Dateisysteme

- Speichercaches leeren
- Anzahl gleichzeitig geöffneter Dateien hochsetzen
- Dateien löschen anhand der inode-Nummer - Datei läßt sich nicht löschen
- Benutzer- und Gruppen-Quota unter Redhat-Linux
- mtime, ctime, atime - Was sind die Unterschiede?
- Eine Partition mit mehr als 2TB unter Linux erstellen
- tmpfs - Temporary File System
- chown/chgrp für Symlinks
- Software-RAID wiederherstellen
- mdadm consistency check abbrechen
- NFS-Server exports neu einlesen
- Platzfresser - Die größten Verzeichnisse finden
- Rekursiv in einem Verzeichnis die neuesten Dateien finden
- davfs2 WebDAV Dateisystem mounten

Speichercaches leeren

Wer mehr über die Speicherverwaltung unter Linux wissen will, sollte sich dieses Buch zu Gemüte führen

Über Sinn und Unsinn dieser Funktion (also das Leeren der Speichercaches) läßt sich streiten, da sie eigentlich fürs Debugging und Programmieren gedacht ist. Jedenfalls bietet der 2.6er Kernel die Möglichkeit Speichercaches des Filesystems zu leeren. Vor dem leeren der Caches sollte 'sync' ausgeführt werden. Folgende Möglichkeiten gibt es:

pagecache leeren (Pages, die über einen Eintrag auf der Festplatte verfügen, werden im schnellen RAM vorgehalten):

```
echo 1 > /proc/sys/vm/drop_caches
```

dentrie- und inode-cache leeren (mit deren Hilfe wird ein Dateisystem im Speicher abgebildet):

```
echo 2 > /proc/sys/vm/drop_caches
```

page-, dentrie- und inode-cache leeren:

```
echo 3 > /proc/sys/vm/drop_caches
```

Das top-Tool zeigt diese speziellen Caches nicht gesondert an, aber im htop kann man schön sehen wieviel gerade belegt ist (die Anzeige beim Speicher, h drücken für die Hilfe). Viel Sinn den Cache auf einem Produktivsystem zu leeren machts nicht. Ich vermute sogar eher das Gegenteil, da ja Seiten, die vorher im Cache lagen, bei erneutem Zugriff wieder von der langsamen Platte geholt werden müssen. Und wenn das Betriebssystem den Speicher, der momentan durch den Cache belegt ist, wieder benötigt holt es sich den schon alleine. Aber zumindest wird dann wieder viel freier Speicher im top angezeigt und der Kunde ist beruhigt ;)

Über die Sysctl-Einstellung vm.vfs_cache_pressure läßt sich konfigurieren, wie oft bzw. ab wann der Kernel sich Speicher aus diesen Caches zurückholt. Der Standardwert ist 100. Stellt man einen niedrigeren Wert ein, holt sich der Kernel nicht mehr so viel Speicher zurück. Beim Wert 0 wird gar kein Speicher mehr freigegeben, das kann allerdings schnell zu einem out-of-memory Problem werden. Bei Werten über 100 holt sich der Kernel den Speicher schneller wieder.

Anzahl gleichzeitig geöffneter Dateien hochsetzen

Mit dem Einzug von systemd in Debian funktioniert die Methode über die limits.conf nicht mehr (dort werden trotzdem noch die user level limits gesetzt). Die Anzahl der gleichzeitig geöffneten Files wird nun in der sysctl.conf und/oder über das systemd-unit-script gesetzt.

Systemd-Style, der systemctl edit wird eine Override-Datei in /etc/system/systemd/<service-name> anlegen:

```
systemctl edit mysql.service


[Service]

LimitNOFILE=65536
```

Hier die Version ohne Systemd:

Aktuellen Wert herausfinden:

```
# sysctl fs.file-nr
fs.file-nr = 3264 0 1637554
```

Ich habe den Wert nun einfach x3 genommen und in der sysctl.conf hinterlegt:

```
# set max number of open files
fs.file-max=4912662
```

Aktiviert wird der Eintrag mit `sysctl –system`. Kontrolliert wird das Setting über das Kommando oben.

Für ein Limit im user-level Bereich läßt sich das limits.conf verwenden. Kontrolliert werden kann das Setting über `ulimit -n`.

In /etc/security/limits.conf folgenden Eintrag vornehmen

/etc/security/limits.conf

```
*               soft      nofile           4096
*               hard     nofile           4096
# oder per user
meinuser         soft      nofile           8192
meinuser         hard      nofile           8192
```

Achtung bei openssh-server Versionen kleiner 3.8 ist nach dieser Änderung kein Einloggen via SSH mehr möglich. Folgende Fehlermeldung taucht dabei im Log auf:

```
Nov  8 09:23:03 XXX sshd[18108]: fatal: PAM session setup failed[6]: Permission denied
```

Dies kann behoben werden durch:

- Ein Update auf eine aktuellere Version oder
- Setzen der Parameter `UsePrivilegeSeparation no` und `PAMAuthenticationViaKbdInt yes` in /etc/ssh/sshd_config. Dies ist aber unsicher und nicht empfohlen.

Dateien löschen anhand der inode-Nummer - Datei läßt sich nicht löschen

Eine Datei mit Sonderzeichen oder Leerzeichen läßt sich nicht löschen. Wie löse ich dieses Problem?

Zum Beispiel folgende Datei:

```
[root@bash]# ls -al
drwxr-xr-x  2 root root 4096 2007-10-19 19:56 .
drwxr-x--- 32 root root 4096 2007-10-19 19:47 ..
-rw-r--r--  1 root root    0 2007-10-19 19:56 \abc \+dgh $
```

Das File kann natürlich auch mit „rm -f „\abc \+dgh $“ oder Tab-Completion gelöscht werden, aber es gibt Situationen, da ist das nicht so einfach möglich. Wie lösche ich dann solche Dateien? Ganz einfach:

```
[root@bash]# ls -ail
2479588 drwxr-xr-x  2 root root 4096 2007-10-19 20:01 .
2478209 drwxr-x--- 32 root root 4096 2007-10-19 19:47 ..
2479591 -rw-r--r--  1 root root    0 2007-10-19 20:01 \abc \+dgh $
```

mit „ls -ali“ (a zeigt alle, l die Liste und i die Inode-Nummer der Datei) bekommt man die Inode der Datei heraus. Jetzt kann die Datei mit dem folgenden Befehl gelöscht werden:

```
[root@bash]# find . -inum 2479591 -exec rm -f {} \;
```

Benutzer- und Gruppen-Quota unter Redhat-Linux

1. Quota installieren

```
yum -y install quota
```

2. /etc/fstab editieren, bei den gewünschten Partitionen ,usrquota,grpquota anfügen

```
/dev/VolGroup00/LogVol00 /                       ext3    defaults,usrquota,grpquota        1 1
LABEL=/boot              /boot                   ext3    defaults        1 2
tmpfs                    /dev/shm                tmpfs   defaults        0 0
devpts                   /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                    /sys                    sysfs   defaults        0 0
proc                     /proc                   proc    defaults        0 0
/dev/VolGroup00/LogVol01 swap                    swap    defaults        0 0
```

3. nun folgende Befehle ausführen (natürlich nur für die gewünschten Partitionen, hier das Beispiel für /):

```
touch /aquota.user /aquota.group
chmod 600 /aquota.*
mount -o remount /
quotacheck -avugm
quotaon -avug
```

4. mittels edquota werden die Quotas für User und Gruppen gesetzt, einfach die Werte für Blocks oder Inodes in dieses Files eintragen und speichern. Wie man den richtigen Wert für Blocks herausfindet steht unter Blocks berechnen.

```
$ edquota -u magenbrot
Disk quotas for user magenbrot (uid 500):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/mapper/VolGroup00-LogVol00       1552          0          0         56        0        0

$ edquota -g magenbrot
Disk quotas for group magenbrot (gid 500):
  Filesystem                   blocks       soft       hard     inodes     soft     hard
  /dev/mapper/VolGroup00-LogVol00       1548          0          0         55        0        0
```

5. mit edquota wird ebenfalls die grace-period gesetzt, nach deren Ablauf das Softlimit forciert wird:

```
$ edquota -t
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
  Filesystem             Block grace period     Inode grace period
  /dev/mapper/VolGroup00-LogVol00                 7days                   7days
```

6. mittels repquota die aktiven Quotas anzeigen:

```
*** Report for user quotas on device /dev/mapper/VolGroup00-LogVol00
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      -- 4499920       0       0         133113     0     0
```

```
daemon    --      20       0       0              3     0     0
lp        --      16       0       0              2     0     0
news      --    7460       0       0            183     0     0
uucp      --     168       0       0              1     0     0
games     --     344       0       0             86     0     0
rpm       --   50260       0       0             67     0     0
apache    --      24       0       0              3     0     0
avahi     --      20       0       0              3     0     0
rpcuser   --       8       0       0              1     0     0
named     --      56       0       0              7     0     0
smmsp     --      16       0       0              2     0     0
ntp       --      16       0       0              2     0     0
webalizer --      32       0       0              4     0     0
squid     --      16       0       0              2     0     0
dovecot   --       8       0       0              1     0     0
xfs       --       4       0       0              1     0     0
hsqldb    --      20       0       0              2     0     0
magenbrot --    1552       0       0             56     0     0
```

mtime, ctime, atime - Was sind die Unterschiede?

Unter Linux besitzt jede Datei 3 Zeitstempel: mtime, ctime und atime.

- atime (access time) ist der Zeitpunkt des letzten Lesezugriffs.
- ctime ist der Zeitpunkt der letzten Änderung an der Inode der Datei
- mtime ist der Zeitpunkt der letzten Änderung am Inhalt der Datei

mtime und ctime ändern sich immer wenn etwas am Inhalt der Datei gemacht wird. ctime ändert sich auch, wenn z.B. der Besitzer der Datei geändert wird, am Inhalt jedoch nichts gemacht wird.

Eine Partition mit mehr als 2TB unter Linux erstellen

Die Festplatten werden immer größer, der verfügbare Speicherplatz in Storages sowieso. Mit dem Standardtool fdisk kann man leider keine Partitionen größer als 2TB erstellen. Hier kommt jetzt das Programm parted ins Spiel. parted unterstützt die Erstellung von Intel EFI/GPT Partitionstabellen. EFI soll mal das normale PC BIOS ablösen, das mit dem orginal IBM PC vor Ewigkeiten eingeführt wurde und den heutigen Anforderungen und Möglichkeiten nicht mehr entspricht.

Vor dem Einsatz von GPT muss sichergestellt werden, dass es vom Kernel unterstützt wird. Anwender von Redhat Enterprise Linux und CentOS haben hier keine Probleme, die Unterstützung ist vorhanden. User von Debian oder Ubuntu müssen sich einen eigenen Kernel bauen:

```
File Systems
   Partition Types
     [*] Advanced partition selection
     [*] EFI GUID Partition support (NEW)
```

Ein GPT-Volume wird nun folgendermaßen erstellt:

```
# parted /dev/sdx

GNU Parted 1.6.19
Using /dev/sdx
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)

> mklabel gpt
> mkpart primary 0 3001G
> quit

# fdisk -l
Platte /dev/sdx: 2800.9 GByte, 2800975872000 Byte
255 Köpfe, 63 Sektoren/Spuren, 340532 Zylinder
Einheiten = Zylinder von 16065 * 512 = 8225280 Bytes

   Gerät Boot      Start         End      Blocks   Id  System
/dev/sdb1               1      267350  2147483647+  ee  EFI GPT

# mkfs.ext3 /dev/sdx1
```

Nach dem Formatieren kann die Platte ganz normal gemounted werden.

tmpfs - Temporary File System

tmpfs ist ein Dateisystem, welches (virtuellen) Arbeitsspeicher statt Festplatten als Speicher verwendet. Im Vergleich zu ramdisk entfallen hier viele Schritte (Anlegen der Ramdisk über Kernel-Parameter, formatieren und mounten). Über einen einfachen Mount-Befehl kann ein solcher Bereich erzeugt und ins Dateisystem eingehängt werden.

Mounten über die Kommandozeile:

```
mount -t tmpfs -o size=20m none /mnt/bla
```

Mounten über die Datei /etc/fstab:

```
none /mnt/tmpfs tmpfs defaults,size=20m,mode=775,uid=<user>,gid=<group> 0 0
```

Ich habe bei mir z.B. die status.dat von Icinga (Nagios-Fork) ausgelagert, da diese auf einer umfangreichen Installation, wo zusätzlich noch gerne in der Weboberfläche rumgemeiert wird, sekündlich geschrieben und auch ziemlich umfangreich werden kann. Das nimmt dann schon ziemlich Last vom Server weg.

chown/chgrp für Symlinks

für das Ändern des Besitzer oder der Gruppe eines Symlinks gibts einen speziellen Parameter für chown oder chgrp.

Mit -h oder –no-dereference läßt sich einschalten, dass direkt der Symlink angefasst werden soll anstatt der referenzierten Datei:

```
# cd /var/lib/mailman
# ls -l

total 24K
lrwxrwxrwx   1 root root   24 May 19 16:40 Mailman -> /usr/lib/mailman/Mailman
drwxrwsr-x   4 list list 4.0K May 19 16:40 archives
lrwxrwxrwx   1 root root   20 May 19 16:40 bin -> /usr/lib/mailman/bin
lrwxrwxrwx   1 root root   24 May 19 16:40 cgi-bin -> /usr/lib/cgi-bin/mailman
lrwxrwxrwx   1 root root   21 May 19 16:40 cron -> /usr/lib/mailman/cron
drwxrwsr-x   2 list list 4.0K Jun 17 10:40 data
lrwxrwxrwx   1 root root   25 May 19 16:40 icons -> /usr/share/images/mailman
drwxrwsr-x 128 list list 4.0K Jun 17 10:38 lists
lrwxrwxrwx   1 root root   17 May 19 16:40 locks -> /var/lock/mailman
lrwxrwxrwx   1 root root   16 May 19 16:40 logs -> /var/log/mailman
lrwxrwxrwx   1 root root   21 May 19 16:40 mail -> /usr/lib/mailman/mail
drwxrwsr-x  38 list list 4.0K May 19 16:40 messages
drwxrwsr-x   2 list list 4.0K Feb 16 21:14 qfiles
lrwxrwxrwx   1 root root   24 May 19 16:40 scripts -> /usr/lib/mailman/scripts
drwxrwsr-x   2 list list 4.0K Feb 16 21:14 spam
lrwxrwxrwx   1 root root   12 May 19 16:40 templates -> /etc/mailman

# chgrp -h list *
# ls -l

total 24K
lrwxrwxrwx   1 root list   24 May 19 16:40 Mailman -> /usr/lib/mailman/Mailman
drwxrwsr-x   4 list list 4.0K May 19 16:40 archives
lrwxrwxrwx   1 root list   20 May 19 16:40 bin -> /usr/lib/mailman/bin
lrwxrwxrwx   1 root list   24 May 19 16:40 cgi-bin -> /usr/lib/cgi-bin/mailman
lrwxrwxrwx   1 root list   21 May 19 16:40 cron -> /usr/lib/mailman/cron
drwxrwsr-x   2 list list 4.0K Jun 17 10:40 data
lrwxrwxrwx   1 root list   25 May 19 16:40 icons -> /usr/share/images/mailman
drwxrwsr-x 128 list list 4.0K Jun 17 10:38 lists
lrwxrwxrwx   1 root list   17 May 19 16:40 locks -> /var/lock/mailman
lrwxrwxrwx   1 root list   16 May 19 16:40 logs -> /var/log/mailman
lrwxrwxrwx   1 root list   21 May 19 16:40 mail -> /usr/lib/mailman/mail
drwxrwsr-x  38 list list 4.0K May 19 16:40 messages
drwxrwsr-x   2 list list 4.0K Feb 16 21:14 qfiles
lrwxrwxrwx   1 root list   24 May 19 16:40 scripts -> /usr/lib/mailman/scripts
drwxrwsr-x   2 list list 4.0K Feb 16 21:14 spam
lrwxrwxrwx   1 root list   12 May 19 16:40 templates -> /etc/mailman
```

Software-RAID wiederherstellen

Arrayzustand prüfen:

```
# cat /proc/mdstat
```

zu jedem mdX device gehoeren mehrere sdX (oder xvd oder nvme, je Hardware), ausserdem sollte da [UU] stehen. Wenn nicht ist was faul.

Partition in Ordnung:

```
md1 : active raid1 sdb1[1] sda1[0]
      48064 blocks [2/2] [UU]
```

Eine Platte fehlt:

```
md0 : active raid1 sda2[0]
      2096384 blocks [2/1] [U_]
```

Wenn sich das Teil nur verhaspelt hat, kann man die fehlende Platte mit mdadm wieder hinzufügen:

```
# mdadm [RAID-Device] --add [Platte]
```

im obigen Beispiel:

```
# mdadm /dev/md0 --add /dev/sdb2
```

Sollte eine Platte getauscht worden sein, muss vorher die Partitionstabelle von der funktionierenden Platte geholt und auf die neue Festplatte kopiert werden:

```
# sfdisk -d /dev/sda | sfdisk /dev/sdb
```

jetzt kann das RAID wieder zusammengefügt werden:

```
# mdadm /dev/md1 --add /dev/sdb1
```

Die Kernel-Defaults für die minimale und maximale Wiederherstellungsgeschwindigkeit sind bei aktuellen Platten/SSDs deutlich zu niedrig. Sofern das System nicht schon wieder produktiv unter Last steht, kann man die Limits wie folgt anschauen, bzw. (schrittweise) hoch setzen:

```
## Anzeigen:
# sysctl dev.raid.speed_limit_min
dev.raid.speed_limit_min = 1000

# sysctl dev.raid.speed_limit_max
dev.raid.speed_limit_max = 200000

## Setzen:
# sysctl -w dev.raid.speed_limit_min=50000
dev.raid.speed_limit_min = 50000

# sysctl -w dev.raid.speed_limit_max=2000000
```

```
dev.raid.speed_limit_max = 2000000
```

Anschließend muss noch der Grub in den Bootsektor der neuen Disk geschrieben werden. Sofern das System noch bootet und nicht in einem Rettungssystem wiederhergestellt wurde, dann läßt sich der Grub einfach mit folgendem Kommando einrichten:

```
# grub-install /dev/sdb
```

Falls das System im Rescue-Mode läuft, wird es evtl. etwas umständlicher. grub-mkdevicemap erzeugt die Devicemap neu, da sich die UUID der Platte ja geändert hat. Für die ganze Prozedur sollten die Platten gemountet und per chroot in das Orginalsystem gewechselt werden.

```
# mount /dev/mdX /mnt
# mount proc sys dev
# chroot
# grub-mkdevicemap -n
# grub-install /dev/sdb
```

mdadm consistency check abbrechen

Unter Debian wird standardmäßig jeden ersten Sonntag im Monat ein Check über alle eingerichteten Software-RAIDs gefahren. Bei sehr großen Festplatten kann das auch mal länger dauern und den normalen Serverbetrieb behindern.

Der Check läßt sich folgendermaßen abbrechen (md3 durch das gewünschte Softraid Device ersetzen):

```
echo idle > /sys/block/md3/md/sync_action
```

NFS-Server exports neu einlesen

nach einer Änderung in /etc/exports auf einem NFS-Server lassen sich neue oder gelöschte Exports mit diesem Kommando neu laden:

```
exportfs -ra
```

Alternativ, falls es das Kommando exportfs nicht gibt, funktioniert auch ein -HUP an nfsd:

```
killall -HUP nfsd
```

Platzfresser - Die größten Verzeichnisse finden

Die Disk läuft voll, was ist schuld? Kann man was aufräumen?

Für einen schnellen Überblick kann schonmal das Tool „ncdu“ helfen. Nach der Installation ist es sofort bereit:

```
$ sudo apt-get update
$ sudo apt-get install ncdu

$ ncdu

ncdu 1.11 ~ Use the arrow keys to navigate, press ? for help
--- /home/magenbrot ----------------
   16,6 GiB [##########] /.local
   11,2 GiB [######    ] /nextcloud
.   5,6 GiB [###       ] /.cache
  609,1 MiB [          ] /.config
  366,7 MiB [          ] /software
  146,5 MiB [          ] /magenbrot-tools
   34,4 MiB [          ] /.linuxmint
   29,4 MiB [          ] /.purple
```

Alternativ per du:

```
# Verzeichnisse inkl. Unterverzeichnisse anzeigen
du -h * | sort -hr | head -n 500
```

oder

```
# Nur die Hauptverzeichnisse anzeigen
du -h --max-depth=1 | sort -hr | head -n 500
```

Rekursiv in einem Verzeichnis die neuesten Dateien finden

Die Webseite funktioniert plötzlich nicht mehr? Die Applikation mag nicht mehr bauen? Ihr habt kein Versionskontrollsystem?

Dann kann euch `find` helfen, die zuletzt geänderten Dateien in einem Verzeichnis inkl. den Unterverzeichnissen, anzuzeigen. Das folgende Kommando listet die 20 zuletzt geänderten Dateien auf:

```
cd /var/www
find -type f -print0 | xargs -0 stat --format '%Y :%y %n' | sort -nr | cut -d: -f2- | head -n 20
```

davfs2 WebDAV Dateisystem mounten

Installation

```
apt update
apt install davfs2
```

Konfiguration

In der `/etc/fstab` diesen Eintrag hinterlegen:

```
# WebDAV cloud.ovtec.it
https://nextcloud.example.com/remote.php/dav/files/myuser/ /home/myuser/cloud davfs
_netdev,user,uid=2342,gid=2342,conf=/etc/davfs2/davfs2.conf 0 0
```

In der Datei `/etc/davfs2/davfs2.conf` die beiden Parameter anpassen:

```
use_locks         0
cache_size        250 # MiByte
```

In der Datei `/etc/davfs2/secrets` die Zugangsdaten eintragen, z.B.:

```
/home/myuser/cloud myuser password
```

Jetzt sollte es mit "`mount /home/myuser/cloud`" gemounted werden können.