**Engineer**

# RAID

☰ flere disker som kjører som én, oftest for å tåle diskfeil

## Hva det er

RAID kjører flere disker som én blokkenhet, slik at gruppen blir raskere, større eller vanskeligere å ta knekken på enn noen av diskene den består av. Navnet kommer fra en teknisk rapport fra University of California, Berkeley, i 1987, skrevet av David Patterson, Garth Gibson og Randy Katz: *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. Linux skriver nivåene med små bokstaver og uten mellomrom, raid1 eller raid5, slik md-driveren rapporterer dem.

Rapporten tar utgangspunkt i et misforhold. Prosessorer og minne ble stadig raskere, og sammendraget advarer om at gevinsten går tapt hvis inn- og utdata ikke holder tritt. Den stiller Single Large Expensive Disk, SLED, der kapasiteten hadde vokst raskt mens ytelsen bare var blitt beskjedent bedre, opp mot oppsett bygd av de billige diskene som ble laget for personlige datamaskiner, og venter at slike oppsett skal slå den store disken omtrent ti ganger på hastighet, pålitelighet, strømforbruk og rom for vekst. Sammenligningen er konkret: en IBM 3380 for stormaskiner, en Fujitsu Super Eagle for minimaskiner og den 100 MB store Conner CP 3100 for personlige datamaskiner.

Så kommer haken, i et avsnitt forfatterne selv kaller de dårlige nyhetene. Hvis disker svikter uavhengig av hverandre og med konstant rate, noe rapporten påpeker at diskprodusentene forutsetter når de oppgir en middeltid til feil (MTTF), er MTTF for hele oppsettet lik MTTF for én disk delt på antall disker. Hundre CP 3100, hver oppgitt til 30 000 timer, gir 300 timer, under to uker; tusen gir 30 timer. Rapporten konkluderer rett ut: uten feiltoleranse er store oppsett av billige disker for upålitelige til å være til nytte.

Svaret er redundans. Diskene deles i pålitelighetsgrupper, hver med ekstra kontrolldisker som rommer redundant informasjon. Når en disk svikter, byttes den ut, og innholdet bygges opp igjen på den nye disken fra denne

informasjonen; tiden det tar, er middeltiden til reparasjon (MTTR), og varme reservedisker (hot standby spares), som kobles inn elektronisk idet en disk dør, gjør den kortere. Deretter legger rapporten fram fem måter å organisere diskene på og kaller hver av dem et nivå. Nivå 1 speiler hver disk. Nivå 2 retter feil med en Hamming-kode, slik minnesystemer gjorde. Nivå 3 klarer seg med én kontrolldisk per gruppe. Nivå 4 lar små lese- og skriveoperasjoner gå til ulike disker samtidig. Nivå 5 sprer data og kontrollinformasjon over alle diskene, slik at ingen enkelt kontrolldisk holder skrivingen tilbake.

Praksis har utvidet listen. mdadm, verktøyet som styrer programvare-RAID i Linux, godtar blant annet linear, raid0 (eller stripe), raid1, raid4, raid5, raid6 og raid10. Striping uten redundans, som i dag kalles nivå 0, finnes ikke i inndelingen fra 1987, som begynner med speilede disker. Rapporten lot det stå åpent om et oppsett bygges i maskinvare eller i programvare, og kalte det et rent spørsmål om kostnad og nytte. Linux md er programvarevarianten; et kontrollerkort som viser operativsystemet én stor disk, er maskinvarevarianten.

Bruk begrepet i en beslutning

Bruk «Hva er RAID?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Sikkerhetskopier som er testet i stedet for tatt for gitt, og en skriftlig plan for morgenen da en server ikke kommer opp igjen.

**Se tjenestene våre** Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Automatiserte 100 kritiske sikkerhetskopier — 100 kritiske sikkerhetskopier automatisert på tvers av databasene

**Automatiserte 100 kritiske sikkerhetskopier** Ta kontakt →

## Hvorfor det er viktig

En disk slites ut, og på en server med én disk stopper feilen maskinen og overlater dataene til den sikkerhetskopien som måtte finnes. Rapporten åpner avsnittet om pålitelighet nettopp med dette: disker er så upålitelige at systemansvarlige tar sikkerhetskopi av alt hver natt. Med redundans blir en død disk en reparasjonsjobb i stedet for et avbrudd. Oppsettet fortsetter på diskene som er igjen, i en tilstand Linux kaller degraded (degradert), og kommer tilbake til full styrke når en erstatning er lagt til og bygd opp.

Den beskyttelsen varer bare så lenge de gjenværende diskene gjør det. Regnestykkene i rapporten regner data som tapt når en disk til i samme gruppe svikter før den første er reparert, så lengden på reparasjonsvinduet betyr like mye som antallet kontrolldisker. Med varme reservedisker og korte reparasjoner

blir tallene imponerende: rapporten beregner at et nivå 5-oppsett med 1 000 disker, i grupper på ti og med 5 prosent reservedisker, fortsatt ville ha en MTTF på 90 år. Et annet sted advarer den om at et jordskjelv eller et strømstøt kan få diskene i ett oppsett til å svikte samtidig, og formlene forutsetter at det ikke skjer.

Valget av nivå er en avveining mellom kapasitet, hastighet og pris. Rapportens egen konklusjon er at speilede disker passer for transaksjonsbehandling som bruker høyst halvparten av lagringen, mens nivå 5 kommer best ut når mer av kapasiteten trengs, for arbeid på superdatamaskiner eller for en blanding av slikt arbeid og transaksjonsbehandling.

Når redundansen tar slutt, gjenstår gjenoppretting. Porteføljeoppføringen om datagjenoppretting fra SD-kort, harddisker, SSD-er og RAID behandler et ødelagt RAID-oppsett som en oppgave for seg, skilt fra en død SSD eller en sviktende disk med plater, der framgangsmåten avhenger av mediet.

## Slik ser det ut i praksis

Ta en server med én partisjon på hver av to disker, `/dev/sda1` og `/dev/sdb1`, som skal speiles til et datavolum. Med Linux md bygger tre kommandoer oppsettet, legger et filsystem på det og fører det inn i konfigurasjonen:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
mke2fs -t ext4 -L data /dev/md0
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Den første oppretter `/dev/md0`, et RAID1-oppsett av de to partisjonene, og kjernen starter en resynkronisering (resync) som regner ut redundansen på nytt, slik den gjør etter enhver oppretting. Den andre formaterer oppsettet nøyaktig som den ville formatert en enkelt disk; filsystemet ser én enhet. Den tredje legger en linje som beskriver oppsettet, til i konfigurasjonsfilen til mdadm, som ligger i `/etc/mdadm.conf` eller `/etc/mdadm/mdadm.conf`, og manualen ber om at utdataene gås gjennom før bruk, siden de kan inneholde detaljer ingen vil ha der.

Tre filer viser hvordan oppsettet har det:

```
cat /proc/mdstat
cat /sys/block/md0/md/degraded
cat /sys/block/md0/md/sync_action
```

`/proc/mdstat` lister alle aktive md-enheter med tilstanden deres, og mdadm leser den for å finne oppsett og følge gjenoppbyggingen. Hvert oppsett har også en katalog under `/sys/block/` , og der inneholder `degraded` antallet medlemmer som mangler: 0 for et helt oppsett, 1 når én disk er borte. `sync_action` viser `idle` når ingenting skjer, `resync` etter oppretting eller en ukontrollert nedstengning, `recover` mens en varm reservedisk overtar innholdet til et tapt medlem, og `check` eller `repair` under en full kontroll. Å skrive `check` til den starter denne kontrollen, som leser hver blokk i oppsettet:

```
echo check > /sys/block/md0/md/sync_action
cat /sys/block/md0/md/mismatch_cnt
```

Etterpå oppgir `mismatch_cnt` antallet sektorer som ble skrevet om, eller som ville ha blitt det. Kjernedokumentasjonen påpeker at de fleste nivåene arbeider med sider og ikke sektorer, så tallet kan bli større enn antallet faktiske feil.

Noen må også få høre om en svikt. `mdadm --monitor` spør oppsettene med jevne mellomrom og gjør det den legger merke til, om til hendelser, som den kan sende med e-post til en adresse eller gi videre til et program. På et system som styres av systemd, peker manualen på `mdmonitor.service` som hovedveien, og på `MAILADDR` i `mdadm.conf` som innstillingen som må fylles ut for e-post. Bare fem hendelser sender e-post: `Fail` , `FailSpare` , `DegradedArray` , `SparesMissing` og `TestMessage` . Den siste finnes for å bevise at veien virker:

```
MAILADDR ops@example.com
```

```
mdadm --monitor --scan --oneshot --test
mdadm --detail --test /dev/md0
```

Den første sender en testmelding for hvert oppsett den finner, og avslutter. Den andre passer i et skript: med `--test` er avslutningskoden 0 for et friskt oppsett, 1 når minst én enhet har sviktet, 2 når så mange har sviktet at oppsettet er

ubrukelig, og 4 når mdadm traff på en feil under innhentingen av opplysninger om oppsettet.

Diskoppsettet dekker én hendelse, at en disk dør, og ingenting annet. Porteføljeoppføringen om automatisering av sikkerhetskopiering og gjenopprettingstester beskriver en separat løsning for gjenoppretting: krypterte, dedupliserte restic-øyeblikksbilder over SFTP til lagring hos en annen leverandør, systemd-timere og en månedlig arbeidsflyt som gjenoppretter en kjent fil til et midlertidig arbeidsområde og sammenligner bytene. Det er en gjenopprettingstest i løsningsutformingen, ikke dokumentasjon på at den er kjørt. Implementeringsdokumentasjonen datert 2026-10-27 sier at timerne var deaktivert uten et konfigurert sikkerhetskopirepo, og dokumenterer ingen fullført gjenoppretting i drift.

## Hvor det går galt

Et oppsett bygges for én hendelse, at en av diskene dør, og de fleste måtene det mister data på, ligger like ved siden av den hendelsen.

**Oppsettet blir tatt for en sikkerhetskopi.** Et speil skriver hver endring til begge diskene samtidig, også en fil som ble slettet ved et uhell, en tabell som en mislykket migrering fjernet, eller et volum som løsepengevirus har kryptert. Da bærer hvert medlem den samme skaden. Løsningen er en egen sikkerhetskopi som oppbevares utenfor verten, med gjenoppretting som testes etter en fast plan, slik at oppsettet beskytter mot maskinvarefeil og sikkerhetskopien mot alt annet.

**Et degradert oppsett blir ikke oppdaget.** Et degradert oppsett fortsetter å levere data, og ingenting ved det ser annerledes ut før neste disk svikter. mdadm sender ingen e-post med mindre den har en adresse eller et program: med `--scan` og ingen av delene satt opp overvåker den ingenting, ifølge manualen. `DegradedArray` gjelder dessuten bare oppsett som allerede er degraderte når mdadm ser dem første gang; en disk som dør senere, gir en `Fail` -hendelse. Løsningen er å sette `MAILADDR` , forsikre seg om at `mdmonitor.service` kjører, og sende en melding med `--test` for å bevise at e-posten kommer fram.

**Den andre disken svikter under gjenoppbyggingen.** En gjenoppbygging rekonstruerer den manglende disken fra alle de andre, og rapportens modell mister data nettopp når en disk til i gruppen svikter innenfor reparasjonstiden. En død disk som blir stående i en uke, strekker vinduet til en uke. Løsningen er å

korte det ned: ha en varm reservedisk i oppsettet, som mdadm kan sette opp allerede ved opprettingen med `--spare-devices`, bytte døde disker raskt og kjøre `check` etter en fast plan, slik at en disk med problemer blir funnet mens oppsettet ennå kan rette dem.

**Diskene er ikke uavhengige.** Regnestykket for MTTF forutsetter at disker svikter hver for seg, og rapporten nevner selv jordskjelv og strømstøt som tilfeller der de ikke gjør det. To disker i ett kabinett deler strømforsyning, kontroller, rom og bygning. Løsningen er å behandle det beregnede tallet som en øvre grense og dekke felles årsaker med noe utenfor kabinettet, for eksempel en kopi på et annet sted.

**Et RAID 5- eller 6-oppsett er både urent og degradert.** Hvis maskinen stopper ukontrollert mens et medlem mangler, kan pariteten ikke stoles på, og noen blokker kan ikke bygges opp igjen, så kjernedokumentasjonen advarer om korrupsjon som ikke kan oppdages, og md nekter å starte oppsettet. Å tvinge det i gang med `mdadm --assemble --force`, eller for et rotfilsystem med modulparameteren `start_dirty_degraded`, er å godta den risikoen. Løsningen er å velge en konsistenspolicy på forhånd: md kan skrive skriveoperasjonene til en egen journalenhet for nivå 4, 5 og 6, og for nivå 5 tilbyr den en partial parity log, som ifølge dokumentasjonen lukker det såkalte write hole. Når tvang er eneste vei inn, bør dataene regnes som tvilsomme og sjekkes mot en sikkerhetskopi.

**RAID 0 blir regnet som beskyttelse.** md regner raid0- og linear-oppsett blant dem som ikke har redundant informasjon, og overvåkingen i mdadm melder dem med `Wrong-Level`, fordi de ikke støtter noen av operasjonene for svikt, reservedisk og resynkronisering som den holder øye med. Det finnes ingenting å bygge opp et manglende medlem fra. Løsningen er å bruke nivå 0 bare til data som kan kastes eller lages på nytt, og aldri regne det som RAID i den betydningen rapporten fra 1987 la i ordet.

**En maskinvarekontroller skjuler diskene.** Bak et kontrollerkort ser operativsystemet én logisk enhet, og spør man den enheten om diskhelse, når spørsmålet fram til kontrolleren, ikke til diskene. smartctl trenger for eksempel `-d megaraid,N` for å nå disk N bak en MegaRAID-kontroller, og ifølge manualen dekker det også Dell PERC-kort. Løsningen er å overvåke hver fysiske disk gjennom kontrollerens eget grensesnitt, slik at et medlem som begynner å svikte, blir oppdaget mens oppsettet ennå er helt.

# Les mer

- A Case for Redundant Arrays of Inexpensive Disks (RAID) — siden for den tekniske rapporten fra Berkeley, med sammendrag og PDF.
- UCB/CSD-87-391, hele rapporten (PDF) — regnestykket for MTTF, pålitelighetsgruppene og de fem nivåene.
- RAID arrays, dokumentasjonen til Linux-kjernen — filene til md under `/sys`, urene degraderte oppsett og konsistenspolicyer.
- mdadm(8) — oppretting av oppsett, overvåkingshendelser, e-post og avslutningskodene for `--detail --test`.
- smartctl(8), smartmontools — tilgang til disker bak RAID-kontrollere med valget `-d`.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

**Se hva det koster** Ta kontakt →

## Gjør begrepene praktiske

Spør hvordan «Hva er RAID?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

**Se tjenestene våre**

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/raid/