SOFTWARE-RAID1 UNTER SUSE 6.0 BIS 6.2

Versionen:

Autor: Thomas King (king@t-king.de)
V1.0
V1.1
V1.2
V1.3
V1.4
V1.5
V1.6
V1.61
V1.62
V1.63
V1.64
V1.65 14.01.2000 (Bug fixed)

Copyright:

Dieses Dokument darf gemäß der GPL Lizenz verbreitet werden. Das Copyright liegt bei Thomas King.

Danksagung:

- Winfried Forster (Winfried.Forster@tsp-online.de)
- Fritz Spitzer (f.spitzer@geosystems.de)
- LUGS (Linux Users Group Switzerland)
- Alle die mich mit Informationen versorgt haben

Aktuelle Version:

Die aktuellste Version dieses Dokuments bekommt man unter: http://www.t-king.de/linux/raid1.html. Mit dieser Version wurde die „Entwicklung“ dieser Dokumentation eingestellt. Die Entwicklung der mdutils wurde einstellt und statt dessen gibt es das neue Software-Raid Projekt „raidtools“. Unter http://www.t-king.de/linux/software.html habe ich meinen Linux-Magazin Artikel (12/99) zu Software-Raid mit raidtools veröffentlicht.

Vorab:

Dies ist ein kleine Anleitung die die Konfiguration eines Software-Raid1-Systems auf Basis einer SuSE 6.0 bis 6.1 beschreibt. Die Linux-Distribution von SuSE ist unter http://www.suse.de/ erhältlich. Natürlich läßt sich diese Anleitung auch für andere Distributionen anwenden. Die Anleitung soll nicht oder nur am Rande erklären was Raid („Redundant Array of Inexpensive Disks“) genau ist und welche Level es davon gibt. Auch werde ich nur am Rande auf die Vor- und Nachteile von Raid1 (auch mirroring genannt) eingehen. Vielmehr möchte ich hier zeigen wie einfach man Software-Raid1 mit SuSE 6.0 bis 6.2 konfigurieren kann. Die Anleitung soll auch Gedankenanstoß und Einstieg sein.
Die meisten Informationen zu dieser Anleitung habe ich von folgenden HOWTO‘s

- Software-RAID HOWTO von Linas Vepstas (linas@linas.org) v0.51
- Root RAID HOWTO cookbook von Michael A. Robinton (michael@bzs.org) v1.07

und aus sonstiger Literatur (man-pages, mailinglists, Newsgoups, c't, ix, ...). Die HOWTO's kann man unter http://metalab.unc.edu/mdw/HOWTO beziehen. Aber wie immer klingen die Sachen in der Literatur einfacher wie sie dann in Praxis sind. Deshalb ist meine Vorgehensweise für die Konfiguration mehr durch das „try and error"-Prinzip geprägt gewesen. Ich möchte an dieser Stelle auch sehr deutlich darauf hinweisen, das meine Vorgehensweise sicher nicht das Optimum darstellt und das man noch einiges daran verbessern kann. Ich bitte um Verbesserungsvorschläge! Auch sei an dieser Stelle der Hinweis angebracht, das die Software, die für das Raid-System benötigt wird, noch Alpha-Stadium ist. Aber ich hatte in der ganzen Zeit keine Ausfälle des Servers die auf die Raid-Software zurückzuführen waren. Viel mehr lag es an Fehlern des Serveradministrators.

Vor- Nachteile:

Vorteile: Raid1 vs. Non-Raid

- Ausfallsicherheit bei Harddiskfehlern (nicht aber bei Administrationsfehlern oder Stromausfall)

Nachteile: Raid1 vs. Non-Raid

- Es wird der n-fache Harddiskspeicherplatz benötigt da die Daten gleichzeitig auf n Partitionen gespeichert werden
- Etwas langsamere Datenübertragungsraten, da die Raid-Software zur Verarbeitung etwas Zeit braucht
- Man braucht einen ausreichend gut ausgestatteten Server, da die Raid-Software etwas Rechenleistung schluckt (hier PII-333Mhz – 192MB RAM)

Konfiguration:

Bei SuSE 6.0 wird standardmäßig der Kernel 2.0.36 , bei SuSE 6.1 ist der Kernel 2.2.5 und bei SuSE 6.2 der Kernel 2.2.10 (hier wird aber der 2.2.11 verwendet, was von der Konfiguration keinen Unterschied macht) mitgeliefert. Auf diese bezieht sich die folgende Konfiguration. Man muß einen neuen Kernel kompilieren der folgende „Treiber" und „Optionen" aktiviert hat:

- Multiple devices driver support
- RAID-1 (mirroring) mode.
- Treiberunterstützung für die Schnittstellen der Harddisks (SCSI, E-DIE, ...) – aber darüber gibt das Handbuch Auskunft

Folgendes Softwarepaket muß installiert sein: mdutils (aus der Serie ap).
Der Server braucht (mindestens) 2 Harddisks die über die gleich Schnittstelle angesprochen werden (SCSI, E-IDE, ...) und gleich große Partitionen aufweisen. Hier haben wir zwei Seagate Barracuda (ST39173W) mit jeweils 9,1GB die an einem Adaptec 2940UW Controller hängen. (Um die Datensicherheit zu erhöhen kann man natürlich hingehen und jede Festplatte an ein eigenes Kabel und an einen eigenen Controller hängen.)
Die Harddisks sind hier wie folgt partitioniert:

Sda:	sdb:
/ (600MB) [sda1]	/ (600MB) [sdb1]
/internet (4GB) [sda2]	/internet (4GB) [sdb2]
/samba (4,1GB) [sda3]	/samba (4,1GB) [sdb3]

Auf / der Harddisk sda habe ich das ganze eigentliche Linux installiert. Bei uns soll der Server einzig und alleine als File-Server fungieren. Die wichtigen und sicherheitstechnisch relevanten Daten liegen bei uns in den Verzeichnissen internet und samba; diese Partitionen sollen über Raid1 geschützt werden. Die Partition sda1 wird über cron alle 24 Stunden komprimiert

und auf die Partition sdb1 kopiert. Da die Daten auf der sda1-Partition kaum verändert werden und nicht sehr wichtig sind, verwenden wir kein Raid1 sondern kopieren diese und machen zusätzlich noch alle 24 Stunden ein Backup (auf Streamer). Aber rein technisch gesehen ist es möglich auch Root über Raid1 zu schützen (s. HOWTO). Bei der Installation von Linux sollte man keine andere Partitionen außer sda1 mounten. Natürlich ist diese Aufteilung der Platten reine exemplarisch und jeder kann dies seinen Anforderungen anpassen!
Es ist leider nicht möglich eine Swap-Partition auf ein Raid-System aufzusetzen, da ansonsten der Rechner nach einiger Zeit mit einem Kernel Panic abstützt (der „Fehler“ liegt hier im Swap-Code).
Die Partitionen sda2 und sdb2 sowie sda3 und sdb3 sollen zu je einer logischen Einheit zusammengefaßt werden. Dies erfolgt durch den Befehl *mdadd /dev/md0 /dev/sda2 /dev/sdb2* und *mdadd /dev/md1 /dev/sda3 /dev/sdb3*. Danach müssen wir die Datei /etc/mdtab durch folgenden Befehl erzeugen *mdcreate raid1 /dev/md0 /dev/sda2 /dev/sdb2* und *mdcreate raid1 /dev/md1 /dev/sda3 /dev/sdb3*. Die Datei /etc/mdtab wird zum mounten der logischen Devices benötigt. Damit das eigentliche Raid1 konfiguriert werden kann muß noch eine Konfigurationsdatei (etc/raid1.conf) angelegt werden. Wie die Datei auszusehen hat ist im Anhang kurz erklärt. Mit dem Befehl *mkraid -f /etc/raid1.conf* wird dann die Konfiguration vervollständigt. Bis dieser Befehl abgearbeitet ist kann einige Zeit vergehen (bei unserem System etwa 30 min.). Danach sollte man das System neu booten. Beim Hochfahren bekommt man dann schon einige Informationen über das Raid-System. Nun muß man nur noch die „neuen“ Partitionen formatieren (Befehl *mke2fs /dev/md0* und *mke2fs /dev/md1*, wenn man das ext2-Filesystem verwenden möchte). Auch dieser Vorgang hat hier einige Zeit in Anspruch genommen (wieder etwa 30 min.). Damit die neu formatieren „Partitionen“ beim nächsten Booten gemountet werden können, müssen sie noch in die Datei /etc/fstab eingetragen werden. Diese Datei ist auch im Anhang abgedruckt. Um hier das Optimale herauszuholen empfiehlt sich ein Blick in die man-pages.
Nach einem erneuten reboot sollte Ihnen dann das Raid1-System zur Verfügung stehen. Die Konfiguration wäre so weit abgeschlossen.

Fehlerbehebung:

Bugfix für SuSE 6.0 und 6.1:

Im Bootskript /sbin/init.d/boot hat sich im Bereich „maybe we use "Multiple devices". So initialize MD.“ [Zeile 34 ff.] ein Fehler eingeschlichen. Eigentlich ist das Skript unter anderem dafür zuständig das wenn der Befehl mdadd -ar nicht ausgeführt werden kann, das dann der Befehl /sbin/ckraid --fix /etc/raid1.conf ausgeführt wird. Danach sollte das Raid wieder hochkommen.
Der Fehler lag daran, das die Prüfung ob der Befehl mdadd -ar ausgeführt werden konnte nicht funktioniert. Man muß folgende Zeilen ändern:

```
#
# maybe we use "Multiple devices".  So initialize MD.
#
if test -f /etc/mdtab -a -x /sbin/mdadd ; then
    ECHO_RETURN=$rc_done_up
    echo "Initializing Multiple Devices..."
    /sbin/mdadd -ar && MDADD_RETURN=0 || MDADD_RETURN=1
    if test $MDADD_RETURN != 0 ; then
        if test  -x /sbin/ckraid ; then
            echo "Initializing Multiple Devices failed.  Trying to recover
it..."
        for i in /etc/raid?.conf ; do
                /sbin/ckraid --fix $i
            done
```

```
            /sbin/mdadd -ar || ECHO_RETURN=$rc_failed_up
        else
            ECHO_RETURN=$rc_failed_up
        fi
    fi
    echo -e "$ECHO_RETURN"
fi
```

Arbeitet mein Raid-System korrekt:

Mit dem Befehl *cat /proc/mdstat* sieht man ob das Raid-System läuft. Hier sieht die Ausgabe so aus:

```
Personalities : [3 raid1]
read_ahead 120 sectors
md0 : active raid1 sda2 sdb2 4096448 blocks [2/2] [UU]
md1 : active raid1 sda3 sdb3 4168768 blocks [2/2] [UU]
md2 : inactive
md3 : inactive
```

Mit dem Befehl *dmesg* kann man nochmals die Bootmessages anschauen. Hier sieht dies wie folgend aus (nur Raid relevante Messages):

```
...
md driver 0.36.3 MAX_MD_DEV=4, MAX_REAL=8
raid1 personality registered
...
REGISTER_DEV sda2 to md0 done
REGISTER_DEV sdb2 to md0 done
raid1: device 08:02 operational as mirror 0
raid1: device 08:12 operational as mirror 1
raid1: raid set 09:00 active with 2 out of 2 mirrors
md: updating raid superblock on device 08:02, sb_offset == 4096448
md: updating raid superblock on device 08:12, sb_offset == 4096448
REGISTER_DEV sda3 to md1 done
REGISTER_DEV sdb3 to md1 done
raid1: device 08:03 operational as mirror 0
raid1: device 08:13 operational as mirror 1
raid1: raid set 09:01 active with 2 out of 2 mirrors
md: updating raid superblock on device 08:03, sb_offset == 4168768
md: updating raid superblock on device 08:13, sb_offset == 4168768
```

Wenn es zu Fehlern kommt werden diese normalerweise im Syslog „mitgeloggt“. Bei SuSE werden die Syslog-Messages in der Datei /var/log/messages gespeichert. Mit dem Befehl *tail -f /var/log/messages* sieht man die neusten Einträge in die Datei immer gleich.

Raid1 nicht sauber „ungemountet“:

Es gibt einige Situationen in den die Raid1-Devices nicht „ungemountet“ werden können:

- Der Rechner wird ohne vorher heruntergefahren worden zu sein ausgeschaltet (Stromausfall, Versehentlich ausgeschaltet, ...)
- Der Rechner ist abgestürzt (Kernel-Panic, ...)
- ...

Beim nächsten Hochfahren des System erkennt mdrun das die Daten auf den Harddisks nicht synchron sind. Es wird eine Fehlermeldung ausgegeben und vom root verlangt das es sich selber um die Behebung des Fehlers kümmert. Man sollte sich deshalb als root einloggen. Mit *mkraid --fix-superblock -f /etc/raid1.conf* werden die Daten wieder synchronisiert. Danach muß man das System neu booten und der Fehler sollte behoben sein.

Eine Harddisk muß ausgewechselt werden:

Dies kann viele Ursachen habe:

- Haedcrash
- Defekt in der Mechanik
- ...

Damit das Raid-System wieder ordentlich arbeiten kann muß man die defekte Harddisk ausgewechselt werden. Damit die vorhandenen Daten auf der funktionstüchtigen Harddisk gesichert werden können und in das ggf. neue Raid-System zurückgespielt werden können geht man am besten wie folgt vor (es gibt zwei Möglichkeiten):

1. Linux von einer Diskette booten. Von den Daten auf der funktionstüchtigen Harddisk ein Backup machen (Streamer, Dat, ...), denn Vorsicht schadet nie. Mit dem Befehl *dd if=/dev/sda2 of=/dev/sdb2* werden dann die Daten von der funktionstüchtigen Harddisk/Partition (hier sda2) auf die neue Harddisk/Partition überspielt. Danach noch die Raid-Superblocks mit *mkraid -f /etc/raid1.conf --only-superblock* neu schreiben. Nach einem Reboot sollte das Raid-System wieder einwandfrei funktionieren.
2. Linux von einer Diskette booten. Von den Daten auf der funktionstüchtigen Harddisk ein Backup machen (Streamer, Dat, ...), denn Vorsicht schadet nie. Mit dem Befehl *ckraid /etc/raid1.conf --fix --force-source /dev/sda2* werden dann die Daten von der funktionstüchtigen Hardisk/Partition (hier sda2) auf die neue Harddisk/Partition überspielt und die Raidstrukturen werden automatisch angelegt. Nach einem Reboot sollte das Raid-System wieder einwandfrei funktionieren.

Cron-Einträge:

Wie oben schon erwähnt ist die eigentliche Linux-Partition nicht über Raid1 gesichert. Da diese Daten nicht von einer besonderen Wichtigkeit sind und sich kaum ändern werden sie nur alle 24 Stunden gesichert. Nach einigem herumexperimentieren hat sich bei uns folgendes bewährt: Über cron-Eintäge werden ausgewählte Verzeichnisse komprimieren und dann gesichert. Der Vorteil bei dieser Methode ist das man mehrere „Versionen“ der Sicherung speichern kann. Sollte z.B. nach einem Software-Update ein Problem auftreten hat man immer noch die Sicherungen von den vergangenen Tagen (hier werden immer die letzten drei Tage vorgehalten).
Auszug aus der Cron-Tabelle:

```
0     2     *     *     *     rm /sicherung/usr3.tgz; cp /sicherung/usr2.tgz
/sicherung/usr3.tgz; cp /sicherung/usr1.tgz /sicherung/usr2.tgz; mv
/sicherung/usr.tgz /sicherung/usr1.tgz
0     3     *     *     *     tar cvfzlp /sicherung/usr.tgz /usr/*
...
```

Aus Platzgründen habe ich nicht die ganze Tabelle abgedruckt. Da sich die Aufrufe für die anderen Verzeichnisse wiederholen habe ich sie hier nicht zusätzlich aufgelistet.

Nachtrag:

Die SuSE 6.0 bis 6.2 vereinfachen dem Administrator sehr das Einrichten des Raid1-Systems, da man nichts an den Bootscripten ändern muß. Dies machen SuSE 6.0 bis 6.2 von Haus aus. Aber genau hier liegt auch ein Problem. Der Administrator weiß beim Hoch- und Herunterfahren nicht genau was das System macht. Das ist nicht weiter schlimm solange alles ohne Probleme läuft, aber was ist wenn dies nicht mehr der Fall ist? Deshalb empfiehlt es sich die Bootscripte von Hand mal durchzugehen und ggf. eigene Wünsche einzubauen (s. HOWTO‘s).
Auch sollte man einige Versuche unternehmen das Raid-System in Streßsituationen zu bringen (Rechner ohne Herunterfahren rebooten, Kopier- und Schreibvorgänge abbrechen, größe-

re Datenbestände verschieben, ...). Einige dieser Maßnahmen sind natürlich nicht gerade die feine englische Art. Aber der Administrator sollte mit den Verhaltensweisen des Systems in Extremlagen vertraut sein und wissen wie er korrigierend eingreifen kann.
Mit cat
Trotz der erhöhten Datensicherheit durch das Raid1-System sollte man es nicht vernachlässigen regelmäßig Backups zumachen. Auch ein Raid1-System kann einen totalen Datenverlust erleiden (Administrationsfehler, Softwarefehler, Hardwarefehler, ...).

Benchmark:

Ich möchte hier gar nicht allzu viel dazu sagen, ich verweise lieber auf schon vorhandene Berichte:

- Leistungsvergleich zwischen Soft- und Hardware-Raid:
  IX 4/99 Seite 112 ff. (Soft- und Hardware-RAID unter Linux - Schutzbrief)
- Benchmarkangaben eines Beispielsystems:
  http://www.Linux-Consulting.com/Raid/Docs/raid_benchmark.paul.txt
- Benchmarksoftware (zum eigenen Test):
  Bonnie: www.spin.ch/~tpo/bench
- Vergleich zwischen Software-Raid 0, 1 und 5:
  http://plasma-gate.weizmann.ac.il/~fnevgeny/tmp/raid.gif

Weitere Entwicklung:

Die Entwicklung der mdutils wurde eingestellt. Dafür gibt es eine „neue“ Entwicklung. Sie nennt sich raidtools und befindet sich im Moment bei der Versionsnummer 0.90 (http://www.de.kernel.org/pub/linux/daemons/raid/alpha/). Leider sind die raidtools zu den mdutils inkompatibel, d.h. die ein vorhandenes Raid-System (auf Basis von mdutils) kann nicht mit den neuen raidtools „weiterarbeiten“. Das Raid-System muß reorganisiert werden. Aber die neuen raidtools haben sehr vielversprechende neue Ansätze (hot-swap, spare-disks, ...).
Leider wurde mit der SuSE 6.2 noch die alten mdutils ausgeliefert. Ab der Kernel Version 2.2.12 soll der alte Raid-Code auf Basis der mdutils durch den neuen Code der raidtools ersetzt werden.

Anhang:

mdtab:

```
# mdtab entry for /dev/md0
/dev/md0    raid1,4k,0,625c5230     /dev/sda2 /dev/sdb2
# mdtab entry for /dev/md1
/dev/md1    raid1,4k,0,44634bac     /dev/sda3 /dev/sdb3
```

raid1.conf:

```
#RAID-Bereich md0 (/internet)
raiddev     /dev/md0
raid-level  1
nr-raid-disks     2
nr-spare-disks    0
device /dev/sda2
raid-disk   0
device /dev/sdb2
raid-disk   1
```

```
#RAID-Bereich md1 (/samba)
raiddev /dev/md1
raid-level  1
nr-raid-disks     2
nr-spare-disks    0
device /dev/sda3
raid-disk   0
device /dev/sdb3
raid-disk   1
```

fstab:

```
/dev/sda1   /           ext2        defaults          1   1
/dev/scd0   /cdrom      iso9660     ro,noauto,user    0   0
/dev/fd0    /floppy     auto        noauto,user       0   0
proc        /proc       proc        defaults          0   0
# End of YaST-generated fstab lines
# Raid-Devices
/dev/md0    /internet   ext2        defaults          1   2
/dev/md1    /samba      ext2        defaults          1   2
```