

11.6 RAID

Wednesday, April 29, 2020 1:15 AM

I forbindelse med forbedring av disklagring har utviklingen ledet an til en liste med diskoperer uavhengig og i parallell. Med flere diskoperer kan separate I/O-forespørsler håndteres i parallell så lenge den nødvendige dataen ligger på separate diskoper. I tillegg kan en enkel I/O-forespørsel executes i parallell dersom blokken av data som skal aksesseres er distribuert over flere diskoper.

RAID

RAID (Redundant Array of Independent Disks) er et skjema; en standard ordning for et multidisk-databasesystem som består av syv nivåer (indeksert på null). Disse nivåene er ikke hierarkisk ordnet, men utpeker design-arkitekturer som deler tre egenskaper:

1. OS-et ser på RAID som en enkel logisk driver.
2. Dataen distribueres på tvers av de fysiske driverne.
3. Redundant diskkapasitet brukes til å lagre paritetsinformasjon, hvilket garanterer gjenopprettelse av data dersom det oppstår diskfeil.

RAID-strategien bruker flere diskdrivere og distribuerer data på en slik måte at samtidig aksess av data fra flere drivere mulig – dette øker ytelsen av I/O og gir lettere inkrementell økning av kapasitet. Ved bruk av mange enheter øker sjansen for feil, og derfor lagrer RAID paritetsinformasjon for å kunne gjenopprette data. Av de syv nivåene beskrevet brukes 0,1,5 og 6 oftest.

RAID 0

Nivå 0 inneholder ikke redundans for å øke ytelse eller beskytte dataen. I RAID 0 distribueres bruker- og systemdata på tvers av diskoper i listen. Dette medfører at dataen distribuert på forskjellige diskoper kan aksesseres samtidig. Dataen *stripes* på tvers av tilgjengelige diskoper. Bruker- og systemdata kan sees på som om de er lagret på en logisk disk. Denne disken deles opp i *strips* som mappes i Round-Robin-stil til fysiske diskoper i RAID-arrayet. Fordelen er at i et n -disk array så er de første n logiske stripsene fysisk lagret som den første stripen på hver av de n diskene – dette danner en *stripe*. Den andre stripen ligger på den andre stripen i hver disk osv. Dersom en I/O-forespørsel består av flere logiske etterfølgende strips, kan de håndteres i parallell.

RAID 1

På dette nivået oppnås redundans ved en hensiktsmessig duplisering av all data. Dette kalles for *speiling* hvor hver logiske strips mappes til en separat fysisk disk slik at hver disk i arrayet har en speil-disk med samme data. Fordelen med speiling er at lese-forespørsler kan gjøres til begge diskene, hvorpå den med minst søketid og rotasjonsforsinkelser velges. I tillegg må skrive-forespørsler gjøres til begge diskene, men dette kan gjøres i parallell. Siden det ikke er noe paritets-bit, vil dette gå fortere enn for RAID 2-6. Dersom det oppstår en feil er dette også lett å ordne opp i da dataen kan aksesseres fra den andre disken. Ulempen med RAID 1 er overhead. Vi velger å bruke RAID 1 i kritiske systemer som trenger backup av dataen eller i systemer som har mange lese-forespørsler.

RAID 2 & RAID 3

RAID 2 og RAID 3 bruker en parallell aksessteknikk. Her er det vanlig at alle driverne er synkronisert i så måte at hver disk-head er i samme posisjon på et gitt tidspunkt. RAID 2 og 3 bruker også striper, men disse er relativt små.

RAID 2

Bruker en hamming-kode for å korrigere single-bit error og oppdage dobbel-bit error. RAID 2 krever også ganske stor plass, med en logaritmisk overhead på antall data disken inneholder. Den brukes egentlig bare til systemer hvor disk-errorer forekommer ofte, ellers regnes den som overkill.

RAID 3

Bruker kun en enkel redundant disk. I tillegg bruker RAID 3 kun én paritets-bit til å rette opp feil istedenfor en error-korreksjonskode. Siden små strips blir brukt kan RAID 3 oppnå høy dataoverføringsrate. På den andre siden kan kun én I/O-forespørsel gjøres om gangen, hvilket gjør at RAID 3 ikke er så bra for transaksjonsorienterte miljøer.

RAID 4

RAID 4, 5 og 6 benytter en uavhengig aksessteknikk. Siden hver disk opererer uavhengig kan I/O-forespørsler behandles i parallell. I RAID 4-6 er stripsene relativt store, samtidig som få er involvert. RAID 4 benytter en *bit-by-bit*-paritetsblokk. Hver gang en skrivning skjer må både dataen og paritets-biten oppdateres. Dette medfører en potensiell flaskehals.

RAID 5

RAID 5 distribuerer paritets-bitene på alle diskene, som typisk allokeres ved Round-Robin-disiplinen. Distribusjonen av paritetsstrips på driverne unngår en potensiell I/O-flaskehals som RAID 4 har. Dessuten vil ikke tap av en disk i RAID 5 medføre tapt data.

RAID 6

I RAID 6 benyttes to ulike paritetsberegninger som lagres i separate blokker på forskjellige diskoper. Derfor har en RAID 6-array hvor bruker krever N diskoper totalt $N + 2$ diskoper. P og Q er to ulike datasjekk-

algoritmer hvor den ene bruker XOR som i RAID 4 og 5, mens den andre er uavhengig. Dette gjør det mulig å regenerere data selv om to disker med brukerdatafeil oppstår. Fordelen med RAID 6 er at den gir svært høy datatilgjengelighet, hvorav tre disker må feile innen MTTR (mean-time-to-repair, se Kap 1) for at data skal gå tapt. RAID 6 har dog en høy straff for skrijving, fordi hver skrijving påvirker begge de to paritetsblokkene. Sammenlignet med RAID 5 kan RAID 6 ha en overordnet stagnasjon på 30% i skriver-ytelse.

Table 11.4 RAID Levels

Category	Level	Description	Disks Required	Data Availability	Large I/O Data Transfer Capacity	Small I/O Request Rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	$2N$	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; significantly lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

Note: N , number of data disks; m , proportional to $\log N$.