Festplattenverwaltung

Vergrößern, verschlüsseln usw.

- Festplattenverwaltung - fdisk,lvm,dd
- Cryptsetup mit LUKS - Festplattenverschlüsselung
- Logical Volume Manager - LVM
- Software RAID - mdadm
- Beispiel - RAID1 - LUKS encryption - LVM
- hdparm - Energiemanagement

Festplattenverwaltung - fdisk,lvm,dd

Partitionstabelle kopieren

`dd if=/dev/sdc of=/dev/sdd bs=512 count=1`

Parameter erklärt:

if -> input device

of -> output device

bs -> blocksize

count -> Anzahl der Blöcke, die kopiert werden sollen

Festplatte vergrößern

.. mit LVM

Das Programm "fdisk" für die Festplatte starten:

```
/# fdisk /dev/sda


Welcome to fdisk (util-linux 2.33.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.



Command (m for help): p
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disklabel type: dos
Disk identifier: 0xb5d10c66

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1953791  1951744  953M 83 Linux
/dev/sda2       1953792 52428799 50475008 24,1G  5 Extended
/dev/sda5       1955840 52428799 50472960 24,1G 8e Linux LVM
```

Dann sieht man alle aktuellen Partitionen.

Ich habe mich beim Einrichten für folgendes Setup entschieden:

```
-- /dev/sda
 |- /dev/sda1 -> /boot
 |- /dev/sda2 -> LVM partition
  |-/dev/sda5 -> volumegroup (vg01)
   |-vg01 -> logical volume swap (lvswap)
   |-vg01 -> logical volume system (lvsys)
```

Zuerst muss man die Festplatte an sich vergrößern.

Da ich die Virtualisierungssoftware "Proxmox" einsetze ist das total einfach.

Entweder per CLI `qm resize VMID FestplattenID +5G` oder man nutzt die grafische Weboberfläche und fügt über den Punkt "Hardware" -> entsprechende Festplatte -> Knopf oben in der Leiste "resize disk" einfach die Größe hinzu.

Danach kann man sich die neue Größe mit `fdisk -l` anzeigen lassen/kontrollieren.

Nun muss man die bestehenden Partitionen mittels fdisk /dev/sdX (X steht für euren Festplatten Identifier Buchstabe) löschen und 1 primäre über die gesamte neue Größe und eine logische über die gesamte neue Größe erstellen.

Am Ende noch die Änderungen speichern mit dem Befehl "w" für "write" und dann noch mittels "partprobe" die neuen Festplatteninformationen einlesen lassen.

Dann kann man mit "vgdisplay" überprüfen ob es geklappt hat und es geht weiter mit dem "lvextend".

Da ich die gesamte Größe haben will kann ich einfach sagen er soll alles nehmen was frei ist:

`lvextend -l +100%FREE /dev/vg01/lvsys`

Und mit "lvdisplay" sehe ich nun meine neue Größe:

```
/# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg01/lvswap
  LV Name                lvswap
  VG Name                vg01
  LV UUID                04chsG-gbxM-RrZZ-1wLv-Nj48-tPgE-0GmDIv
  LV Write Access        read/write
  LV Creation host, time debian, 2020-02-17 17:03:48 +0100
  LV Status              available
  # open                 2
  LV Size                <1,86 GiB
  Current LE             476
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:1

  --- Logical volume ---
  LV Path                /dev/vg01/lvsys
  LV Name                lvsys
  VG Name                vg01
  LV UUID                nUBumD-WQ2c-wiMW-fXyS-tvoa-ELZB-rcUA6H
  LV Write Access        read/write
  LV Creation host, time debian, 2020-02-17 17:05:38 +0100
  LV Status              available
  # open                 1
  LV Size                <22,21 GiB
  Current LE             5685
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     256
  Block device           254:0
```

Cryptsetup mit LUKS - Festplattenverschlüsselung

Quelle: https://linuxwiki.de/cryptsetup

1. LUKS-Partition initialisieren

```
cryptsetup luksFormat -c aes-cbc-essiv:sha256 -s 256 /dev/sdc1
```

Ich habe hier z.b. ein RAID1 Device ausgewählt z.b. /dev/md0. Somit ist das gesamte RAID 1 verschlüsselt.

2. LUKS-Partition öffnen und mapping erstellen (/dev/MAPPING)

Öffnen mit Passwort

```
cryptsetup luksOpen /dev/sdc1 cr_crypto
```

LUKS Gerät löschen

Quelle: https://unix.stackexchange.com/a/606231

```
cryptsetup-reencrypt --decrypt
```

USB zum Entschlüsseln

Quelle: https://decatec.de/linux/verschluesselte-festplatte-luks-mit-usb-stick-bei-systemstart-entschluesseln/#USB-Stick_zum_Entschluesseln_einrichten

Hat nur leider bei mir so noch nicht funktioniert.

Andere Anleitung:

https://wiki.ubuntuusers.de/Archiv/System_verschl%C3%BCsseln/Entschl%C3%BCsseln_mit_einem_USB-Schl%C3%BCssel/

Logical Volume Manager - LVM

Grundlagen

Quelle: https://www.thomas-krenn.com/de/wiki/LVM_Grundlagen

Vorgehensweise

Quelle: https://www.howtoforge.com/linux_lvm

Software RAID - mdadm

Software RAID erstellen

http://www.prontosystems.org/tux/software_raid

RAID Device umbenennen

Quelle: https://serverfault.com/questions/267480/how-do-i-rename-an-mdadm-raid-array

Start with `mdadm --detail /dev/md127`:

```
Version : 0.90
Creation Time : Wed Apr 13 20:03:21 2011
Raid Level : raid10
Array Size : 656765952 (626.34 GiB 672.53 GB)
Used Dev Size : 437843968 (417.56 GiB 448.35 GB)
Raid Devices : 3
Total Devices : 2
Preferred Minor : 8
Persistence : Superblock is persistent
```

The first line shows the metadata version used by this array. Now, stop the array:

```
mdadm --stop /dev/md127
mdadm --remove /dev/md127
```

And assemble it again using the new name. If the metadata version is 1.0 or higher, use this:

```
mdadm --assemble /dev/md3 /dev/sd[abcdefghijk]3 --update=name
```

For arrays using old metadata structure (most likely 0.90, as it allows for kernel auto-assembly), use this:

```
mdadm --assemble /dev/md3 --update=super-minor /dev/sd[abcdefghijk]3
```

Beispiel - RAID1 - LUKS encryption - LVM

Quelle: http://hermann-mayer.net/blog/raid1-luks-und-lvm-ersetzen-mein-altes-speichersystem

hdparm - Energiemanagement

Festplatten nach 20 Minuten in Standby schicken

Quelle: https://linuxundich.de/hardware/festplatten-automatisch-im-betrieb-in-den-standby-schalten/

Festplatten automatisch im Betrieb in den Standby schalten

Von

Christoph Langner

-

14. April 2011

74832

68

Teilen

Ich bin gerade dabei einen Rechner mit einem Mix aus Solid-State-Disk (aka SSD) und normaler Festplatte aufzubauen. Der Vorteil der SSD ist natürlich die enorme Geschwindigkeit und der lautlose Betrieb. Allerdings ist der Preis pro GB Speicherkapazität leider ebenso enorm… Um trotzdem ausreichend Speicherplatz im Rechner zu haben, wird daher zusätzlich eine herkömmliche Festplatte verbaut. Leider wird der durch die SSD fast lautlose PC durch die Harddisk wieder deutlich lauter. Mit den passenden Einstellungen kann man jedoch die Platte bei Nichtgebrauch via hdparm automatisch abschalten lassen. Für mich eine gute Lösung um die Musik- oder Videosammlung auszulagern oder große Images von virtuellen Maschinen zu speichern…

Im Endeffekt ist das Ganze relativ leicht umzusetzen. Erfahrenen Linux-Usern müsste ich nur die Schlagworte `hdparm`, die Option `-S XX` und die Konfigurationsdatei `hdparm.conf` nennen. Wer ausführliche Informationen möchte, der sollte einfach weiterlesen □□ Erstmal muss man dazu die Daten der Platte herausfinden, die man bei Nichtgebrauch abschalten möchte. Am einfachsten geht dies über die Laufwerksverwaltung von GNOME, die Ihr über das Menü unter *Einstellungen* | *System* aufrufen könnt. Dort müsstet Ihr einfach nur eure Platte raussuchen und euch das entsprechende Gerät `/dev/sdXY` merken.

In meinem Beispiel arbeite ich also mit der Platte `/dev/sdc`. Wahrscheinlich handelt es sich bei euch um ein anderes Gerät, von daher müsst Ihr die folgenden Hinweise von daher an eure Situation anpassen. Für KDEler gibt es alternativ sicherlich auch in der KDE SC ein entsprechendes Werkzeug und wenn alle Stricke reißen, hilft auf jeden Fall ein `sudo fdisk -l` weiter.

Nun überprüft Ihr am besten erst einmal, ob sich eure Platte überhaupt in den Leerlauf schicken lässt. Alle halbwegs aktuellen Platten und Controller sollten dazu eigentlich in der Lage sein, aber bevor Ihr euch später wundert warum das nicht funktioniert, solltet Ihr erstmal einen kleinen Test machen. Ruft dazu `hdparm` mit der Option `-C` auf...

```
$ sudo hdparm -C /dev/sdc
/dev/sdc:
 drive state is:  active/idle
```

Man sieht, dass hdparm den Status korrekt abrufen konnte und die Platte läuft, man sollte dies auch an ihrem Betriebsgeräusch erkennen können. Nun könnt Ihr die Platte einmal von Hand abschalten...

```
$ sudo hdparm -y /dev/sdc
```

...der Erfolg der Aktion sollte hörbar sein. Optional könnt Ihr erneut mit der Option `-C` nachprüfen, ob die Platte auch wirklich in den Standby geschickt werden konnte.

```
$ sudo hdparm -C /dev/sdc
/dev/sdc:
drive state is:  standby
```

Sobald ihr wieder auf die Platte zugreift (bspw. über den Dateimanager), wird die Platte wieder automatisch anlaufen und ihre Daten preisgeben. Nun wäre es recht umständlich die Platte immer von Hand abzuschalten, besser wäre es, wenn sich die Platte nach einer Gewissen Zeit ohne Zugriffe automatisch abstellen könnte.

Dazu gibt es die Option `-S XX` von Hdparm. Über sie könnt ihr eine Zeitspanne definieren, ab der die Platte abgeschaltet wird. Der Zahlenwert hinter dem Schlüssel drückt nicht direkt die Zeit aus, sondern dient nur als Multiplikator. Ich zitiere einfach mal die man-Page...

> Values from 1 to 240 specify multiples of 5 seconds, yielding timeouts from 5 seconds to 20 minutes. Values from 241 to 251 specify from 1 to 11 units of 30 minutes, yielding timeouts from 30 minutes to 5.5 hours. A value of 252 signifies a timeout of 21 minutes. A value of 253 sets a vendor-defined timeout period between 8 and 12 hours, and the value 254 is reserved. 255 is interpreted as 21 minutes plus 15 seconds. Note that some older drives may have very different interpretations of these values.

Ein Wert von bspw. 60 entspricht daher 60*5 Sekunden, also 300 Sekunden oder fünf Minuten. Zu kurz sollte man das Timeout nicht wählen, da zu häufiges Runter- und wieder Hochdrehen des Motors die Lebensdauer der Platte beeinträchtigen kann.

```
$ sudo hdparm -S 60 /dev/sdc
```

Damit das Ganze automatisch beim Start des Systems ausgeführt wird, solltet Ihr diese Einstellung in der Konfigurationsdatei `/etc/hdparm.conf` speichern. Am besten arbeitet Ihr in dem Fall mit UUIDs, so dass sich der Eintrag immer auf die gewünschte Platte bezieht, selbst wenn sich eure Plattenkonfiguration einmal ändern sollte. Bestimmt die UUID über bspw...

```
$ sudo blkid
[...]
/dev/sdb1: LABEL="data-intern" UUID="ff7f1cdd-431a-4e47-8b90-49737cec82b3" TYPE="ext4"
/dev/sdc1: LABEL="datengrab" UUID="4c9dca33-e3e1-4a6a-a7d4-110cdbb4cfbe" TYPE="ext3"
```

...und fügt dann einfach euren gewünschten Eintrag an das Ende der hdparm.conf an...

```
$ sudo gedit /etc/hdparm.conf
[...]
/dev/disk/by-uuid/4c9dca33-e3e1-4a6a-a7d4-110cdbb4cfbe {
spindown_time = 60
}
```

...ab dem nächsten Neustart sollte die Einstellung automatisch aktiv sein und eure Platte bei Nichtgebrauch abgeschaltet werden. Abschließend noch der Hinweis, dass Ihr beim Arbeiten mit Hdparm wirklich euer Hirn einschalten müsst. Bei der „Schlaffunktion“ müsst Ihr wie bereits erwähnt überdenken, dass zu häufiges Hochdrehen der Platte die Lebensdauer reduzieren kann. Setzt das Intervall daher auf keinen Fall zu kurz.