

# Troubleshooting

Diese Seite widmet sich der Beseitigung von Problemen aller Art am Server oder im Netzwerk. Dabei geht es nicht alleine um den invis-Server. Die hier beschriebenen Mittel und Wege dürfte auch für andere Linux-Admins von Interesse sein.

## Datenträgerprobleme

### RAID Verbund defekt

#### RAID5, die Erste

Der Ausfall einer Festplatte ist dank RAID-Systemen in aller Regel zu verschmerzen. Das Handling von Linux Software-RAID Verbänden ist meist nicht weiter kompliziert. Trotzdem kann es zu Zuständen kommen, die einem schon mal den Angstschweiss auf die Stirn treiben können. Eine entsprechende Situation durfte ich bereits Live erleben. Vorausgegangen war ein Versehen. Aus einem RAID5-Verbund mit insgesamt 6 Festplatten wurde versehentlich eine vollkommen intakte Platte im Betrieb herausgezogen. Eigentlich kein Problem. Platte wieder einstecken und die RAID-Partition dem Verbund wieder hinzufügen:

```
invis:~ # mdadm /dev/md1 -a /dev/sdc1
```

(Die Festplatten, RAID und Partitionsbezeichnungen müssen natürlich an die eigenen Gegebenheiten angepasst werden.)

Danach kontrollieren, ob ein RAID-Resync läuft:

```
invis:~ # cat /proc/mdstat
```

In der Ausgabe sollte der Fortschritt der Synchronisation angezeigt werden. Im besonderen Fall hier war der Rebuild allerdings schon nach wenigen Sekunden scheinbar erfolgreich abgeschlossen (Bei 4TB Platten dauert ein vollständiger Resync normalerweise Stunden.). Zunächst habe ich dem wenig Bedeutung beigemessen, da erstens die Platte nur wenige Sekunden aus dem RAID entfernt war und der Server wegen Wartungsarbeiten gerade nichts zu tun hatte. Es also wenig Zugriffe auf den RAID-Verbund gab.

Die böse Überraschung folgte dann etwa 1 bis 2 Stunden später. Auf dem RAID Verbund liefen einige virtuelle Maschinen, die sich zunehmend sonderbar verhielten. Es war auch nicht mehr möglich diese geordnet herunterzufahren. Nach kurzer Zeit hatte der Linux-Kernel des Servers das zugrunde liegende Dateisystem (glücklicherweise) auf Read-Only umgeschaltet, was natürlich für das merkwürdige Verhalten der VMs mitverantwortlich war. Der erste Reflex war ein Reboot des Servers, der allerdings mitten drin stecken blieb. Der betroffene RAID-Verbund stand nicht mehr zur Verfügung (inactive).

Der erste Versuch den Verbund wieder zum Leben zu erwecken endete damit das plötzlich der Ausfall zweier Platten gemeldet wurde, was bei einem RAID5 natürlich der Supergau wäre. Totalverlust!

In der Verzweiflung versucht man natürlich nach jedem Strohalme zu greifen. Ich habe das System über eine Rescue-Disk gebootet und dann einen zweiten Versuch unternommen den Verbund zu reaktivieren. Wichtig ist, dass er trotz Inaktivität zunächst gestoppt werden:

```
invis:~ # mdadm --stop /dev/md1
```

Ein automatisches Reassemblieren mit:

```
invis:~ # mdadm --assemble --scan
```

führte nicht zum Erfolg. Wer das testet, muss vor dem nächsten Schritt den Verbund wieder wie zuvor gezeigt stoppen.

Erst mit Gewalt unter Angabe aller beteiligten Platten klappte es:

```
invis:~ # mdadm --assemble --force /dev/md1 /dev/sdc1 /dev/sdd1 /dev/sde1 /dev/sdf1 /dev/sg1 /dev/sdh1
```

Wieder war zu beobachten, dass der Resync nach nur wenigen Sekunden abgeschlossen war. Misstrauen war angesagt. Zur endgültigen Beseitigung des Problems habe ich die betroffene Platte wieder aus dem laufenden Verbund entfernt und vollständig neu partitioniert. Erst danach wurde sie dem Verbund wieder hinzugefügt. Danach lief die Resynchronisation erwartungsgemäß einige Stunden.

**Achtung:** Sie müssen sich für diesen Schritt absolut sicher sein welche Platte das Problem verursacht hat. Ich kann mir vorstellen, dass es das finale Ende des RAID5 Verbundes wäre jetzt die falsche Platte aus dem RAID zu nehmen.

Entfernen der Platte (Partition) aus dem Verbund:

```
invis:~ # mdadm /dev/md1 -f /dev/sdc1
invis:~ # mdadm /dev/md1 -r /dev/sdc1
```

Zur Sicherheit ein paar Nuller auf den Anfang schreiben:

```
invis:~ # dd if=/dev/zero of=/dev/sdc
```

Dann mit **fdisk** neu partitionieren, am besten damit beginnen eine neue Partitionstabelle auf die Platte zu schreiben (fdisk Kommando „g“ für eine GPT-Tabelle).

Beim neuen Anlegen einer Partition (fdisk Kommando „n“) auf der Platte erkennen neuere fdisk-Versionen ggf. noch alte Strukturen wie etwa die RAID-Signatur und fragen, ob diese überschrieben werden können. Unbedingt bejahen! Nicht vergessen den Partitionstyp auf Linux-RAID zu setzen (fdisk-Kommando „t“ / Typ „29“ oder „fd“ - je nach fdisk Version).

Danach kann die Platte (Partition) wieder in den RAID-Verbund aufgenommen werden:

```
invis:~ # mdadm /dev/md1 -a /dev/sdc1
```

Kontrolle wieder mit:

```
invis:~ # cat /proc/mdstat
```

Jetzt sollte der Resync je nach Plattengröße einige Stunden dauern.

## RAID5, die Zweite

Auch dies ist ein realer Fall aus der Praxis. Vorausgegangen ist ein schweres Unwetter. Getroffen hat es Augsburg Ende Mai 2024, wenn ich richtig informiert bin, kam es durch Blitzschlag sogar zu schweren Schäden in einem Umspannwerk. Neben dem darauf folgenden Stromausfall, kann es im Netz sogar zu Spannungsspitzen kommen, die auch vom Überspannungsschutz einer USV nicht vollständig kompensiert werden können.

Das Resultat beim Kunden war eine regelrecht aus einem RAID5-Verbund heraus gedroschene Festplatte. Physisch zerstört. Zwar ließ sich die Platte irgend wie noch ansprechen, „*fdisk -l /dev/sda*“ beispielsweise zeigte, ohne Zeitverzögerung, noch die Partitionstabelle an. Allerdings resultierte jeder Versuch mit der Platte zu kommunizieren massive I/O-Fehler im System-Journal. Ein merkwürdiger Zustand irgendwo zwischen intakt und defekt, wobei letzteres eindeutig der Realität entsprach.

Bedingt durch den anschließenden Stromausfall, war der Server irgendwann aus. Nach dem Neustart, reagierte der RAID5-Verbund nicht mehr:

```
0:rescue:~ # cat /proc/mdstat
Personalities :
md0 : inactive sdb1[1](S) sdd1[4](S) sdc1[2](S)
      11721050052 blocks super 1.0

unused devices: <none>
```

In der Liste der beteiligten RAID-Devices fehlt „*/dev/sda1*“ und der Status steht auf „inactive“. Nicht unbedingt das, was man erwartet. Schließlich soll ein RAID ja genau davor schützen, also „degraded“, also beim Wegfall eines beteiligten Devices einfach wieder starten. Leider nicht in diesem Fall. Das fehlende Device, wurde irgendwie immer noch als beteiligt erkannt, aber wegen physischer Fehler auch nicht eingebunden. Henne und Ei Problem. Das RAID5 lies sich nicht „degradiert“ aktivieren, da auf „*/dev/sda1*“ keine Superblock gefunden werden konnte. Allerdings konnte „*/dev/sda1*“ auch nicht mittels:

```
0:rescue:~ # mdadm /dev/md0 -f /dev/sda1
0:rescue:~ # mdadm /dev/md0 -r /dev/sda1
```

als fehlerhaft gekennzeichnet und entfernt werden, da es ja ohnehin nicht mehr da war. Beängstigender Zustand, wirkte so, als müsste der Verbund unter Verlust aller Daten aufgegeben werden, obwohl ja noch 3 von 4 Devices intakt und vorhanden waren – also eigentlich kein Problem für einen RAID5-Verbund.

Ich spare mir hier jetzt die Beschreibung einiger Stunden hilflosen Herumgestochers im Internet.

Erwartet hätte ich, dass der Verbund degradiert einfach wieder startet. Es gibt allerdings eine Kernel-Bootoption, die dem Kernel untersagt einen RAID-Verbund in solchen Fällen einfach wieder zu starten.

Statt dessen bleibt der Verbund inaktiv, was vermutlich sogar dem Schutz der darauf liegenden Daten dient. Dem Admin treibt es allerdings in solchen Momenten den kalten Schweiß auf die Stirn. Im hier beschriebenen Fall war es sogar so, dass ich für eine Schulung in Lübeck engagiert und vertraglich gebunden und der Kunde, eine Anwaltskanzlei in Augsburg, handlungsunfähig war. Satte 800km Entfernung, kein Remote-Zugriff und vor Ort niemand mit adäquaten Kenntnissen. (Ist nicht böse gemeint!)

Beim Versuch irgendwie zu einem Remote-Zugriff zu kommen, gab es einige Hürden aus dem Weg zu räumen. Zwar startete das System aus eigener Kraft noch bis zu einem „emergency mode“ hoch, auch eine Anmeldung als Benutzer „root“ war möglich. Allerdings endeten alle Versuche auf dem System temporär eine Netzwerkschnittstelle zum Leben zu erwecken in unverständlichem Durcheinander. Also musste das System über einen USB-Stick in ein echtes Rescue-System gestartet werden. Den Server aber über einen USB-Stick zu starten erforderte gelinde gesagt ein paar Überredungskünste.

Gelingt das, muss im System eine Netzwerkschnittstelle etabliert, ein Default-Gateway definiert, ein root-Passwort gesetzt, der SSH-Daemon gestartet, und im Router eine Portweiterleitung eingerichtet werden. Alles mit Personen Vor Ort, die in solchen Dingen irgendwo zwischen „nicht trittsicher“ oder „ohne Fachkenntnisse“ sind.

Auch wenn es nicht direkt zum Thema passt hier die erforderlichen Befehle um ein (openSUSE) Rescue-System ins Netz zu bringen:

```
0:rescue:~ # ip addr add 192.168.178.99/24 dev eth0
0:rescue:~ # ip link set dev eth0 up
0:rescue:~ # ip route add default via 192.168.178.1 dev eth0
0:rescue:~ # systemctl start sshd.service
...
0:rescue:~ # passwd
```

Das Beispiel bezieht sich auf das traditionelle IP-Netz hinter einer Fritz-Box. Wie gesagt, es muss dann noch im Router ein Portforwarding des Ports 22/TCP auf die in der ersten Zeile gesetzte IP-Adresse eingerichtet (und, wenn der Schlamassel behoben ist, wieder entfernt) werden.

Zurück zum Thema. Die Lösung des Problems war überraschend einfach.

Wird Google bemüht, liefert das viele gute Tipps und Kommandos zu Tage mit denen das Problem beseitigt werden kann. Leider fehlt oft der Zusammenhang, oder zumindest die richtige Reihenfolge der einzugebenden Kommandos.

Der „degradierte“ Verbund lies sich wie folgt wieder zum Leben erwecken. Schritt 1 war das „Stoppen“ des ohnehin inaktiven RAID-Vebundes:

```
0:rescue:~ # mdadm --stop /dev/md0
```

Das Problem war, dass ich zunächst versucht hatte, denn Verbund ohne brachiale Gewalt wieder zu aktivieren:

```
0:rescue:~ # mdadm --assemble /dev/md0 /dev/sda1 /dev/sdb1 /dev/sdc1
/dev/sdd1
```

```
mdadm: /dev/sdb1 is identified as a member of /dev/md/0, slot 1.
mdadm: /dev/sdc1 is identified as a member of /dev/md/0, slot 2.
mdadm: /dev/sdd1 is identified as a member of /dev/md/0, slot 3.
mdadm: no uptodate device for slot 0 of /dev/md/0
mdadm: added /dev/sdc1 to /dev/md/0 as 2
mdadm: added /dev/sdd1 to /dev/md/0 as 3
mdadm: added /dev/sdb1 to /dev/md/0 as 1
mdadm: failed to RUN_ARRAY /dev/md/0: Input/output error
mdadm: Not enough devices to start the array while not clean - consider --force.
```

Wird das Kommando, wie empfohlen mit der Option „-force“ wiederholt, führt das zu nichts. Aus den Rückmeldungen geht einfach nicht hervor, dass der Verbund zunächst erneut gestoppt werden muss... Das hat ein bisschen Zeit gekostet.

Zum Ziel führt also:

```
0:rescue:~ # mdadm --stop /dev/md0
0:rescue:~ # mdadm --assemble --force /dev/md0 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

Die beiden Befehle also in unmittelbarer Abfolge. Danach startete der Verbund, wenn auch degradiert. Bleibt also die defekte Platte zu tauschen.

## Netzwerkprobleme

## Server-Dienste

From:

<https://wiki.invis-server.org/> - **invis-server.org**

Permanent link:

[https://wiki.invis-server.org/doku.php?id=invis\\_server\\_wiki:troubleshooting&rev=1717099947](https://wiki.invis-server.org/doku.php?id=invis_server_wiki:troubleshooting&rev=1717099947)

Last update: **2024/05/30 20:12**

