Logical Volume Manager

LVM2 unter Linux

Dirk Geschke

Linux User Group Erding

27. November 2013

Gliederung

1. Einleitung
2. Überblick
3. LVM2 unter Linux
4. Praxis
5. Fazit

Allgemeines

- ein Logical Volume Manager (LVM) abstrahiert Datenspeicher
- Zusammenfassung mehrere Festplatten oder RAID-Systeme zu einem logischen Laufwerk
- kann neu partitioniert werden.
- Online Vergrößerungen und bedingt auch Verkleinerungen sind möglich.
- Größte Schwierigkeit: Nomenklatur!

Nomenklatur

Physical Volume hierbei handelt es sich in der Regel um eine Festplatte, Festplattenpartition oder einfach ein RAID-System: **PV**

Physical Entity Teil eines **PV**, Gruppe von Blöcken, auch *Physical Partition* (**PP**) genannt. Andere irreführende Namen: *stripes* oder *chunks*: **PE**

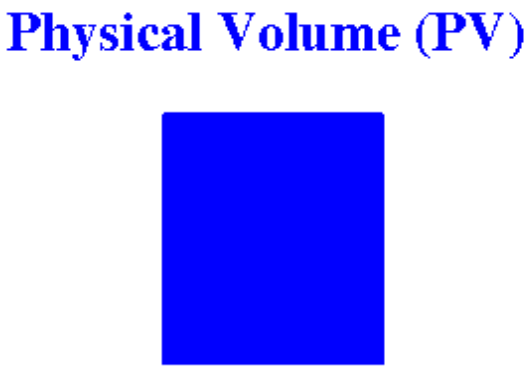
Volume Group Gruppierung von **PV**s zu einer logischen Einheit: **VG**

Logical Volume Teil einer **VG**, logische Partition: **LV**

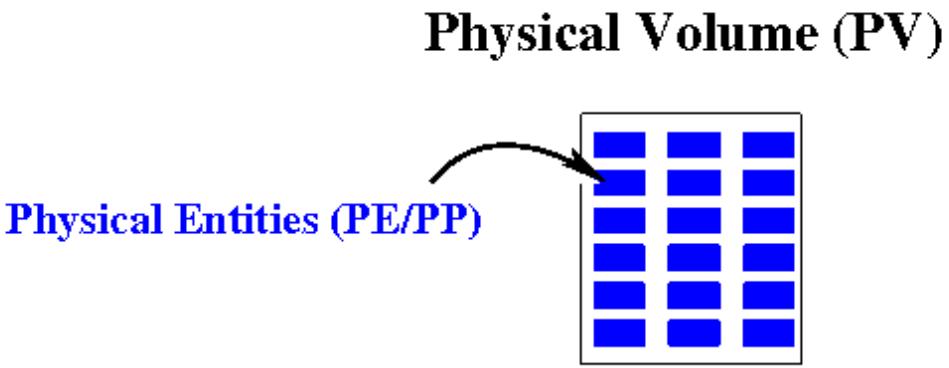
Logical Entity Teil einer **LV**, analogon zu **PE**: **LE** (**LP)**

Überblick

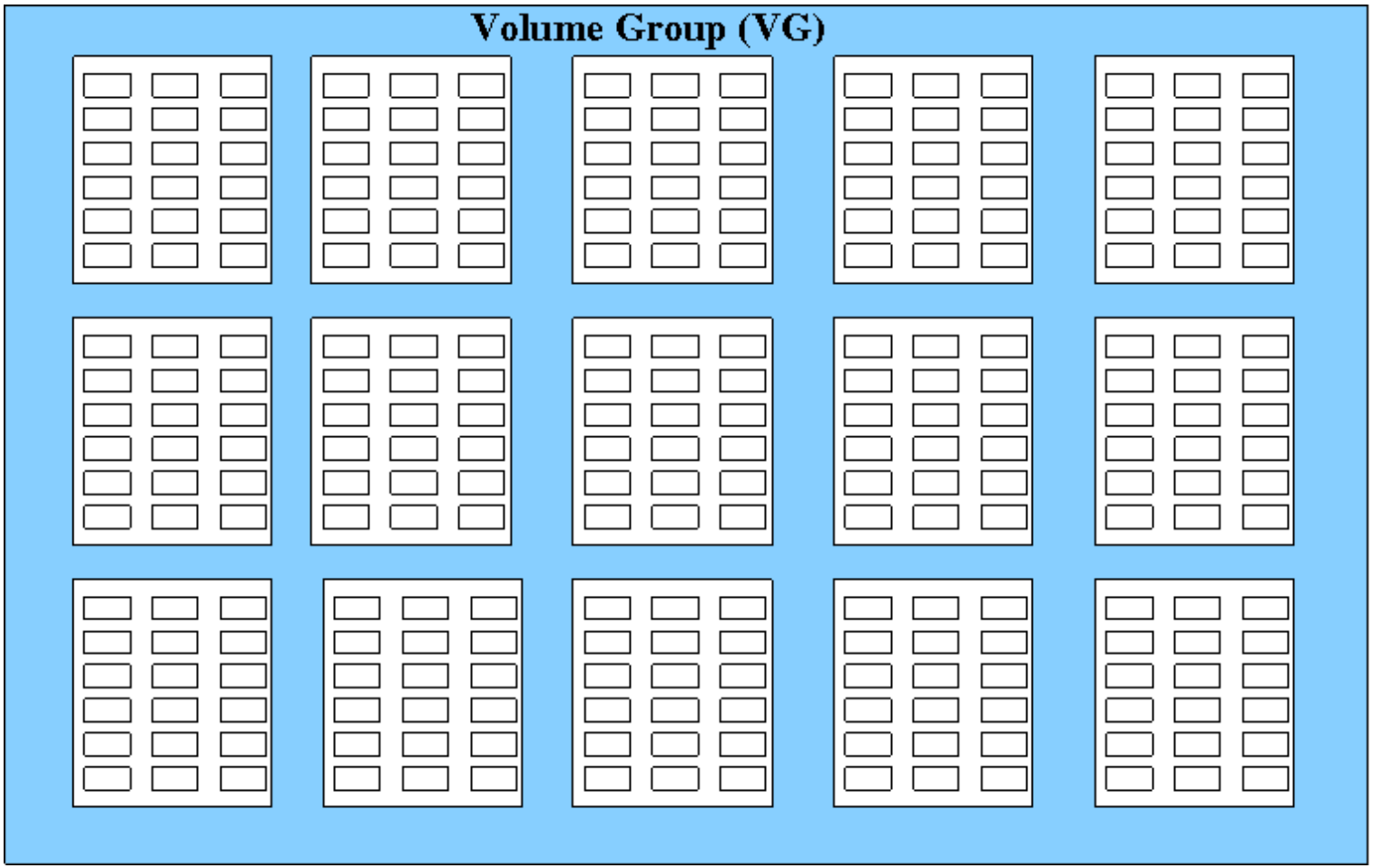
Physical Volume



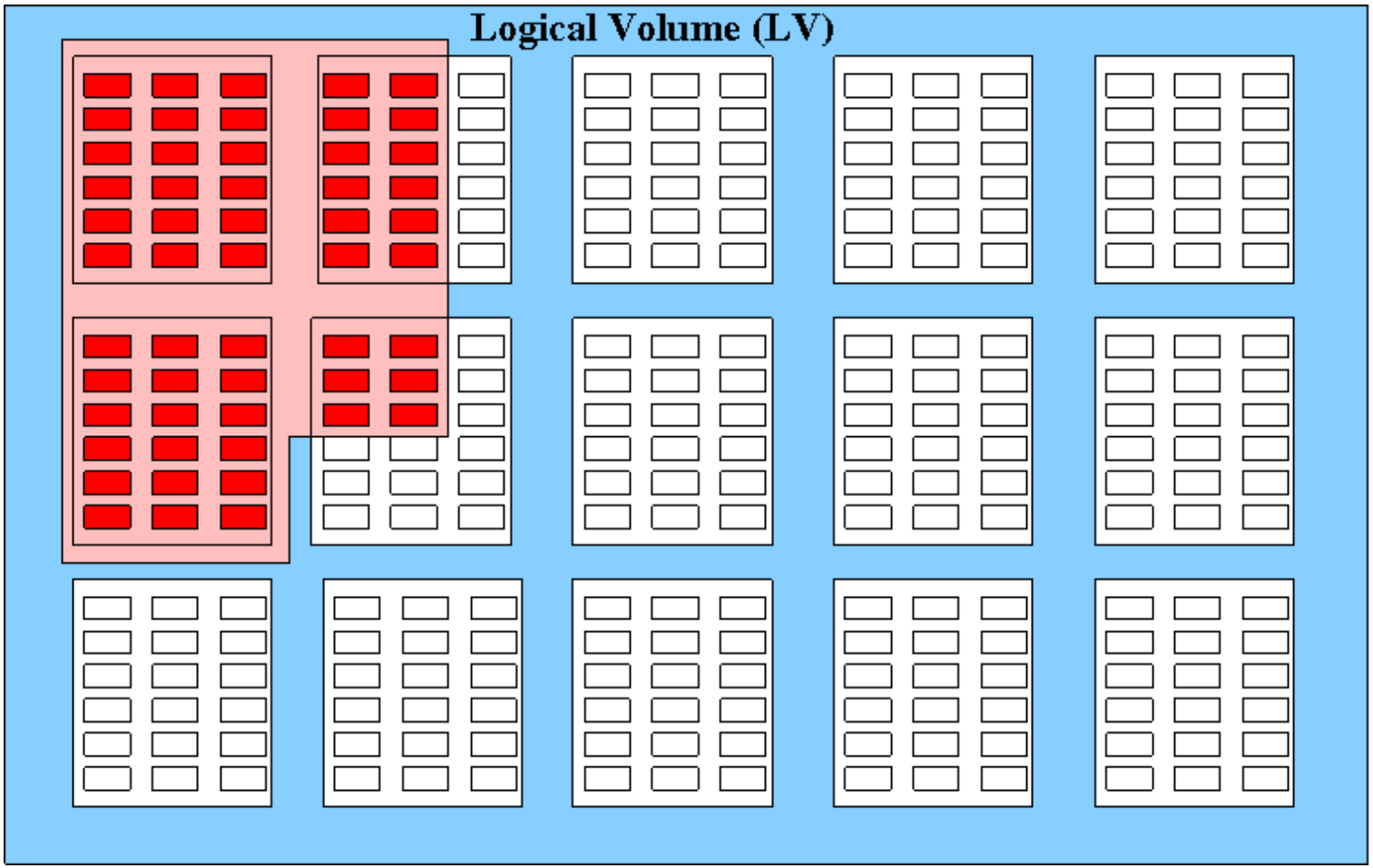
Physical Entity



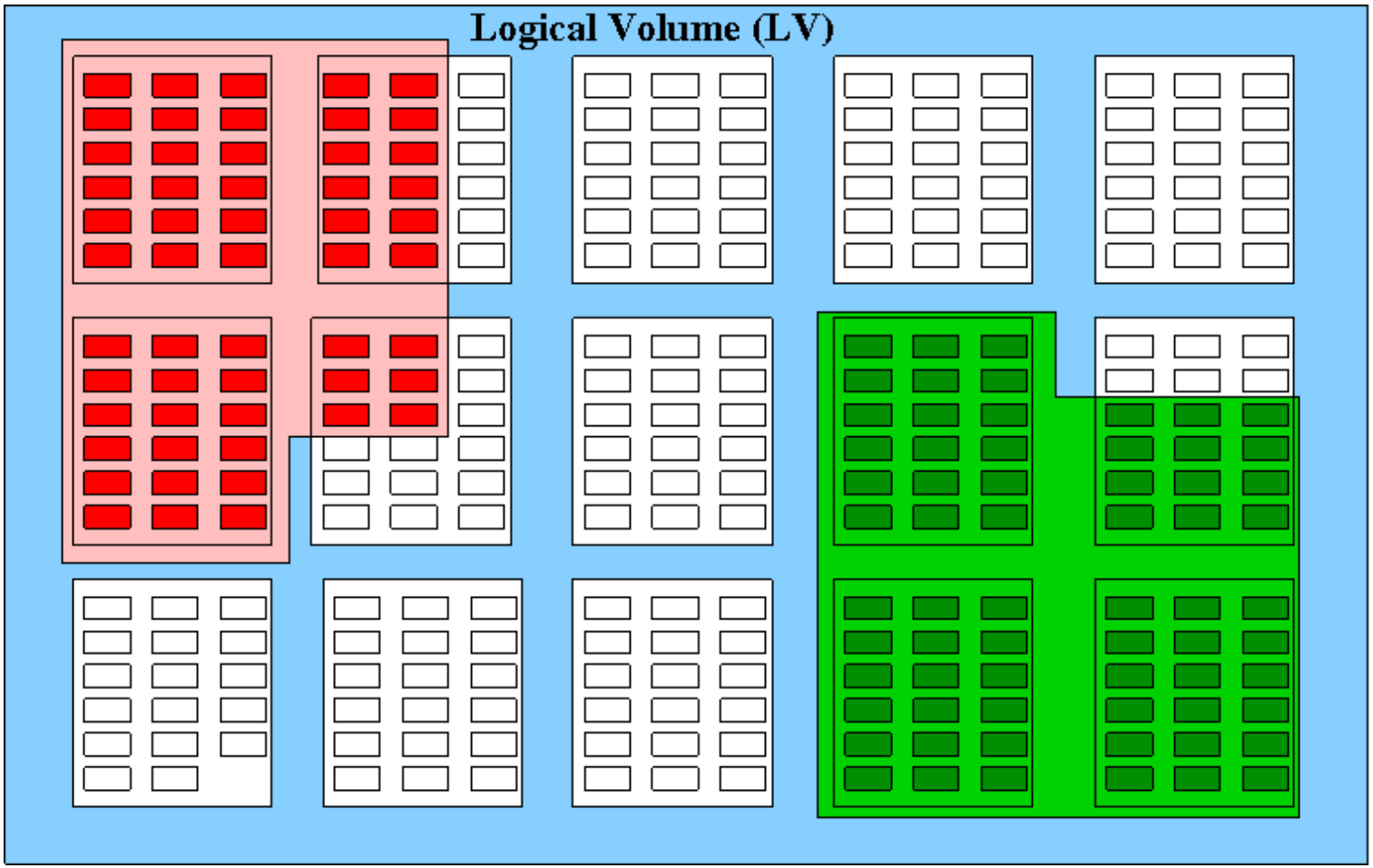
Volume Group



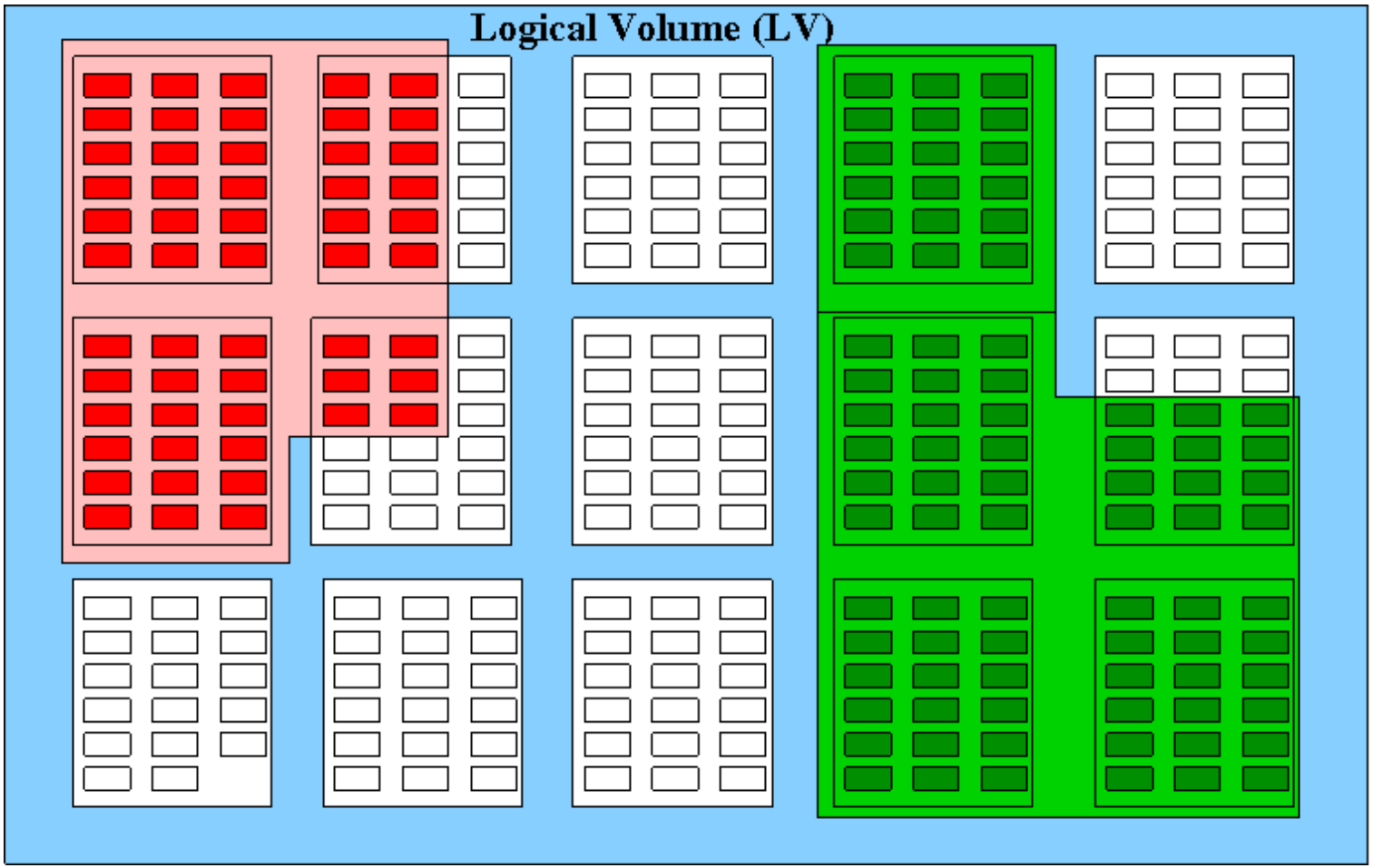
Logical Volume



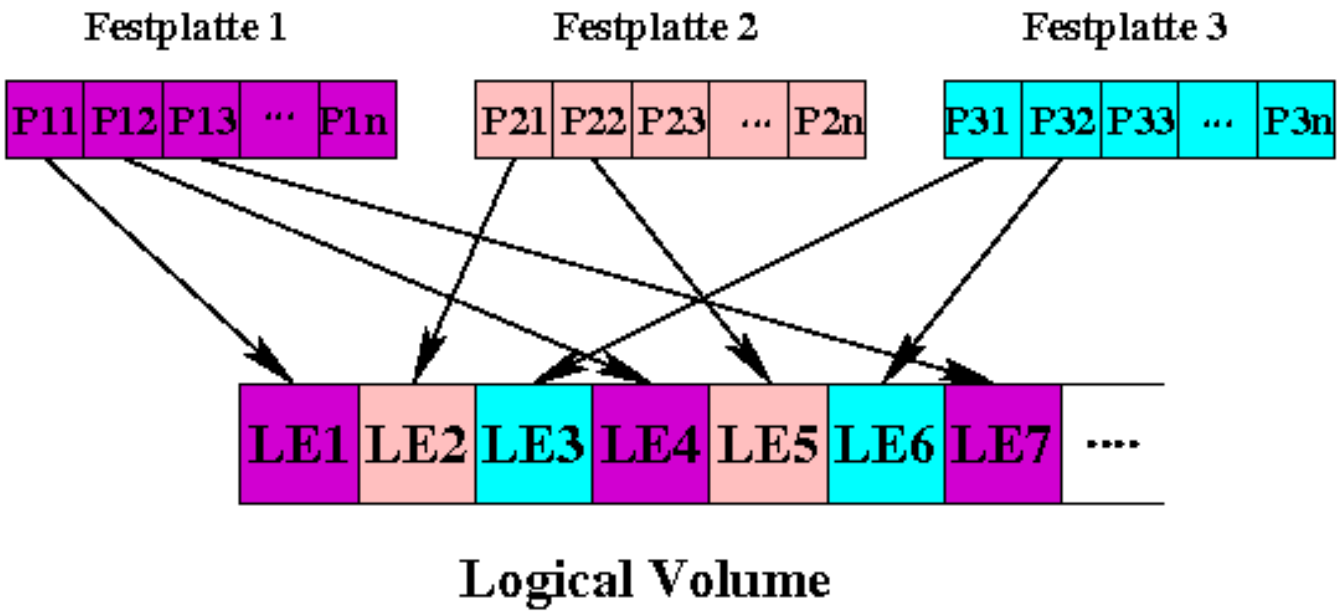
2. Logical Volume



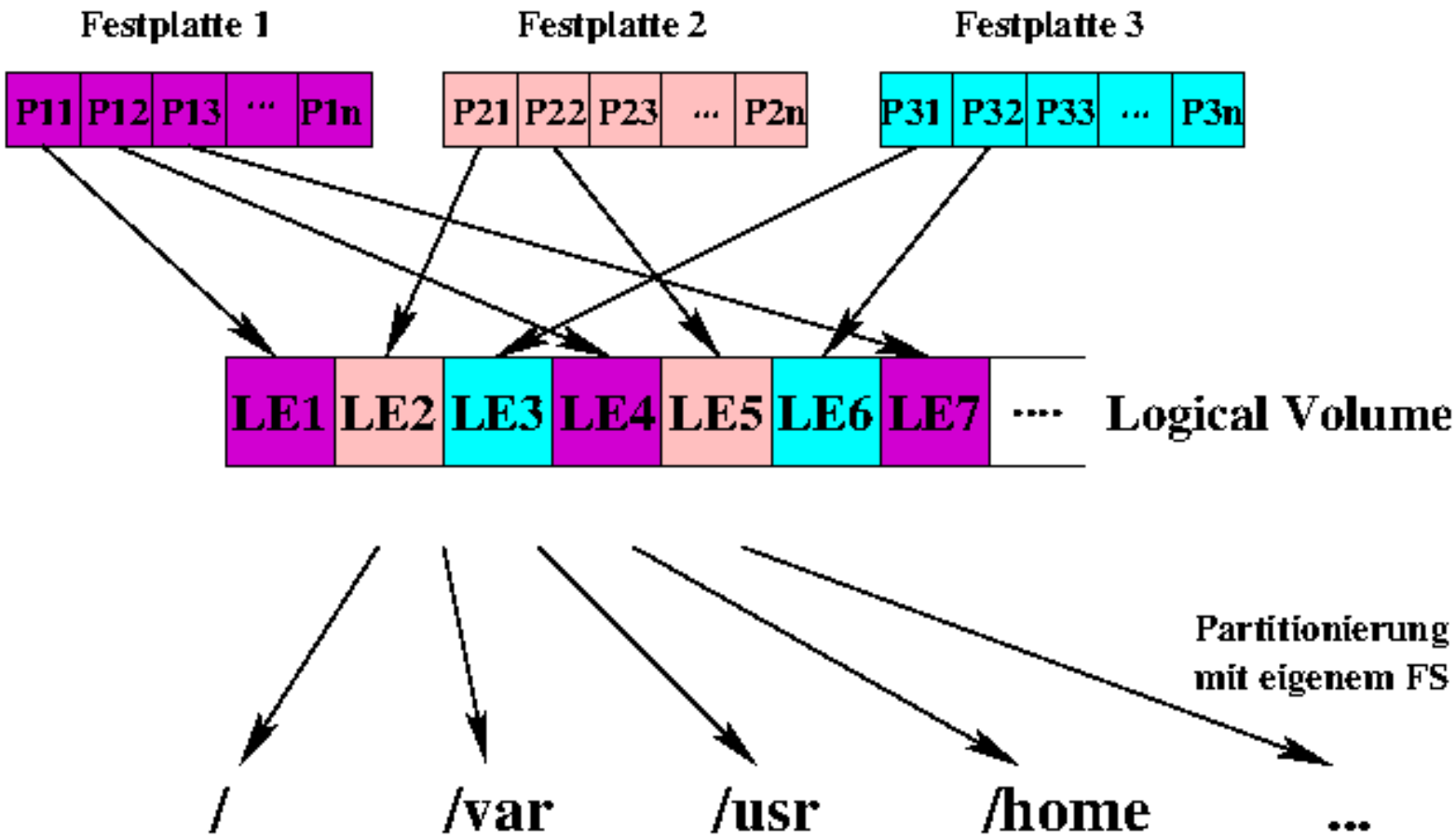
Erweiterung Logical Volume



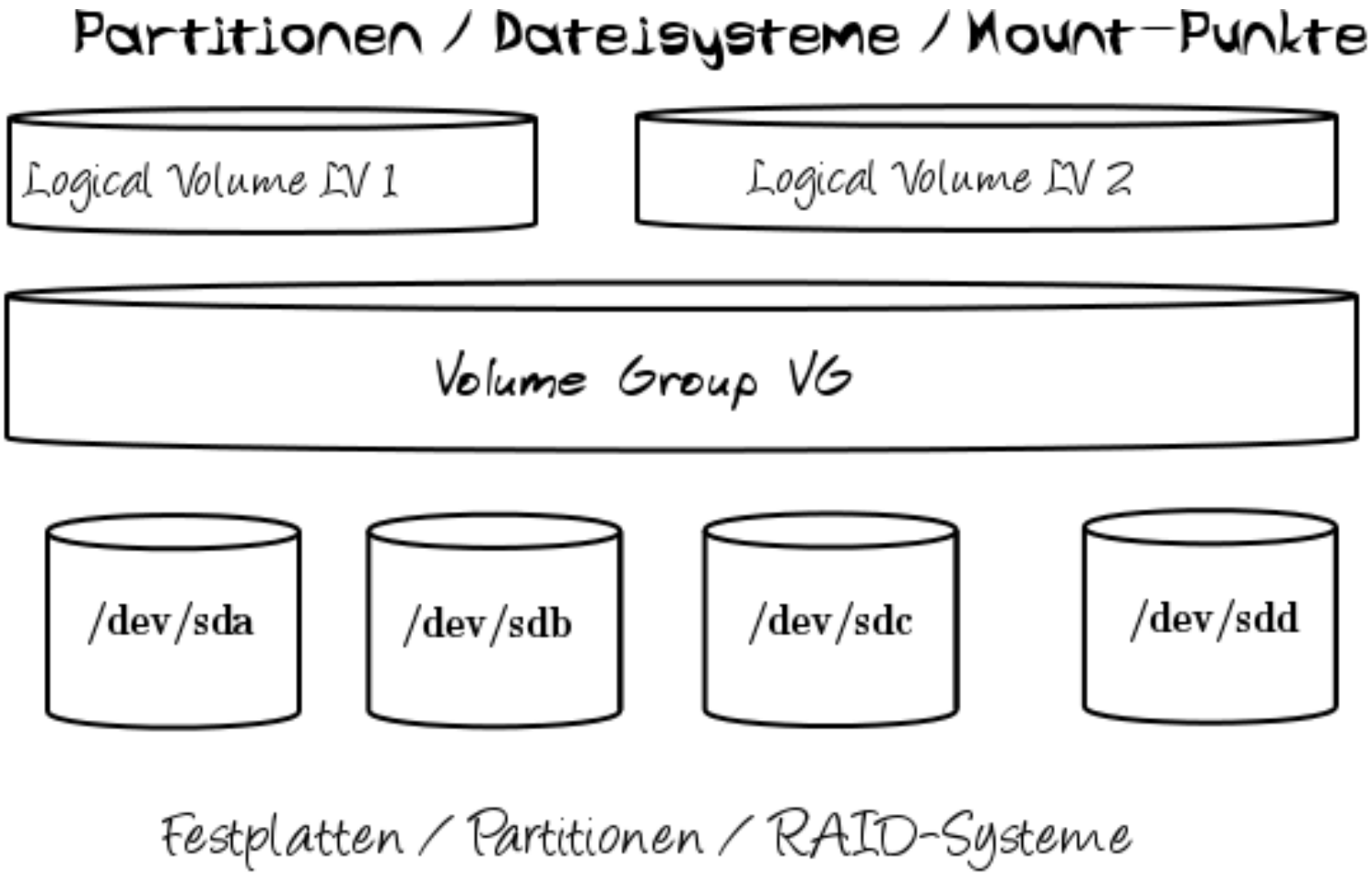
Zuordnung der Blöcke



Zuordnung der Blöcke



Alternative Sichtweise



LVM: Vor- und Nachteile

Vorteile:

+ unabhängig von Hardwarebegrenzungen
+ leicht erweiterbar

Nachteile:

- wird leicht unübersichtlich
- keine Ausfallsicherheit
- muss beim Booten berücksichtigt werden (**grub2, initramfs** oder separates /**boot**)

LVM2 unter Linux

Features

- vollständiger LVM mit vielen Userland-Tools
- Kernel-Part: DM - Device Mapper Support
- Striping–Möglichkeit bei mehreren PVs
- Snapshots sind möglich
- LVM-RAID Levels: 0, 1, 4, 5, 6, 10
- Thin Provisioning (*overcommit*)

Praxis

Anlegen eines Physical Volumes

```
# pvcreate /dev/sda /dev/sdb
Writing physical volume data to disk "/dev/sda"
Physical volume "/dev/sda" successfully created
Writing physical volume data to disk "/dev/sdb"
Physical volume "/dev/sdb" successfully created
```

Nach **PV**s kann gesucht werden:

```
# pvscan
PV /dev/sda lvm2 [10.00 GiB]
PV /dev/sdb lvm2 [10.00 GiB]
Total: 2 [20.00 GiB] / in use: 0 [0 ] / in no VG: 2 [2
GiB]
```

Ansehen einer PV

```
# pvdisplay /dev/sda
"/dev/sda" is a new physical volume of "10.00 GiB"
-- NEW Physical volume --
PV Name /dev/sda
VG Name
PV Size 10.00 GiB
Allocatable NO
PE Size 0
Total PE 0
Free PE 0
Allocated PE 0
PV UUID 3jvb0T-DS21-z5q9-RNMA-4mfA-6k4u-RGsHHI
```

Alternativ mit **pvs** möglich

Anlegen einer Volume Group

```
# vgcreate lug-vg /dev/sda /dev/sdb
Volume group "lug-vg" successfully created
```

Suchen von Volume Groups ist ebenfalls leicht

```
# vgscan
Reading all physical volumes. This may take a while...
Found volume group "lug-vg" using metadata type lvm2
```

Ansehen des PV erneut

```
# pvdisplay /dev/sda
-- Physical volume --
PV Name /dev/sda
VG Name lug-vg
PV Size 10.00 GiB / not usable 4.00 MiB
Allocatable yes
PE Size 4.00 MiB
Total PE 2559
Free PE 2559
Allocated PE 0
PV UUID 3jvb0T-DS21-z5q9-RNMA-4mfA-6k4u-RGsHHI
```

Ansehen der VG (gekürzt)

```
# vgdisplay lug-vg
-- Volume group --
VG Name lug-vg
Format lvm2
VG Access read/write
VG Status resizable
Cur PV 2
Act PV 2
VG Size 19.99 GiB
PE Size 4.00 MiB
Total PE 5118
Alloc PE / Size 0 / 0
Free PE / Size 5118 / 19.99 GiB
VG UUID KJpKaL-447W-qqmt-Arl3-Eq4a-iwKi-ylQ12X
```

Anlegen eines Logical Volumes

```
# lvcreate -n lug-lv -L1g lug-vg
Logical volume "lug-lv" created
```

Scannen ist ebenfalls wieder leicht:

```
# lvscan
ACTIVE '/dev/lug-vg/lug-lv' [1.00 GiB] inherit
```

Ansehen mit **lvs** ist ähnlich (gekürzt):

```
# lvs
LV     VG     Attr      LSize
lug-lv lug-vg -wi-a-- 1.00g
```

Ansehen eines Logical Volumes in einer VG

```
# lvdisplay /dev/lug-vg/lug-lv
-- Logical volume --
LV Path /dev/lug-vg/lug-lv
LV Name lug-lv
VG Name lug-vg
LV UUID rPtcAa-3VOm-A20e-H9Yp-z0yY-CKSJ-sTBgdm
LV Write Access read/write
LV Status available
LV Size 1.00 GiB
Current LE 256
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:0
```

Formatieren, einhängen, testen

```
# mkfs -t xfs /dev/lug-vg/lug-lv
meta-data=/dev/lug-vg/lug-lv isize=256 agcount=4, ...
```

Einhängen:

```
# mount /dev/lug-vg/lug-lv /mnt
SGI XFS with ACLs, security attributes, realtime, larg
block/inode numbers, no debug enabled
SGI XFS Quota Management subsystem
XFS (dm-0): Mounting Filesystem
XFS (dm-0): Ending clean mount
```

Testen:

```
# dd if=/dev/zero of=/mnt/dd bs=64M count=10
671088640 bytes (671 MB) copied, 108.845 s, 6.2 MB/s
```

Striping

Zerlegen:

```
# umount /mnt
# lvremove /dev/lug-vg/lug-lv
Do you really want to remove active logical volume lug
[y/n]: y
Logical volume "lug-lv" successfully removed
```

Erzeugen eines *striping* LV:

```
# lvcreate --stripes 2 -n lug-lv -L1g lug-vg
Using default stripesize 64.00 KiB
Logical volume "lug-lv" created
```

Formatieren, mounten und erneutes Testen:

```
# dd if=/dev/zero of=/mnt/dd bs=64M count=10
671088640 bytes (671 MB) copied, 54.9124 s, 12.2 MB/s
```

Vergrößern

```
# lvextend -L+1g /dev/lug-vg/lug-lv
Using stripesize of last segment 64.00 KiB
Extending logical volume lug-lv to 2.00 GiB
Logical volume lug-lv successfully resized
```

$\Longrightarrow$ nicht vergessen: Dateisystem muss auch vergrößert werden!

```
# xfs_growfs /mnt
...
data blocks changed from 262016 to 524288
```

$\Longrightarrow$ theoretisch ist auch verkleinern möglich, hängt aber vom Dateisystem ab!

Vergrößern der VG

Auch Volume Groups können vergrößert werden:

```
# pvcreate /dev/sdc /dev/sdd
Writing physical volume data to disk "/dev/sdc"
Physical volume "/dev/sdc" successfully created
Writing physical volume data to disk "/dev/sdd"
Physical volume "/dev/sdd" successfully created

# vgextend lug-vg /dev/sdc /dev/sdd
Volume group "lug-vg" successfully extended

# vgs
VG      #PV #LV #SN Attr   VSize  VFree
lug-vg    4   1   0 wz-n- 39.98g 37.98g
```

Snapshots

- Snapshots sind ein LV mit eigener Größe
- es werden nur die Deltas kopiert, also **C**opy–**o**n–**W**rite
- per default sind die Snaphots read-write
- können als Spielwiese verwendet werden
- Snaphosts können wieder zurückgespielt werden
- gut für Erstellung von Backups!
- volle Snapshot-LVs werden automatisch deaktiviert!

Snapshots

```
# lvcreate -L 1G -s -n lug-snap /dev/lug-vg/lug-lv
Logical volume "lug-snap" created
# ls -l /mnt
-rw-r--r- 1 root root 671088640 Nov 26 15:00 dd
# cp /boot/vmlinuz /mnt
# ls -l /mnt
tooal 658132
-rw-r--r- 1 root root 671088640 Nov 26 15:00 dd
-rw-r--r- 1 root root   2835648 Nov 26 18:33 vmlinuz
# mount -o nouuid /dev/lug-vg/lug-snap /mnt2
# ls -l /mnt2
total 655360
-rw-r--r- 1 root root 671088640 Nov 26 15:00 dd
```

Snapshots: zurück auf Los

```
# lvs
LV       VG     Attr      LSize Pool Origin Data%
lug-lv   lug-vg owi-aos- 2.00g
lug-snap lug-vg swi-aos- 1.00g      lug-lv   0.46
# umount /mnt /mnt2
# lvconvert --merge /dev/lug-vg/lug-snap
Merging of volume lug-snap started.
lug-lv: Merged: 0.4%
lug-lv: Merged: 0.0%
Merge of snapshot into logical volume lug-lv has finis
Logical volume "lug-snap" successfully removed
# mount /dev/lug-vg/lug-lv /mnt
# ls -l /mnt
total 655360
-rw-r-r- 1 root root 671088640 Nov 26 15:00 dd
```

LVM–RAID

Vorbereitungen: Altes LV löschen, VG mit 6 Festplatten anlegen:

```
# lvremove -f lug-vg/lug-lv
Logical volume "lug-lv" successfully removed
# pvcreate /dev/sde /dev/sdf
Writing physical volume data to disk "/dev/sde"
Physical volume "/dev/sde" successfully created
Writing physical volume data to disk "/dev/sdf"
Physical volume "/dev/sdf" successfully created
# vgextend lug-vg /dev/sde /dev/sdf
Volume group "lug-vg" successfully extended
# vgs
VG      #PV #LV #SN Attr   VSize  VFree
lug-vg    6   0   0 wz-n- 59.98g 59.98g
```

LVM–RAID

```
# lvcreate -i 4 -L 10g --type raid6 -n lug-lv lug-vg
Using default stripesize 64.00 KiB
Logical volume "lug-lv" created
```

Im **dmesg** gibt es Details:

```
[ 4918.757094] RAID conf printout:
[ 4918.757102] -- level:6 rd:6 wd:6
[ 4918.757107] disk 0, o:1, dev:dm-1
[ 4918.757110] disk 1, o:1, dev:dm-3
[ 4918.757113] disk 2, o:1, dev:dm-5
[ 4918.757116] disk 3, o:1, dev:dm-7
[ 4918.757119] disk 4, o:1, dev:dm-9
[ 4918.757122] disk 5, o:1, dev:dm-11
```

Testen des RAIDs

```
# echo 1 > /sys/block/sda/device/delete
# lvs
Couldn't find device with uuid 3jvb0T-DS21-z5q9-RNMA-4
LV     VG     Attr     LSize  Pool ...
lug-lv lug-vg rwi-aor- 10.00g
```

aber **dmesg** zeigt es erst bei Zugriff etwas:

```
md/raid:mdX: Disk failure on dm-1, disabling device.
md/raid:mdX: Operation continuing on 5 devices.
```

Rescan scsi devices:

```
# echo "- - -" > /sys/class/scsi_host/host0/scan
```

Testen des RAIDs

dmesg listet /dev/sda wieder als /dev/sdg

```
sd 0:0:0:0: Attached scsi generic sg0 type 0 ...
```

und das Physical Volume wird anhand der UUID wiedergefunden:

```
# pvscan
PV /dev/sdg VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
PV /dev/sdb VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
PV /dev/sdc VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
PV /dev/sdd VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
PV /dev/sde VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
PV /dev/sdf VG lug-vg lvm2 [10.00 GiB / 7.49 GiB free]
Total: 6 [59.98 GiB] / in use: 6 [59.98 GiB] / in no V
0 [0 ]
```

Testen des RAIDs

Allerdings:

```
# dmsetup status lug–vg-lug–lv
0 20971520 raid raid6_zr 6 DAAAAA 5242880/5242880
```

Hier bedeutet:

- A Alive und in-sync
- a Alive aber nicht in-sync
- D Dead, ausgefallen

Das passt noch nicht, aber so:

```
# lvconvert --repair /dev/lug-vg/lug-lv
Attempt to replace failed RAID images (requires full d
resync)? [y/n]: y
Faulty devices in lug-vg/lug-lv successfully replaced.
```

Thin Provisioning

- es wird ein Pool vom Typ **thin-pool** erstellt
- daraus können LVs mit Typ **thin** erstellt werden
- *overcommit*-Analogon für Logical Volumes
- Größe der LVs kann größer als der Pool sein
- Daten werden erst bereitgestellt, wenn sie benötigt werden
- unterstützt das Dateisystem **discard**, können die Blöcke wieder freigegeben werden

$\Longrightarrow$ Könnte für VServer (OpenVZ, LXC) interessant sein (ISP)

Fazit

Zum Abschluss . . .

- LVM ist eigentlich ein einfaches Konzept
- man muss neu nachdenken: PV, PE, VG, LV, LE
- viele Tools für zahlreiche Anpassungen
- hohe Komplexität dennoch vorhanden
- Vorteile der LVM2-Erweiterungen wie RAID und Thin Provisioning sind fraglich.
- Dennoch ein sehr nützliches Tool!

$\Longrightarrow$ **Frage: Ist es wirklich sinnvoll alles zu vereinen?**

Tja, und nun?

Wie geht es weiter?

- Dezember-Stammtisch? 4. Mittwoch ist Weihnachten...
- Vorträge? Ideen wären z.B.:
  - Linux–Container (LXC, OpenVZ)
  - `systemd`
  - `cgroups`
  - Spielen mit `strace, /proc, /sys, lsof, netstat, ...`
  - Linux–HA–Konzepte
  - Netzwerkmonitoring: Argus, Smokeping, ...
  - Arduino, Raspberry Pi, ...
  - ...
- Sonstiges?