BERUFSAKADEMIE | MANNHEIM
university of cooperative education | staatliche studienakademie

AUSARBEITUNGEN ZUM THEMA A6:

LVM

VON

TIMO BÖLLINGER

UND

DOMINIC ECKART

DOZENT:

PROF. TISCHHHAUSER

MANNHEIM 2004

INHALTSVERZEICHNIS

1. LOGICAL VOLUME MANAGEMENT – EINFÜHRUNG

LVM ist eine Zwischenschicht zwischen dem Betriebssystem und der Hardware, es wird nicht mehr das Volumen als physikalische Instanz angesprochen sondern es wird ein virtuelles Volumen adressiert. Hinter diesem virtuellen Volumen steht eine Logik, die das virtuelle Volumen auf reale Volumen abbildet.
Diese Zwischensicht ermöglicht es die Abhängigkeit zwischen Hardwareseite und Speicherverwaltung zu trennen.
Bei einem Desktop System ohne ein LVM gibt es physikalische Laufwerke, diese enthalten Partitionen, auf diesen gibt es wiederum ein Dateisystem. Bei Datenbankservern z.B. Oracle kann es sein, dass auf der Partition kein Dateisystem existiert. Diese Partitionen werden und können nicht vom OS verwaltet werden sondern werden direkt von Oracle verwaltet. Diesen Typ von Partitionen nennt man raw.

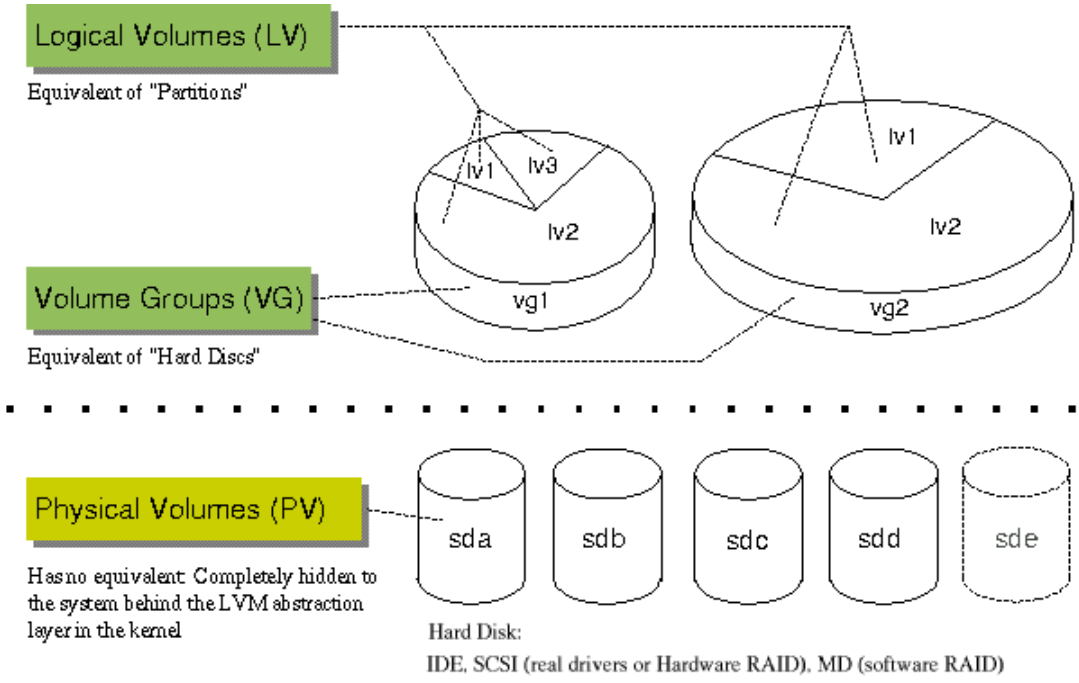

Abbildung 1; Quelle: www.suse.de

Ein LVM fast physikalische Objekt (Festplatten aber auch RAID-Systeme) nicht mehr als unzertrennliche Einheit auf, sondern erzeugt aus ihnen einen Speicherpool (Volume Groups).

- Die Volume Group entspricht aus Sicht es OS der Festplatte.
- Die Partitionen entsprechen den Logical Volumes.

1.1. WAS KANN LVM?

- **Unabhängigkeit von der Hardware**: Das System ist unabhängig von der Hardware die unter dem LVM steckt, so können z.B. Festplatten getauscht werden während Daten gelesen werden, dies geschieht durch die logische Zwischenschicht.
- **Daten verschieben**: Durch das LVM ist es möglich im laufendem Betrieb dannen von einem physikalischen Medium auf ein anderes zu verschieben ohne, dass das System etwas davon merkt.
- **Speicherabbilder**: LVM bietet die Möglichkeit ein schreibgeschütztes Abbild der LV zu machen und dieses dem System unter einem Alias zur Verfügung zu stellen. Dies kann z.B. zur Datensicherung dienen.

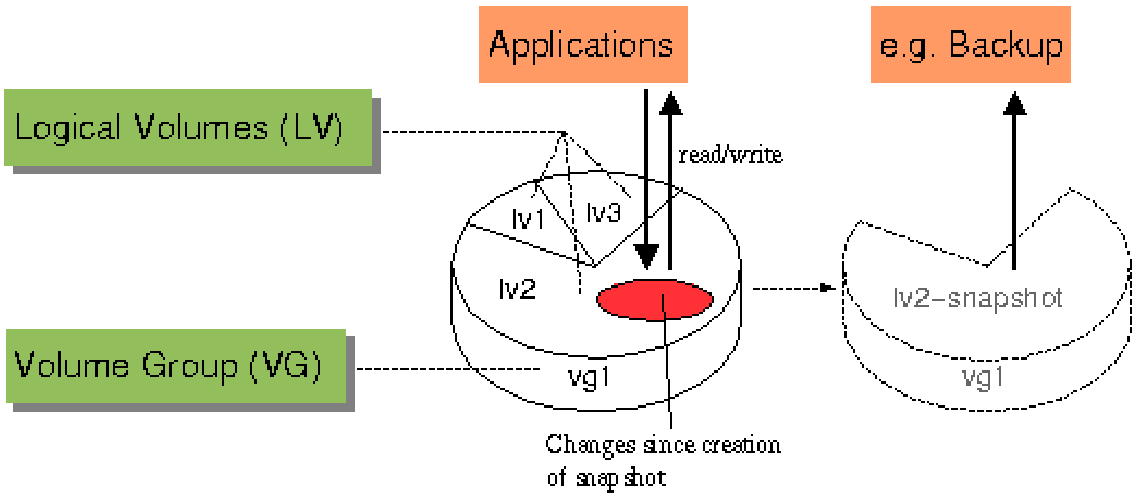

Abbildung 2; Quelle: www.suse.de

- **Größe ändern**: Durch die logische Zwischensicht ist es auch kein Problem ein Partition (LV) im laufenden Betrieb neu zu dimensionieren, dies hängt aber vom Dateisystem ab. Das Dateisystem muss eine dynamische Größenänderung im Betrieb zulassen, also sein Dateisystem-Tabellen entsprechend anpassen können. Falls dies nicht möglich ist muss die Partition aus- und eingehängt (Unmount und Mount) werden.
- **Volume Groups**: Ziel ist es mehrere physikalische Festplatten als ein Speicherobjekt zu verwenden. Es werden beide Platten wie eine Große angesprochen, dazu gibt es zwei Möglichkeiten dies zu tun:

- **Concatenation (Verkettung)**: Die Daten werden zuerst auf die eine Platte geschrieben, wenn diese voll ist wird einfach auf die zweite Platte weitergeschrieben.
- **Striping (streifend):** Die Daten werden abwechselnd auf die eine oder die andre Platte geschrieben. Vergleiche dazu RAID, ein Vorteil zu Raid ist die Unabhängigkeit vom Standort der Platten im System dazu weiter unten mehr.

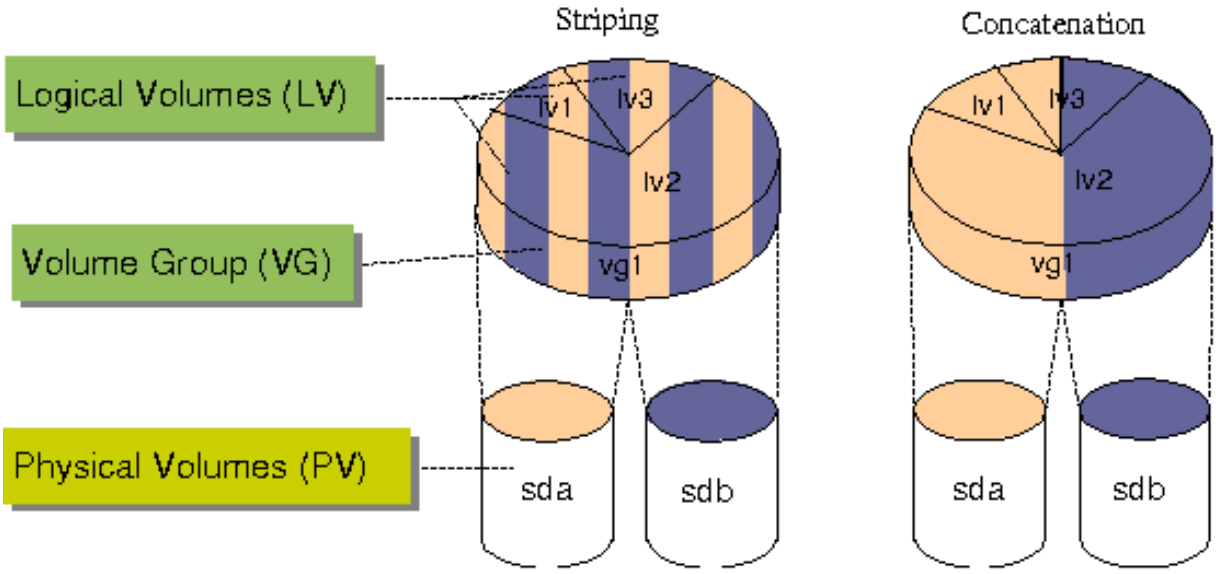

Abbildung 3; Quelle: www.suse.de

- **Unabhängigkeit vom Standort der Platten im System:** Bei einem System ohne LVM wird die Reihenfolge der Platten beim Booten durch den Standort im System generiert, z.B. an welchem IDE-Kanal und ob sie master oder Slave ist. Wenn man die Platten vertauscht ändern sie somit auch Ihre Bezeichnung im System (z.B. hda, hbb). LVM bezeichnet seine Platten nicht durch Bezeichner wie hda, die beim Booten vergeben werden, sondern hat einen speziellen Datenblock (Konfigurationsblock) auf jeder Festplatte. Dies ermöglicht eine genau Zuordnung der einzelnen Platten zu den Volume Groups usw. .

1.2. WIE FUNKTIONIERT LVM?

Die Festplatte oder das physikalische Medium wir als PV (physical Volume) bezeichnet, es ist in PEs (Physical Extents) aufgeteilt, diese sind mit Blöcken auf einer Festplatte zu vergleichen. Auf einem PV sind alle PEs gleich groß, die Große ist wählbar. Jede PE hat eine eindeutige ID.

Die Verwaltungsdaten des LV werden im VGDA (Volume Group Descriptor Area) gespeichert, dieser enthält Daten wie:

- die Nummer der PV in der VG
- die Größe der PEs
- welcher VG gehört das PV an
- welche PEs dieser PV gehören welcher LV an

Eine VG darf:

- aus maximal 255 PVs bestehen
- ein bis 255 LVs haben, diese dürfen auf beliebigen PVs der VGs gespeichert sein

Beim Systemstart sucht der Treiber des LVM auf den Platten nach den VGDA, dann werden die VGs und LVs aktiviert. Die Informationen der VGDA werden benötigt um zu wissen wie die Daten auf die PVs gemappt werden müssen.

Das LVs wiederum wird in LEs unterteilt, deren Größe entspricht der Größe der PEs auf der entsprechenden PV. Die Les sind auch indiziert. Jede LE entspricht genau einer PE auf der PV, dies entspricht einem 1:1 Mapping. Dabei sind die PE-Indizes nicht global eindeutig sondern nur für eine PV. Daher ums die PE in irgend einer weise in die Beziehung zur PV gebracht werden.
Wenn jetzt nun ein LE adressiert wird, erfolgt das Adressmapping über die eindeutige ID des PVs im VG und der eindeutigen ID der PE in der PV. Dazu existieren Mapping-Tabellen.

Beispiel: Anhand Abbildung 4 (Quelle: www.suse.de)
Eine Anwendung greift auf das LV lv3, Byte 254.123 zu;
dementsprechend ist das LE #62 (62*4MB=253.952, #63 wäre das nächste LE).
Laut Mapping-Tabelle sind die LEs 0 -~500 des LV lv3 auf dem PV1 gespeichert.
Dabei setzt sich die Nummer des PEs aus der Anzahl der PEs von lv1 auf PV1 und der Anzahl der PEs von lv2 auf PV1 plus 62 zusammen.

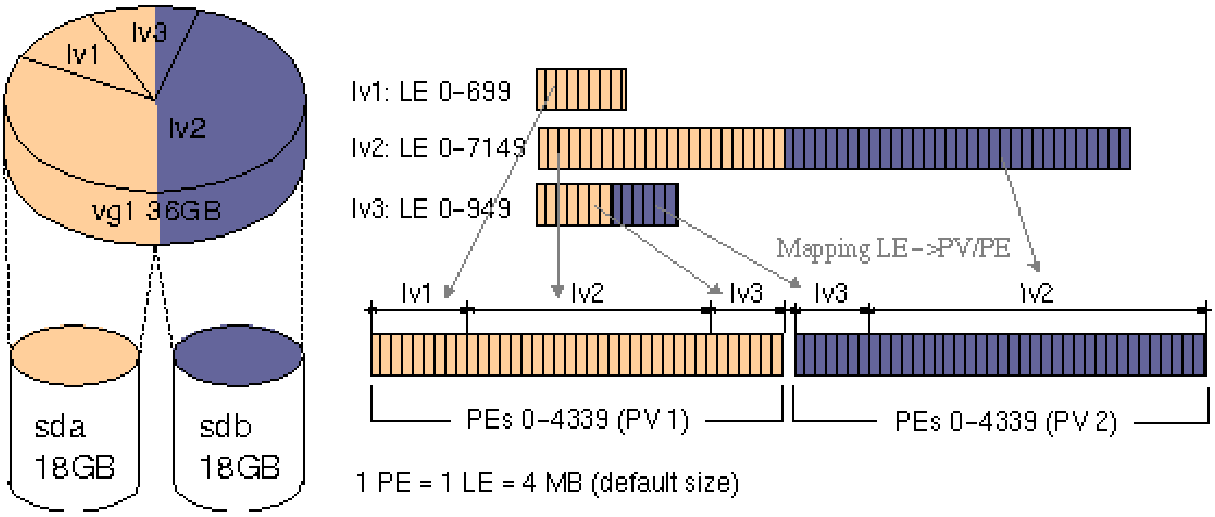

Abbildung 5; Quelle: www.suse.de

1.3. RAID

Der Begriff RAID steht ursprünglich für „Redundant Array of Independent Disks“ seit die Festplattenpreise immer stärker gesunken sind hat sich auch der Begriff „Redundant Array of Inexpensive Disks“ eingebürgert. RAID ist eine Technik, die es ermöglich ein Verbund von Festplatten als ein Speichermedium anzusprechen. Dabei beschränk sich RAID nicht darauf lediglich größere Speichermedien dem System zu einem relativ günstigen Preis zur Verfügung zu stellen. Sondern schützt je nach RAID-Level auch vor Festplattenfehlern im Gegensatz zu einzelnen Festplatten.

1.3.1. LINEAR (APPEND) MODE

Der Pool von Festplatten wird als ein großes Speichermedium verwendet und linear (erst die 1.Platte bis sie voll ist, dann die 2.Platte usw.) beschrieben. Dieses Verfahren bietet keinen Geschwindigkeitsvorteil und gewährleistet keine Datensicherheit beim Ausfall einer Festplatte.

1.3.2. RAID-0 (STRIPING) MODE

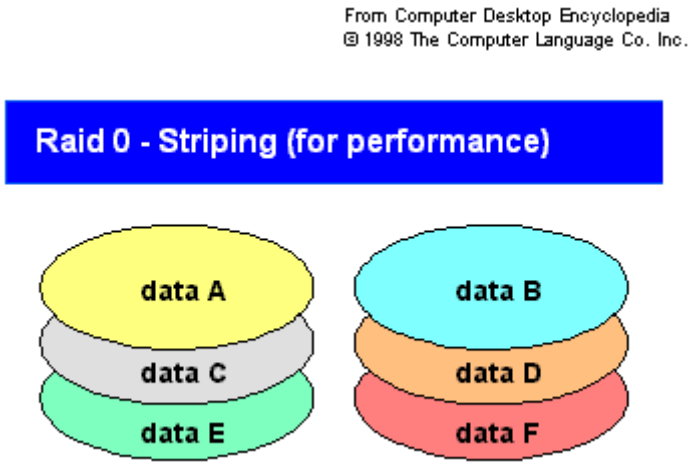

Abbildung 6; Quelle: http://www.personal.psu.edu/

Im Unterschied zum Linear (Append) Mode werden die Daten nicht linear sondern parallel geschrieben. Dadurch wird ein deutlicher Zuwachs der Datenrate erzielt. Dieses Verfahren gewährleistet durch die Verteilung der Daten keine Datensicherheit. Ein Festplattenausfall ist hier besonders schwerwiegend da die Daten nicht linear über die Festplatten verteilt sind.

1.3.3. RAID-1 (MIRRORING) MODE

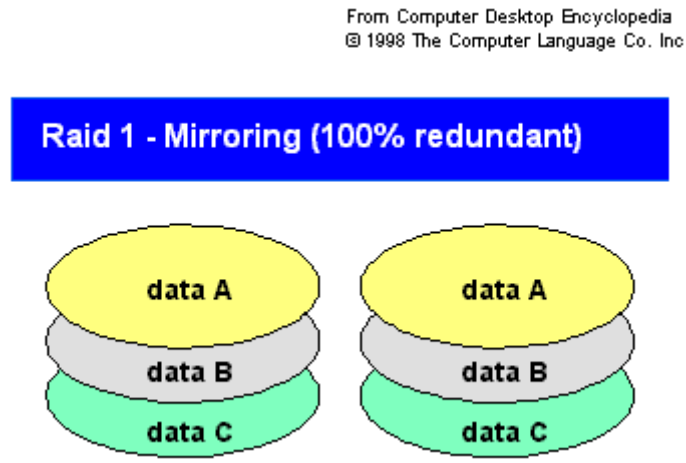

Abbildung 7; Quelle: http://www.personal.psu.edu/

Die Daten werden parallel auf zwei gleich große Festplatten geschrieben. Die Redundanz fordert doppelt so viel Speicherplatz wie für die Daten benötigt würde. Große Ausfallsicherheit wird erzielt aber kein Geschwindigkeitsvorteil.

1.3.4. RAID-5 (STRIPING & DISTRIBUTED PARITY) MODE

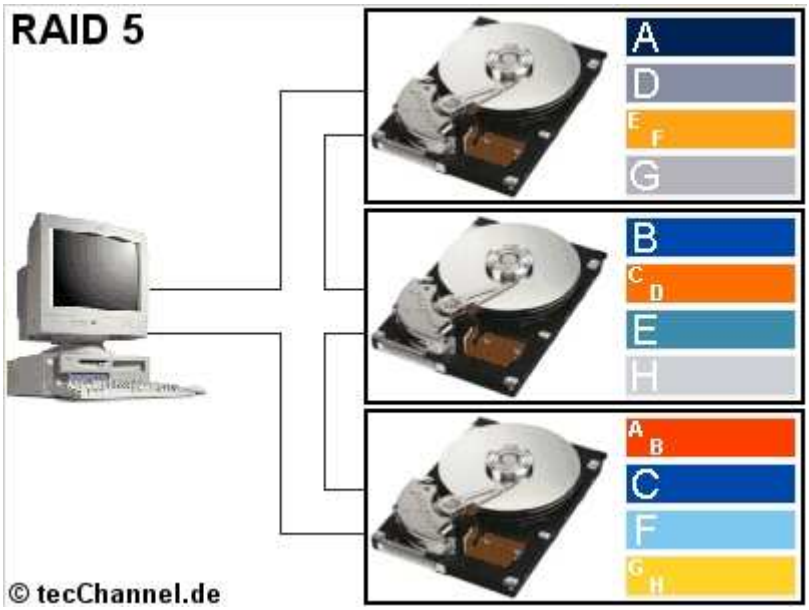

Abbildung 8; Quelle: www.tecchannel.de

Paritätsinformationen werden mit den Nutzdaten gemeinsam auf die Platten verteilt. Auf zwei Platten liegen die Blöcke mit Nutzdaten, auf einer die berechnete Parität der beiden Blöcke und dies rotiert. Der resultierende Kapazitätsverlust ist wesentlich geringer als bei RAID 1.Das schreiben dauert durch die Paritätsberechung länger, dafür steigt die Lesegeschwindigkeit im Gegensatz zu einer einzelnen Festplatte.

1.4. LVM VERSUS RAID

LVM kann ein RAID-System „nicht“ ersetzen, dies liegt schon daran, dass RAID und LVM auf verschiedenen Ebenen arbeiten. RAID auf Hardware-Ebene und LVM auf der Software-Ebene.

Ein RAID-System ist meistens an eine bestimmte Hardware gebunden und somit herstellerabhängig. Die Anzahl der Festplatten die ein RAID-Kontroller verwalten kann ist beschränkter als bei LVM, welches bis zu 255 Platten verwalten kann.

Die Größenänderung eines RAID-Systems ist nur eingeschränkt möglich, bei LVM kann ein Speichermedium an einem beliebigen Port zu einer VG hinzugefügt werden. So bietet sich sogar die Möglichkeit ein volles RAID-System (PV) um eine Platte aus einem anderen RAID-System (PV) zu erweitern, indem man die Platte in die bestehende VG einbindet.

Die meisten Änderungen können bei LVM im Gegensatz zu RAID sogar im laufenden Betrieb geändert werden.

1.5. AUFWANDSBETRACHTUNG FÜR EIN LVM

Bei jedem Zugriff auf die Festplatte muss die Adresse, die das OS liefert auf die reelle Position auf einer der Platten des LVM gemappt werden. Dazu dient wie schon erwähnt eine Tabelle, die Umsetzung der Adressen geschieht im Vergleich zu einem I/O Zugriff auf die Platte sehr schnell und ist deshalb vernachlässigbar.
Die Tabellen für die Unsetzung (enthalten in der VGDA) stehen auf den einzelnen PVs, dieser Datenbereich ist sehr klein ca. 128k. Diese Datenmenge kann meist im Cache der CPU gehalten werden, dazu kommt, dass die Tabellen im Normalfall Read-Only sind. Sie werden nur bei Änderungen im LVM angepasst.

1.6. LVM 2

Es gibt drei wesentliche Änderungen von LVM 1 zu LVM 2.

- Der Device Mapper ermöglicht es ein neues Blockdevice auf ein bestehendes Device aufzusetzen.
- Das Metadatenformat wurde hinsichtlich Stabilität und Effizienz grundlegend überarbeitet.
- Durch die lvm.conf kann der Administrator das verhalten der Device beeinflussen.

LVM 2 ist abwärtskompatibel zu LVM 1.

2. ANDERE VOLUME MANAGEMENT SYTEME

Es bleibt zu erwähnen, dass es neben LVM weitere logische Volume Managementsysteme gibt. Bespielsweise EVMS, ein von IBM betreutes Open Source Projekt oder Vinum, welches im Rahmen OpenBSD entwickelt wird.
Grundlegend ist der Funktionsumfang gleich, lediglich Implementierungen und Namensgebungen unterscheiden sich.

3. GLOSSAR

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

4. QUELLENVERZEICHNIS

http://www.linuxhaven.de/dlhp/HOWTO/DE-LVM-HOWTO.html

http://www.linuxhaven.de/dlhp/HOWTO/DE-Software-RAID-HOWTO.html

http://www.suse.de/de/whitepapers/lvm/lvm1.html

http://www.suse.de/de/whitepapers/lvm/lvm2.html

Linux Magazin 3/2004, S. 54-57

http://www.personal.psu.edu/users/c/l/clw220/IST220pres/raid.htm