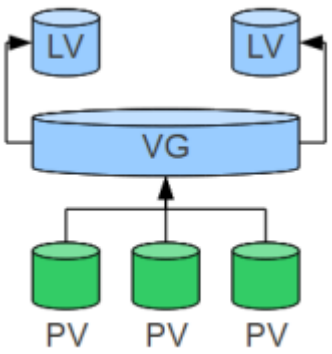
LVM Grundlagen

Der Logical Volume Manager (LVM) ermöglicht eine abstrahierte und flexible Verwaltung von Datenspeichern eines Computer-Systems. Gegenüber dem traditionellem Weg über Partitionen wird eine weitere, höhere Abstraktionsschicht eingeführt, die eine einfache und effiziente Verwaltung und Konfiguration der Datenträger ermöglichen soll. So können zum Beispiel mehrere Festplatten zu einem Logical Volume zusammengefasst und als ein gesamter Datenspeicher verwendet werden. Eine weitere Vereinfachung bietet die Möglichkeit der Gruppierung von Logical Volumes in Volume Groups, denen auch Bezeichnungen gegeben werden können. Durch diese Einordnungen und Namensgebungen ensteht für den Administrator auch ein logischer Bezug zu den sich in den Logical Volumes befindlichen Daten.

Dieser Artikel zeigt wie LVM unter Linux implementiert ist.

Physical Volumes (PVs)

Ein PV wird immer durch ein Block-Device repräsentiert - z.B. eine Festplatte (Device) oder eine Partition. Wird eine Partition bzw. ein Device als PV initialistiert, so wird das Block-Device mit einem LVM-Label gekennzeichent und mit Metadaten versehen. Das LVM-Label identifiziert das Block-Device als PV, es beinhaltet den Random Unique Identifier (UUID), es speichert die Größe des Devices in Bytes und es zeichnet auf, wo sich die Metadaten befinden.



Die Metadaten werden für die genaue Beschreibung der Konfiguration der Volume Groups benötigt. Standardmäßig wird eine identische Kopie der Metadaten auf den PVs einer VG gespeichert. Beim Anlegen einer PV ist es möglich zu definieren, dass 0,1 oder 2 Kopien von Metadaten gespeichert werden - ist diese Zahl einmal definiert, kann sie nicht mehr geändert werden. Die mehrfache Speicherung der Metadaten soll vor einer irrtümlichen Überschreibung schützen. Auch die Größe des Metadatenbereichs kann beim Anlegen eines PVs geändert werden. Unter Umständen kann dieser Bereich nämlich zu klein werden, was zu Problemen führen kann, da die Größe nachträglich nicht mehr geändert werden kann: LVM VG vgname metadata too large for circular buffer beheben.

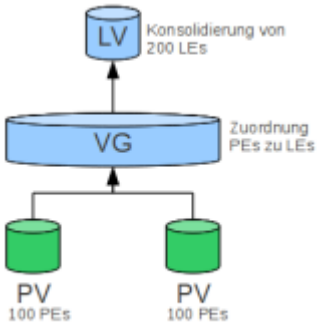
Ein PV wird bei der Erstellung in lauter kleine, gleich große Stückchen unterteilt, den sogenannten Physical Extents (PEs). Diese PEs sind die kleinst mögliche Datenmenge, die allokiert werden kann und beträgt standardmäßig 4 MiB, kann aber beim anlegen der darauf aufsetzenden Volume Group geändert werden. Seit dem lvm2-Format ist die Anzahl der PEs nicht mehr beschränkt, laut Man-Page von vgcreate kann eine hohe Anzahl an PEs die bereit gestellten Tools verlangsamen. Die I/O-Performance wird aber nicht beeinträchtigt.

Volume Groups (VGs)

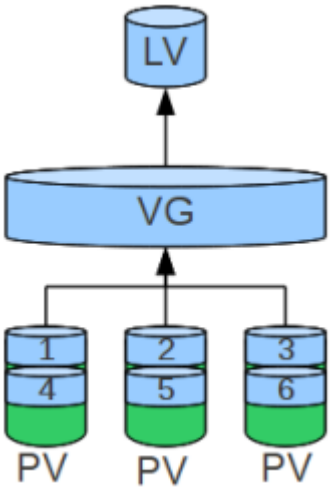
Eine oder mehrerere PVs können zu einer oder mehreren VGs zusammengefasst werden. Diese VG stellt einen Pool an Datenspeicher dar, aus dem später Logical Volumes allokiert werden können. Der verfügbare Speicher einer VG wird dabei in kleine, gleich groß bleibende Teile unterteilt - den Logical Extents. Diese stellen das Pendant zu den PEs auf PV-Ebene dar. Die Aufgabe der VG ist es, die **Logical Extents** den Physical Extents richtig zuzuordnen. Durch diese Gruppierung in einer Zwischenschicht können auch PEs von unterschiedlichen Devices in einer VG zusammengefasst werden.

Logical Volumes (LVs)

VGs werden in eine oder mehrere LVs aufgeteilt, auf denen anschließend die Dateisysteme aufsetzen. Von LVs können drei verschiedene Arten angelegt werden:



- Linear Volumes: Eine geordnete Zusammenfassung von PEs zu einem LV. Die Größe der darunter liegenden PVs muss nicht gleich sein, sie werden konsolidiert an die LVs weiter gegeben. Des weiteren muss ein LV nicht zwangsweise aus einer VG bestehen. Es können genauso mehrere LV aus einer VG heraus kreiert werden, solange die VG noch Speicherplatz zur Verfügung stellen kann. Die Größen der erzeugten LV können dabei auch unterschiedlich sein, haben minimal aber die Größe eines PE.



- Striped Volumes: Wie bei RAID 0 werden die Daten des LVs abwechselnd auf die die PVs aufgeteilt. Dies kann unter Umständen die Performance erhöhen, nicht aber die Ausfallssicherheit. Aufbau eines striped LVs Die Abbildung „Aufbau eines striped LV“ zeigt, wie die Stripes 1 bis 6 sequentiell auf die darunterliegenden PVs geschrieben werden. Dabei kann es sein, dass einzelne I/O-Operationen parallel getätigt werden können und somit der Schreibvorgang schneller erledigt wird.
- Mirrored Volumes: Ähnlich zu RAID 1 werden bei Mirrored Volumes die Daten auf die darunter liegenden Devices gespiegelt. Es können dabei auch mehrere Spiegelungen auf verschiedene, sogenannte „Legs“ aufgeteilt werden. Das zu kopierende Device wird dabei in sogenannte „Regions“ unterteilt, die bei Änderungen synchronisiert werden. Die Größe dieser Regions ist standardmäßig 512KB, kann aber auch verändert werden.

LVM und Device Mapper

Seit dem Linux Kernel 2.6 ist LVM mithilfe des Device Mappers implementiert. LVM verwendet dabei das linear Device Mapper Target.

Vorteile bei der Verwendung von LVMs