# Software RAID

- Kurzakte –

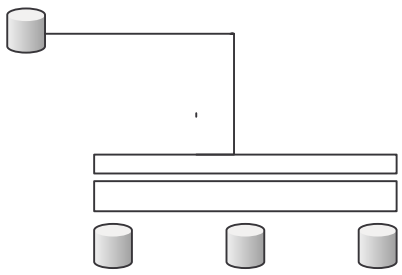
| | |
|---|---|
| ***Installation*** | |
| • Pakete raidtools (Management) und dmraid (Anzeige der RAID-Settings)<br>• mdadm (umfasst Funktionalität der oberen Pakete)<br>• Voraussetzungen: mindestens Kernel 2.4 mit RAID-Unterstützung (ältere patchbar) und raid-autodetect<br>Test durch das Vorhandensein der Datei /proc/mdstat | |
| ***Besondere Kennzeichen*** | |
| • Verwaltungsschicht zwischen Dateisystem und Hardware<br>• Nachteile von Software- gegenüber Hardware-RAID sind die geringere Performance, die spürbare Mehrbelastung der CPU, die mit den Treibern nicht voll ausnutzbare Parallelisierung von Zugriffen<br>• Achtung: auch die "Hard"-Varianten auf den Motherboards arbeiten mit Treibern – kein Unterschied zu Soft! | |
| ***Konfiguration*** | |
| /etc/raidtab | • Hauptkonfiguatonsdatei zur Festlegung des RAID-Levels, persistenten Superblocks und der physikalischen Geräte<br>• Musterdateien in /usr/share/doc/packages |
| ***Programme zum RAID-Management*** | |
| mkraid /dev/md0 | • initialisiert das Array, schreibt persistenten Superblock, startet Array<br>• liest Informationen aus /etc/raidtab<br>• bei Veränderungen bewirkt mkraid ein Update des RAID-Verbunds<br>• brutales Überschreiben alter Superblockdaten mit Option -R |
| mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb6 /dev/sdc5 (gleiche Wirkung w. o.) | |
| raidstop /dev/md0<br>mdadm –S /dev/md0 | • Stoppen des RAID-Devices (vorher RAID-Device aushängen) |
| raidstart /dev/md0<br>mdadm –R /dev/md0 | • (Re)Start des RAID-Devices |
| raidsetfaulty /dev/md0 /dev/sdb1 | • markiert die Partition hdd1 im laufenden RAID-Verbund als defekt |
| raidhotremove /dev/md0 /dev/sdc2<br>mdadm /dev/md1 -r /dev/sdc2 | • Entfernen hotplugfähiger gecrashter Platten aus dem laufendem RAID-Verbund<br>• eine vorhandene spare disk wird sofort integriert<br>• die ausgefallene RAID-Partition wird als neue spare disk wieder eingefügt<br>• intakte Disks können nicht entfernt werden |
| raidhotadd /dev/md1 /dev/sdc2<br>mdadm /dev/md1 -a /dev/sdc2 | • Einarbeiten der ausgetauschten und intakten Platte ins Array und Aktualisierung der persistenten Superblöcke<br>• dadurch werden bei korrekt aktualisierter /etc/raidtab keine Daten gefährdet (im Gegensatz zur Nutzung von mkraid) |
| ***Funktionsprüfung des Arrays*** | |
| /var/log/messages | • bei failures können tonnenweise Meldungen erzeugt werden |
| /proc/mdstat | • existiert bei einem RAID-fähigem Kernel<br>• Anzeige des RAID-Levels und der integrierten Devices<br>• faulty devices wreden angezeigt durch ein führendes F |
| lsraid -a /dev/md0<br>mdadm --detail /dev/md0 | • Check des RAID-Arays<br>• Fehler werden lautstart angezeigt |
| mdadm --monitor --mail=root@localhost --delay=1800 /dev/md2 & mdadm läuft als Daemon im Monitor-Mode und checkt Array md2 alle 30 min, bei kritischen Events und Failures erfolgt eine Warnung per email | |
| ***Voraussetzungen zum automatischn Start (Erkennen) des RAID-Verbundes beim Booten*** | |
| • »Autodetect RAID partitions« ist im Kernel aktiviert.<br>• »persistent-superblock 1« ist in der Datei /etc/raidtab für das Device gesetzt.<br>• Der Partitionstyp, der für das RAID genutzten Partitionen, muss auf »0xFD« gesetzt werden. | |
| ***(8), Dokumentation*** | |
| • man: raidtab (5), ckraid (8), lsraid (8), mkraid (8), raid0run (8), raidadd (8), raidreconf (8), raidrun (8), raidstart (8), raidstop (8), dmraid (8), md (4), mdadm.conf (5), mdadm<br>• HowTo: /usr/share/doc/packages/raidtools | |

# RAID – Checkliste zum Erstellen des RAID-Verbunds

| *RAID-Verbund auf der Kommandozeile erstellen oder verändern* |
|---|
| (1) ein bereits aktives Array aushängen (**umount /dev/md0**) und deaktivieren (**raidstop /dev/md0**)<br>(2) Erzeugen der Partitionen (primär) auf den Festplatten (fdisk Option n) des zukünftigen Arrays und Vergabe des Partitionstyps FD (fdisk Option t und Schreiben mit w) RAID-Array darf nicht aktiv sein →raidstop /dev/md0<br>(3) Konfiguration der /etc/raidtab mit persistentem Superblock<br>(4) Initialisieren und Starten des Arrays **mkraid /dev/md0**<br>ab jetzt beginnt die Rekonstruktion<br>diese ist transparent, das Array kann trotz dieser Aktivität sofort genutzt werden (formatieren und mounten)<br>(5) Kontrolle in **/proc/mdstat**<br>(6) Erzeugen des Dateisystems mit **mkfs.ext3 /dev/md0**<br>(7) Mounten des Arrays in den Linux-Dateibaum (**mount /dev/md0 /media/raid**) |

| *RAID-Verbund mit Yast erstellen* |
|---|
| (1) Erstellen der RAID-Partitionen jeweils durch Auswahl der Festplatte → Anlegen → primäre Partition → Auswahl von „nicht formatieren“ →Auswahl der Dateisystem-ID „0xFD Linux RAID“ → evtl. Größe festlegen<br>(2) Hinzufügen zum Array mit Button „RAID“ → RAID anlegen<br>a. Schritt 1: Festlegen des RAID-Typs (Levels)<br>b. Schritt 2: Hinzufügen der RAID-Partitionen vom Typ FD<br>c. Schritt 3: Aufbringen des Dateisystems auf das RAID-Device mit persistentem Superblock und Angabe des Mountpoints (Chunk-Größe=32 für Performance-Gewinn bei Leveln 0 und 5) → Optionen für ext3: Schrittweite für veränderte Chunkgröße berechnen = stride (bei chunk =32 und Blockgröße=4096 ist stride=8) mit „Anwenden“ den RAID Verbund zusammenfügen<br>(3) spare disks müssen per Hand in die /etc/raidtab eingetragen werden |

einfache Beipielskonfiguration:



# RAID – Checkliste zur Reparatur des RAID-Verbunds

| *sichere Wiederherstellung eines nicht hotplugfähigen Arrays nach Failure* |
|---|
| (1) Simulation eines Plattendefekts durch Abziehen des Spannungskabels von der Festplatte oder mit dem Kommando **raidsetfaulty /dev/md0 /dev/sdb1** oder **mdadm --manage --set-faulty /dev/md0 /dev/sdb2** |
| (2) Anzeige des Recoveries und der defekten Platte mit **watch cat /proc/mdstat** (defekte Partition wird mit (F) gekennzeichnet)<br>das Recovery abwarten |
| (3) Herunterfahren des Rechners und Austauschen der Festplatte und Neustart |
| (4) Erstellen einer RAID-Partition mit dem Typ 0xFD<br>(RAID darf nicht aktiv sein!!! → **umount /dev/md0** und **raidstop /dev/md0**) |
| (5) Anpassen der /etc/raidtab, falls die neue Partitionsbezeichnung von der alten abweicht |
| (6) Aktivierung des RAID-Verbundes mit **raidstart /dev/md0** |
| (7) Einarbeitung der neuen Partition in den Verbund und Schreiben von neuen persistenten Superblöcken mit **raidhotadd /dev/md0 /dev/sdb1** ohne die Daten zu gefährden (**mkraid** wäre auch möglich, zerstört aber die Daten im Verbund) |
| (8) Inspektion des Verlaufs mit **watch cat /proc/mdstat** bis zur Meldung „sync finished“ |
| (9) eneutes Mounten in den Verzeichnisbaum |

| *sichere Wiederherstellung eines nicht hotplugfähigen Arrays nach Failure mit spare disk* |
|---|
| (1) Simulation eines Plattendefekts durch Abziehen des Spannungskabels von der Festplatte oder mit dem Kommando **raidsetfaulty /dev/md0 /dev/sdb1** oder **mdadm --manage --set-faulty /dev/md0 /dev/sdb2** |
| (2) jetzt beginnt das recovery über den RAID-Daemon md0_raidx |
| (3) Anzeige des Recoveries und der defekten Platte mit **watch cat /proc/mdstat** (defekte Partition wird mit (F) gekennzeichnet)<br>das Ende des recovery-Prozesses abwarten<br>damit ist unser RAID wieder arbeitsfähig! (Status „good“ bei **lsraid –a /dev/md0**)<br>Reparatur irgendwann später: |
| (4) mit **umount /dev/md0** und **raidstop /dev/md0** wird der aktuelle Zustand auf die Partitionen geschrieben |
| (5) ein **raidstart -a** und **cat /proc/mdstat** zeigt, dass die defekte Partition /dev/sdc1 völlig fehlt, die spare disk hat diese Rolle im Verbund eingenommen |
| (6) Herunterfahren des Rechners und Austauschen der Festplatte und Neustart |
| (7) Erstellen einer RAID-Partition mit dem Typ 0xFD<br>(RAID darf nicht aktiv sein!!!) |
| (8) keine Panik, die spare disk ist noch nicht da:<br>Anpassen der Partitionen in der /etc/raidtab, denn die alte spare disk ist ja jetzt eine raid-disk und die ausgetauschte, intakte Platte wird zur spare-disk |
| (9) Aktivierung des RAID-Verbundes mit **raidstart /dev/md0** |
| (10) Einarbeitung der neuen Partition in den Verbund und Schreiben von neuen persistenten Superblöcken mit **raidhotadd /dev/md0 /dev/sdb1** ohne die Daten zu gefährden (**mkraid** wäre auch möglich, zerstört aber die Daten im Verbund) |
| (11) Inspektion des Verlaufs mit **watch cat /proc/mdstat** bis zur Meldung „sync finished“ |
| (12) Schreiben der neuen persistenten Superblöcke mit **raidstop /dev/md0** |
| (13) eneutes Mounten in den Verzeichnisbaum nach **raidstart /dev/md0** |

| *hotplugfähiges Array nach Failure wieder herstellen* |
|---|
| (1) Simulation eines Plattendefekts durch Abziehen des Spannungskabels von der Festplatte oder mit dem Kommando **raidsetfaulty /dev/md0 /dev/sdb1** oder<br>**mdadm --manage --set-faulty /dev/md1 /dev/sdc2** |
| (2) Anzeige der defekten Platte mit **cat /proc/mdstat** (defekte partition wird mit (F) gekennzeichnet) |
| (3) Aushängen einer kaputten Platte:<br>• **cat /proc/mdstat** aktueller Zustand<br>• **mdadm /dev/md0 -f /dev/sdb1** sdb1 als failed markieren<br>• **mdadm /dev/md0 -r /dev/sdb1** sdb1 aus Verbund entfernen<br>eine evtl. vorhandene spare disk wird sofort eingearbeitet |
| (4) Paritätsinformationen auf neuer Platte wiederherstellen:<br>• **mdadm /dev/md0 -a /dev/sdb2**<br>• mit **watch cat /proc/mdstat** lässt sich Vorgang verfolgen |

## /etc/raidtab - Beispiele für verschiedene RAID-Level

Linear Mode (einfaches Anhängen)

```
raiddev /dev/md0
        raid-level      linear
        nr-raid-disks   2
        chunk-size      32          # irrelevant
        persistent-superblock 1
        device          /dev/sdb6   # Größe der Partitionen egal
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

RAID-0 (parallele Nutzung)

```
raiddev /dev/md0
        raid-level      0
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size     4
        device          /dev/sdb6   # annähernd gleiche Größe sinnvoll
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

RAID-1 (Spiegelung) – die identische Datenhaltung ist hier durch separates Einhängen sichtbar (Kontrolle)

```
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        nr-spare-disks  1
        persistent-superblock 1
        device          /dev/sdb6   # annähernd gleiche Größe sinnvoll
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
        device          /dev/sdd5   # ab diesem Level möglich
        spare-disk      0
```

**mdadm --create -v /dev/md0 --level=1 --raid-devices=2 /dev/sda3 /dev/sdb1 ...**

RAID-5 (parallele Nutzung mit Parität)

```
        raiddev /dev/md0
        raid-level      5
        nr-raid-disks   4
        nr-spare-disks  1
        persistent-superblock 1
        parity-algorithm        left-symmetric # einzig sinnvoll
        chunk-size      32
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb1
        raid-disk       1
        device          /dev/sdc1
        raid-disk       2
        device          /dev/sdd1
        raid-disk       3
        device          /dev/sdf1
        spare-disk     0
```

Falls auf das RAID-Device ein ext-Dateisystem aufgebracht wird, kann man die Performance mit dem speziellen parameter stride erhöhen:
Wenn die chunk-size=32 und die Blockgröße des ext-Dateisystems 4096 KB beträgt, wird der Wert stride=8 gesetzt. Dies bestimmt die zu verteilende Blockgröße auf den RAID-Verbund.

Bsp.: **mkfs.ext3 -b 4096 -R stride=8 /dev/md0**