

Workshop

Logical Volumen Management

Dynamisches einrichten und verwalten von Festplattenplatz mit dem Einsatz von logischen Volumen.

[Chemnitzer Linux-Tage 2011](#)

Marek Walther

2011-03-18

Inhaltsverzeichnis

Versionen und Unterschiede.....	2
LVM Grundlagen.....	3
LVM Praxis.....	5
Vorbereiten von physikalischen Volumen.....	6
Abfragen von physikalischen Volumen.....	7
Erstellen von Volumengruppen.....	8
Abfragen von Volumengruppen-Informationen.....	10
Entfernen von Volumengruppen.....	12
Erstellen von logischen Volumen.....	13
Abfragen von Volumen-Information.....	14
Entfernen von logischen Volumen.....	15
Verwenden von logischen Volumen.....	16
Administrative Wartungsarbeiten.....	17
Statusinformation abfragen.....	17
Ändern der Speichergröße von Volumengruppen.....	19
Erweitern von Volumengruppen.....	20
Verkleinern einer Volumengruppe.....	22
Ändern der Speichergröße von logischen Volumen.....	25
Vergrößern von logischen Volumen.....	26
Verkleinern von logischen Volumen.....	28
Spezialitäten unter LVM.....	30
Snapshots für logische Volumen.....	30
Logical Volumen mit Stripe Sets.....	35
Logische Volumen mit Spiegelung.....	39
Anhang.....	41
Abbildungsverzeichnis.....	41
Tabellenverzeichnis.....	41
Literaturverzeichnis.....	41

LVM2 unter Linux ist eine Abstraktionsschicht auf Kernelebene, die eine dynamische Verwaltung von physikalischem Festplattenplatz ermöglicht. Aus historischen Gründen lässt sich der Speicherplatz einer Festplatte nur in feste Bereiche, den Partitionen, aufteilen. Eine Änderung der vorgenommenen Struktur bedingt eine komplette Neupartitionierung der Festplatte und damit einen „Verlust“ der auf ihr abgelegten Dateisysteme. Als weiteres Hindernis kann die Partitionstabelle einer Festplatte maximal 4 primäre oder 3 primäre und eine erweiterte Partition aufnehmen. Die erweiterte Partition enthält eine eigene Partitionstabelle, die je nach Größe 6 bis 12 logische Partitionen aufnehmen kann. Auch wenn es heute brauchbare Werkzeuge gibt, die bei einer Verkleinerung oder Umpartitionierung die vorhandenen Dateisysteme anpassen oder verschieben, ist deren Handhabung für den administrativen Betrieb aufwändig, unflexibel und mit einem Offlinebetrieb des betroffenen Systems verbunden. LVM2 hebt diese Beschränkungen auf und flexibilisiert die Zuweisung von Speicherplatz. Es kann durch das spätere Hinzufügen von weiterem Festplattenspeicher erweitert werden und fast alle Wartungs- und Bearbeitungsfunktionen sind im Onlinebetrieb des Systems durchführbar. Diese Eigenschaften und ein paar weitere Spezialitäten senken den Arbeitsaufwand und den Stressfaktor in der Administration erheblich.

Versionen und Unterschiede

Logical Volumen Management ist bei Linux seit der Kernelversion 2.2 im Einsatz und weit verbreitet. Die erste Version hat gegenüber der Version 2 einige Einschränkungen und es fehlen ihr einige Funktionen. Ein großer Fortschritt ist die Aufhebung der maximalen PE-Zuweisungen an Volumen. Die physikalischen Extents (PE) bilden die Granularität, mit der den logischen Volumen Speicherplatz zugewiesen wird. Hierbei gab es bisher die Beschränkung, dass einem logischen Volumen maximal 65535 PEs zugewiesen werden konnten, was die maximale Größe eines Volumens bei der PE-Standardgröße von 4MB auf 256GB beschränkte. Hinzu kommen Probleme mit der Performance bei einer hohen Zahl zugewiesener physikalischer Extents (PE). Neben der Verbesserung der Performance besitzt LVM2 die Möglichkeit, Metadaten in den physikalischen Volumen zu hinterlegen und bietet Raum für Erweiterungen und dem Clustering von Volumengruppen im Netzwerk.

LVM Grundlagen

Aufbau und Funktionsweise lassen sich an einem einfachen hierarchischen Model verdeutlichen.

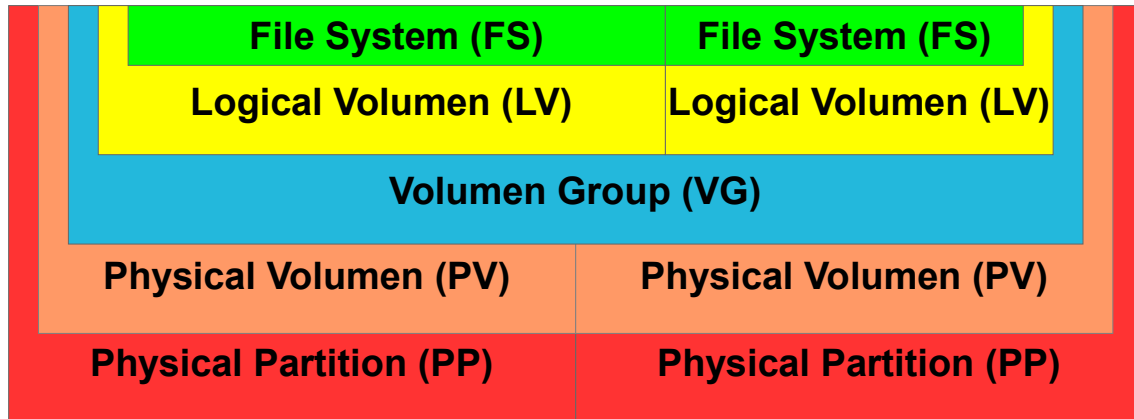


Abbildung 1: Modelaufbau Logical Volumen Management

Auf der unteren Ebene befinden sich die physikalischen Partitionen (PP). Auch wenn der verfügbare Speicher später dynamisch zugewiesen werden soll, müssen die Daten auf physikalischen Partitionen, Festplatten oder einem Raid-Verbund abgelegt werden. Es ist möglich, hier eine Festplatte ohne Partitionierung zu verwenden. Aus administrativen Gründen sollte aber eine Partition für die Verwendung eingerichtet sein. Denn eine Festplatte ohne Partitionierung sieht leer aus und verleitet dazu, partitioniert und genutzt zu werden.

Die physikalischen Partitionen (PP) bilden die Basis für das physikalische Volumen (PV). Diese kapseln den Speicherplatz in Verwaltungseinheiten, den physikalischer Extents (PE). Sie nehmen die Daten auf und bilden die Granularität, mit der Speicherplatz an die logischen Volumen zugewiesen wird. Damit beeinflussen sie auch maßgeblich die maximale Volumengröße und die Performance. Das physikalische Volumen hält in Metabereichen Platz für die spätere Volumengruppenkonfiguration bereit.

Die Volumengruppe stellt den ihr anvertrauten Speicherplatz über die physikalischen Verwaltungseinheiten den logischen Volumen zur Verfügung. Sie organisiert ihn in logische Verwaltungseinheiten, den logischen Extents (LE). Die Größe dieser Verwaltungseinheiten entspricht der Größe der physikalischen Extents (PE), die beim Anlegen der Volumengruppe festgelegt wird. Hierbei ist es möglich, einer Volumengruppe mehrere physikalische Volumen (PV) zuzuweisen und damit den verfügbaren Gesamtspeicherplatz zu vergrößern. Physikalische Volumen (PV) können der Volumengruppe auch zu einem späteren Zeitpunkt hinzugefügt werden, womit der Speicherplatz der Gruppe bei Bedarf erweitert werden kann.

Das Logische Volumen (LV) wird innerhalb einer Volumengruppe angelegt und später vom System als Blockgerät angesprochen. Damit dient es als flexibler Ersatz für Partitionen, Blockgeräte und Laufwerke und kann mit einem Dateisystem versehen oder durch eine zusätzliche Verschlüsselung gesichert werden.

Die Speicherzuteilung erfolgt durch eine interne Verknüpfung der vorhandenen Verwaltungseinheiten. Die Basis bilden die physikalischen Verwaltungseinheiten, deren Größe bei der Zuordnung des physikalischen Volumens zu einer Volumengruppe festgelegt wird. Die Volumengruppe erhält bei der Zuordnung die Autorität über die physikalischen Verwaltungseinheiten und verknüpft diese beim Anlegen eines logischen Volumen entsprechend auf die logischen Verwaltungseinheiten des logischen Volumens.

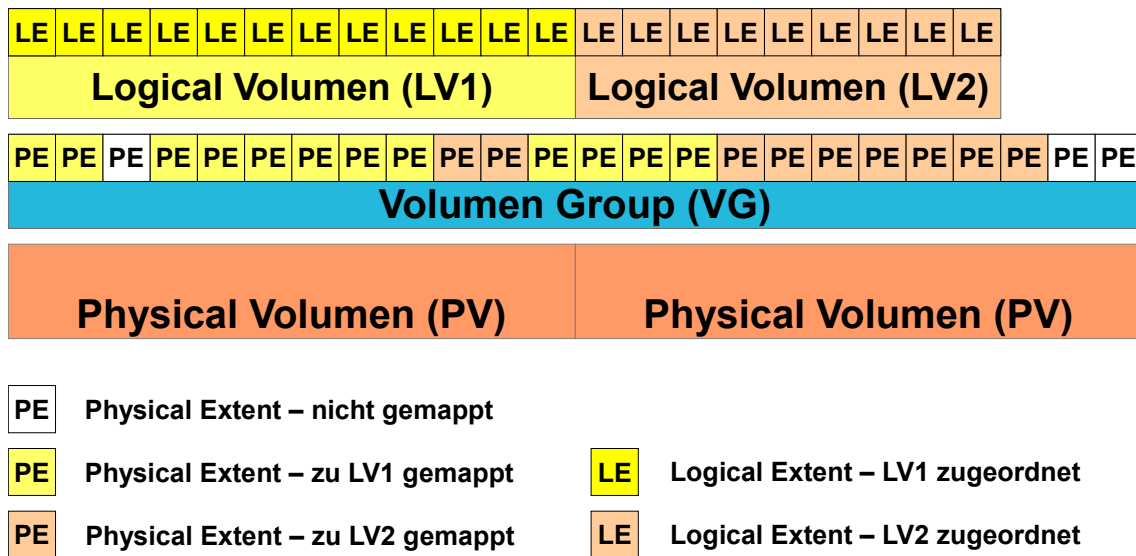


Abbildung 2: Verknüpfen von Verwaltungseinheiten

Die Verknüpfung der Verwaltungseinheiten ist beim Zugriff auf das logische Volumen transparent. Die Zuteilung und damit die physikalische Lage der Verwaltungseinheiten erfolgt in Abhängigkeit einer *Zuweisungsrichtlinie*, die für Volumengruppen oder logische Volumen eingestellt werden kann. Die Standardrichtlinie für Volumengruppen ist die Richtlinie „*Normal*“ und enthält ausgewogene Regeln. Die „*contiguous*“ Richtlinie versucht neu angeforderte Verwaltungseinheiten in der Nähe bereits vorhandener Verwaltungseinheiten zusammenhängend abzubilden, was Kopfbewegungen der Festplatte reduziert und die Zugriffsgeschwindigkeit erhöht. Die Richtlinie „*cling*“ versucht neu angeforderte Verwaltungseinheiten auf dem gleichen physikalischen Volumen abzubilden, auf dem sich bereit Verwaltungseinheiten des anfordernden Volumens befinden. Volumen können damit auf einer physikalischen Partition ihre Daten ablegen. Im Gegensatz zur Richtlinie „*normal*“ unterstützt diese Richtlinie das Einrichten von Stripe Sets auf Volumengruppen mit nur einem physikalischen Volumen. Logische Volumen besitzen als Standard die Zuweisungsrichtlinie „*inherit*“ sie übernehmen damit die Richtlinien-

konfiguration ihrer Volumengruppe. Die Zuweisungsrichtlinie kann mit den entsprechenden Werkzeugen geändert werden.

LVM Praxis

Nachdem einige theoretischen Grundlagen geschaffen wurden, gilt es diese mit der Praxis zu verfestigen. Hierbei ist es hilfreich, die Elemente, verschiedene administrative Tätigkeiten und verfügbare Werkzeuge aus dem LVM-Werkzeugkasten in einer Matrix zusammenzufassen. Die Tabelle wurde um eine Spalte mit einer subjektiven Einschätzung der administrativen Relevanz ergänzt.

	<i>Physikalisches Volumen (PV)</i>	<i>Volumengruppe (VG)</i>	<i>Logisches Volumen (LV)</i>	<i>Adm. Relevanz</i>
<i>Entfernen (create)</i>	pvcreate	vgcreate	lvcreate	* * * * *
<i>Entfernen (remove)</i>		vgremove	lvremove	* * * * *
<i>Vergrößern (extend)</i>	-/-	vgextend	lvextend , lvresize	* * * * *
<i>Verkleinern (reduce)</i>	-/-	vgreduce	lvreduce , lvresize	* * * * *
<i>Umbenennen (rename)</i>	-/-	vgrename	lvrename	*
<i>Anzeigen (display)</i>	pvdisplay	vgdisplay	lvdisplay	* * * * *
<i>Ändern (change)</i>	pvchange	-/-	lvchange	* *
<i>Verschieben (move)</i>	pvmove	-/-	-/-	* *
<i>Zusammenführen (merge)</i>	-/-	vgmerge	-/-	*
<i>Exportieren (export)</i>	-/-	vgexport	-/-	*
<i>Importieren (import)</i>	-/-	vgimport	-/-	*
<i>Status</i>	pvs	vgs	lvs	

Tabelle 1: LVM-Werkzeugmatrix

Der Einsatz von LVM2 auf einer lokalen Maschine ist relativ einfach. Als Voraussetzung müssen die LVM-Module für den Kernel kompiliert, geladen und der LVM-Werkzeugkasten installiert sein. Ersteres sollte heute der Standard bei allen größeren Distributionen sein und Letzteres muss gegebenenfalls nach Art des Hauses nachinstalliert werden. Unter Debian und Ubuntu ist das mit einer einfachen Zeile durchführbar.

```
~:~# apt-get install lvm2
```

Im nächsten Schritt müssen Datenträger oder Partitionen zur Aufnahme der Daten bereitgestellt werden. Hier bietet sich eine Partition oder ein RAID-Verbund an, der auch für eine erhöhte Zuverlässigkeit sorgen kann. Für einen Test können auch externe über USB angebundene Laufwerke verwendet werden. Hierbei ist aber auf die USB-Reihenfolge zu achten, da die

Laufwerke beim Booten in der Slot-Reihenfolge eingebunden werden, was eventuell zu einem Laufwerkschaos führen kann. Deshalb sollten die Laufwerke im ausgeschalteten Zustand einstecken und dann den Rechner gestartet werden. Für die durchgeführten Übungen kommen drei 2GB USB-Sticks zum Einsatz.

Vorbereiten von physikalischen Volumen

Nachdem die Voraussetzungen geschaffen wurden und Laufwerke bereitstehen, müssen die physikalischen Volumen vorbereitet und eingerichtet werden. Die Vorbereitung ist hierbei von der Art des Mediums abhängig, bei dem es sich um ein Block-Gerät handeln muss. Bei den vorliegenden Beispielen kommen Laufwerke zum Einsatz, die partitioniert werden müssen. Auf zwei der verwendeten Datenträgern sind jeweils zwei Partitionen anzulegen. Die größere Partition, mit einer Größe von ca. 1GB, wird als physikalisches Volumen eingerichtet und bekommt den Partitionstyp *0x8e* für „Logical Volumen Management“. Zu Kontrollzwecken ist eine kleinere Partition mit ca. 512MB und den Standardtyp *0x83* für „Linux-Partition“ anzulegen. Die Partitionierung kann nach Art der verwendeten Distribution vorgenommen werden und sollte wie folgt aussehen:

```
debian:~# sfdisk /dev/sdd -d
# partition table of /dev/sdd
unit: sectors

/dev/sdd1 : start=      62, size=  2002662, Id=8e
/dev/sdd2 : start=  2002724, size=  1003284, Id=83
/dev/sdd3 : start=      0, size=      0, Id= 0
/dev/sdd4 : start=      0, size=      0, Id= 0

debian:~# sfdisk /dev/sde -d
# partition table of /dev/sde
unit: sectors

/dev/sde1 : start=      62, size=  2002662, Id=8e
/dev/sde2 : start=  2002724, size=  1003284, Id=83
/dev/sde3 : start=      0, size=      0, Id= 0
/dev/sde4 : start=      0, size=      0, Id= 0
```

Damit die angelegten Partitionen unter LVM2 genutzt werden können, müssen sie mit dem Werkzeug „*pvcreate*“ vorbereitet werden. Das Werkzeug erwartet den Pfad und den Namen aller zu bearbeitenden physikalischen Partitionen. Es hinterlegt im zweiten Sektor der Partitionen eine interne Bezeichnung, mit der sie auch bei einer unterschiedlichen Laufwerkseinbindung sicher zugeordnet werden können. Dieser Fall kann auftreten, wenn sich die physikalische Anzahl der Blockgeräte ändert, den der Kernel bindet diese in der Reihenfolge ein, in der sie von den Treibern erkannt werden. Im zweiten Sektor ist die interne Bezeichnung vor einem versehentlichen löschen durch die Einrichtung eines MBR oder Boot-Loaders geschützt. Zusätzlich wird ein Metadatenbereich eingerichtet, in die die Volumengruppe später die Konfiguration ablegt. Der Metadatenbereich ist im Standard einmal vorhanden und wird direkt hinter dem Sektor mit der Bezeichnung abgelegt. Falls gewünscht, kann man das Werkzeug mittels Option anweisen, eine weitere Kopie dieses Bereichs am Ende anzulegen. Dies kann im

Fälle einer Datenwiederherstellung einen unschätzbaren Wert haben. Die Anzahl der Metadatenbereiche kann zu einem späteren Zeitpunkt nicht mehr verändert werden und es sind maximal zwei Metabereiche möglich. Ist eine Veränderung gewünscht, muss das physikalische Volumen aufgehoben und mittels „*pvcreeate*“ neu vorbereitet werden, was den Verlust der darauf abgelegten Daten zur Folge hat. Für die Übungen werden zwei physikalische Volumen auf unterschiedlichen Datenträgern mit „*pvcreeate*“ vorbereitet.

```
debian:~# pvcreeate /dev/sdd1 /dev/sde1 --metadacopies 2
Physical volume "/dev/sdd1" successfully created
Physical volume "/dev/sde1" successfully created
```

Hier werden die Partitionen /dev/sdd1 und /dev/sde1 als physikalisches Volumen vorbereitet, und die Metadaten in jeweils zwei Bereichen der Partitionen vorgehalten.

<i>Option</i>	<i>Beschreibung</i>
-f	Mache das was ich sage, ohne weiteres Nachfragen.
-ff	Erzwingt eine Reinitialisierung eines in eine Volumengruppe eingebundenen physikalischen Volumen.
-y	Alle Nachfragen automatisch mit Ja beantworten.
-z	Löschen der ersten 4 Sektoren.
- -metadacopies VALUE	Festlegen der Anzahl der reservierten Metadatenbereiche. Die Standardgröße ist 1 und legt einen Metadatenbereich am Anfang des physikalischen Volumens fest. Der Wert 2 reserviert einen Metadatenbereich am Anfang und am Ende des physikalischen Volumens. Mit dem Wert 0 wird auf einen Metadatenbereich für die Volumengruppenkonfiguration verzichtet.

Tabelle 2: Werkzeugoptionen für *pvcreeate*

Abfragen von physikalischen Volumen

Informationen über den Zustand des physikalischen Volumens können mit dem Werkzeug „*pvdisplay*“ abgerufen werden.

An der Ausgabe lässt sich leicht erkennen, dass die physikalischen Volumen in keine Volumengruppe eingebunden sind und die Größe der physikalischen Verwaltungseinheiten (PE) noch nicht festgelegt wurden.

```

debian:~# pvdisk /dev/sdd1 /dev/sde1
"/dev/sdd1" is a new physical volume of "977,86 MB"
--- NEW Physical volume ---
PV Name                /dev/sdd1
VG Name
PV Size                977,86 MB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               BtJlrx-MWdf-NZZl-WIHN-8lrR-lKHa-V3zKec

"/dev/sde1" is a new physical volume of "977,86 MB"
--- NEW Physical volume ---
PV Name                /dev/sde1
VG Name
PV Size                977,86 MB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               MRIGsr-k0sa-2YYI-HZ2C-Hcuy-7pq2-wCEuMI

```

<i>Attribut</i>	<i>Beschreibung</i>
PE Name	Name und Pfad der Gerätedatei der verwendeten physikalischen Partition.
VG Name	Name der Volumengruppe, dem das physikalische Volumen zugeordnet ist.
PV Size	Größe des physikalischen Volumens.
Allocatable	
PE Size (KByte)	Größe der physikalischen Verwaltungseinheit. Wird bei der Zuordnung zu einer Volumengruppe festgelegt.
Total PE	Anzahl der physikalischen Verwaltungseinheiten, in die das physikalische Volumen unterteilt ist.
Free PE	Anzahl der physikalischen Verwaltungseinheiten, die noch nicht als logische Verwaltungseinheit einem logischen Volumen zugeordnet wurden.
Allocated PE	Anzahl der physikalischer Verwaltungseinheiten, die über die Volumengruppe bereits einem logischen Volumen zugeordnet wurden.
PV UUID	Universally Unique Identifier - Unabhängige Volumen- oder Datenträgerbezeichnung. Wird beim Vorbereiten des physikalischen Volumens automatisch generiert.

Tabelle 3: Attribute physikalischer Volumen

Erstellen von Volumengruppen

Beim Erstellen einer Volumengruppe ist ein Name zu wählen, mit dem diese im System bekanntgemacht wird. Es sind die physikalischen Volumen anzugeben, die der Volumengruppe zugeordnet werden sollen. Beim Anlegen der Volumengruppe wird die Größe der physikalischen

Verwaltungseinheiten festgelegt. Diese Einstellung kann später nicht mehr verändert werden und beträgt als Standardgröße 4 Megabyte. Diese Größe stellt die Granularität dar, in denen der Speicher an logische Volumen verteilt wird, und kann Auswirkungen auf die Performance haben. Da wir heute Laufwerke im Terabyte-Bereich haben und gegebenenfalls größere logische Volumen planen, kann dieser Wert ohne Weiteres auf 8 oder 16 Megabyte angehoben werden. Zusätzlich kann die maximale Anzahl, der physikalischen Volumen und der logischen Volumen die in dieser Gruppe verwaltet werden können, festgelegt werden. Zum Anlegen einer neuen Volumengruppe kommt das Werkzeug „*vgcreate*“ zum Einsatz.

```
debian:~# vgcreate -s 2m vg01 /dev/sdd1
Volume group "vg01" successfully created
debian:~# vgcreate -s 2m vg02 /dev/sde1
Volume group "vg02" successfully created
```

Dieses Beispiel legt zwei Volumengruppe mit den Namen „vg01“ und „vg02“ an, denen die Blockgeräte /dev/sdd1 und /dev/sde1 als physikalisches Volumen zugewiesen werden. Die Größe der PE-Verwaltungseinheiten wird auf 2 Megabyte festgelegt.

<i>Option</i>	<i>Beschreibung</i>
-s VALUE	Festlegen der Verwaltungseinheitsgröße
-p VALUE	Festlegen der maximalen Anzahl physikalischen Volumen
-l VALUE	Festlegen der maximalen Anzahl der verwaltbaren logischen Volumen

Tabelle 4: Werkzeugoptionen für vgcreate

Eine Prüfung, der physikalischen Volumen offenbart, dass die Volumen wie gewünscht der neuen Volumengruppe zugeordnet und auch die Größe und Anzahl der physikalischer Verwaltungseinheiten festgelegt wurden.

```

debian:~# pvdisplay /dev/sdd1 /dev/sde1
--- Physical volume ---
PV Name                /dev/sdd1
VG Name                vg01
PV Size                977,86 MB / not usable 1,86 MB
Allocatable           yes
PE Size (KByte)       2048
Total PE              488
Free PE               488
Allocated PE          0
PV UUID               BtJlrX-MWdf-NZZl-WIHN-8lrR-lKHa-V3zKec

"/dev/sde1" is a new physical volume of "977,86 MB"
--- NEW Physical volume ---
PV Name                /dev/sde1
VG Name                vg02
PV Size                977,86 MB / not usable 1,86 MB
Allocatable           yes
PE Size (KByte)       2048
Total PE              488
Free PE               488
Allocated PE          0
PV UUID               MRIGsr-k0sa-2YYI-HZ2C-Hcuy-7pq2-wCEuMI

```

Abfragen von Volumengruppen-Informationen

Der aktuellen Status und die Konfiguration einer Volumengruppen kann mit dem Werkzeug „*vgdisplay*“ abgefragt werden. Das Werkzeug erwartet als Argument den Namen der abzufragenden Volumengruppe. Mit der Option *-v* wird die zusätzliche Ausgabe der Konfiguration aller in einer Volumengruppe enthaltenen logischen Volumen angefordert.

```

debian:~# vgdisplay vg01
--- Volume group ---
VG Name                vg01
System ID
Format                lvm2
Metadata Areas        2
Metadata Sequence No  1
VG Access              read/write
VG Status              resizable
MAX LV                0
Cur LV                0
Open LV                0
Max PV                0
Cur PV                1
Act PV                1
VG Size                976,00 MB
PE Size                2,00 MB
Total PE              488
Alloc PE / Size       0 / 0
Free PE / Size        488 / 976,00 MB
VG UUID                RZ27X2-kyaU-9Yf3-jZ0s-UfbU-20Ts-juoKer

```

Hier offenbart sich die genutzte und verfügbare Kapazität der Volumengruppe. Auch die Anzahl der eingebundenen physikalischen Volumen und ein Überblick der logischen Volumen, kann abgelesen werden. Die Volumengruppe hat ein physikalisches Volumen eingebunden. Sie hat eine Speichergröße von 976MB, die noch vollständig ungenutzt ist. Es wurden noch keine logischen Volumen angelegt und es sind auch keine Volumen in dieser Gruppe geöffnet.

<i>Attribut</i>	<i>Beschreibung</i>
VG Name	Bezeichnung der Volumengruppe.
System ID	
Format	Format der Volumengruppe, Version des verwendeten LVM.
Metadata Areas	Anzahl der Metadatenbereiche für die Volumengruppenkonfiguration.
Metadata Sequence No	
VG Access	Gibt an, ob in dieser Gruppe Daten gelesen oder geschrieben werden können.
VG Status	Gibt den Status der Volumengruppe an.
MAX LV	Festgelegte Anzahl der maximal verwaltbaren logischen Volumen.
Cur LV	Anzahl der angelegten logischen Volumen in dieser Volumengruppe.
Open LV	Anzahl der geöffneten logischen Volumen in dieser Volumengruppe.
Max PV	Anzahl der maximal zuweisbaren physikalischen Volumen.
Cur PV	Anzahl der zugewiesenen physikalischen Volumen.
Act PV	Anzahl der aktiven physikalischen Volumen.
VG Size	Speichergröße der Volumengruppe.
PE Size	Größe der physikalischer Verwaltungseinheiten.
Total PE	Anzahl der vorhandenen physikalischen Verwaltungseinheiten in der Volumengruppe
Alloc PE / Size	Anzahl der an logische Volumen zugewiesenen physikalischen Verwaltungseinheiten mit der Angabe der Speichergröße.
Free PE / Size	Anzahl der noch verfügbaren physikalischen Verwaltungseinheiten mit der Angabe der Speichergröße.
VG UUID	Universally Unique Identifier - Unabhängige Volumen oder Datenträgerbezeichnung. Wird beim Anlegen der Volumengruppe automatisch generiert.

Tabelle 5: Attribute von Volumengruppen

Entfernen von Volumengruppen

Wird eine Volumengruppe nicht mehr benötigt, kann sie aufgelöst werden um die physikalischen Volumen wieder freizugeben. Bei diesem Vorgang gehen alle logischen Volumen und ihre Daten verloren. Das Werkzeug „*vgremove*“ benötigt den Namen der Volumengruppe und besitzt eine Option zum Simulieren des Vorgangs. Enthält die Volumengruppe logische Volumen, entfernt es diese und fragt hierbei bei jedem Volumen nach.

```
debian:~# vgremove vg01 --test
Test mode: Metadata will NOT be updated.
Do you really want to remove volume group "vg01" containing 1 logical volumes?
[y/n]: y
Do you really want to remove active logical volume "lv-linear"? [y/n]: n
Logical volume "lv-linear" not removed
Command failed with status code 5.
```

Aufgrund des Nachfragens konnte im letzten Moment der Verlust des logischen Volumens „*lv-linear*“ verhindert werden. Jetzt wird die Volumengruppe „*vg02*“ endgültig gelöscht.

Da die Gruppe keine logischen Volumen enthalten hat, wurde auch keine weitere Sicherheitsabfrage vom Werkzeug gestellt.

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-t	--test	Das Entfernen der Volumengruppe nur simulieren.
-v	--verbose	Ein paar mehr Informationen über den Vorgang ausgeben.
-f	--force	Mache das, was ich sage. Ohne weiteres Nachfragen.

Tabelle 6: Werkzeugoptionen von *vgremove*

Erstellen von logischen Volumen

Logische Volumen werden mit dem Werkzeug „*lvcreate*“ erzeugt. Beim Erstellen wird die Größe des neuen Volumen und der Name der Volumengruppe, in der das Volumen erstellt werden soll, erwartet. Zusätzlich muss über die Werkzeugeoptionen ein Name für das neue logische Volumen angegeben werden, der dieses eindeutig kennzeichnet. Beim linearen Erstellen des logischen Volumens werden ihm freie PE-Verwaltungseinheiten aus der Volumengruppe zugewiesen. Es erfolgt hierbei keine direkte Zuweisung oder Aufteilung auf die eingebundenen physikalischen Volumen.

```

debian:~# lvcreate -L 250m -n lv-linear vg01
Logical volume "lv-linear" created
debian:~# lvcreate -L 250m -n lv-nc vg01
Logical volume "lv-linear" created
    
```

Das Beispiel erstellt in der Volumengruppe „vg01“ zwei neue logischen Volumen mit der Größe von 250 Megabyte und den Namen „lv-linear“ und „lv-nc“ in linearer Weise an.

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-L VALUE	-size VALUE	Zu reservierende Größe des neuen Volumens in Byte. Für größere Volumen können die Suffixe <i>M</i> (egabyte), <i>G</i> (igabyte), <i>T</i> (errabyte), <i>P</i> (etabyte) oder <i>E</i> (xabyte) angehängt werden.
-l		Zu reservierende Größe des neuen Volumens in logischen Verwaltungseinheiten. Mit einem Suffix ist es möglich, die Angabe in Prozent relativ zur Volumengruppe „%VG“, physikalischer Volumen „(%PV)“ oder den freien Einheiten in logischen Verwaltungseinheiten „(%FREE)“ in einer Volumengruppe anzugeben.
-n	-name VALUE	Name des logischen Volumens.
-r VALUE	-readahead VALUE	Festlegen der Sektoren, die vorausschauend geladen werden sollen. Der Standardwert ist „Auto“. Er legt den Wert automatisch bei Starten des Volumens fest. Der Wert „None“ deaktiviert das vorausschauende Laden von Sektoren. Werte zwischen 2 bis 120 sind für Volumen im LVM1-Format gültig.
-p [r/rw]	-permission [r/rw]	Legt fest, ob das Volumen geschrieben oder nur gelesen werden darf.

Tabelle 7: Optionen von *lvcreate*

Abfragen von Volumen-Information

Information über die Konfiguration eines logischen Volumens liefert das Werkzeug „*lvdisplay*“. Zusätzlich kann das Werkzeug „*vgdisplay*“ mit der Verbose-Option (-v) verwendet werden, um die Konfiguration vorhandener logischen Volumen auszugeben.

```

debian:~# lvdisplay vg01/lv-linear
--- Logical volume ---
LV Name                /dev/vg01/lv-linear
VG Name                vg01
LV UUID                z5DibN-wyGJ-vMnY-A8qn-G1cb-c5V1-06dh97
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                250,00 MB
Current LE             125
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device           254:0
  
```

<i>Attribut</i>	<i>Beschreibung</i>
LV Name	Bezeichnung und Pfad der Gerätedatei des logischen Volumens.
VG Name	
LV UUID	Universall Unique Identifier - Unabhängige Volumen oder Datenträgerbezeichnung. Wird beim Anlegen des logischen Volumens automatisch generiert.
LV Write Access	Aktueller Lese-/Schreibstatus.
LV Status	Status des logischen Volumens .
# open	
LV Size	Speichergröße des logischen Volumens.
Current LE	Anzahl der zugewiesen logischen Verwaltungseinheiten.
Segments	Anzahl der Segmente aus der das Volumen besteht.
Allocation	Zuweisungsrichtlinie für neue Verwaltungseinheiten.
Read ahead sectors	Status des vorausschauenden Lesens von Sektoren.
- currently set to	Anzahl der Sektoren, die vorausschauend gelesen werden.
Block device	Major und Minor des verwendeten Blockgerätes.

Tabelle 8: Attribute logischer Volumen

Entfernen von logischen Volumen

Logische Volumen, die nicht mehr benötigt werden, können wieder aus der Volumengruppe entfernt werden um die belegte logischen Verwaltungseinheiten freizugeben. Für diese Aufgabe steht das Werkzeug „*lvremove*“ zur Verfügung. Es erwartet den Pfad zur Gerätedatei des logischen Volumens und verfügt über eine Option für einen Testmodus, in dem das Entfernen nur simuliert wird.

```
debian:~# lvremove /dev/vg01/lv-nc --test -v
Test mode: Metadata will NOT be updated.
Using logical volume(s) on command line
Do you really want to remove active logical volume "lv-nc"? [y/n]: y
Test mode: Skipping archiving of volume group.
Found volume group "vg01"
Found volume group "vg01"
Releasing logical volume "lv-nc"
Test mode: Skipping volume group backup.
Logical volume "lv-nc" successfully removed
Test mode: Wiping internal cache
Wiping internal VG cache
```

```
debian:~# lvremove /dev/vg01/lv-nc
Do you really want to remove active logical volume "lv-nc"? [y/n]: y
Logical volume "lv-nc" successfully removed
```

Die zusätzliche Verbose-Option sorgt für ein paar weitere Information. Das Werkzeug fragt interaktiv nach, ob man es mit dem Löschen auch wirklich ernst meint. Da das der Fall ist, wird das logische Volumen jetzt komplett entfernt.

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-t	--test	Das Entfernen des logischen Volumens nur simulieren.
-v	--verbose	Ein paar zusätzliche Informationen über den Vorgang ausgeben.
-f	--force	Mache, was ich sage. Ohne weiteres Nachfragen.

Tabelle 9: Optionen von *lvremove*

Verwenden von logischen Volumen

Logische Volumen werden als normale Blockgeräte im System verwendet und eingebunden und müssen vor ihrer Verwendung mit einem Dateisystem vorbereitet werden. Welches Dateisystem zum Einsatz kommt, ist abhängig vom Einsatzszenario des Volumens. Im Beispiel wird das Volumen mit einem Standard Ext2-Dateisystem formatiert und an den neu erzeugten Mountpoint „/mnt/linear“ angebunden.

```
debian:~# mkfs -t ext2 /dev/vg01/lv-linear
mke2fs 1.41.3 (12-Oct-2008)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=1024 (log=0)
Fragmentgröße=1024 (log=0)
64000 Inodes, 256000 Blöcke
12800 Blöcke (5.00%) reserviert für den Superuser
Erster Datenblock=1
Maximale Dateisystem-Blöcke=67371008
32 Blockgruppen
8192 Blöcke pro Gruppe, 8192 Fragmente pro Gruppe
2000 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Schreibe Inode-Tabellen: erledigt
Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt

Das Dateisystem wird automatisch nach jeweils 38 Einhäng-Vorgängen bzw.
alle 180 Tage überprüft, je nachdem, was zuerst eintritt. Verwendbar mit
tune2fs -c oder -t .
debian:~# mkdir /mnt/linear
debian:~# mount /dev/vg01/lv-linear /mnt/linear
debian:~# df
Dateisystem          1KBlöcke   Benutzt Verfügbar Ben% Eingehängt auf
/dev/sda2             6870656   1359592   5162052   21% /
tmpfs                 253144    0         253144    0% /lib/init/rw
udev                 10240     764       9476     8% /dev
tmpfs                 253144    0         253144    0% /dev/shm
/dev/mapper/vg01-lv--linear
247919              2062     233057    1% /mnt/linear
```

Administrative Wartungsarbeiten

Linux-Server sind langlebig, und während ihrer Betriebszeit pflegeleicht. Solange man nicht anfängt, an ihnen herum zuschrauben. Aber irgendwann kommt der Tag im Leben eines jeden Servers, wo es Zeit für ein paar Upgrades oder andere positive Veränderungen wird. Jetzt schlägt die Stunde für LVM und man kann die Früchte ernten, die man beim Aufsetzen müheselig ausgesät hat. Auch wenn viele Wartungsarbeiten am LVM2 online durchgeführt werden können, ist es nicht hilfreich sie in der Daten-Rush-Hour in Angriff zu nehmen. Eine Datensicherung aller zu bearbeitenden Volumen oder Volumengruppen sollte obligatorisch vor den Arbeiten durchgeführt und seltene Operationen im Labor getestet werden.

Statusinformation abfragen

Die Display-Werkzeuge für die drei Grundelemente geben eine große Menge an Information heraus. In der Regel möchte sich der Systemverwalter aber einen kurzen und schnellen Überblick über die Volumenkonfiguration verschaffen. Diese ist mit den Statuswerkzeugen „pvs“, „vgs“ und „lvs“ möglich.

```

debian:~# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sdd1   vg01   lvm2 a-    976,00M 726,00M
/dev/sde1           lvm2 --    977,86M 977,86M
debian:~# vgs
VG  #PV #LV #SN Attr   VSize   VFree
vg01  1  1  0 wz--n- 976,00M 726,00M
debian:~# lvs
LV          VG      Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv-linear  vg01   -wi-a- 250,00M
    
```

Neben den allgemeinen Informationen zeigen die Werkzeuge über die Attribute einige spezielle Zustände der Volumen und der Volumengruppe an. Die Attribute verteilen sich hierbei auf mehrere Zeichen, die jeweils eine Kategorie zusammenfassen.

<i>Zeichen</i>	<i>Attribut</i>	<i>Beschreibung</i>
1	a	
2		

Tabelle 10: pvs - Attribute physikalischer Volumen

Zeichen	Attribut	Beschreibung
1	<i>r</i>	<i>Zugriffsberechtigung</i> – die Volumengruppe kann nur gelesen werden.
1	<i>w</i>	<i>Zugriffsberechtigung</i> – die Volumengruppe ist beschreibbar, das impliziert die Lesbarkeit der Volumengruppe,
2	<i>z</i>	Die Gruppe kann vergrößert werden.
3	<i>x</i>	Die Gruppe ist exportiert.
4	<i>p</i>	
5	<i>c</i>	<i>Zuweisungsrichtlinie</i> – contiguous
5	<i>l</i>	<i>Zuweisungsrichtlinie</i> – cling
5	<i>n</i>	<i>Zuweisungsrichtlinie</i> – normal
5	<i>a</i>	<i>Zuweisungsrichtlinie</i> – anywhere
5	<i>i</i>	<i>Zuweisungsrichtlinie</i> – inherit
6	<i>c</i>	Die Volumengruppe ist Teil eines Clusters.
1	<i>m</i>	<i>Volumen-Typ</i> – gespiegeltes Volumen.
1	<i>M</i>	<i>Volumen-Typ</i> – gespiegeltes Volumen ohne initiale Synchronisation.
1	<i>o</i>	<i>Volumen-Typ</i> –
1	<i>p</i>	<i>Volumen-Typ</i> – Zugehörige PEs werden verschoben.
1	<i>s</i>	<i>Volumen-Typ</i> – Volumen ist ein Snapshot.
1	<i>S</i>	<i>Volumen-Typ</i> – Volumen ist ein ungültiges Snapshot.
1	<i>v</i>	<i>Volumen-Typ</i> – Virtual
1	<i>i</i>	<i>Volumen-Typ</i> – Spiegelimage
1	<i>I</i>	<i>Volumen-Typ</i> – Spiegelimage ohne Synchronisation.
1	<i>c</i>	<i>Volumen-Typ</i> – befindet sich in Konvertierung.
2	<i>r</i>	<i>Zugriffsberechtigung</i> – Volumen kann gelesen werden.
2	<i>w</i>	<i>Zugriffsberechtigung</i> – Volumen kann geschrieben werden, das impliziert automatisch die Lesbarkeit des Volumens.
3	<i>c</i>	<i>Zuweisungsrichtlinie</i> – contiguous
3	<i>l</i>	<i>Zuweisungsrichtlinie</i> – cling
3	<i>n</i>	<i>Zuweisungsrichtlinie</i> – normal
3	<i>a</i>	<i>Zuweisungsrichtlinie</i> – anywhere
3	<i>i</i>	<i>Zuweisungsrichtlinie</i> – inherited
4	<i>m</i>	Fixierter Spiegel.
5	<i>a</i>	<i>Status</i> – das Volumen ist aktiv.
5	<i>s</i>	<i>Status</i> – das Volumen ist suspendiert.
5	<i>I</i>	<i>Status</i> – das Volumen ist ungültiges Snapshot.

Zeichen	Attribut	Beschreibung
1	r	<i>Zugriffsberechtigung</i> – die Volumengruppe kann nur gelesen werden.
1	w	<i>Zugriffsberechtigung</i> – die Volumengruppe ist beschreibbar, das impliziert die Lesbarkeit der Volumengruppe,
5	S	<i>Status</i> – das Volumen ist ein ungültiges suspendiertes Snapshot.
5	d	<i>Status</i> – ein gemapptes Gerät ist ohne gültige Tabelle vorhanden.
5	i	<i>Status</i> – ein gemapptes Gerät ist mit einer inaktiven Tabelle vorhanden.
6	o	Volumen ist geöffnet.

Tabelle 11: lvs - Attribute logischer Volumen

Ändern der Speichergröße von Volumengruppen

Wird der Speicherplatz in der Volumengruppe knapp, muss schnell Nachschub her. Hierfür ist eine Festplatte oder ein Raid-Verbund ins System zu integrieren und als physikalisches Volumen vorzubereiten. Dieses wird der Volumengruppe zugewiesen, die den frischen Nachschub an physikalischen Verwaltungseinheiten für die Speicherzuteilung an ihre logischen Volumen über logische Verwaltungseinheiten verwendet. Hierbei gilt es, folgendes Problem im Auge zu behalten: Die Volumengruppe verteilt die Daten ähnlich einem JBOD-Verbund linear über mehrere physikalische Volumen. Fällt eines der physikalischen Volumen aus, bedeutet das das Ende der gespeicherten Daten. Sollten die physikalischen Volumen bisher nicht über ein redundanten Raid-Verbund realisiert sein, ist spätestens jetzt der Zeitpunkt gekommen, ernsthaft darüber nachzudenken.

Erweitern von Volumengruppen

Die Erweiterung einer Volumengruppe erfolgt durch das Hinzufügen frischer physikalischer Volumen. Hierfür kommt das Werkzeug „*vgextend*“ zum Einsatz.

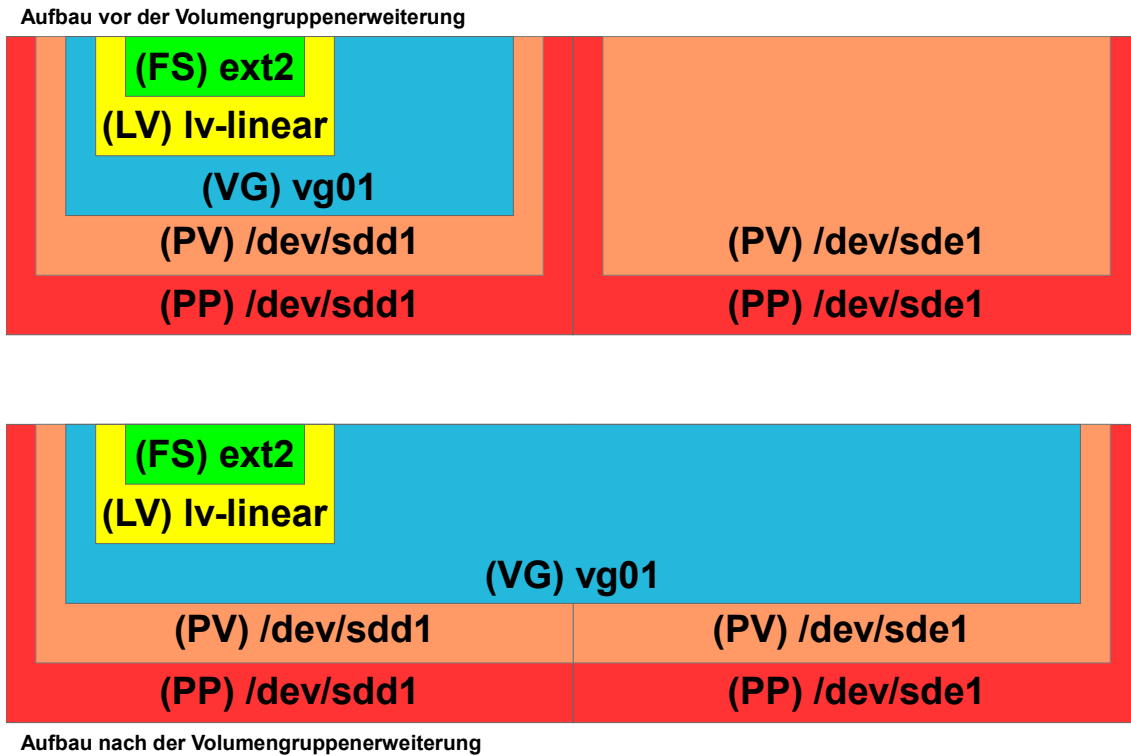


Abbildung 3: Volumengruppenerweiterung

```
debian:~# vgextend vg01 /dev/sde1
Volume group "vg01" successfully extended
```

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-t	--test	Trockenlauf ohne Änderung durchführen.
-v	--verbose	Erweiterte Information bei der Ausführung ausgeben

Tabelle 12: Optionen für "vgextend"

Das Ergebnis kann über die Volumengruppen-Information abgerufen werden.

```
debian:~# vgdisplay vg01
--- Volume group ---
VG Name                vg01
System ID
Format                 lvm2
Metadata Areas         4
Metadata Sequence No   5
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
Open LV                 1
Max PV                  0
Cur PV                 2
Act PV                  2
VG Size                 1,91 GB
PE Size                 2,00 MB
Total PE                976
Alloc PE / Size         125 / 250,00 MB
Free PE / Size           851 / 1,66 GB
VG UUID                 RZ27X2-kyaU-9Yf3-jz0s-UfbU-20Ts-juoKer
```

Ein Blick verrät, dass der Gruppe jetzt die doppelte Anzahl an logischen Extents und der doppelte Speicherplatz zur Verfügung steht. Die Anzahl der eingebundenen logischen Volumen hat sich auf zwei erhöht und es stehen vier Metadatenbereiche für Kopien der Gruppenkonfiguration zur Verfügung. Die Gruppe enthält weiterhin nur ein logisches Volumen mit 250MB Größe.

Verkleinern einer Volumengruppe

Das Verkleinern einer Volumengruppe erfolgt analog zum Vergrößern, die gewünschte Anzahl an physikalischen Volumen ist aus der Gruppe zu entfernen. Bevor dieses durchgeführt wird, ist zu prüfen, dass keine physikalischen Verwaltungseinheiten des zu entfernenden physikalischen Volumens mehr in Verwendung sind. Ist das nicht der Fall, dann sind die betreffenden physikalischen Verwaltungseinheiten auf das in der Gruppe verbleibende physikalische Volumen zu verschieben. Vorausgesetzt, es steht dort ausreichend freier Platz zur Verfügung.

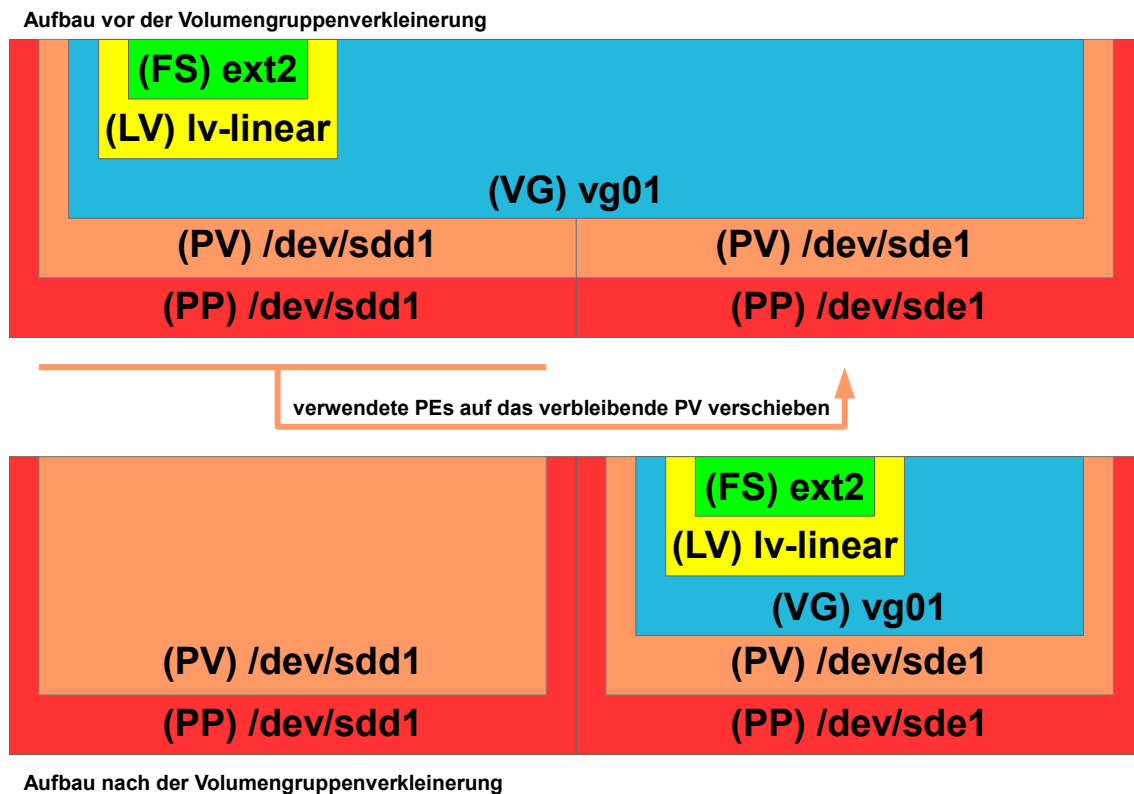


Abbildung 4: Volumengruppenverkleinerung

Im Beispiel soll das physikalische Volumen „dev/sdd1“ aus der Volumengruppe „vg01“ entfernt werden. Der Status der physikalischen Volumen legt die nächsten Schritte und ihre Reihenfolge fest.

```
debian:~# pvs
PV          VG   Fmt  Attr  PSize   PFree
/dev/sdd1   vg01 lvm2 a-   976,00M 726,00M
/dev/sde1   vg01 lvm2 a-   976,00M 976,00M
```

Hier lässt sich ablesen, dass das zu entfernende physikalische Volumen „dev/sdd1“ zu ca. 250MB belegt ist und das verbleibende physikalische Volumen „dev/sde1“ ausreichend freien Platz besitzt. Jetzt können die physikalischer Verwaltungseinheiten mithilfe des Werkzeugs

„pvmove“ ihren Platz wechseln. Das Werkzeug erwartet als erstes Argument die Quelle und als zweites Argument das Ziel der zu verschiebenden physikalischer Verwaltungseinheiten. Das Werkzeug verfügt über eine Option, die das Verschieben in einen Hintergrundprozess verschiebt. Ich halte das für keine gute Idee, da dann mühselig geprüft werden muss, ob der Vorgang abgeschlossen wurde. Mit der Test-Option kann der Vorgang einmal im Trockenlauf getestet werden.

```
debian:~# pvmove /dev/sdd1 /dev/sde1
/dev/sdd1: Moved: 12,0%
/dev/sdd1: Moved: 21,6%
/dev/sdd1: Moved: 31,2%
/dev/sdd1: Moved: 38,4%
/dev/sdd1: Moved: 45,6%
/dev/sdd1: Moved: 52,0%
/dev/sdd1: Moved: 61,6%
/dev/sdd1: Moved: 69,6%
/dev/sdd1: Moved: 76,8%
/dev/sdd1: Moved: 84,0%
/dev/sdd1: Moved: 91,2%
/dev/sdd1: Moved: 98,4%
/dev/sdd1: Moved: 100,0%
```

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-b	--background	Verschiebevorgang in einen Hintergrundprozess schieben.
-i VALUE	--interval VALUE	
-t	--test	Trockenlauf ohne Änderungen durchführen.
-n VALUE	--name VALUE	
-v	--verbose	Erweiterte Information bei der Ausführung ausgeben.

Tabelle 13: ausgewählte Optionen von pvmove

Im nächsten Schritt wird das physikalische Volumen „/dev/sdd1“ mithilfe des Werkzeugs „vgreduce“ aus der Volumengruppe „vg01“ entfernt.

```
debian:~# vgreduce vg01 /dev/sdd1
Removed "/dev/sdd1" from volume group "vg01"
```

<i>Option</i>	<i>Option</i>	<i>Beschreibung</i>
-a	--all	
-t	--test	Trockenlauf ohne Änderungen durchführen.
-v	--verbose	Erweiterte Information bei der Ausführung ausgeben.
	--mirroronly	
	--removemissing	

Tabelle 14: Optionen von *vgreduce*

Zum Abschluss ein Überblick über die aktuelle Volumengruppen-Information der Volumengruppe „vg01“ und den Status der physikalischen Volumen:

```

debian:~# vgdisplay vg01
--- Volume group ---
VG Name                vg01
System ID
Format                 lvm2
Metadata Areas        2
Metadata Sequence No  9
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               1
Max PV                 0
Cur PV                1
Act PV                1
VG Size                976,00 MB
PE Size                2,00 MB
Total PE              488
Alloc PE / Size       125 / 250,00 MB
Free PE / Size        363 / 726,00 MB
VG UUID                RZ27X2-kyaU-9Yf3-jZ0s-UfbU-20Ts-juoKer

debian:~# pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sdd1   lvm2  --   977,86M 977,86M
/dev/sde1   vg01  lvm2 a-    976,00M 726,00M

```

Am Status lässt sich mit einem Blick ablesen, dass das physikalische Volumen „/dev/sdd1“ nicht mehr verwendet wird und das Volumen „/dev/sde1“ alleine der Volumengruppe „vg01“ zugewiesen ist. Die Volumengruppeninformation verrät, dass der Gruppe nur noch 976MB und ein physikalisches Volumen zur Verfügung steht.

Ändern der Speichergröße von logischen Volumen

Die Größenänderung vorhandener logischer Volumen ist eine Standardoperation der Verwaltung und kann in der Regel im Onlinebetrieb des Volumens durchgeführt werden, wobei es einiges zu beachten gibt. Das Volumen bildet nur den Container für ein darin enthaltenes Dateisystem. Daraus folgt, dass eine Vergrößerung dieses Containers in der Regel unproblematisch durchgeführt werden kann, beim Verkleinern aber vorher das Dateisystem entsprechend verkleinert werden muss. Die Anpassung des Dateisystems wird nicht von den LVM-Werkzeugen abgedeckt. Hier sind die zum verwendeten Dateisystem passenden Werkzeuge einzusetzen. Ob eine Anpassung des Dateisystems überhaupt möglich ist, hängt stark vom eingesetzten Dateisystem ab. Einige Dateisysteme, wie beispielsweise das professionell eingesetzte XFS, besitzen keine Werkzeuge zur Verkleinerung. Andere Dateisysteme wie EXT2 oder EXT3 können nur offline verkleinert werden und benötigen vorher eine Dateisystemprüfung, die natürlich auch nur offline durchgeführt werden kann. Eine Offlineänderung kann auch von der Kernelversion, Dateisystemversion oder der Version des vorhandenen Werkzeugkastens für die Dateisystemwartung abhängen. Derzeitig zeichnet sich aber ab, dass alle modernen Dateisysteme zumindestens eine Vergrößerung im Onlinebetrieb ermöglichen. Beim Offlinebetrieb muss das entsprechende Dateisystem aus dem Verzeichnisbaum demontiert werden. Das ist nicht unproblematisch, da hierfür keine betroffenen Dateien oder Verzeichnisse mehr im Zugriff sein dürfen. Bei einem Server kann das bedeuten, das er für die Wartungsmaßnahme im Runlevel 1 gestartet werden muss. In diesem Fall steht er nicht mehr für seine eigentlichen Aufgaben zur Verfügung, was häufig nur in der Nacht oder am Wochenende zu bewerkstelligen ist. Aus diesem Grund sollte bei der Installation das Dateisystem mit Bedacht gewählt und die Fähigkeit für die Onlineänderung getestet werden. Ein weiteres Problem stellt die Fragmentierung der Volumengruppe bei Größenänderungen logischer Volumen dar. logische Volumen sind für das Dateisystem transparent und erscheinen hier linear an einem Stück. Dieses stimmt auch, solange noch keine Änderung der Volumengröße durchgeführt wurde. Bei einer Erweiterung reserviert

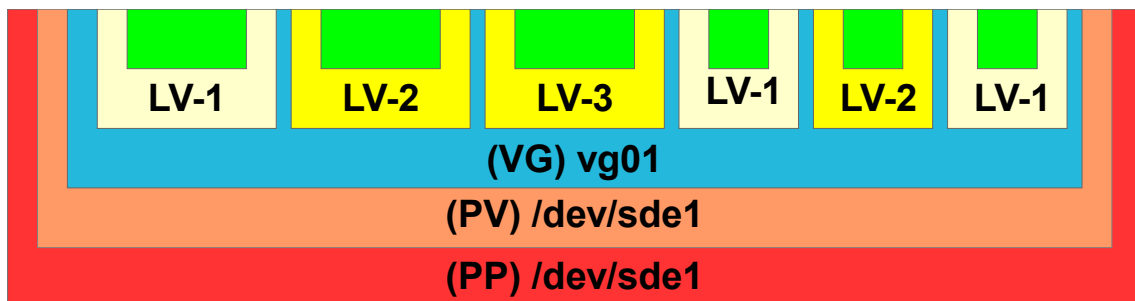


Abbildung 5: Fragmentierung einer Volumengruppe

die Volumengruppe freie logische Verwaltungseinheiten, die sich von der Geometrie aber nicht mehr in der Nähe des Volumens befinden müssen. Eine häufige Anpassung von Volumengrößen führt deshalb zu einer starken Fragmentierung innerhalb der Volumengruppe und kann zu

Performanceeinbrüchen beim Zugriff führen. Vergrößerungen sind deshalb großzügig durchzuführen, während auf Verkleinerungen möglichst zu verzichten ist.

Vergrößern von logischen Volumen

Beim Vergrößern logischer Volumen muss als erster Schritt das logische Volumen auf die gewünschte Größe erweitert werden. Danach kann das enthaltene Dateisystem auf die neue Größe angepasst werden. Hierbei ist es hilfreich, dass die Werkzeuge für die Dateisystemvergrößerung in der Lage sind, das Dateisystem automatisch auf die maximale Größe des Containers aufzublasen.

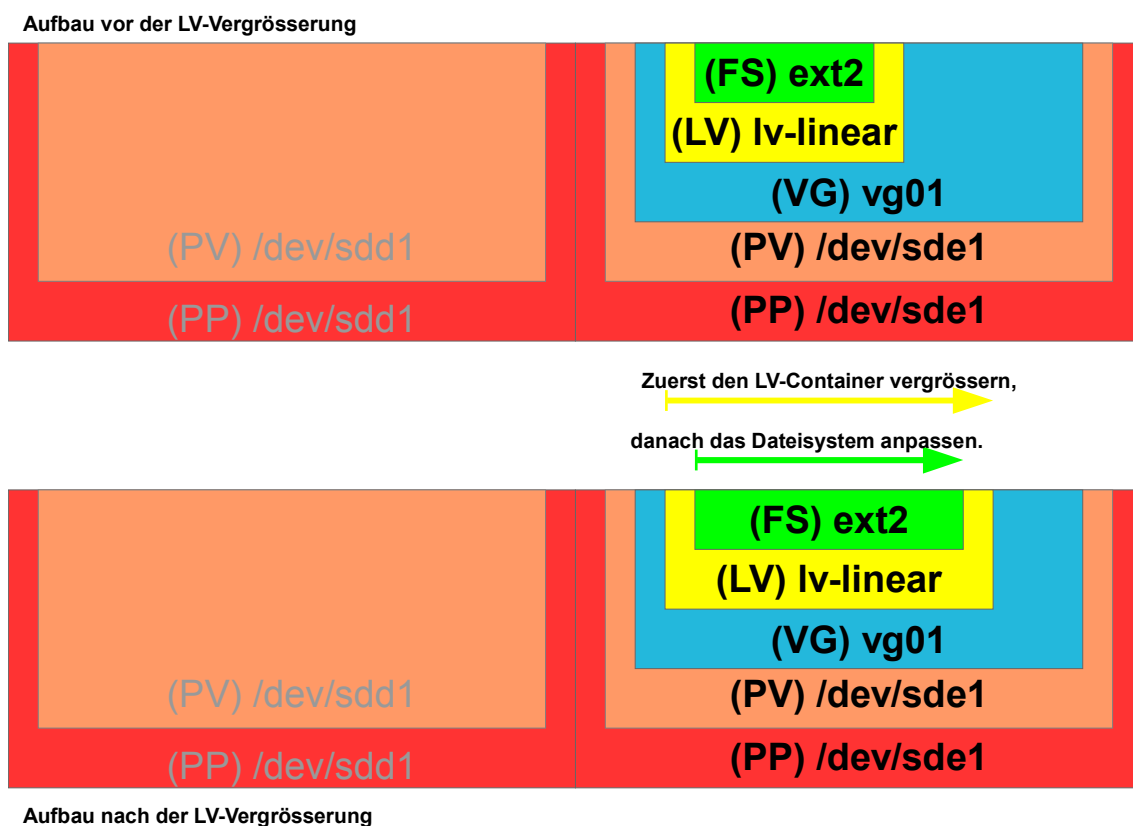


Abbildung 6: Vergrößern von logischer Volumen

Als Werkzeug für die Volumenvergrößerung kommt „*lvresize*“ zum Einsatz. Das Werkzeug für die Größenänderung des Dateisystems ist vom verwendeten Dateisystem abhängig. Im Beispiel kommt das Dateisystem ext2 zum Einsatz, hierfür ist das Werkzeug „*resize2fs*“ zuständig.

Im Beispiel wird zuerst der Status der logischen Volumen und der Volumengruppe abgefragt. Danach wird das logische Volumen „*lv-linear*“ auf 400MB vergrößert, was durch die Abfrage des Volumenstatus bestätigt wird. Das enthalte Dateisystem, das über „*/mnt/linear*“ eingebunden ist, besitzt aber laut Abfrage weiterhin nur ca. 250MB. Im nächsten Schritt soll deshalb das

Dateisystem an die neue Containergröße angepasst werden. Leider unterstützt die eingesetzte Kernelversion oder das verwendete Werkzeug nicht die Onlinevergrößerung, was in einer Spätschicht für den Administrator enden kann.

```
debian:~# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 1 1 0 wz--n- 976,00M 726,00M
debian:~# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lv-linear vg01 -wi-ao 250,00M
debian:~# lvresize /dev/vg01/lv-linear -L 400m
Extending logical volume lv-linear to 400,00 MB
Logical volume lv-linear successfully resized
debian:~# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lv-linear vg01 -wi-ao 400,00M
debian:~# df /mnt/linear
Dateisystem 1K Blöcke Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
247919 2062 233057 1% /mnt/linear
debian:~# resize2fs /dev/vg01/lv-linear
resize2fs 1.41.3 (12-Oct-2008)
Das Dateisystem auf /dev/vg01/lv-linear ist auf /mnt/linear eingehängt; Online-
Größenveränderung nötig
old desc_blocks = 1, new_desc_blocks = 2
resize2fs: Der Kernel unterstützt die Online-Vergrößerung nicht
```

```
debian:~# umount /mnt/linear
debian:~# e2fsck -f /dev/vg01/lv-linear
e2fsck 1.41.3 (12-Oct-2008)
Durchgang 1: Prüfe Inodes, Blocks, und Größen
Durchgang 2: Prüfe Verzeichnis Struktur
Durchgang 3: Prüfe Verzeichnis Verknüpfungen
Durchgang 4: Überprüfe die Referenzzähler
Durchgang 5: Überprüfe Gruppe Zusammenfassung
/dev/vg01/lv-linear: 11/64000 Dateien (0.0% nicht zusammenhängend), 10143/256000
Blöcke
debian:~# resize2fs /dev/vg01/lv-linear
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/vg01/lv-linear to 409600 (1k) blocks.
Das Dateisystem auf /dev/vg01/lv-linear ist nun 409600 Blöcke groß.
debian:~# df /mnt/linear
debian:~# mount /dev/vg01/lv-linear /mnt/linear
Dateisystem 1K Blöcke Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
396972 2309 374183 1% /mnt/linear
```

In Folge der Nachlässigkeit bei der Dateisystemauswahl und eines unterlassenen Vergrößerungstest bei der Installation muss das Dateisystem demontiert werden, was zum Glück auch auf Anhieb klappt. Eine Überprüfung des Dateisystems wird durchgeführt, ohne die das Vergrößerungswerkzeug seine Arbeit verweigern würde. Mit dem Werkzeug „*resize2fs*“ wird das Dateisystem automatisch auf die neue Containergröße erweitert. Zum Abschluss wird das Dateisystem wieder im Verzeichnisbaum montiert und eine Abfrage zeigt, dass dieses jetzt auf ca. 400MB vergrößert wurde.

Verkleinern von logischen Volumen

Die Verkleinerung logischer Volumen erfolgt analog der Vergrößerung. Hierbei ist zuerst das Dateisystem deutlich unter die Zielgröße zu verkleinern. Das ist zum einen notwendig da es zusätzlichen Speicher für seine Metainformation benötigt, der häufig bei der Verkleinerung nicht mit berücksichtigt wird. Zum anderen ist das Dateisystem für die LVM-Werkzeuge transparent und unbekannt. Sie wissen nichts über das Dateisystem, und wenn sie angewiesen werden, das Volumen zu verkleinern, wird das ohne Rücksicht auf den Inhalt des Volumens durchgeführt. Wird hierbei das Ende des Dateisystems abgeschnitten, kann das einen Dateiverlust nach sich ziehen. Nach der Verkleinerung des Volumens wird das Dateisystem wieder auf die maximal mögliche Größe angepasst.

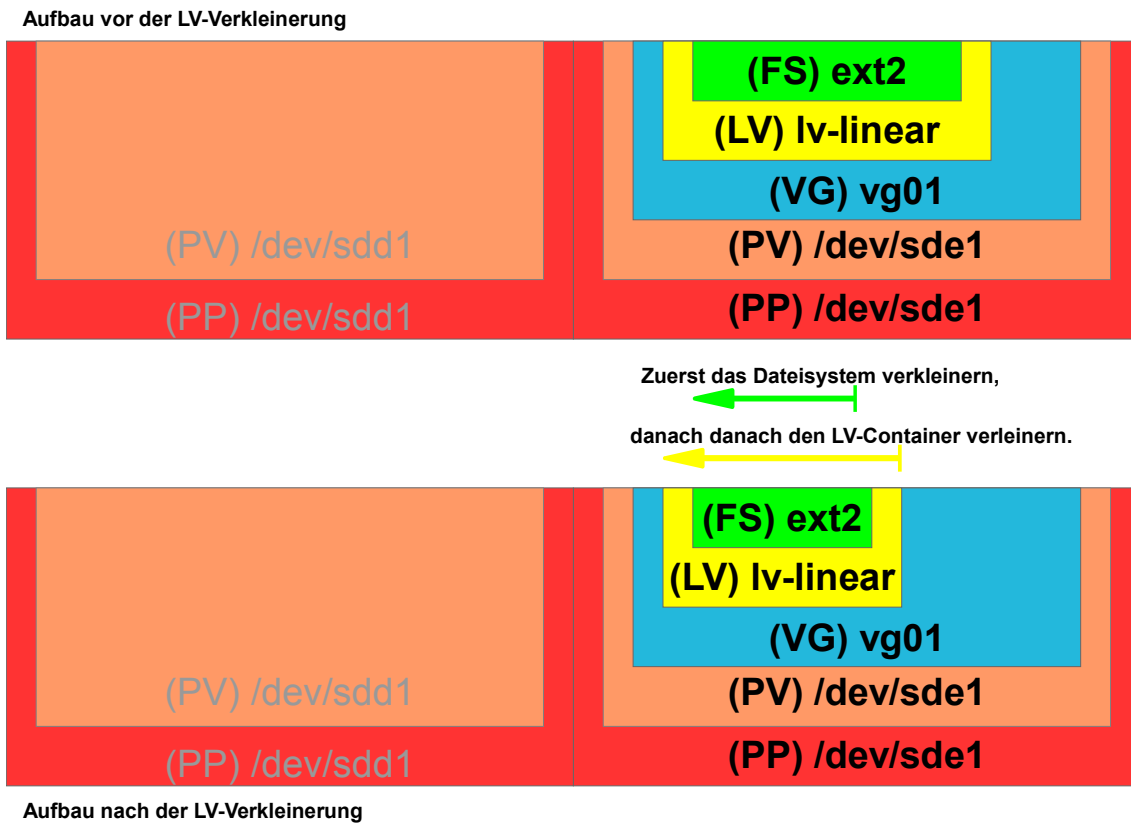


Abbildung 7: Verkleinern von logischen Volumen

Im Beispiel soll das logische Volumen „lv-linear“ auf 150MB verkleinert werden. Für die Volumenänderung kommt das Werkzeug „*lvresize*“ und für die Änderungen am EXT2-Dateisystem die Werkzeuge „*resize2fs*“ und „*fsck*“ zum Einsatz. Alternativ kann für die Verkleinerung des logischen Volumens auch das Werkzeug „*lvreduce*“ verwendet werden. Da sich das gewählte Dateisystem nur im Offline-Betrieb verkleinern lässt, muss es aus dem Verzeichnisbaum demontiert werden. Hierfür kann bei einer exzessiven Nutzung ein Neustart des Servers im Runlevel 1 notwendig sein.

```
debian:~# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg01  1  1  0 wz--n- 976,00M 576,00M
debian:~# lvs
LV      VG   Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv-linear vg01 -wi-ao 400,00M
debian:~# df /mnt/linear
Dateisystem          1KBlöcke   Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
                        396972      2309    374183    1% /mnt/linear
```

Im ersten Schritt wird der Zustand der Volumen und des Dateisystems abgefragt. Das betreffende Volumen hat eine Größe von 400MB und beinhaltet ein Dateisystem, das mit ca. 2,5MB belegt ist. Damit steht fest, dass eine Verkleinerung auf die gewünschte Größe möglich ist.

```
debian:~# umount /mnt/linear
debian:~# fsck /dev/vg01/lv-linear -f
fsck 1.41.3 (12-Oct-2008)
e2fsck 1.41.3 (12-Oct-2008)
Durchgang 1: Prüfe Inodes, Blocks, und GrÃ¶Ã¶en
Durchgang 2: Prüfe Verzeichnis Struktur
Durchgang 3: Prüfe Verzeichnis VerknÃ¼pfungen
Durchgang 4: Überprüfe die ReferenzzÃ¼hler
Durchgang 5: Überprüfe Gruppe Zusammenfassung
/dev/vg01/lv-linear: 11/100000 Dateien (0.0% nicht zusammenhängend), 14937/409600
Blöcke
debian:~# resize2fs /dev/vg01/lv-linear 80m
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/vg01/lv-linear to 81920 (1k) blocks.
Das Dateisystem auf /dev/vg01/lv-linear ist nun 81920 Blöcke groß.
```

Im nächsten Schritt wird das betroffene Dateisystem aus dem Verzeichnisbaum demontiert und eine Überprüfung des Dateisystems erzwungen. Danach wird das Dateisystem verkleinert. Damit das Dateisystem beim Verkleinern des Volumens nicht beschädigt wird, liegt die neue Größe deutlich unter der Wunschgröße von 150MB, aber auch deutlich über dem aktuell Füllstand des Dateisystems.

```

debian:~# lvresize /dev/vg01/lv-linear -L 150m
WARNING: Reducing active logical volume to 150,00 MB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv-linear? [y/n]: y
Reducing logical volume lv-linear to 150,00 MB
Logical volume lv-linear successfully resized
debian:~# resize2fs /dev/vg01/lv-linear
resize2fs 1.41.3 (12-Oct-2008)
Resizing the filesystem on /dev/vg01/lv-linear to 153600 (1k) blocks.
Das Dateisystem auf /dev/vg01/lv-linear ist nun 153600 Blöcke groß.
debian:~# mount /dev/vg01/lv-linear /mnt/linear
debian:~# lvs
LV          VG      Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv-linear  vg01   -wi-ao 150,00M
debian:~# df /mnt/linear
Dateisystem          1KBBlöcke   Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
                    148799         1550    139569    2% /mnt/linear

```

Im letzten Schritt wird das logische Volumen auf die Wunschgröße verkleinert und das Dateisystem automatisch an die maximal mögliche Größe angepasst. Das Dateisystem wird wieder im Dateibaum montiert und eine Statusabfrage bestätigt eine Volumengröße von 150MB und eine Dateisystemgröße von ca. 150MB.

Spezialitäten unter LVM

Neben dem Vorteil der abstrakten Speicherverwaltung bietet LVM2 einige Spezialitäten an. Mit Snap Shots kann der Datenzustand logischer Volumen temporär eingefroren werden, was besonders bei einer Datensicherung von Dateien von Vorteil sein kann. Zusätzlich sind Raid ähnliche Funktionen verfügbar. Logische Volumen können gespiegelt oder als Stripe Set über mehrere physikalische Volumen verteilt werden. Auch wenn damit weder Raid-Controller oder Softwareraid zu ersetzen sind, kann diese Funktionalität den administrativen Alltag gelegentlich vereinfachen.

Snapshots für logische Volumen

Snapshots ermöglichen, im laufenden Betrieb ein Zustandsabbild eines logischen Volumen einzufrieren. Dieses ist schnell eingerichtet, benötigt gegenüber einem Image wenig zusätzlichen Speicher in der Volumengruppe und ist für die Anwender völlig transparent. Das originale logische Volumen funktioniert hierbei weiterhin wie gewohnt, sodass der Anwender keine Änderung feststellen wird. Das bedeutet, er kann weiterhin Dateien anlegen, löschen oder ändern und alle Operationen werden scheinbar wie immer durchgeführt. Über das Snapshot steht allerdings der Datenstand zum Zeitpunkt des Einfrierens zur Verfügung. Leider überdauert ein Snapshot keinen Neustart des Servers und ist nicht für die forensische Sicherung geeignet. Für eine dateikonsistente Datensicherung im laufenden Betrieb oder zur Synchronisation ist es ideal geeignet.

Snapshots werden in der Volumengruppe als Volumen angelegt und mit dem zugehörigen logischen Volumen verbunden. Über die zugehörige Gerätedatei kann auf den Datenstand des betreffenden Volumens zum Zeitpunkt der Snapshot-Erstellung zugegriffen werden. Beim Erstellen ist eine Speichergröße eines Puffers anzugeben, in dem alle Änderungen, die von den Anwendern beim Arbeiten an den physikalischen Verwaltungseinheiten durchführen werden, landen. Die Puffergröße ist für die geplante Lebensdauer des Snapshot ausreichend zu dimensionieren. Bei der Festlegung der Snapshotgröße ist es wichtig zu beachten, dass die Arbeitsgranularität der Größe der physikalischen Verwaltungseinheiten entspricht. Wird der Platz knapp, lassen sich Snapshots mit den entsprechenden Werkzeugen zur Volumengrößenänderung vergrößern. Läuft der reservierte Puffer über, wird das betreffende Snapshot automatisch aufgelöst und alle Änderungen ins logische Volumen übertragen. Dieses gilt auch, wenn das Snapshot manuell vom Administrator aufgelöst wird.

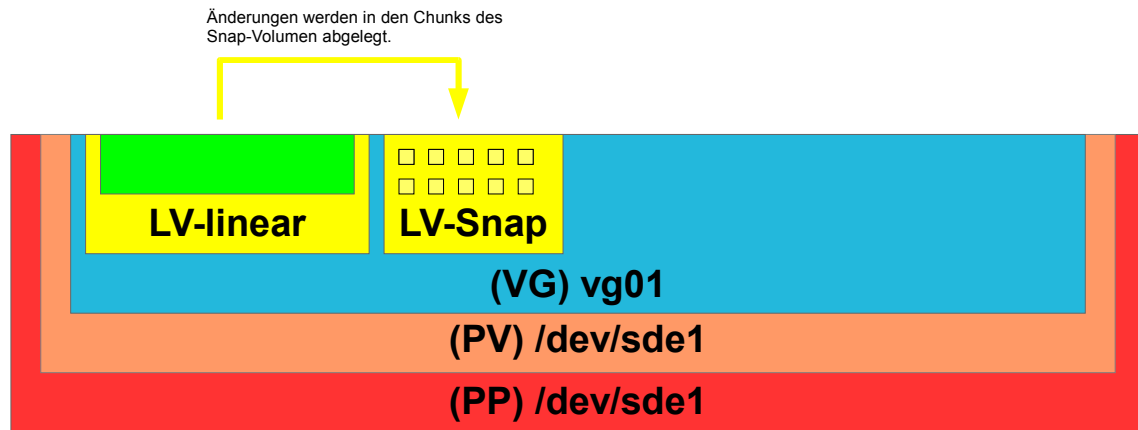


Abbildung 8: Funktion von Snap Shots

Im Beispiel wird für das linear angelegte Volumen ein Snapshot erzeugt und getestet. Hierbei kommen die bereits bekannten Werkzeuge „*lvcreate*“ zum Erstellen des Snapshot-Volumens, „*lvresize*“ zur Größenänderung des Volumens und „*lvremove*“ zum Auflösen des Snapshots zum Einsatz.

```

debian:~# lvs
LV      VG      Attr  LSize   Origin Snap%   Move Log Copy%  Convert
lv-linear vg01   -wi-ao 150,00M
lv-stripe vg01   -wi-ao 400,00M
debian:~# echo "Hallo Welt" > /mnt/linear/test.txt
debian:~# lvcreate -s -n snap -L 20m /dev/vg01/lv-linear
Logical volume "snap" created
debian:~# lvs
LV      VG      Attr  LSize   Origin  Snap%   Move Log Copy%  Convert
lv-linear vg01   owi-ao 150,00M
lv-stripe vg01   -wi-ao 400,00M
snap    vg01   swi-ao 20,00M  lv-linear  0,04
debian:~# mkdir /mnt/snap
debian:~# mount /dev/vg01/snap /mnt/snap
debian:~# df /mnt/*
Dateisystem          1K Blöcke   Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
148799             1551    139568    2% /mnt/linear
/dev/mapper/vg01-snap
148799             1551    139568    2% /mnt/snap
/dev/mapper/vg01-lv--stripe
396672             2318    373874    1% /mnt/strip

```

Im ersten Schritt zeigt der Status die augenblickliche Konfiguration der Volumen an. Das betreffende logische Volumen ist an den Mountpoint „/mnt/linear“ gebunden, hier wird ein Text in eine Testdatei geschrieben. Jetzt wird das Snapshot für das Volumen erzeugt. Das Werkzeug „*lvcreate*“ erhält hierfür die Option „-s“ und das Argument, für welches bestehende logische Volumen das Snapshot angelegt werden soll. Das Snapshot bekommt eine Größe von 20Mb und hört auf den Namen „snap“. Der danach abgerufene Volumenstatus listet das Snapshot auf. Aus der Auflistung geht neben der Größe, auch die prozentual verbrauchte Speichermenge und der Name des originalen Volumens, an den das Snapshot gebunden ist, hervor. Um das Snapshot zu testen, wird der Mountpoint „/mnt/snap“ angelegt und das Snap-Volumen mit diesem montiert. Die Auflistung der montierten Volumen offenbart gleiche Werte für das Originalvolumen unter „/mnt/linear“ und das Snap-Volumen unter „/mnt/snap“. Als Unterschied ist das Snap-Volumen nicht beschreibbar eingebunden, sodass über den betreffenden Mountpoint keine Änderungen vorgenommen werden können.

```

debian:~# echo "Hallo du schöne Welt" > /mnt/linear/test.txt
debian:~# cat /mnt/linear/test.txt
Hallo du schöne Welt
debian:~# cat /mnt/snap/test.txt
Hallo Welt
debian:~# lvs
LV      VG      Attr  LSize   Origin  Snap%   Move Log Copy%  Convert
lv-linear vg01   owi-ao 150,00M
lv-stripe vg01   -wi-ao 400,00M
snap    vg01   swi-ao 20,00M  lv-linear  0,12

```

Im nächsten Schritt wird der Inhalt der vorher angelegten Testdatei geändert, was über die Ausgabe der Datei bestätigt wird. Über das montierte Snap-Volumen ist aber weiterhin der alte Stand zum Zeitpunkt der Erstellung des Snapshots abrufbar. Aus diesem Grund eignen sich Snapshots hervorragend für transparente Datensicherungen auf Dateibasis, von denen der Anwender nichts mitbekommen soll. Die Ausgabe des Volumenstatus bestätigt, dass sich der Speicherverbrauch auf dem Snap-Volumen erhöht hat. Die geänderten Informationen sind damit im Puffer des Snap-Volumens abgelegt worden.

```
debian:~# lvs vg01/snap
--- Logical volume ---
LV Name                /dev/vg01/snap
VG Name                vg01
LV UUID                ulx8oj-Y6HR-Bomy-rzeW-LF7h-WcGw-vW8IB1
LV Write Access        read/write
LV snapshot status     active destination for /dev/vg01/lv-linear
LV Status              available
# open                 1
LV Size                150,00 MB
Current LE             75
COW-table size         20,00 MB
COW-table LE          10
Allocated to snapshot 0,12%
Snapshot chunk size    4,00 KB
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    256
Block device           254:2
```

Erweiterte Information über das Snap-Volumen liefert das Werkzeug „lvs“. Hier sind der Status und das Zielvolumen inklusive der Stammdaten des Zielvolumens wie Größe und Anzahl der logischen Verwaltungseinheiten abrufbar. Die Funktionalität des Snapshots erfolgt durch das Mappen geänderter Information. Dieses verwaltet das Snap-Volumen über die COW-Tabelle, ihr steht hierfür der reservierte Speicherplatz zur Verfügung. Der verfügbare Speicher wird in der Tabelle über Chunks verwaltet, die in der Beispielkonfiguration eine definierte Größe von 4kb haben. Das bedeutet, Änderungen werden mit einer Granularität von 4kb abgelegt und verwaltet. Die Chunk-Größe kann beim Anlegen des Snapshots festgelegt werden und als gute Wahl gilt die Blockgröße des darüberliegenden Dateisystems, das häufig eine Blockgröße von 4kb aufweist. Bei der vorherigen Änderung hat sich der Speicherverbrauch um 0,08% erhöht, was einer Größe von 16kb oder vier Chunks entspricht. Die vorgenommene Änderung ist mit Sicherheit kleiner als 1kb, aber warum wurden vier Chunks aus dem knappen Snapshot Speicher verwendet? Die Antwort ist ganz einfach. Das über das Volumen gelegte Dateisystem hat vier Änderungen durchgeführt. Zuerst hat es die neuen Daten im Dateisystem abgelegt und danach zum Erzeugen der neuen Datei eine Inode reserviert, was schon zwei Änderungen entspricht. Dann wurde der Verzeichniseintrag für die neue Datei erzeugt und mit der neuen Inode verknüpft, damit ist die Datei im Dateisystem verfügbar. Zum Schluss wurde die alte Inode, die auf die alten Daten zeigt, freigegeben und so der alte Inhalt gelöscht. Damit wurden vom Dateisystem vier Änderungen an unterschiedlichen Blöcken vorgenommen, die vom LVM in vier Chunks abgelegt wurden. Da LVM den verfügbaren Speicher transparent zur Verfügung stellt,

besitzt es kein Wissen über den Aufbau und die Struktur des drüberliegenden Dateisystems. Es kann nur Änderungen in den Blöcken erfassen und diese mappen, den Sinn dieser Änderungen kann es nicht erfassen. Bei der Größenfestlegung des Snap-Volumens sollte dieses bedacht und ausreichend Platz vorgesehen werden. Fällt trotz der gewissenhaften Planung auf, dass der Platz im Snap-Volumen knapp wird, kann das Volumen ohne Probleme vergrößert werden. Läuft das Snap-Volumen über, wird dieses automatisch aufgelöst und die gemappten Änderungen werden in das Zielvolumen übertragen.

```
debian:~# lvresize -L 50m vg01/snap
Extending logical volume snap to 50,00 MB
Logical volume snap successfully resized
debian:~# lvs
LV          VG      Attr      LSize   Origin      Snap%   Move Log Copy%   Convert
lv-linear  vg01   owi-ao   150,00M
lv-stripe  vg01   -wi-ao   400,00M
snap       vg01   swi-ao   50,00M  lv-linear   0,05
```

Für die Vergrößerung kommt das Werkzeug „*lvresize*“ zum Einsatz. Der Status zeigt an, dass die Vergrößerung wie beabsichtigt durchgeführt wurde und das Snap-Volumen ausreichend Platz zur Verfügung hat. Jetzt wird es Zeit das Snap-Volumen aufzulösen.

```
debian:~# umount /mnt/snap/
debian:~# lvremove vg01/snap
Do you really want to remove active logical volume "snap"? [y/n]: y
Logical volume "snap" successfully removed
debian:~# cat /mnt/linear/test.txt
Hallo du schöne Welt
```

Vor dem Auflösen wird das Volumen vom Mountpoint getrennt und danach mit dem Werkzeug „*lvremove*“ aufgelöst. Bei diesem Vorgang wird eine interaktive Bestätigung abgefragt, die mit der Option „-s“ unterdrückt werden kann. Eine Prüfung der Datei bestätigt, dass die vorherige Änderung an der Datei beim Auflösen übernommen worden ist.

Logical Volumen mit Stripe Sets

Zur Geschwindigkeitssteigerung bietet LVM2 die Möglichkeit, logische Volumen als Stripe Set über mehrere physikalischen Volumen innerhalb einer Volumengruppe zu verteilen. Hierbei werden die Daten in Streifen geschnitten und gleichmäßig über die eingebundenen physikalischen Volumen verteilt. Die Konstellation entspricht hierbei einer RAID0-Konfiguration und bringt einen ähnlichen Leistungsschub, aber auch die bekannte Gefahr eines erhöhten Datenverlustes mit sich. Bei der Einrichtung des logischen Volumens sind die Anzahl und Größe der Streifen anzugeben, in die die Daten geschnitten werden sollen. Hierbei ergibt sich die maximale Anzahl aus der Anzahl der zugeordneten physikalische Volumen in der Volumengruppe. Die Größe der Streifen kann zwischen 8KB und 256KB festgelegt werden und kann in Abhängigkeit der durchschnittlichen Dateigröße die Leistung beeinflussen. Wie beim RAID0 kann hier, für eine ausgewogene Leistung eine Streifenlänge von 64KB gewählt werden. Die Leistung wird sich hierbei um den Faktor der verteilten Streifen erhöhen. Die Ausfallwahrscheinlichkeit steigt ebenfalls um den Faktor der Streifen und der Ausfall eines physikalischen Volumens hat den Verlust des gesamten Datenbestandes im Stripe Set zur Folge.

Ist für die Volumengruppe die Zuweisungsrichtlinie „normal“ aktiv, wird das Einrichten mehrerer Streifen auf nur einem physikalischen Volumen verweigert, da es zu einem starken

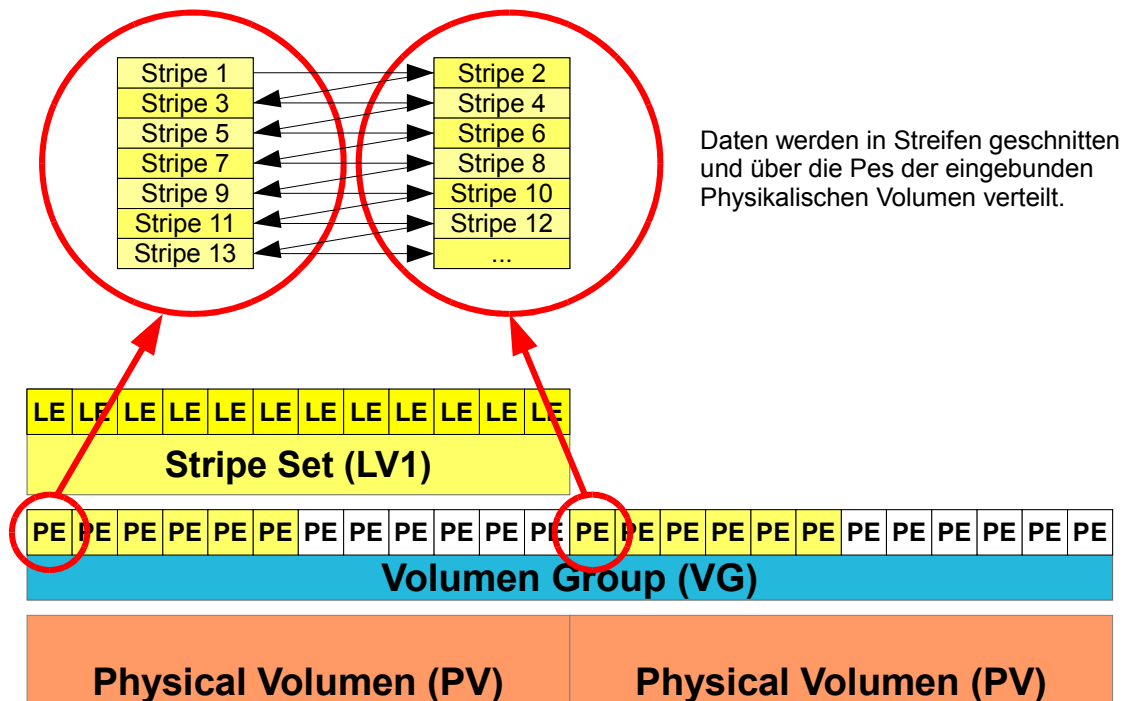


Abbildung 9: Funktion von Stripe Set

Leistungseinbruch des Volumens führen würde. Soll dieses Verhalten erzwungen werden, muss für die Volumengruppe die Richtlinie „anywhere“ aktiviert werden.

Im Beispiel wird ein neues logisches Volumen als „Stripe Set“ über zwei physikalische Volumen angelegt. Im Anschluss wird die Leistung des Volumens ermittelt und mit der Leistung des linearen Volumens verglichen.

```

debian:~# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 1 1 0 wz--n- 976,00M 826,00M
debian:~# pvs
PV VG Fmt Attr PSize PFree
/dev/sdb1 vg01 lvm2 a- 976,00M 826,00M
/dev/sde1 vg01 lvm2 -- 977,86M 977,86M
debian:~# vgextend vg01 /dev/sde1
Volume group "vg01" successfully extended
debian:~# lvcreate -i2 -I64 -L400m -n lv-stripe /dev/vg01
Logical volume "lv-stripe" created
debian:~# pvs
PV VG Fmt Attr PSize PFree
/dev/sdb1 vg01 lvm2 a- 976,00M 626,00M
/dev/sde1 vg01 lvm2 a- 976,00M 776,00M
debian:~# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 2 2 0 wz--n- 1,91G 1,37G
debian:~# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lv-linear vg01 -wi-a- 150,00M
lv-stripe vg01 -wi-a- 400,00M

```

Im ersten Schritt wird der Status der Volumen und der Volumengruppen ermittelt. Die Volumengruppe „vg01“ verfügt über ein physikalisches Volumen. Sie wird um das physikalische Volumen „/dev/sde1“ erweitert, das noch ungenutzt zur Verfügung steht. Nach der Erweiterung wird das logische Volumen „lv-stripe“ in der Volumengruppe „vg01“ mit einer Größe von 400Mb angelegt. Das Volumen wird als Stripe Set über 2 Streifen mit einer Streifengröße von 64Kb aufgeteilt. Die Statusabfrage zum Abschluss bestätigt die Erweiterung der Volumengruppe und die Nutzung der physikalischen Volumen.

```

debian:~# mkdir /mnt/stripe
debian:~# mke2fs /dev/vg01/lv-stripe
mke2fs 1.41.3 (12-Oct-2008)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=1024 (log=0)
Fragmentgröße=1024 (log=0)
102400 Inodes, 409600 Blöcke
20480 Blöcke (5.00%) reserviert für den Superuser
Erster Datenblock=1
Maximale Dateisystem-Blöcke=67633152
50 Blockgruppen
8192 Blöcke pro Gruppe, 8192 Fragmente pro Gruppe
2048 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Schreibe Inode-Tabellen: erledigt
Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt

Das Dateisystem wird automatisch nach jeweils 20 Einhäng-Vorgängen bzw.
alle 180 Tage überprüft, je nachdem, was zuerst eintritt. Veränderbar mit
tune2fs -c oder -t .
debian:~# mount /dev/vg01/lv-stripe /mnt/stripe

```

Im zweiten Schritt wird ein Mountpoint angelegt, das logische Volumen mit einem ext2 Dateisystem versehen und auf dem Mountpoint montiert.

```
debian:~# df /mnt/*
Dateisystem      1K Blöcke    Benutzt Verfügbar Ben% Eingehängt auf
/dev/mapper/vg01-lv--linear
                  148799      1550    139569    2% /mnt/linear
/dev/mapper/vg01-lv--stripe
                  396672      2318    373874    1% /mnt/strip
debian:~# free
              total        used        free      shared    buffers     cached
Mem:           28852        27616        1236           0         1568        11968
-/+ buffers/cache:    14080        14772
Swap:          979924         8296        971628
```

Vor dem Leistungsvergleich werden noch einmal die Daten abgefragt. Beide Dateisysteme sind im Verzeichnisbaum montiert und haben die eingerichtete Größe. Als Arbeitsspeicher stehen dem System nur ca. 32Mb zur Verfügung. Das ist wichtig, da nicht benutzter Speicher vom Betriebssystem automatisch als Cache verwendet wird, was das Messergebnis verfälschen würde.

```
debian:~# bonnie++ -d /mnt/linear/ -s 130m -u 0
Using uid:0, gid:0.
Writing withputc()...done
Writing intelligently...done
Rewriting...done
Reading withgetc()...done
Reading intelligently...done
start'em...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version 1.03d
-----Sequential Output----- --Sequential Input- --Random-
-Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine      Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
debian       130M 1257  8 1588  0 1026  0 12255 73 16023  2 89.7  1
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
    16 2807 99 +++++ +++ 5570 14 2440 80 4440 50 10664 99
```

Für den Test kommt das Werkzeug „*Bonnie++*“ zum Einsatz, das ggf über die Paketverwaltung nachinstalliert werden muss. Als Erstes wird das lineare Volumen mit 130Mb getestet. Hierbei treten die Schwächen der eingesetzten USB-Speicher zutage, die sich insbesondere bei einem Vergleich der Schreib- zur Leserate zeigen. Beim blockweise Schreiben kommt das Volumen auf ca. 16Mb/Sek, beim Lesen auf ca. 1.5Mb/Sek.

```

debian:~# bonnie++ -d /mnt/striped/ -s 130m -u 0
Using uid:0, gid:0.
Writing with putc()...done
Writing intelligently...done
Rewriting...done
Reading with getc()...done
Reading intelligently...done
start 'em...done...done...done...
Create files in sequential order...done.
Stat files in sequential order...done.
Delete files in sequential order...done.
Create files in random order...done.
Stat files in random order...done.
Delete files in random order...done.
Version 1.03d      -----Sequential Output----- --Sequential Input- --Random-
                  -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine           Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
debian            130M  2482  17  2745  1  2038  1 13354  81 30828  8 108.7  0
-----Sequential Create----- -----Random Create-----
-Create-- --Read--- -Delete-- -Create-- --Read--- -Delete--
files /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
      16 2383  86 +++++ +++ 8614  14  1538  52  9459 100 10839 100

```

2011-03-18 | 2011-03-15_Logical_Volumen_Management.txt | Marek Walther

Im letzten Abschnitt wird das Stripe Set Volumen getestet. Hier zeigt sich gegenüber dem linearen Volumen nahezu eine Verdoppelung der Lese- und Schreibraten. Das war zu erwarten, können die Daten doch gleichzeitig über zwei physikalische Volumen weggeschrieben werden. Interessant ist hierbei der Einbruch beim zufälligem Erzeugen von Daten und der gleichbleibende Wert beim Löschen zufällig angelegter Daten. Diese Werte könne mit der hohen Prozessor-Auslassung beim Zugriff auf die USB-Speicher im Zusammenhang stehen.

Abschließend kann festgehalten werden, dass das Stripe-Set wie erwartet funktioniert, die Schreib- und Leseleistung verbessert und dass USB-Flash-Speicher nicht wirklich als leistungsstarke Speicherlösung zu gebrauchen sind. Die Technik lässt sich gut einsetzen, um nachträglich mehr Leistung für ein Dateisystem bereit zustellen. Generell ist es aber besser für ein solches Unterfangen den Unterbau des physikalischen Volumens mittel Software RAID einzurichten. Dieses kann bei der Erweiterung eines Servers auch nachträglich durch den Umzug der physikalischen Verwaltungseinheiten erfolgen und der Leistungsschub kommt hierbei allen logischen Volumen in der Volumengruppe zugute.

Logische Volumen mit Spiegelung

Zum Erhöhen der Ausfallsicherheit bietet LVM die Möglichkeit der vereinfachten Spiegelung logischer Volumen an. Hierbei wird ein lineares logisches Volumen als Master angelegt, das automatisch auf ein oder mehrere Spiegelvolumen repliziert wird. Die Volumengruppe muss für die Anwendung über ausreichende physikalische Volumen verfügen, da jedes in der Spiegelung angelegte Volumen auf einem eigenen physikalischen Volumen abgelegt wird. Für die Spiegelung verwendet LVM eine eigene Granularität, die „*Regio*“ genannt wird und als Standardwert eine Größe von 512kb enthält. Dieser Wert entspricht der normalen Sektorgröße von blockorientierten Datenträgern. Damit die Laufwerksleistung nicht eingeschränkt wird, sollten alle eingebunden physikalischen Volumen die gleiche Sektorgröße besitzen und die Regiogröße sollte auf diese Sektorgröße eingestellt sein. In der Vergangenheit war dieses kein Problem, da die Sektorgröße bei Festplatten aus Tradition 512kb betrug. In der Zukunft werden die Hersteller diese Größe auf 4kb anheben, da sie so die Speichergröße moderner Festplatten halbwegs kompatibel erhöhen können. Physikalische Volumen mit unterschiedlichen Sektorgrößen führen beim Laufwerk mit den 4kb Sektoren zu einem vierfach höheren Overhead, was diese Platte extrem ausbremsen wird. Die von den Herstellern angebotene 512kb Emulation ist keine Abhilfe, da sich an der Problematik nichts ändert und sie nur gegenüber dem Betriebssystem verschwiegen wird. Die Regiogröße kann beim Anlegen mit der Option „-r“ angepasst werden. Die Replizierung erfolgt immer vom Mastervolumen auf die Slavevolumen und der augenblickliche Status der Regios wird in einem Log festgehalten. Für dieses Log gibt es zwei Ablageformen. Als Standard wird dieses in einem eigenen Volumen abgelegt, was in der Volumengruppe ein weiteres physikalisches Volumen benötigt. Damit sind in der Volumengruppe für ein Volumen mit einem Spiegel und einem Volumenlog, drei physikalische Volumen notwendig. Als Alternative kann das Log im Speicher angelegt werden. Hierbei geht das Log beim Ausschalten des Systems verloren und LVM ist der aktuelle Status der Regios nicht bekannt. Als Folge wird es das gesamte Mastervolumen auf die Slavevolumen replizieren, was bei großen Volumen zu einem Leistungseinbruch während der Replikation führt.

Im Beispiel wird ein logisches Volumen als Spiegelvolumen mit einer Kapazität von 50Mb eingerichtet und bereitgestellt.

```

debian:~# pvs
PV          VG      Fmt  Attr PSize   PFree
/dev/sdb1  vg01  lvm2 a-   976,00M 626,00M
/dev/sde1  vg01  lvm2 a-   976,00M 776,00M
debian:~# vgs
VG      #PV #LV #SN Attr   VSize VFree
vg01    2   2   0 wz--n- 1,91G 1,37G
debian:~# lvs
LV          VG      Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv-linear  vg01  -wi-ao 150,00M
lv-stripe  vg01  -wi-ao 400,00M
debian:~# lvcreate -L 50 -m1 --corelog -n mirror-core vg01
Logical volume "mirror-core" created

debian:~# lvs
LV          VG      Attr   LSize   Origin Snap%   Move Log Copy%   Convert
lv-linear  vg01  -wi-ao 150,00M
lv-stripe  vg01  -wi-ao 400,00M
mirror-core vg01  mwi-a- 50,00M                                     40,00
debian:~# vgs
VG      #PV #LV #SN Attr   VSize VFree
vg01    2   3   0 wz--n- 1,91G 1,27G
debian:~# pvs
PV          VG      Fmt  Attr PSize   PFree
/dev/sdb1  vg01  lvm2 a-   976,00M 576,00M
/dev/sde1  vg01  lvm2 a-   976,00M 726,00M
debian:~# lvdisplay vg01/mirror-core
--- Logical volume ---
LV Name                /dev/vg01/mirror-core
VG Name                 vg01
LV UUID                 e9TrkM-qYGC-Tysv-gnMH-EZ0D-lMvs-Hjgou9
LV Write Access        read/write
LV Status               available
# open                  0
LV Size                 50,00 MB
Current LE              25
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            254:4

```

Im ersten Schritt wird der Status der Volumen ausgegeben und geprüft, dass die Volumengruppe bereits über zwei physikalische Volumen verfügt und dass auf beiden physikalischen Volumen noch ausreichend Platz für zwei Volumen vorhanden ist. Danach wird ein neues logisches Volumen als Spiegelvolumen eingerichtet. Das Werkzeug „*lvcreate*“ bekommt dafür die Option „*-m*“ mit der Anzahl der anzulegenden Volumenspiegel übergeben. Eine nachträgliche Prüfung der Volumen zeigt, dass das neue Volumen angelegt und der notwendige Speicher auf beiden physikalischen Volumen reserviert wurde. Dass es sich hierbei um einen Spiegel handelt, kann man nur den Attributen der Ausgabe des Werkzeugs „*lvs*“ entnehmen.

Anhang

Abbildungsverzeichnis

Abbildung 1: Modelaufbau Logical Volumen Management.....	3
Abbildung 2: Verknüpfen von Verwaltungseinheiten.....	4
Abbildung 3: Volumengruppenerweiterung.....	20
Abbildung 4: Volumengruppenverkleinerung.....	22
Abbildung 5: Fragmentierung einer Volumengruppe.....	25
Abbildung 6: Vergrößern von logischer Volumen.....	26
Abbildung 7: Verkleinern von logischen Volumen.....	28
Abbildung 8: Funktion von Snap Shots.....	31
Abbildung 9: Funktion von Stripe Set.....	35

Tabellenverzeichnis

Tabelle 1: LVM-Werkzeugmatrix.....	5
Tabelle 2: Werkzeugoptionen für pvcreate.....	7
Tabelle 3: Attribute physikalischer Volumen.....	8
Tabelle 4: Werkzeugoptionen für vgcreate.....	9
Tabelle 5: Attribute von Volumengruppen.....	11
Tabelle 6: Werkzeugoptionen von vgremove.....	12
Tabelle 7: Optionen von lvcreate.....	13
Tabelle 8: Attribute logischer Volumen.....	14
Tabelle 9: Optionen von lvremove.....	15
Tabelle 10: pvs - Attribute physikalischer Volumen.....	17
Tabelle 11: lvs - Attribute logischer Volumen.....	19
Tabelle 12: Optionen für "vgextend".....	20
Tabelle 13: ausgewählte Optionen von pvmove.....	23
Tabelle 14: Optionen von vgreduce.....	24

Literaturverzeichnis

[1][2][3][4]

- 1: Carsten Wiese , Volumenkontrolle , 1.10.2006, <http://www.linux-magazin.de/Online-Artikel/Volumenkontrolle>
- 2: , LVM2 Manpage, , <http://linux.die.net/man/8/lvm>
- 3: , LVM2 Resource Page, , <http://sourceware.org/lvm2/>
- 4: AJ Lewis, LVM HOWTO, , <http://tldp.org/HOWTO/LVM-HOWTO/index.html>