Software-Raid mit Logical Volume Manager ausrüsten

Datenkoppler

Wenn die Festplatte wieder einmal voll ist, möchte man am liebsten einfach eine zweite nahtlos einfügen. Der Logical Volume Manager erfüllt diesen Wunsch, er fasst mehrere Blockdevices zu einem homogenen Speicherpool zusammen, der sich leicht erweitern lässt. Carsten Wiese

Die großen Kapazitätssteigerungen bei Festplatten in den letzten Jahren haben nicht das Problem der Datenspeicherung vereinfacht, vielmehr wachsen die Datenbestände allenthalben in immer größerem Tempo. Gerade bei sorgsam aufgeteilten Systemen, auf denen Benutzer- und Anwendungsdaten auf verschiedenen Laufwerken gespeichert sind, fehlt es immer wieder an Platz. Der Einbau einer neuen Festplatte hilft nur bedingt – im schlimmsten Fall müssen die Daten je nach verfügbarem Platz ständig hin und her verschoben werden.

Ein Ausweg sind Raid-Systeme, die mehrere Festplatten zu einem Blockdevice zusammenfassen. Zusätzlich bieten sie Ausfallsicherheit. Die Umstellung eines Linux-Systems auf ein Software-Raid-5 ohne Neuinstallation beschreibt der Artikel aus [1]. Damit ist das Linux-System jedoch auf die Größe des Raids zementiert, das Auslagern der Benutzer-Homes oder die Erweiterung durch größere Festplatten erfordert in jedem Fall ein weiteres Raid oder einzelne, dann ungesicherte Festplatten. Die beschriebenen Verteilungsprobleme stellen sich dann erneut.

Mit dem Logical Volume Manager (LVM) gewinnt der Administrator die nötige Flexibilität, um etwa im Betrieb die Gesamtkapazität zu erhöhen oder einzelne Systemteile abzutrennen. Dazu schiebt er das LVM zwischen Dateisystem und Raid ein. Das Besondere eines LVM ist, dass die Größe des Speicherplatzes des Systems zur Laufzeit vergrößert oder verkleinert werden kann – sofern das Dateisystem dies erlaubt. Software-Raid und Logical Volume Manager bilden somit eine gute Kombination aus Hochverfügbarkeit und Skalierbarkeit.

Die Qual der Wahl

Für Linux gibt es zurzeit zwei sehr gute Volume-Manager-Implementationen: Zuerst den Logical Volume Manager, der seit 1997 von Heinz Mauelshagen entwickelt wird und sich stark an den LVM von HP-UX anlehnt. Die zweite ist das Enterprise Volume Management System (EVMS) [2] von IBM, das seit 2002 unter der GPL steht. Die Entwicklung zweier Volume-Manager hat für einige Diskussionen in der Linux-Gemeinde gesorgt, viele fragen, ob man beide Projekte nicht besser zusammenlegen solle. Für den 2.6er Kernel hat Linus Torvalds entschieden, LVM in der Version 2 (LVM 2) aufzunehmen.

Parallelentwicklung

IBM entwickelt EVMS dennoch parallel weiter. Der Kasten „Exkurs: EVMS“ enthält eine Einführung in dieses umfangreiche Werkzeug, das nicht nur Linux-LVM beherrscht. Ausführlich beschäftigt sich der Artikel jedoch mit dem LVM, der im Kernel 2.4 der meisten Distributionen bereits enthalten ist. Der Plain-vanilla-Kernel 2.4 arbeitet noch mit Version 1 des LVM, zum Umstieg auf LVM 2.00.08 muss das Paket von [3] eingebaut werden. Es enthält neben dem Kernelpatch auch die passende Version der LVM-Administrationsprogramme. Zusätzlich ist das Patch für den Device Mapper von [3] erforderlich.

Bei den aktuellen Distributionen ist Suse Linux 9.0 bezüglich Volume-Manager führend: Der Suse-Kernel enthält bereits LVM, LVM 2 und EVMS. Fedora Linux bringt es immerhin auf LVM und LVM 2, bei Red Hat gibt es in der Professional

Abkürzungen	
LE	Logical Extent
LV	Logical Volume
LVM	Logical Volume Manager
PE	Physical Extent
PV	Physical Volume
VG	Volume Group
VGDA	Volume Group Descriptor Area

Workstation und Enterprise nur LVM. Die folgenden Beispiele funktionieren mit der LVM-Version 1, die Neuerungen von Version 2 sind im Kasten „Neu in LVM 2“ beschrieben.

LVM auf Software-Raid

Das LVM-System basiert auf drei Stufen: dem Physical Volume (PV), der Volume Group (VG) und dem Logical Volume (LV). Um etwas Licht in die babylonische Begriffsverwirrung zu bringen, zeigt die Grafik in Abbildung 1 die einzelnen Abstraktionsschichten.

Die Basis sind in der Schicht 1 die physikalischen Festplatten, hier also drei IDE-Festplatten. Daraus setzt sich das Multiple Device (»/dev/md3«) des Software-Raid in der zweiten Schicht zusammen. Man spricht hier trotzdem vom Physical Volume (PV), denn letztlich emuliert das Software-Raid ja ein physikalisches Laufwerk. Bei einem Hardware-Raid fasst der Raid-Controller die Schicht 1 intern zu einem physikalischen Laufwerk (Blockdevice) zusammen.

Am Anfang jedes Physical Volume ist im ersten Physical Extent die Volume Group Descriptor Area (VGDA) gespeichert. Sie enthält die Metadaten der LVM-Konfiguration. Die VGDA ist in vier Bereiche unterteilt und lässt sich mit der Partitionstabelle einer normalen Festplatte vergleichen. In dem Verzeichnis »/etc /lvmconf« wird ein automatisches Backup der VGDA angelegt.

Die Volume Group »asterix« in Schicht 3 fasst ein oder mehrere Physical Volumes zusammen, in diesem Beispiel nur »/dev/md3«. Es entsteht also gleichsam ein neues Laufwerk, das sich über alle Physical Volumes verteilt. Die Volume Group ist bereits flexibel und lässt sich jederzeit um weitere Physical Volumes erweitern.

In der vierten Schicht liegen die Logical Volumes, im Beispiel sind das »mp3« und »doc«. Jedes Logical Volume ist ein separates Blockdevice mit dem Namen des jeweiligen LV, hier »/dev/asterix /mp3« und »/dev/asterix/doc«, und lässt sich nachträglich vergrößern oder verkleinern. Die beiden Logical Volumes werden wie jedes andere Blockdevice auch mit einem Dateisystem versehen und dann eingebunden (Schicht 5).

Die kleinste Einheit

Die Schichten zwei, drei und vier lassen sich noch weiter sezieren. Die kleinste physikalische Speichereinheit in einem LVM-System ist das Physical Extent (PE). Jedes Physical Volume wird beim Anlegen einer Volume Group in gleich große PEs unterteilt, Standard sind 32 MByte pro PE. Insgesamt kann ein Physical Volume 65536 PEs enthalten, was maximal 2 TByte große Volumes ermöglicht. Jedes Physical Extent bekommt eine ID, die Nummerierung beginnt in jedem Physical Volume bei 0 – damit hat jedes PE innerhalb eines Volumes eine eindeutige ID.

Analog dazu ist jedes Logical Volume (LV) in Logical Extents (LE) unterteilt. Die Größe der Logical und Physical Extents einer Volume Group ist stets gleich, ein Logical Extent ist also standardmäßig ebenfalls 32 MByte groß. Wie schon die PEs werden auch die Logical Extents bei 0 beginnend durchnummeriert und enthalten einen Verweis auf das zugehörige Physical Extent. Die LEs werden eins zu eins auf die PEs abgebildet, jedes Logical Extent hat genau ein zugehöriges Physical Extent. Abbildung 2 zeigt, wie die beiden Logical Volumes »mp3« und »doc« auf dem Physical Volume »/dev/md3« aus der Volume Group »asterix« abgebildet werden.

Wenn eine Anwendung auf ein bestimmtes Byte im Logical Volume »doc« zugreift, wird zunächst die ID des Logical Extents ermittelt, in dem sich die Daten befinden. Anhand der Zuordnung der Logical Extents zu den Physical Extents – das Logical Volume »doc« arbeitet auf der Volume Group »asterix« und ist auf dem Physical Volume »/dev/md3« abgebildet – ermittelt der LVM die ID des zugehörigen Physical Extents und kann auf die entsprechende Position in »/dev /md3« zugreifen.

Stars and Striping

Es gibt zwei Methoden, die Daten auf den Physical Volumes zu verteilen: Linear und Striping. Standardmäßig verwendet LVM den Linear Mode, er ist für Volume Groups geeignet, die nur aus einem Physical Volume mit einer Festplatte oder einem Raid-System bestehen.

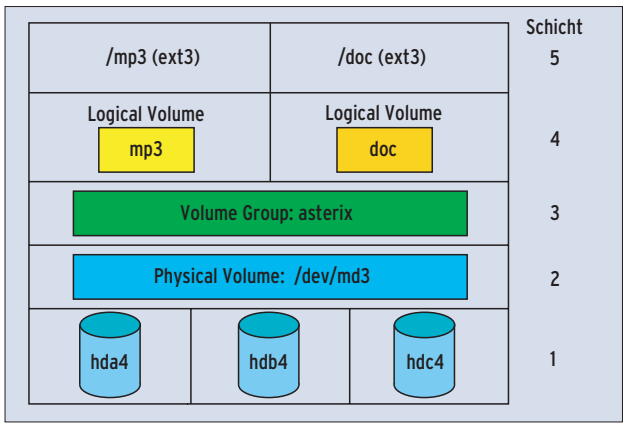

Abbildung 1: Die Volume Group fasst ein oder mehrere Physical Volumes zu einem Datenpool zusammen. Die Logical Volumes bedienen sich aus dem Pool und sind ganz normale Blockdevices.

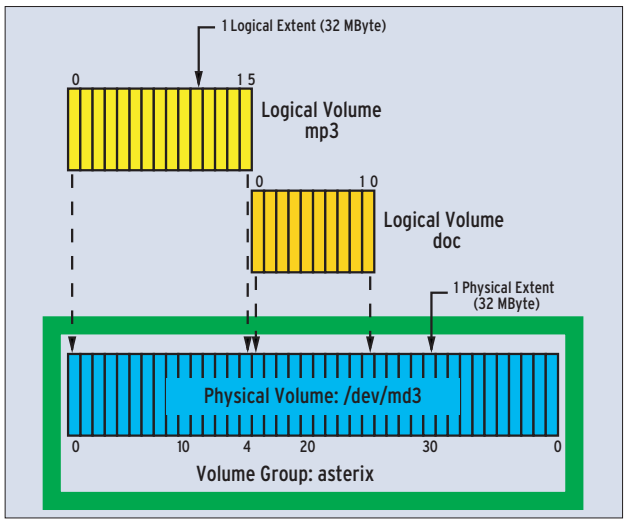

Abbildung 2: Die Logical Extents haben die gleiche Größe wie die zugeordneten Physical Extents. Im Linear Mode liegen die beiden Logical Volumes direkt hintereinander auf dem Physical Volume.

Die Logical Extents werden hierbei linear aufsteigend mit den Physical Extents verknüpft.
Bei Volume Groups, die sich über mehrere Physical Volumes verteilen, kann Striping interessant sein. Benachbarte Logical Extents sind dann mit Physical Extents aus zwei verschiedenen Physical Volumes verknüpft. Wird später gleichzeitig auf Daten benachbarter LEs zugegriffen, kommen die Daten von zwei verschiedenen Laufwerken, was die Datentransferrate im Einzelfall erhöhen kann. Dafür sind solche Logical Volumes nachträglich nicht vergrößerbar.
In der Praxis werden fast nur lineare Logical Volumes verwendet, der Befehl

```
lvcreate -L300 -nmp3 asterix
```

erzeugt das 320 MByte große LV »mp3« in der Volume Group »asterix«. Für den Striping-Modus müssen zusätzlich der Parameter »-i« und die Anzahl der Streifen angegeben werden, etwa »-i2« bei zwei Physical Volumes.
Das Logical Volume »mp3« sollte zwar 300 MByte und nicht 320 MByte groß sein, es werden aber nur ganze Logical Extents verwendet. Die Partition wird daher automatisch auf das nächste Vielfache von 32 MByte vergrößert.

Werkzeuge

Das LVM-Paket enthält eine ganze Reihe von Administrationstools, mit denen die verschiedenen Schichten erzeugt und manipuliert werden. Die Programme »vgcreate«, »vgdisplay«, »vgchange« und »vgremove« sind zum Beispiel für Volume Groups zuständig. Für Backup und Restore der Volume-Group-Konfigurationsdateien gibt es »vgcfgbackup« und »vgbfgrestore«; »vgreduce« und »vgextend« verkleinern oder vergrößern eine Volume Group.
Für den Transfer von einem Rechner zum anderen gibt es »vgexport« und »vgimport«. Mehrere Volume Groups teilen oder zusammenfügen ist Aufgabe der Tools »vgsplit« und »vgmerge«, »vgscan« durchsucht Blockdevices nach (verloren gegangenen) Volume Groups, »vgrename« ändert die Namen.
Für die beiden anderen Abstraktionsstufen, Physical und Logical Volumes, gibt es zum Teil ähnliche Tools, die sich im Namen nur durch die ersten zwei Buchstaben unterscheiden: pv steht für Physical Volume, lv für Logical Volume.

Bauplan

Das Physical Volume ist ein Software-Raid-5 und besteht aus – wie in Abbildung 1 zu sehen – drei IDE-Festplatten. Die Daten aus »/mp3« und »/doc« sollen auf dem Raid ausgelagert werden, LVM empfiehlt sich, um für zukünftige Vergrößerungen gewappnet zu sein. Den Aufbau eines Software-Raid-5 aus drei Festplatten erklärt der bereits erwähnte Artikel aus [1], die folgenden Beispiele setzen ein fertig initialisiertes Raid-System »/dev/md3« voraus.
Im Kernel müssen in der Kategorie »Multiple device support« die Einträge »RAID support« sowie »Logical volume manager (LVM) support« aktiviert sein (Abbildung 4). Wird die LVM-Unterstützung modular eingebunden, ist das Modul mittels »lvmcreate_initrd« an die Initial RAM Disk (»initrd«) anzuhängen.

Volumes anlegen

Physical Volume ist das Software-Raid-5 »/dev/md3«, der Befehl »pvcreate /dev/md3« bereitet das Blockdevice entsprechend vor. Zur Kontrolle zeigt »pvdisplay /dev/md3« die Daten des Physical Volume an. Die Volume Group »asterix« entsteht mit dem Befehl »vgcreate asterix /dev/md3«, bei mehreren Physical Volumes wird der Aufruf um die Namen der Blockdevices erweitert.
Die vierte Schicht, die beiden Logical Volumes »mp3« und »doc«, entstehen mit den Befehlen »lvcreate -L300 -nmp3 asterix« und »lvcreate -L200 -ndoc asterix«. Die tatsächliche Größe der Logical Volumes beträgt 320 MByte respektive 224 MByte, da LVM nur ganze Logical Extents mit je 32 MByte verwendet.
Im Verzeichnis »/dev/asterix« sollten jetzt drei neue Devices liegen: das Character Device »group« und die beiden Blockdevices »mp3« und »doc«. Die Dateisysteme entstehen – wie bei Blockdevices üblich – über die Befehle »mkfs.ext3 /dev/asterix/mp3« und »mkfs.ext3 /dev/asterix/doc«.
Auch beim Mounten unterscheiden sich die Logical Volumes nicht von anderen Blockdevices, allerdings sind zuvor »vgscan« und »vgchange -ay« auszuführen und beim Herunterfahren »vgchange -an«. Am besten werden diese Befehle in den Start-Stop-Skripten »/etc/init.d/boot« und »/etc/init.d/halt« untergebracht, je nach Distribution.
Wenn das Logical Volume »mp3« voll läuft, wird es einfach mittels »lvextend

Exkurs: EVMS

Das Enterprise Volume Management System (EVMS) von IBM besteht aus einem Plugin-Modell, in das sich einzelne Module als Erweiterungen sehr einfach einfügen lassen. Es ist kompatibel zum LVM, integriert Software-Raid (Multiple Devices) und unterstützt die gängigen Linux-Dateisysteme. Auch Bad Block Relocation (BBR) und Cluster Support sind keine Fremdworte für das EVMS.
Das grafische Admin-Interface von EVMS (Abbildung 3) gibt es auch als Textmodus-Variante. Die Terminologie ist etwas anders als beim LVM, so werden zum Beispiel Physical Volumes zu Segments, die Volume Group heißt Containers und hinter dem Begriff Regions verbergen sich die Logical Volumes. Die gute Dokumentation auf der Projekt-Homepage [2] erleichtert den Einstieg.

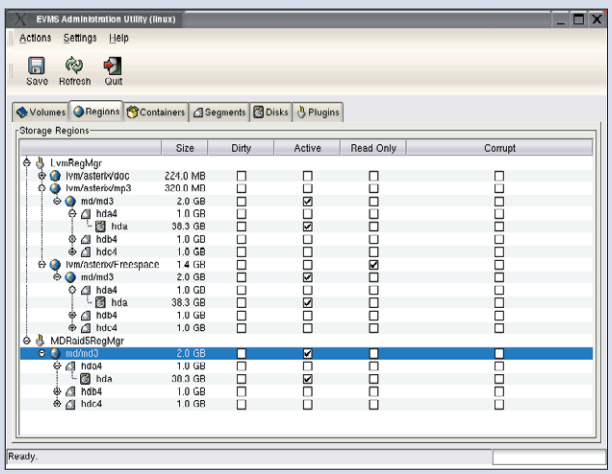

Abbildung 3: Das EVMS lässt sich per GUI oder über die Textkonsole komfortabel verwalten. Die Bezeichnungen unterscheiden sich jedoch von denen bei LVM.

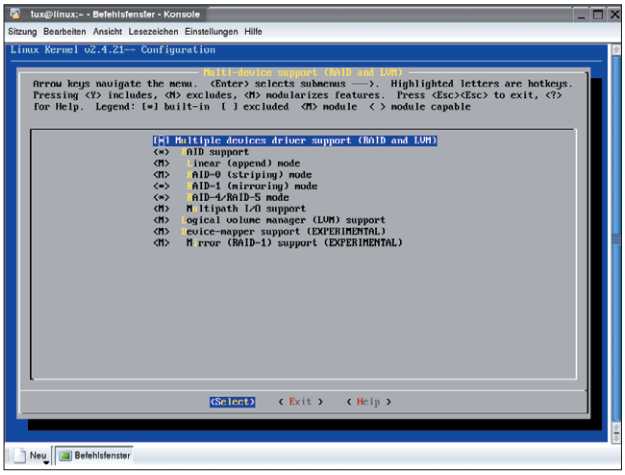

Abbildung 4: Der Logical Volume Manager (LVM, Version 1) ist im Kernel 2.4 standardmäßig enthalten. Für LVM 2 muss der Kernel zunächst gepatcht werden.

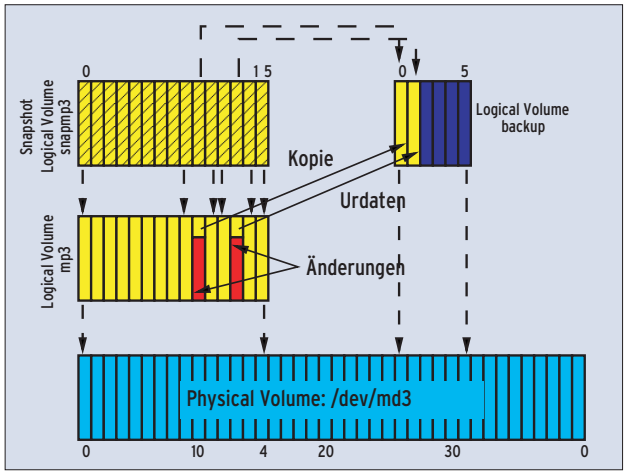

Abbildung 5: Beim Snapshot entsteht ein schreibgeschütztes Logical Volume, etwa für ein Backup, während das ursprüngliche LV weiterhin beschreibbar bleibt.

-L 500M /dev/asterix/mp3« auf 500 MByte erweitert. Anschließend passt man das Dateisystem per »resize2fs /dev /asterix/mp3« an.

Schmankerl: Der Snapshot

Die Snapshot-Funktion ist besonders für Backups interessant, sie ermöglicht es, zu jedem Zeitpunkt ein Alias eines Logical Volume anzulegen, das sich nicht mehr verändert. Das so entstandene Snapshot Logical Volume ist eine exakte, eingefrorene Kopie (frozen Image). Für ein Backup wird das Snapshot Logical Volume nur lesbar gemountet und anschließend gesichert. Dabei bleibt das ursprüngliche Logical Volume weiterhin beschreibbar – wenn dort ein Logical Extent geändert wird, speichert der LVM eine Kopie des ursprünglichen Datensatzes auf dem Snapshot Volume (siehe Abbildung 5).

Das Snapshot Logical Volume muss also genug Pufferspeicher (zugeordnet) haben, um die gleiche Datenmenge aufzunehmen, die während des Backups auf das ursprüngliche Volume geschrieben wird. Dazu legt man am besten eine separate Volume Group »operator« an, die zum Beispiel auf einer einzelnen Festplatte liegt. Läuft das Snapshot Volume voll oder startet der Rechner neu, verschwindet das Snapshot Volume automatisch. Die echten Daten werden jedoch nach wie vor auf das ursprüngliche Logical Volume geschrieben, ein Datenverlust entsteht beim Überlaufen des Snapshot-LV also nicht. Eine detaillierte, allerdings englische Anleitung zum Backup per Snapshot Volumes gibt es unter [4] im Kapitel 11.4.

Root-Dateisystem mit LVM und Striping-Risiken

Zwei Howtos unter [5] und [6] beschreiben, wie auf einem Software-Raid mit LVM ein Root-Filesystem eingerichtet wird. Das mag in einigen Fällen sinnvoll sein, jedoch steigt der Administrationsaufwand gewaltig und bei einem Systemcrash wird es ziemlich kompliziert. Man sollte nicht einfach alles auf LVM umstellen, sondern abwägen, wo ein LVM wirklich sinnvoll ist.

Auch ist gut zu überlegen, ob ein Logical Volume im Striping-Modus überhaupt einen Geschwindigkeitsvorteil bringt, denn der Nachteil, die Größe nachträglich nicht mehr verändern zu können, wiegt sehr schwer. Ein lineares LVM hat dieses Problem nicht und muss nicht erheblich langsamer sein.

Den Ausfall einer Festplatte in einem Striped-LVM fängt das Raid-System ab. Ohne Raid kann ein Plattencrash schnell zum Super-GAU werden, wenn etwa ein Striped Logical Volume komplett über zwei oder mehr Physical Volumes verteilt ist, dann ist der gesamte Datenbestand inkonsistent, während bei Linear Logical Volumes nur die Daten des jeweiligen LV betroffen sind. *(mdö)* ■

Neu in LVM 2

Die drei wesentlichen Änderungen im LVM 2 sind der Device Mapper, das neue Metadatenformat und die Konfigurationsdatei »lvm.conf«. Mit dem Device Mapper ist es möglich, ein neues Blockdevice auf ein existierendes Device aufzusetzen. Das Metadatenformat im LVM 2 ist gegenüber dem alten stabiler und effizienter gestaltet.

Mit der »lvm.conf« hat der Administrator die Möglichkeit, Parameter für die einzelnen Devices einzutragen und das Backup der Metadaten sowie den Umfang der Logfiles anzupassen. LVM 2 ist abwärts kompatibel zur Version 1.x, der Befehl »vgconvert -M2 asterix« konvertiert die Metadaten der Volume Group »asterix« in das neue Format.

Infos

[1] Carsten Wiese, „Volks-Raid“: Linux-Magazin 07/03, S. 50; [http://www.linux-magazin.de/Artikel/ausgabe/2003/07/volksraid/volksraid.html]

[2] Enterprise Volume Management System (EVMS): [http://evms.sf.net]

[3] LVM2-Patch und Device Mapper: [http://www.sistina.com/products_lvm_download.htm]

[4] LVM-Howto: [http://tldp.org/HOWTO/LVM-HOWTO/]

[5] Root-on-LVM-on-Raid-Howto: [http://www.midhgard.it/docs/lvm/html/index.html]

[6] Root-on-Raid- und Root-on-LVM-on-Raid-Howto: [http://karaolides.com/computing/HOWTO/lvmraid]

Der Autor

Carsten Wiese arbeitet als Systemintegrator bei der Höft und Wessel AG in Hannover. Er beschäftigt sich, neben vielen anderen Aufgaben, mit Raid-Systemen und Hochverfügbarkeitslösungen, leider nicht ausschließlich unter Linux.