Engineer

RAID

☰ flere diske, der kører som én, typisk for at tåle et nedbrud

Hvad det er

RAID lader flere diske arbejde som én blokenhed, så gruppen kan blive hurtigere, større eller sværere at slå ihjel end hver enkelt af dem. Navnet stammer fra en teknisk rapport fra University of California, Berkeley, skrevet i 1987 af David Patterson, Garth Gibson og Randy Katz: *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. Linux skriver niveauerne med små bogstaver og uden mellemrum, raid1 eller raid5, sådan som md-driveren selv angiver dem.

Rapporten tager udgangspunkt i en skævhed. Processorer og hukommelse blev hurtigere, og resuméet advarer om, at gevinsten går til spilde, hvis input og output ikke følger med. Den stiller Single Large Expensive Disk, SLED, hvis kapacitet var vokset hurtigt, mens ydelsen kun var blevet lidt bedre, op mod arrays bygget af de billige diske, der blev lavet til personlige computere. Dem forventer den slår den store disk cirka ti gange på hastighed, pålidelighed, strømforbrug og plads til at vokse. Sammenligningen er konkret: en IBM 3380 til mainframes, en Fujitsu Super Eagle til minicomputere og Conners CP 3100 på 100 MB til pc'er.

Så kommer hagen, i et afsnit forfatterne selv kalder de dårlige nyheder. Hvis diske svigter uafhængigt af hinanden og med en konstant rate, hvilket rapporten bemærker er det, diskproducenterne går ud fra, når de opgiver en middeltid til fejl (MTTF), er et arrays MTTF lig med én disks divideret med antallet af diske. Hundrede CP 3100'ere, hver opgivet til 30.000 timer, giver 300 timer, under to uger; tusind giver 30 timer. Rapporten konkluderer, at store arrays af billige diske uden fejltolerance er for upålidelige til at være til nytte.

Svaret er redundans. Diskene deles i pålidelighedsgrupper, hver med ekstra kontroldiske, der rummer redundant information. Når en disk svigter, bliver den skiftet ud, og dens indhold genopbygges på den nye disk ud fra den information;

tiden, det tager, er middeltiden til reparation (MTTR), og varme reservediske (hot standby spares), der kobles ind elektronisk, når en disk dør, gør den kortere. Rapporten beskriver derefter fem måder at organisere diskene på og kalder hver af dem et niveau. Niveau 1 spejler hver disk. Niveau 2 retter fejl med en Hamming-kode, sådan som hukommelsessystemer gjorde. Niveau 3 klarer sig med en enkelt kontroldisk pr. gruppe. Niveau 4 lader små læsninger og skrivninger gå til forskellige diske på samme tid. Niveau 5 spreder data og kontrolinformation over alle diske, så ingen enkelt kontroldisk bremser skrivningerne.

Praksis har føjet mere til listen. mdadm, værktøjet der styrer software-RAID på Linux, accepterer blandt andet linear, raid0 (eller stripe), raid1, raid4, raid5, raid6 og raid10. Striping uden redundans, som i dag kaldes niveau 0, findes ikke i inddelingen fra 1987, der begynder med spejlede diske. Rapporten lod det stå åbent, om et array bygges i hardware eller i software, og kaldte det alene et spørgsmål om omkostninger og udbytte. Linux md er softwareudgaven; et controllerkort, der viser operativsystemet én stor disk, er hardwareudgaven.

Brug begrebet i en beslutning

Brug »Hvad er RAID?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Sikkerhedskopier, der er testet og ikke bare antaget at virke, og en nedskrevet plan for den morgen, en server ikke starter igen.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har automatiseret 100 kritiske databackups — 100 kritiske backups af databaserne automatiseret

Har automatiseret 100 kritiske databackups Kontakt os →

Hvorfor det betyder noget

En disk slides op, og på en server med kun én disk standser maskinen, når den går, og data er overladt til den backup, der nu engang findes. Rapporten åbner sit afsnit om pålidelighed netop med det: diske er så upålidelige, at systemadministratorer tager backup af alt hver nat. Med redundans bliver en død disk et reparationsjob i stedet for et nedbrud. Arrayet fortsætter på de diske, der er tilbage, i en tilstand Linux kalder degraded, og er fuldt intakt igen, når en ny disk er sat i og genopbygget.

Den beskyttelse holder kun, så længe de resterende diske gør. Rapportens regnestykke regner data for tabt, når en disk mere i samme gruppe svigter, før den første er repareret, så længden af reparationsvinduet betyder lige så meget

som antallet af kontroldiske. Med varme reservediske og korte reparationer bliver tallene imponerende: rapporten beregner, at et niveau 5-array på 1.000 diske, i grupper på ti med 5 procent reservediske, stadig ville have en MTTF på 90 år. Et andet sted advarer den om, at et jordskælv eller et strømstød kan få diskene i ét array til at svigte samtidig, og dens formler forudsætter, at det ikke sker.

Valget af niveau er en afvejning mellem kapacitet, hastighed og pris. Rapportens egen konklusion er, at spejlede diske passer til transaktionsbehandling, der højst bruger halvdelen af lagerpladsen, og at niveau 5 ser bedst ud, når mere af kapaciteten skal udnyttes, til supercomputerarbejde eller til en blanding af det og transaktionsbehandling.

Når redundansen slipper op, er gendannelse det, der er tilbage. Porteføljesiden om datagendannelse fra SD-kort, harddiske, SSD'er og RAID behandler et sammenbrudt RAID-array som en opgave for sig, adskilt fra en død SSD eller et pladedrev, der er gået i stykker, og fremgangsmåden afhænger af mediet.

Sådan ser det ud i praksis

Tag en server med én partition på hver af to diske, `/dev/sda1` og `/dev/sdb1`, der skal spejles til et datavolumen. Med Linux md bygger tre kommandoer arrayet, lægger et filsystem på det og skriver det ind i konfigurationen:

```
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 opretter `/dev/md0`, et RAID1-array af de to partitioner, og kernen går i gang med en resync, der beregner redundansen, som den gør efter enhver oprettelse. Den anden formaterer arrayet præcis som en enkelt disk; filsystemet ser én enhed. Den tredje tilføjer en linje, der beskriver arrayet, til mdadm's konfigurationsfil, som ligger i `/etc/mdadm.conf` eller `/etc/mdadm/mdadm.conf`, og manualen beder om, at det output bliver gennemset før brug, da det kan indeholde detaljer, ingen ønsker at have stående der.

Tre filer viser, hvordan arrayet har det:

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

`/proc/mdstat` viser alle aktive md-enheder med deres tilstand, og mdadm læser den for at finde arrays og følge genopbygningen. Hvert array har også en mappe under `/sys/block/`, og i den holder `degraded` antallet af manglende medlemmer: 0 for et helt array, 1 når én disk er væk. `sync_action` står på `idle`, når intet sker, `resync` efter oprettelse eller en uren nedlukning, `recover`, mens en varm reservedisk overtager indholdet fra et tabt medlem, og `check` eller `repair` under en fuld kontrol. Skriver man `check` til den, starter den kontrol, som læser hver eneste blok i arrayet:

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

Bagefter angiver `mismatch_cnt`, hvor mange sektorer der blev skrevet om, eller som ville være blevet det. Kernens dokumentation bemærker, at de fleste niveauer arbejder med sider frem for sektorer, så tallet kan være større end antallet af reelle fejl.

Nogen skal også høre om et nedbrud. `mdadm --monitor` tjekker arrayene med faste mellemrum og gør det, den bemærker, til hændelser, som den kan sende med mail til en adresse eller give videre til et program. På et system, der kører systemd, nævner manualen `mdmonitor.service` som hovedvejen og `MAILADDR` i `mdadm.conf` som den indstilling, der skal udfyldes for at få mail. Kun fem hændelser sender mail: `Fail`, `FailSpare`, `DegradedArray`, `SparesMissing` og `TestMessage`. Den sidste findes for at bevise, at vejen virker:

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

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

Den første sender en testbesked for hvert array, den finder, og afslutter så. Den anden passer til et script: med `--test` er dens exitstatus 0 for et sundt array, 1 når mindst én enhed er svigtet, 2 når så mange er svigtet, at arrayet ikke kan bruges, og 4 når mdadm løb ind i en fejl under indsamlingen af oplysninger om arrayet.

Arrayet dækker én hændelse, at en disk dør, og intet andet. Porteføljesiden om automatiseret sikkerhedskopiering og gendannelsestest beskriver et separat gendannelsesdesign: krypterede, deduplikerede restic-snapshots over SFTP til lager hos en anden udbyder, systemd-timere og et månedligt workflow, som gendanner en kendt fil til et midlertidigt arbejdsområde og sammenligner dens bytes. Det er en test af gendannelse i designet, ikke dokumentation for, at den er kørt. Implementeringsbeskrivelsen dateret 2026-10-27 angiver, at timerne var deaktiveret uden et konfigureret backuplager, og dokumenterer ingen gennemført gendannelse i drift.

Hvor det går galt

Et array er bygget til én hændelse, at en af dets diske dør, og de fleste måder, det mister data på, ligger lige ved siden af den hændelse.

Arrayet bliver forvekslet med en backup. Et spejl skriver hver ændring til begge diske på én gang, og det gælder også en fil, der er slettet ved en fejl, en tabel, som en dårlig migrering har droppet, eller et volumen, som ransomware har krypteret. Hvert medlem bærer så den samme skade. Løsningen er en separat backup, der opbevares uden for værten, med gendannelser, der bliver testet efter en fast plan, så arrayet beskytter mod hardwarefejl og backuppen mod alt andet.

Et degraderet array bliver ikke opdaget. Et degraderet array bliver ved med at levere data, og intet ved det ser anderledes ud, før den næste disk svigter. mdadm sender ingen mail, medmindre den har en adresse eller et program: med `--scan` og ingen af delene sat op siger manualen, at den ikke overvåger noget. `DegradedArray` dækker desuden kun arrays, der allerede er degraderede, når mdadm ser dem første gang; en disk, der dør senere, giver en `Fail` -hændelse. Løsningen er at sætte `MAILADDR` , sikre at `mdmonitor.service` kører, og sende en `--test` -besked for at bevise, at mailen kommer frem.

Den anden disk svigter under genopbygningen. En genopbygning rekonstruerer den manglende disk ud fra alle de andre, og rapportens model mister data netop, når en disk mere i gruppen svigter inden for reparationstiden. En død disk, der bliver siddende i en uge, strækker vinduet til en uge. Løsningen er at gøre det kortere: hold en varm reservedisk i arrayet, som mdadm kan sætte op ved oprettelsen med `--spare-devices`, skift døde diske hurtigt ud, og kør en planlagt `check`, så en disk med problemer bliver fundet, mens arrayet stadig kan rette dem.

Diskene er ikke uafhængige. MTTF-regnestykket forudsætter, at diske svigter hver for sig, og rapporten nævner selv et jordskælv og et strømstød som tilfælde, hvor de ikke gør. To diske i samme kabinet deler strømforsyning, controller, rum og bygning. Løsningen er at læse det beregnede tal som en øvre grænse og dække de fælles årsager med noget uden for kassen, for eksempel en kopi et andet sted.

Et RAID 5- eller 6-array er både urent og degraderet. Går maskinen ned uden en ren nedlukning, mens et medlem mangler, kan man ikke stole på pariteten, og nogle blokke kan ikke genopbygges. Kernens dokumentation advarer derfor om korruption, der ikke kan opdages, og md nægter at starte arrayet. At tvinge det i gang med `mdadm --assemble --force`, eller for et rodfilsystem med modulparameteren `start_dirty_degraded`, betyder at acceptere den risiko. Løsningen er at vælge en konsistenspolitik på forhånd: md kan logge skrivninger til en journalenhed for niveau 4, 5 og 6, og for niveau 5 tilbyder den en partial parity log, som ifølge dokumentationen lukker det såkaldte write hole. Er tvang den eneste vej ind, skal data behandles som mistænkelige og tjekkes mod en backup.

RAID 0 bliver regnet som beskyttelse. md opfører raid0- og linear-arrays som arrays uden redundansinformation, og mdadm's overvågning melder dem med `Wrong-Level`, fordi de ikke understøtter nogen af de operationer for fejl, reservediske og resync, som den holder øje med. Der er intet at genopbygge et manglende medlem ud fra. Løsningen er kun at bruge niveau 0 til data, der kan smides væk eller skabes igen, og aldrig at regne det for RAID i den betydning, rapporten fra 1987 giver ordet.

En hardwarecontroller skjuler diskene. Bag et controllerkort ser operativsystemet én logisk enhed, og et helbredstjek rettet mod den enhed når controlleren, ikke drevene. smartctl skal for eksempel have `-d megaraid,N` for at nå disk N bag en MegaRAID-controller, og ifølge manualen dækker det også Dell

PERC-kort. Løsningen er at overvåge hver fysisk disk gennem controllerens egen grænseflade, så et medlem, der er ved at svigte, bliver opdaget, mens arrayet stadig er helt.

Læs mere

- A Case for Redundant Arrays of Inexpensive Disks (RAID) — siden for den tekniske rapport fra Berkeley med resumé og PDF.
- UCB/CSD-87-391, hele rapporten (PDF) — MTTF-regnestykket, pålidelighedsgrupperne og de fem niveauer.
- RAID arrays, Linux-kernens dokumentation — md's filer under `/sys`, urene degraderede arrays og konsistenspolitikker.
- mdadm(8) — oprettelse af arrays, overvågningshændelser, mail og exitkoderne for `--detail --test`.
- smartctl(8), smartmontools — adgang til diske bag RAID-controllere med tilvalget `-d`.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er RAID?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/raid/