpartition und eine Sparse-Datei verwendet. Zum Testen der Performance werden IOzone und Bonnie++ eingesetzt.

IOzone
IOzone ist ein Tool mit dem in einer Reihe von unterschiedlichen Tests die Lese- und Schreib-Geschwindigkeit überprüft werden kann. Es wird hier zur Überprüfung der sequentiellen Lese- und Schreibgeschwindigkeit verwendet.

Bonnie++
Bonnie++ dient wie IOzone als Tool zum Testen von Festplatten. Es wird hier zur Überprüfung der sequentiellen Lese- und Schreibgeschwindigkeit sowie zum Testen des wahlfreien Zugriffs verwendet.

2.6.2 Testergebnisse

Es gibt bei den Testergebnissen keinen klaren Gewinner oder Verlierer. Im Großen und Ganzen fallen bei den Ergebnissen unter den einzelnen Copy-on-Write Verfahren keine bemerkenswerten Unterschiede auf.

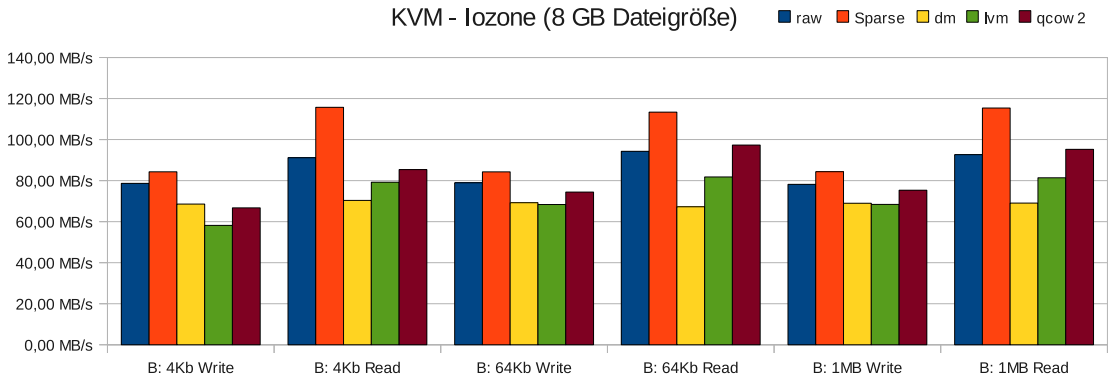

Abbildung 2.3: Iozone-kvm-8gb

Die Abbildung 2.3 zeigt, dass mit KVM qcow2 gegenüber den anderen Copy-on-Write-Techniken einen Geschwindigkeitsvorteil beim sequentiellen Lesen und Schreiben hat. LVM-Snapshots und dm-snapshots liegen hingegen ungefähr gleich auf.

Abbildung 2.4 ist zu entnehmen, dass qcow2 wie auch bei den sequentiellen Tests vor LVM-Snapshots und dm-snapshots liegt. Der Unterschied zu der echten Fest-

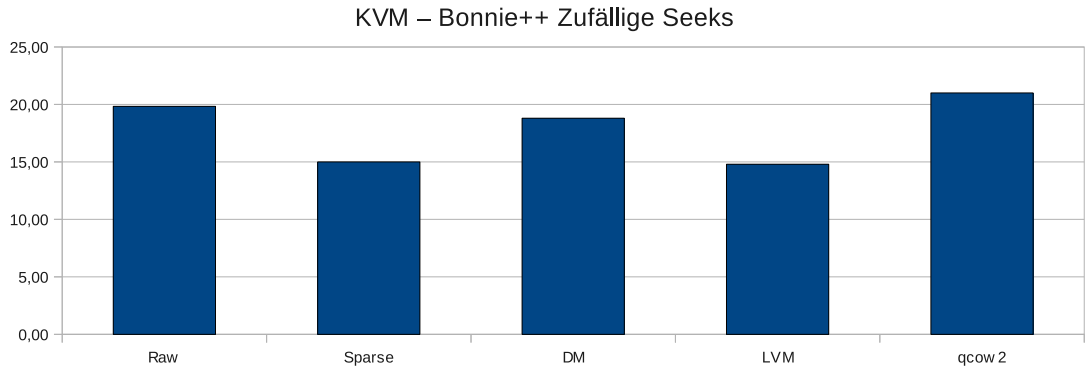

Abbildung 2.4: bonnie-kvm-random-seek

plattenpartition ist in beiden Tests sehr gering. Die guten Werte von qcow2 sowohl beim sequentiellen als auch beim zufälligem Zugriff auf die Festplatte, hängen mit der direkten Integration in KVM zusammen.

In Xen schneiden die dm-snapshots besser ab als LVM-Snapshots und vhd beim sequentiellen Lesen und Schreiben, wie in Abbildung 2.5 zu sehen ist.

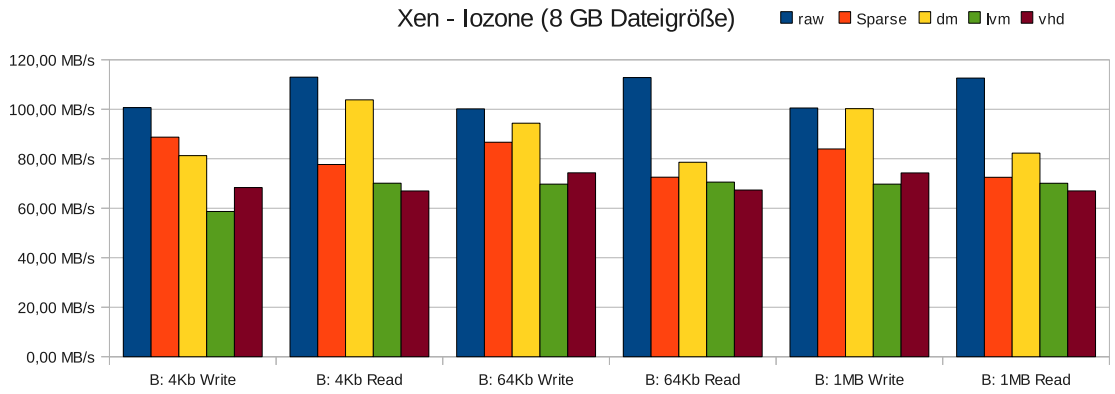

Abbildung 2.5: Iozone-xen-8gb

Beim zufälligen Zugriff auf die Festplatte ist unter Xen vhd abgeschlagen hinter LVM-Snapshots und dm-snapshots. Diese sind ungefähr gleichauf und liegen nicht weit hinter dem Sparse-Image und der Festplattenpartition (Abbildung 2.6). Das

mittelmäßige Abschneiden des Imageformats vhd verwundert, da Citrix, die treibende Kraft der Xen Weiterentwicklung, eine optimierte vhd-Unterstützung entwickelt hat. [Cro]

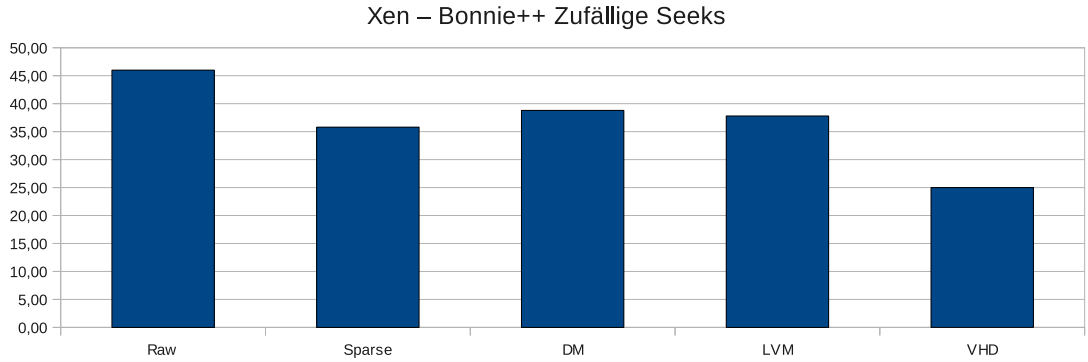

Abbildung 2.6: bonnie-xen-random-seek

Die Testergebnisse zeigen, dass es Geschwindigkeitsunterschiede zwischen den Copy-on-Write-Technniken gibt. Diese Unterschiede in der Geschwindigkeit sind aber nicht so gravierend, dass man einzelne Copy-on-Write-Lösungen aufgrund der Performance-Tests kategorisch ausschließen müsste. Dennoch sind besonders die Vorteile von qcow2 in Verbindung mit KVM zu erwähnen. Für Xen gibt es kein Image-Format, dass ähnliche Testergebnisse wie qcow2 in Verbindung mit KVM vorweisen kann.

2.7 Fazit

Aufgrund des unterschiedlichen Verhaltens der Copy-on-Write-Techniken mit KVM und Xen, wird auch für die beiden Virtualisierungslösungen ein jeweiliges Fazit gezogen.

2.7.1 KVM

Unter KVM gibt es die Alternativen dm-snapshots LVM-Snapshots oder qcow2. Das von Microsoft entwickelte vhd kommt nicht in Frage. KVM unterstützt zwar das vhd-Format, jedoch nicht die Copy-on-Write-Funktion des Formats.

Die effizienteste Lösung für Copy-on-Write mit KVM ist qcow2. Dafür gibt es mehrere Gründe. Das qcow2-Format ist Teil des qemu-Projekts und damit sehr gut in dem darauf basierendem KVM integriert. Durch die gute Integration werden sehr gute Performance-Werte erreicht. Außerdem lässt es sich im Gegensatz zu dm-snapshots und LVM-Snapshots sehr leicht einrichten.

2.7.2 Xen

Die für Xen zur Verfügung stehenden Copy-on-Write-Formate sind dm-snapshots, LVM-snapshots und vhd. Xen ünterstützte in einigen vergangenen Versionen qcow2, diese Unterstützung ist jedoch nicht in der aktuellen Version enthalten (Version 4.0.1). [qco]

Für Xen ist die beste Copy-on-Write-Lösung vhd. Es ist zwar laut der Performance-Tests nicht die schnellste Lösung, hat aber wesentliche Vorteile gegenüber dm-snapshots und LVM-Snapshots. Es werden keine Änderungen am Xen-Quelltext benötigt, wie es bei dm-snapshots der Fall ist. Die Funktion der Live-Migration ist mit vhd leichter zu realisieren als mit LVM-Snapshots und dm-snapshots. Die in dieser Arbeit verwendete Lösung ist vhd. Falls Xen in den nächsten Versionen wieder qcow2 unterstützt, sollte jedoch die Verwendung von qcow2 auch unter Xen geprüft werden. [rac]

3 Analyse Verteilung von Images

3.1 Multicast

3.2 BitTorrent

3.3 NFS

4 Synthese

4.1 Konzept

4.2 Realisierung einer Komplettlösung

5 Zusammenfassung und Ausblick

Glossar

Open Source
: Open Source Software.

Literaturverzeichnis

[Bau] Baun, Christian: *Vorlesung Systemsoftware*. http://jonathan.sv.hs-mannheim.de/~c.baun/SYS0708/Skript/folien_sys_vorlesung_13_WS0708.pdf, Abruf: 31.10.2010

[Bro] Brož, Milan: *Device mapper*. http://mbroz.fedorapeople.org/talks/DeviceMapperBasics/dm.pdf, Abruf: 17.10.2010

[Cro] Crosby, Simon: *We've Open Sourced Our Optimized VHD Support*. http://community.citrix.com/x/OYKiAw, Abruf: 11.11.2010

[dmk] *Device-mapper snapshot support*. http://www.kernel.org/doc/Documentation/device-mapper/snapshot.txt, Abruf: 17.10.2010

[lvma] *Linux LVM-HOWTO*. http://www.selflinux.org/selflinux/html/lvm01.html, Abruf: 18.10.2010

[lvmb] *LVM2 Resource Page*. http://sourceware.org/lvm2/, Abruf: 18.10.2010

[lvmc] *What is Logical Volume Management?* http://tldp.org/HOWTO/LVM-HOWTO/whatisvolman.html, Abruf: 18.10.2010

[McL] McLoughlin, Mark: *The QCOW2 Image Format*. http://people.gnome.org/~markmc/qcow-image-format.html, Abruf: 18.10.2010

[mso] *Microsoft Open Specification Promise*. https://www.microsoft.com/interop/osp/default.mspx, Abruf: 18.10.2010

[Prz] Przywara, André: *Virtualization Primer*. http://www.andrep.de/virtual/, Abruf: 01.11.2010

[qco] *Qcow2 Support*. http://lists.xensource.com/archives/html/xen-devel/2010-11/msg00256.html, Abruf: 14.11.2010

[qem] *QEMU Emulator User Documentation*. http://wiki.qemu.org/download/qemu-doc.html#disk_005fimages, Abruf: 18.10.2010

[rac] *Race condition in /etc/xen/scripts/block*. http://lists.xensource.com/archives/html/xen-devel/2010-07/msg00827.html, Abruf: 14.11.2010

[spa] *Sparse files*. http://www.lrdev.com/lr/unix/sparsefile.html, Abruf: 18.10.2010

[vhd] *Virtual Hard Disk Image Format Specification*. http://technet.microsoft.com/en-us/virtualserver/bb676673.aspx, Abruf: 18.10.2010

[vmw] VMware: *VMware Benchmarking Approval Process*. http://www.vmware.com/pdf/benchmarking_approval_process.pdf, Abruf: 05.11.2010

Tabellenverzeichnis

Abbildungsverzeichnis

Listings