Software-RAID mit raidtools

Software-RAID

von Thomas King

RAID? Was ist RAID? Welche Einsatzmöglichkeiten gibt es? Und wie lässt sich Software-Raid unter Linux realisieren? - Diese Fragen soll dieser Artikel beantworten.

RAID

An der University of California, Berkeley, U.S.A., wurde der Begriff RAID 1987 geprägt. RAID steht für "Redundant Array of Inexpensive Disks". Mit RAID werden mehrere unabhängige Block Devices (meist IDE- oder SCSI-Festplatten) zu einer großen logischen Einheit zusammengeschlossen (siehe Abbildung 1). Auf dem Array werden nicht nur die eigentlichen Daten gespeichert, sondern es werden auch Redundanz-Daten gespeichert. Diese Redundanz-Daten sind entweder Parity- Daten, die aus mehreren Datenblöcken berechnet werden, oder eine Kopie der eigentlichen Daten.

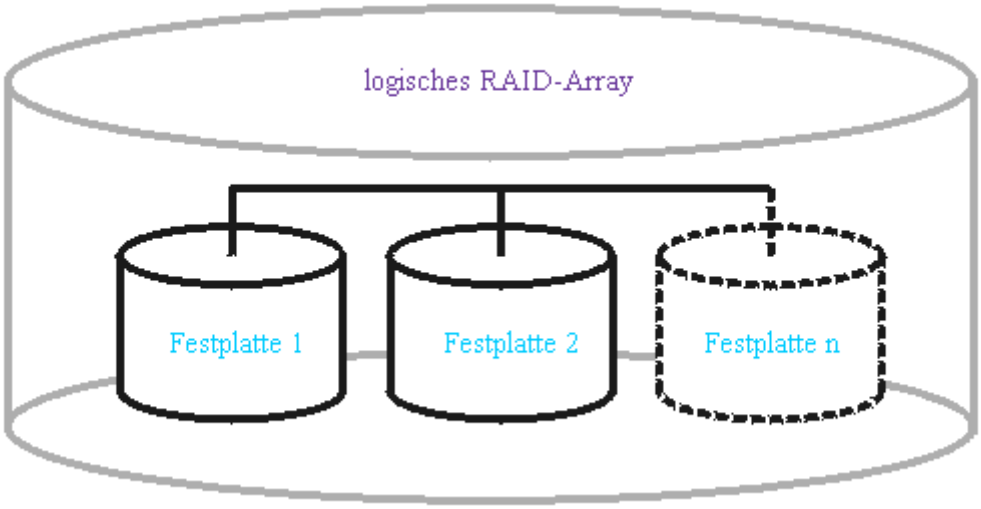

Abbildung 1: Schematischer Aufbau von RAID

RAID-Levels

Die einzelnen RAID-Levels wurden (fast) alle von der University of California, Berkeley, U.S.A., spezifiziert. Aus Platzgründen führe ich hier nur die RAID-Levels auf die von Linux und gängigen Hardwarekonfigurationen unterstützt werden.

Linear mode

- Zwei oder mehr Festplatten werden zu einer logischen Einheit zusammengefasst. Die Daten werden zuerst auf die erste Festplatte geschrieben bis die Speicherkapazität erschöpft ist. Anschließend wird die zweite Festplatte mit Daten belegt. Dieses Schema wird auch bei mehr als zwei Platten entsprechend angewandt.
- Es werden keine Redundanz-Informationen angelegt, d.h. fällt eine Festplatte aus, dann sind alle

Daten verloren.
- Die Lese- und Schreibperformance auf das Array ist normalerweise nicht besser als auf die einzelnen Festplatten. Nur wenn ein Benutzer von der ersten Platte und ein anderer Benutzer von der zweiten Platte ließt, kommt es zu einem Performancevorteil.

RAID-0 oder Data Stripping

- Bei RAID-0 werden wie beim Linear-Mode zwei oder mehr Festplatten zu einer logischen Einheit zusammengefasst. Der Unterschied ist, dass die Schreib- und Lesezugriffe parallel auf die einzelnen Platten verteilt werden. Da die Daten parallel auf beide Platten geschrieben werden, sind die Platten immer gleich belegt. Daraus resultiert, dass die Partitionsgrößen der einzelnen Platten exakt gleich groß sein müssen.
- Wie beim Linear-Mode werden hier auch keine Redundanz-Informationen angelegt, d.h. fällt eine Festplatte aus sind alle Daten verloren.
- Die Lese- und Schreibperformance ist aufgrund der Parallelität höher als bei einzelnen Festplatten. Wenn das Bussystem der Platten es zulässt, ist der Datendurchsatz (MB/s) für Lese- und Schreibzugriffe annähernd gleich *(Anzahl Platten)*(Durchsatz einer Platte).*

RAID-1 oder Drive Mirroring

- Das ist das erste RAID-Level das Redundanz-Informationen erzeugt und speichert. RAID-1 kann mit zwei oder mehr Datenfestplatten und keiner oder mehreren Ersatzplatten erzeugt werden. Dieser RAID-Mode spiegelt auf allen Datenfestplatten die gleichen Informationen. Dadurch müssen die Partitionen gleich groß sein.
- Fallen alle Platten bis auf eine aus, so sind die Daten trotzdem noch intakt. Wenn eine oder mehrere Ersatzfestplatten vorhanden sind und der Ausfall einer oder mehrerer Datenfestplatten vom IDE- oder SCSI-Controller erkannt wird, so startet automatisch die Reorganisation des Arrays.
- Die Leseperformance erhöht sich auf annähernd *(Anzahl Platten)*(Durchsatz einer Platte),* da die Lesezugriffe parallel erfolgen können. Die Schreibperformance ist geringfügig langsamer als auf eine Platte, da die CPU die Daten auf jede einzelnen Festplatte des Arrays schicken muss.

RAID-4 oder Block Striping mit Parity-Laufwerk

- Dieses RAID-Level wird recht selten verwendet. RAID-4 kann bei drei oder mehr Festplatten benutzt werden. Die Parity-Daten werden auf einer Festplatte gespeichert, die eigentlichen Daten werden auf den restlichen Platten des Arrays nach dem RAID-0 Verfahren gespeichert. Da eine Platte für die Parity-Daten verwendet wird berechnet sich die Speicherkapazität des RAID-Arrays *(Anzahl Platten - 1)*(kleinste Platte im Array).* Wie bei RAID-0 sollten die Partition gleich groß sein.
- Fällt eine Platte aus, so können die Daten mit Hilfe der Parity-Daten rekonstruiert werden. Fallen zwei Platten gleichzeitig aus, so sind die Daten verloren.
- Der Grund weshalb dieses Level nicht öfters verwendet wird ist der, dass die Parity-Informationen auf einer extra Festplatte gespeichert werden. Diese Parity-Informationen müssen bei jedem Schreibvorgang aktualisiert werden. Dies macht das ganze RAID-System maximal genau gleich schnell wie die Parity-Festplatte. Hat man aber mehrere langsame und eine schnelle Festplatte, so kann dieses RAID-Level sehr nützlich sein.

RAID-5 oder Block Striping mit verteilter Parity

- Dieses RAID-Level wird eingesetzt wenn um eine große Anzahl von Platten zusammenfügen und gleichzeitig eine hohe Ausfallsicherheit erreichen möchte. RAID-5 kann bei drei oder mehr Datenfestplatten und keiner oder mehreren Ersatzfestplatten benutzt werden. Die Speicherkapazität des Arrays berechnet sich gleich wie bei RAID-4. Der große Unterschied

zwischen RAID-5 und RAID-4 ist, dass man den Flaschenhals Parity-Festplatte dadurch umgeht, dass jede Festplatte neben den eigentlichen Daten auch Parity-Daten speichert.
- Fällt eine Platte aus, so sind die Daten dank der Parity-Informationen intakt. Ist eine Ersatzfestplatte vorhanden so wird die Rekonstruktion des Arrays sofort nach Feststellen des Plattenausfalls begonnen. Fallen gleichzeitig zwei oder mehr Festplatten aus, so sind die Daten verloren.
- Die Lese- und Schreibperformance ist bei RAID-5 deutlich höher als bei einer einzelnen Platte aus dem Array. Es lässt sich aber kein genauer Wert festlegen, da dies von unterschiedlichen Faktoren abhängt.

RAID-10 oder Mirrored Striping Array

- Dieses RAID-Level ist eine Kombination aus RAID-1 und RAID-0. Üblicherweise werden hier 4 Festplatten verwendet, da RAID-10 aus 2 RAID-1 Arrays besteht, die dann zu einem RAID-0 System zusammengefügt werden. Es sind aber auch andere Anordnungen von Festplatten denkbar.
- Fällt eine Platte aus, so sind die Daten dank der redundanten Speicherung intakt. Ist eine Ersatzfestplatte vorhanden so wird die Reorganisation des Arrays gestartet. Fallen mehr als eine Platte gleichzeitig aus, so kommt es darauf an das die einzelnen RAID-1 Arrays funktionstüchtig bleiben. Dann besteht für die Daten keine Gefahr. Ist eines der RAID-1 Arrays nicht mehr funktionstüchtig, so sind die Daten verloren.
- Die Lese- und Schreibperformance ist bei RAID-10 deutlich höher wie bei einer einzelnen Platte aus dem Array. Es lässt sich aber kein genauer Wert festlegen, da dies von unterschiedlichen Faktoren abhängt. Da keine Parity-Informationen berechnet werden müssen, sind auch die Schreibzugriffe sehr schnell.

Anwendungsgebiet

Sie fragen sich sicher warum es so viele RAID-Levels gibt? Dies hat den Grund, dass man für jedes spezielle Problem eine passende Lösung anbieten möchte. Aus diesem Grund gibt es auch nicht "die ultimative RAID-Lösung". Je nach Problemstellung wird man ein anderes RAID-Level als Lösung in Betracht ziehen. RAID wird überall dort eingesetzt wo ein erhöhtes Maß an Datensicherheit verlangt wird oder der Datendurchsatz deutlich erhöht werden muss.
Normalerweise werden Hardware-Controller für die Verwaltung des RAID-Systems verwendet. Da Hardware im Gegensatz zu GPL-Software immer deutlich teurer ist, gibt es unter Linux eine Software-RAID Lösung. Hierbei wird versucht die Funktionen des Hardware-Controllers durch Software nachzubilden.

RAID Setup

Software

Damit man unter Linux Software-RAID einsetzen kann, benötigt man folgende Dinge:

- Einen Kernel in der Version 2.0.x oder 2.2.x (wobei einige 2.2.x Versionen nicht unterstützt werden, so z.B. 2.2.5, 2.2.8 und 2.2.9).
- Das RAID Patch für den Kernel
- Die eigentlichen raidtools

Den Linux-Kernel kann man von [1] beziehen. Das RAID Patch und die raidtools bekommt man von [2]. In den aktuellen Kernelversionen ist eine Software-RAID Unterstützung integriert. Diese basiert aber auf den veralteten mdutils. Die mdutils werden nicht mehr weiterentwickelt und bieten weniger Features als die raidtools, weshalb man die mdutils nicht mehr einsetzten sollte. In der Kernelversion 2.2.12 sollte eigentlich der alte RAID-Code durch den neuen ersetzt werden. Leider hat sich Linus

Torvalds dazu entschieden, den RAID-Code erst in einer späteren Version in den Kernel aufzunehmen. Ich lese die Linux-RAID Mailinglist [3] mittlerweile etwas über ein Jahr mit und habe noch nie Berichte gelesen in denen die RAID-Software für einen Ausfall des Arrays verantwortlich gemacht werden konnte. Die Software ist trotz ihres frühen Entwicklungsstadiums stabil. Zur Zeit ist die Version 0145 des RAID Patches und die Version 0.90 der raidtools aktuell.

Den entpackten Kernel Patch kopiert man nach */usr/src* und spielt dann den Patch in den Quellcode mit

```
# patch -p0 < raid0145-19990824-2.2.11
```

ein. Damit RAID verwendet werden kann muss man in der Kernelkonfiguration (unter "Block devices") die entsprechenden RAID-Einträge aktivieren. Damit die Änderungen an der Konfiguration wirksam werden, muss man den Kernel neu kompilieren und installieren. Nach einem Reboot sollte die Datei */proc/mdstat* vorhanden sein. In dieser Datei werden alle Informationen über das RAID-System angezeigt. Die Datei kann man mit

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

ansehen.

Die raidtools sollte man in ein Verzeichnis entpacken. Das GNU autoconf System ruft man mit dem Befehl

```
# ./configure
```

auf. Dieses Programm sollte sich ohne eine Fehlermeldung beenden. Mit einem

```
# make all
```

kompiliert man die raidtools und mit

```
# make install
```

werden sie installiert.

Konfigurationsdatei */etc/raidtab*

Die raidtools werden über die Konfigurationsdatei */etc/raidtab* konfiguriert. Diese Datei darf nach einem erfolgreichen Erstellen eines RAID-Systems nicht gelöscht werden, da sie im Falle einer Reorganisation des Arrays benötigt wird. Auf die Konfigurationsdatei wird in den folgenden Abschnitten Bezug genommen. Deshalb soll an dieser Stelle etwas Licht zu denn einzelnen Parametern der Konfigurationsdatei gebracht werden. Die meisten verwendeten Parameter (raiddev, raidlevel, device, raid-disk, spare-disk, ...) sind selbsterklärend und aus Platzgründen möchte ich auf diese nicht eingehen. Auf die sich nicht selbsterklärenden Parameter möchte ich an dieser Stelle etwas genauer eingehen.

persistent-superblock
Der "Persistent-Superblock" wird bei der Erstellung des RAID-Arrays an den Anfang der teilnehmenden Festplatten geschrieben. Der "Superblock" enthält die Konfigurationsinformationen des RAID-Systems. Er wird zur automatischen RAID-Erkennung des Kernels und zum Booten eines RAID-System benötigt.

chunk-size
Möchte man beispielsweise auf ein RAID-0-Array mit zwei Festplatten ein Byte schreiben, so würden nach unserer Definition auf jede Festplatte genau 4 Bits geschrieben werden. Da heutige Hardware dies nicht unterstützt, wird mit dem Parameter Chunk-Size die kleinste "atomische" Einheit, die eine

Festplatte speichern kann, festgelegt. Der Parameter Chunk-Size kann bei jedem RAID-Level, außer dem "Linear Mode", verwendet werden. Der Wert des Parameters gibt immer eine Zahl in KByte an. Um die optimale Performance aus dem RAID-System herausholen zu können sollte man mit diesem Parameter experimentieren. Je nach Größe der Dateien die man hauptsächlich auf dem Array speichert sollte man die Chunk-Size wählen.

parity-algorithm
Der Parameter Parity-Algorithm ist nur bei RAID-5 zulässig. Er definiert den Algorhythmus der festlegt, an welchen Platz die Partity-Informationen gespeichert werden. *left-symmetric* ist der Standardwert bei Festplatten.

Konfiguration von Linear mode

Wenn man zwei oder mehr Festplatten hat (die jeweilige Speicherkapazität spielt hier keine Rolle) und diese zu einer logischen Einheit zusammenfügen möchte, muss die Konfigurationsdatei */etc/raidtab* wie folgt aussehen:

```
raiddev               /dev/md0
raid-level            linear
nr-raid-disks         2
persistent-superblock 1
device                /dev/sdb1
raid-disk             0
device                /dev/sdc1
raid-disk             1
```

Wie oben schon erwähnt können hier keine Ersatzfestplatten definiert werden, d.h. fällt eine Platte aus, sind alle Daten verloren.
Der Aufruf von

```
# mkraid /dev/md0
```

schreibt die Superblocks, erzeugt und startet das Array. Nachdem das Array erfolgreich erzeugt und gestartet wurde sollte dies in der Datei */proc/mdstat* angezeigt werden. Mit dem Befehl

```
# mke2fs /dev/md0
```

wird das Array formatiert. Um das Array in das Verzeichnis */linear* einzuhängen muss man folgenden Befehl aufrufen:

```
# mount -t ext2 /dev/md0 /linear
```

Wie man sieht kann man mit einem RAID-System genauso umgehen wie mit anderen Speichermedien.

Konfiguration von RAID-0

Die */etc/raidtab* für RAID-0 muss wie folgt aussehen:

```
raiddev               /dev/md0
raid-level            0
nr-raid-disks         2
persistent-superblock 1
chunk-size            4
device                /dev/sdb1
raid-disk             0
device                /dev/sdc1
raid-disk             1
```

Auch hier lassen sich keine Ersatzfestplatten definieren, d.h. fällt eine Platte aus, sind die Daten des ganzen Arrays verloren.

Wie man das Array anlegt, formatiert und mountet können Sie oben im Abschnitt *Konfiguration von Linear mode* nachlesen.

Konfiguration von RAID-1

Wenn man Daten auf zwei oder mehr Festplatten spiegeln möchte, muss die Speicherkapazität der einzelnen Partitionen gleich groß sein. Hier sieht man die Konfigurationsdatei */etc/raidtab* für RAID-1 ohne Ersatzfestplatten:

```
raiddev               /dev/md0
raid-level            1
nr-raid-disks         2
nr-spare-disks        0
chunk-size            4
persistent-superblock 1
device                /dev/sdb1
raid-disk             0
device                /dev/sdc1
raid-disk             1
```

Möchte man Ersatzfestplatten in das Array aufnehmen muss die Konfigurationsdatei wie folgt aussehen:

```
raiddev               /dev/md0
raid-level            1
nr-raid-disks         2
nr-spare-disks        1
chunk-size            4
persistent-superblock 1
device                /dev/sdb1
raid-disk             0
device                /dev/sdc1
raid-disk             1
device                /dev/sdd1
spare-disk            0
```

Wie man das Array anlegt, formatiert und mountet können Sie oben im Abschnitt *Konfiguration von Linear mode* nachlesen.

Konfiguration von RAID-4

Möchte man ein RAID-4-System ohne Ersatzfestplatten konfigurieren muss die Konfigurationsdatei */etc/raidtab* wie folgt aussehen:

```
raiddev               /dev/md0
raid-level            4
nr-raid-disks         4
nr-spare-disks        0
persistent-superblock 1
chunk-size            32
device                /dev/sdb1
raid-disk             0
device                /dev/sdc1
raid-disk             1
device                /dev/sdd1
raid-disk             2
device                /dev/sde1
raid-disk             3
```

Möchte man Ersatzfestplatten verwenden muss man die Konfigurationsdatei um folgende Zeilen ergänzen:

```
device                /dev/sdf1
spare-disk            0
```

Die Nummer hinter dem Eintrag *nr-spare-disks* muß dann natürlich auf *1* geändert werden.
Mit dem Aufruf

```
# mkraid /dev/md0
```

wird das Array initialisiert. Für RAID-4 und 5 hat das Programm mke2fs eine spezielle Option. Diese Option speichert ext2 spezifische Daten an einem besseren Platz auf dem RAID-System. Möchte man ein ext2-Dateisystem mit einer Blockgröße von 4KB anlegen, so muss der Aufruf von mke2fs bei dieser Konfiguration wie folgt aussehen:

```
# mke2fs -b 4096 -R stride=8 /dev/md0
```

Der Wert des Parameters *-R stride* ist das Ergebnis von Chunk-Size geteilt durch die Blockgröße des Dateisystems. Verwendet man diesen Parameter nicht, so verschenkt man einiges an Leistung des RAID-Systems.
Mounten lässt sich ein solche erstelltes Array wie üblich:

```
# mount -t ext2 /dev/md0 /raid4
```

Konfiguration von RAID-5

Damit man ein RAID-5-Array ohne Ersatzfestplatten erstellen kann muss die Konfigurationsdatei */etc/raidtab* folgende Einträge aufweisen:

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

Möchte man Ersatzfestplatten verwenden muss man die Konfigurationsdatei um folgende Zeilen ergänzen:

```
device                /dev/sdf1
spare-disk            0
```

Die Nummer hinter dem Eintrag *nr-spare-disks* muß dann natürlich auf *1* geändert werden.
Wie man das Array anlegt, formatiert und mountet können Sie oben im Abschnitt *Konfiguration von RAID-4* nachlesen.

Konfiguration von RAID-10

Um ein RAID-10-Array aufbauen zu können, muss man zuerst die RAID-1-Arrays erstellen. Wie man dies zu bewerkstelligen ist, steht im Abschnitt *Konfiguration von RAID-1*. Die einzelnen RAID-1-Arrays werden mit RAID-0 zu einer logischen Einheit zusammengefasst. In der Konfigurationsdatei muss man statt der Partitionen die Gerätenamen der RAID-1-Arrays angeben. Im Abschnitt *Konfiguration von RAID-0* sieht man wie eine Konfigurationsdatei für RAID-0 aussehen muss.

Automatische Erkennung

Wenn die automatische Erkennung aktiviert und konfiguriert ist, erkennt der Kernel das RAID-System automatisch und startet dieses. Damit die automatische Erkennung funktioniert muss dies im Kernel aktiviert sein und ein RAID-Array mit einem Persistent-Superblock vorhanden sein. Der Kernel erkennt die RAID-Partitionen an dem speziellen Partitionstyp *0xfd*. Den Partitionstyp kann man mit *fdisk* festlegen:

```
# fdisk /dev/sda
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): fd
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Nach einem Reboot sollte der Kernel das RAID-System erkennen und automatisch starten. Automatisch gestartete RAID-Arrays werden beim Herunterfahren des Rechners auch wieder automatisch gestoppt. Bei dieser Lösung benötigt man keine Startskripte die das RAID-System verwalten. Damit keine Probleme auftreten sollte man diese Startskripte löschen.

Rekonstruktion

Wenn es während des Betriebs des RAID-System zum Ausfall einer Festplatte kommt, startet die Rekonstruktion sobald der Fehler von der Software (IDE-, SCSI-Treiber) erkannt wurde und eine Ersatzfestplatte vorhanden ist. Ist keine Ersatzfestplatte vorhanden, so sollte man den Rechner herunterfahren (Hot-Swapping funktioniert bei IDE- und den meisten SCSI-Treibern nicht). Die defekte Festplatte durch eine funktionierende ersetzten und den Computer wieder anschalten. Die Software erkennt dann automatisch, dass eine neue Festplatte vorhanden ist und beginnt mit der Rekonstruktion der Daten. Bei der Festplatte die neu ins RAID-Array genommen wurde muss man darauf achten, dass sie schon richtig partitioniert ist und der Partition-Typ richtig gesetzt ist. Wie man hier klar sehen kann ist es von Vorteil eine Ersatzfestplatte von Anfang an ins Array zu integrieren. Natürlich funktioniert die Rekonstruktion nur bei den RAID-Levels, bei denen Redundanz-Daten gespeichert werden.

Weiteres

Sollte man nachdem Lesen dieses Artikel noch weitergehende Fragen haben, so lohnt es sich auf jeden Fall die "The Software-RAID HOWTO" von Jakob Østergaard zu lesen. Man findet die HOWTO unter [4].
Ich möchte mich an dieser Stelle bei all denen Bedanken die sich an der Entwicklung von Software-RAID beteiligen.

Literatur und Bezugsquellen

[1] ftp://ftp.de.kernel.org/pub/linux/kernel
[2] ftp://ftp.de.kernel.org/pub/linux/daemons/raid/alpha
[3] linux-raid@vger.rutgers.edu
[4] http://ostenfeld.dk/~jakob/Software-RAID.HOWTO

Der Autor

Thomas King ist Student der Wirtschaftsinformatik und beschäftigt sich viel mit Linux. Seine Leidenschaft zu Linux hat vor über drei Jahren begonnen und wächst noch immer jeden Tag. :-)
Zu erreichen ist er unter king@t-king.de oder http://www.t-king.de/.