A case for redundant array of inexpensive disks (RAID)

Siddharth Bhat (20161105)

March 12, 2020

- Moore's law: Transitors in a chip $\times 2$ every 2 years: $\equiv 2^{\frac{\text{Year}-1964}{2}}$
- Bits stored / inch $\times 10$ every 10 years: $\equiv 10^{\frac{\text{Year}-1971}{10}}$
- SLED (Single Large expensive magnetic disks) cannot keep pace with CPUs!
- *Speedup* $\equiv$?? Amdahl's law

Inexpensive Disks

	IBM 3380	Conners CP3100
Price	$ 135000	**$1000**
Power/box(Watt)	6600 W	**10 W**
IO/sec (max)	**50 ops/s**	30 ops/s
IO/sec (typical)	**30 ops/s**	20 ops/s
Data capacity (MB)	**7500 MB**	100 MB
Time to failure (rated)	30,000 hours	30,000 hours
Time to failure (practice)	100,000 hours	Unk

Array of Inexpensive Disks

	IBM 3380	135 × Conners CP3100
Price	$135000	$135000
Power/box(Watt)	6600 W	**1350 W**
IO/sec (max)	50 ops/s	**4050 ops/s**
IO/sec (typical)	30 ops/s	**2700 ops/s**
Data capacity (MB)	7500 MB	**13500 MB**
Time to failure (rated)	**30,000 hours**	100 hours
Time to failure (practice)	**100,000 hours**	Unk

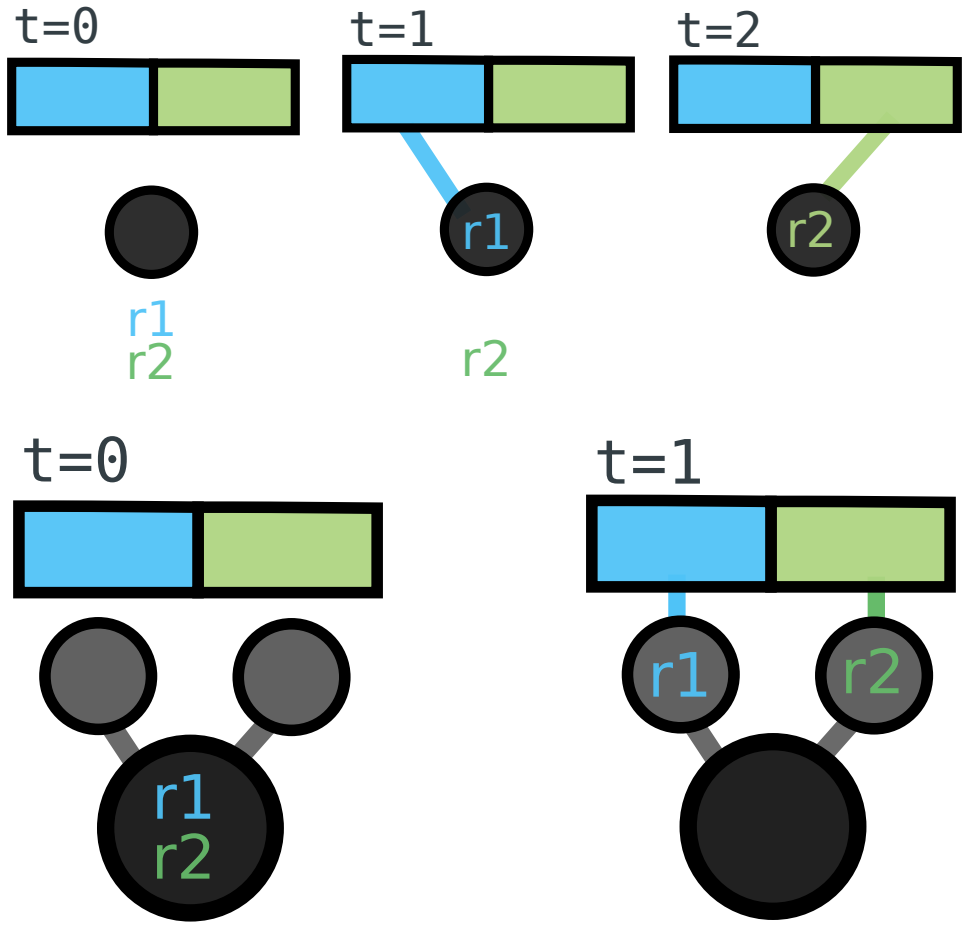
The ugly

Mean time to failure of any of 135 disks: 30,000 hours / 135 = 100 hours

Redundant Array of Inexpensive Disks

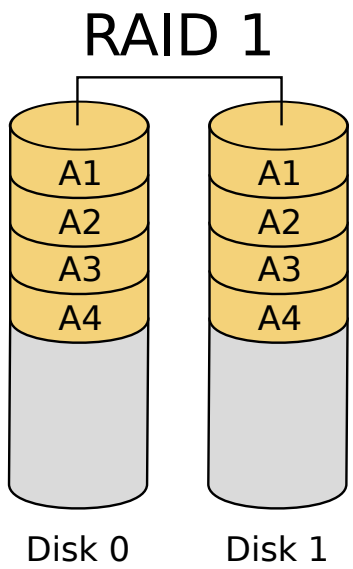
- Extra check disks store redundant information.
- Replace failed disk & replicate from check disk.

Pleasing side-effects of Redundancy



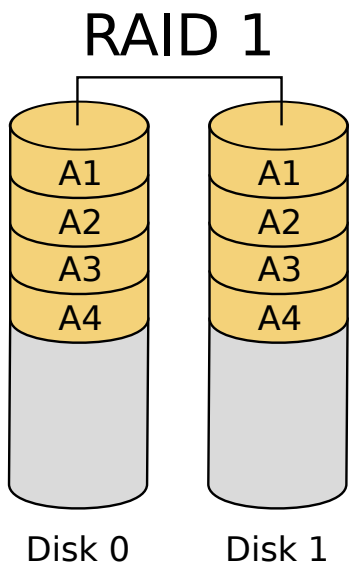
Parallel reads & writes are enabled to random sections of disk.

RAID 1: Mirrored Disks



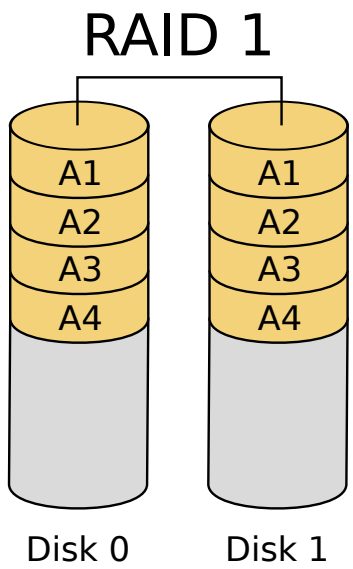
- All data is duplicated across all disks

RAID 1: Mirrored Disks



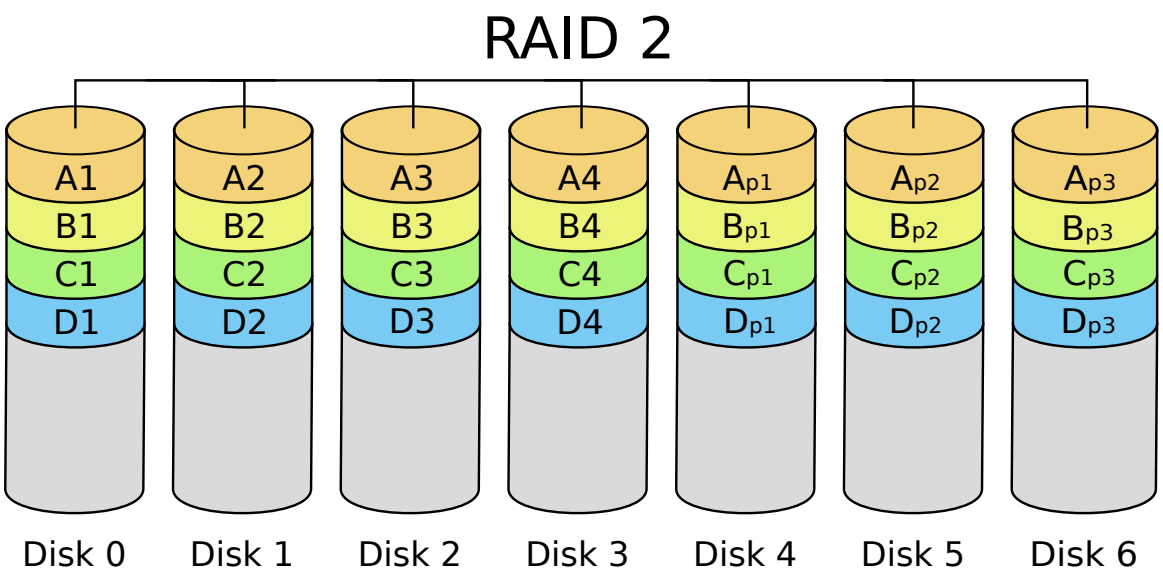
- All data is duplicated across all disks
- Writes are expensive: need to be replicated!

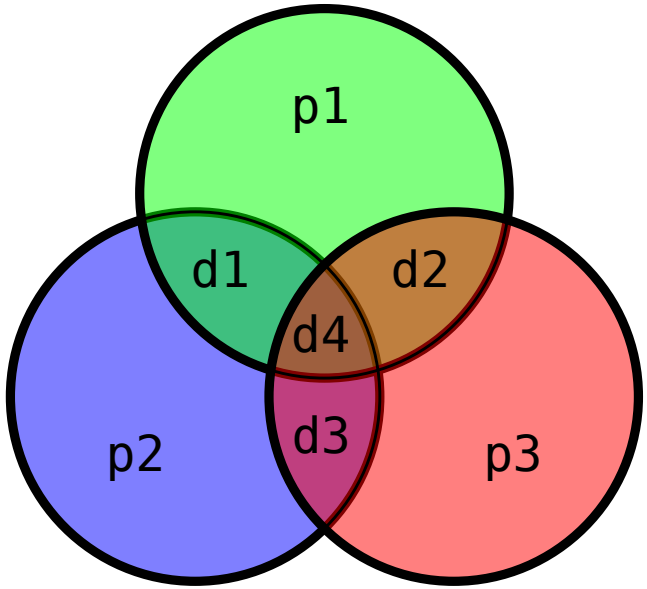
RAID 1: Mirrored Disks



- All data is duplicated across all disks
- Writes are expensive: need to be replicated!
- Disk space utilization: **50%**.

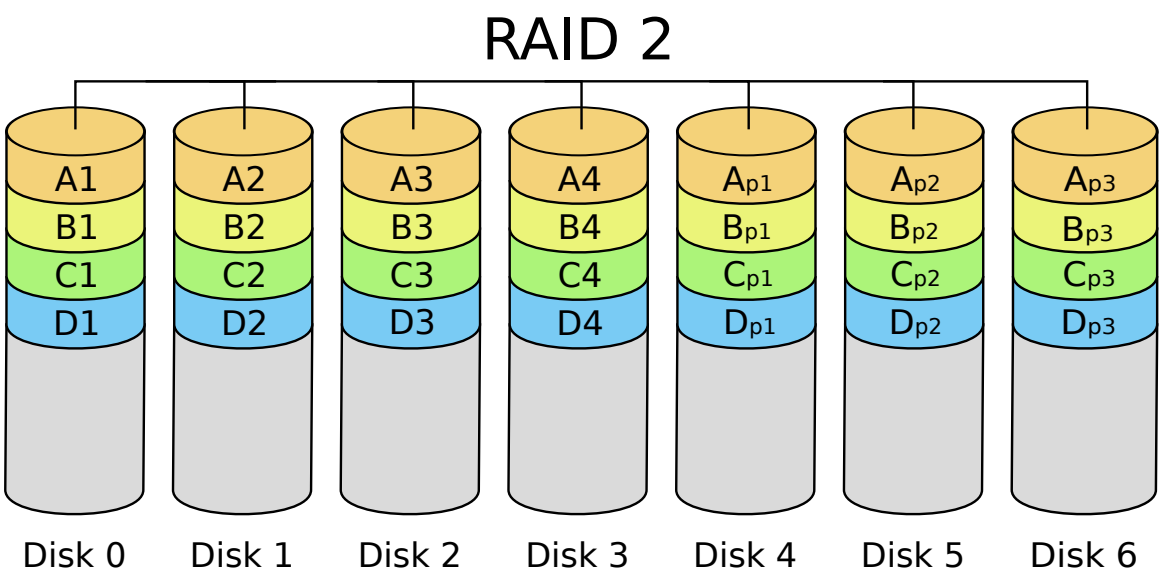
RAID 2: Bit level strining + Hamming ECC

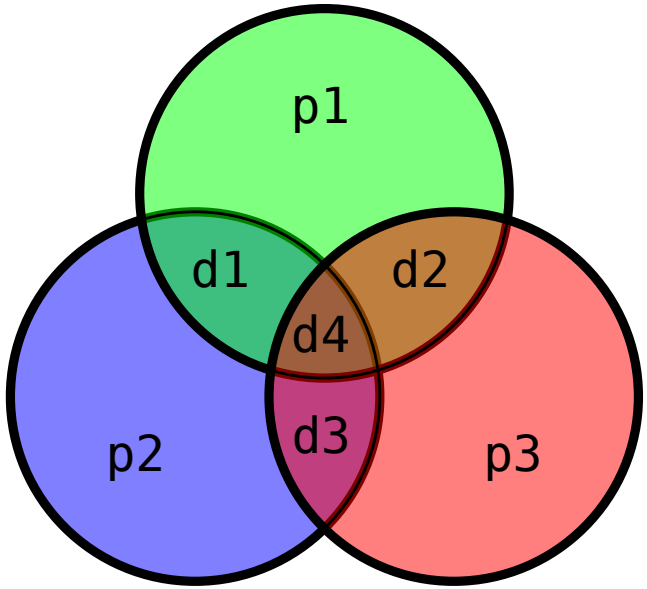




- Use **bit-level** striping (consecutive bits $\rightarrow$ different disks).
- Use Hamming code **per bit**.

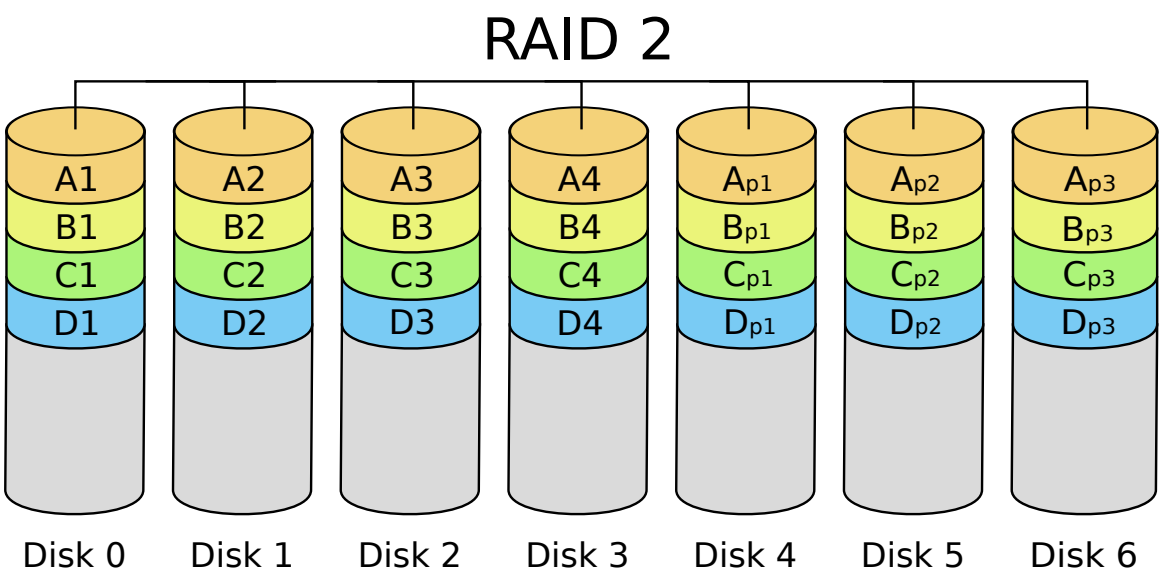
RAID 2: Bit level strining + Hamming ECC

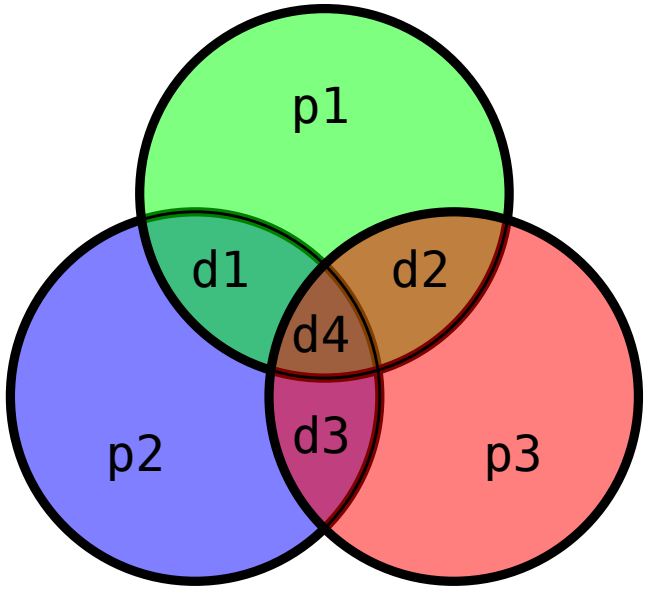




- Use **bit-level** striping (consecutive bits $\rightarrow$ different disks).
- Use Hamming code **per bit**.
- All disks need to read same bit simultaneously: bitwise error checking.

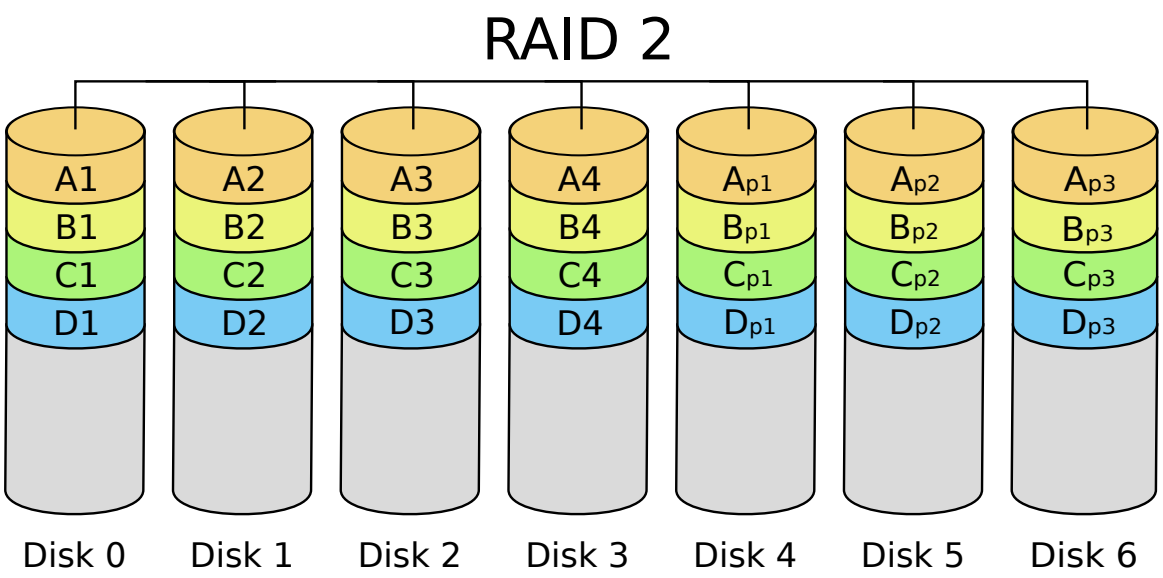
RAID 2: Bit level strining + Hamming ECC

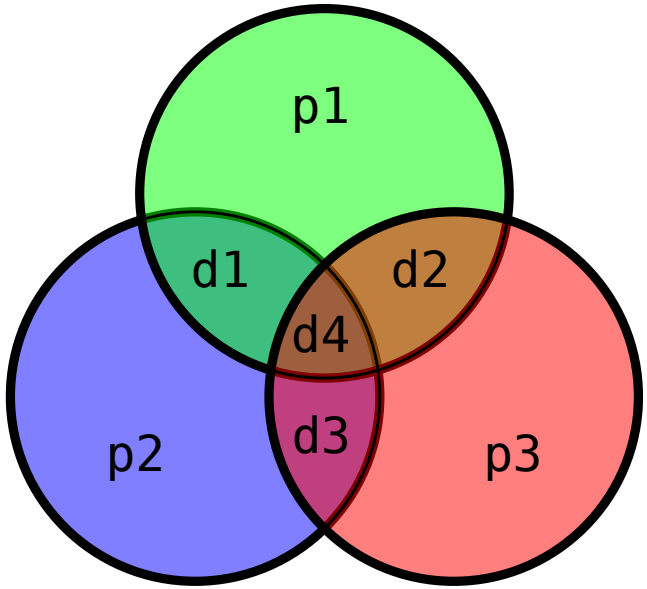




- Use **bit-level** striping (consecutive bits $\rightarrow$ different disks).
- Use Hamming code **per bit**.
- All disks need to read same bit simultaneously: bitwise error checking.
- So, **Cannot service multiple requests at once**.

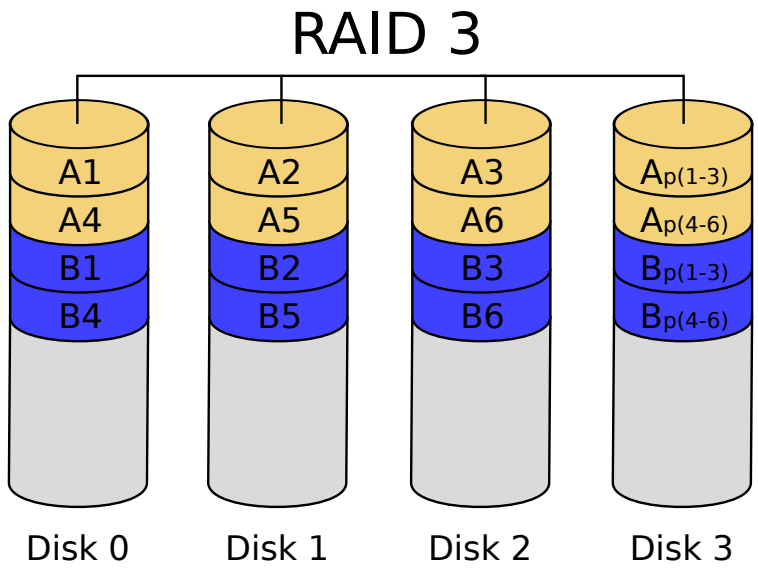
RAID 2: Bit level strining + Hamming ECC





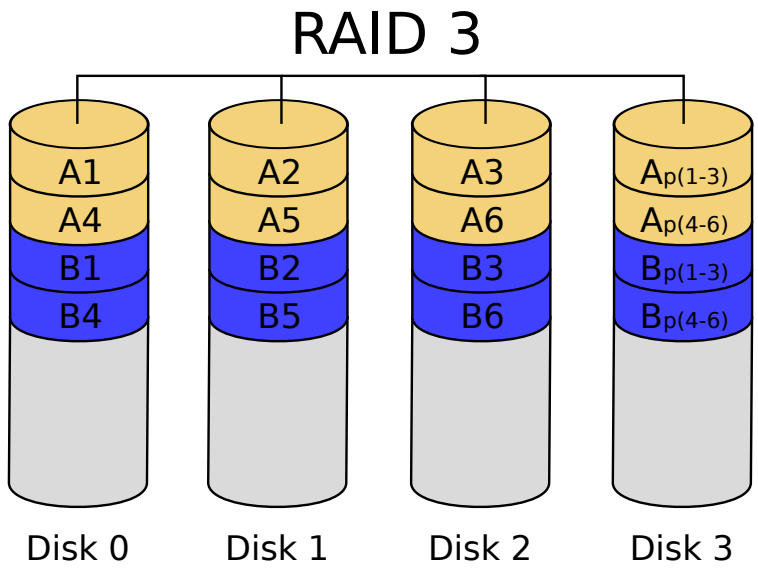
- Use **bit-level** striping (consecutive bits $\rightarrow$ different disks).
- Use Hamming code **per bit**.
- All disks need to read same bit simultaneously: bitwise error checking.
- So, **Cannot service multiple requests at once**.
- Not used anymore.

RAID 3: Byte level striping + parity ECC



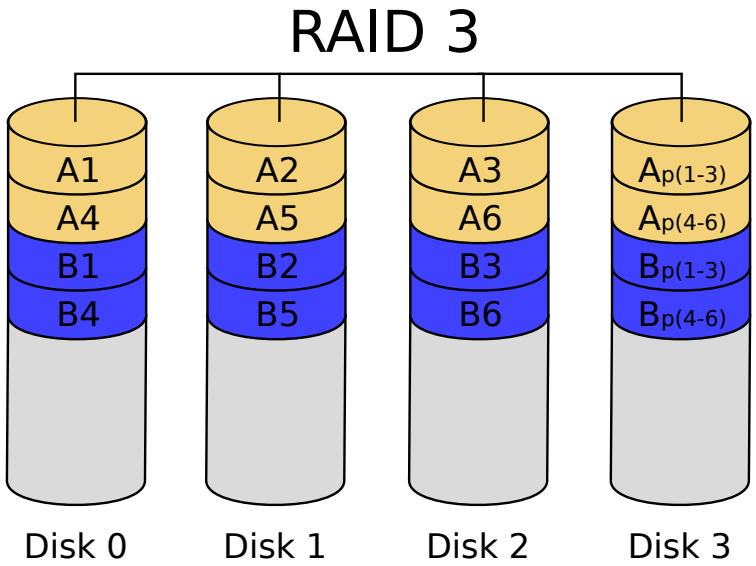
- Use **byte-level** striping (consecutive bytes $\rightarrow$ different disks).

RAID 3: Byte level striping + parity ECC



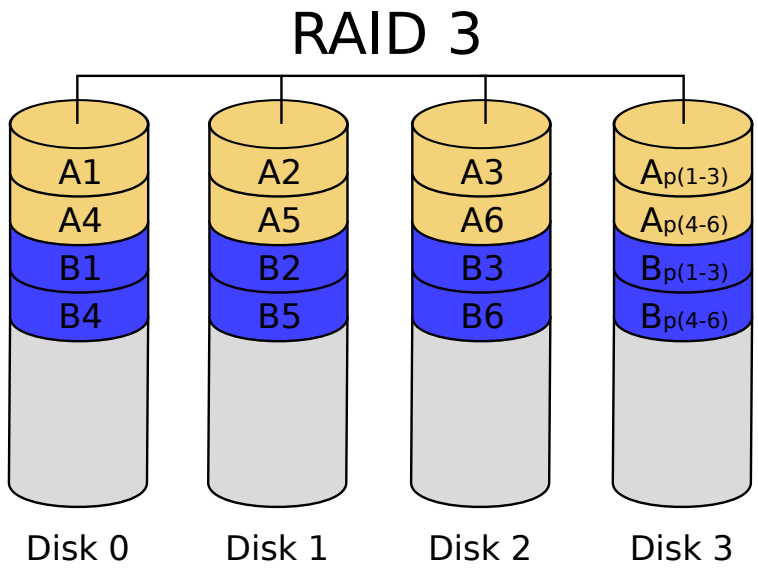
- Use **byte-level** striping (consecutive bytes $\rightarrow$ different disks).
- I/O needs **synchronized spindles**.

RAID 3: Byte level striping + parity ECC



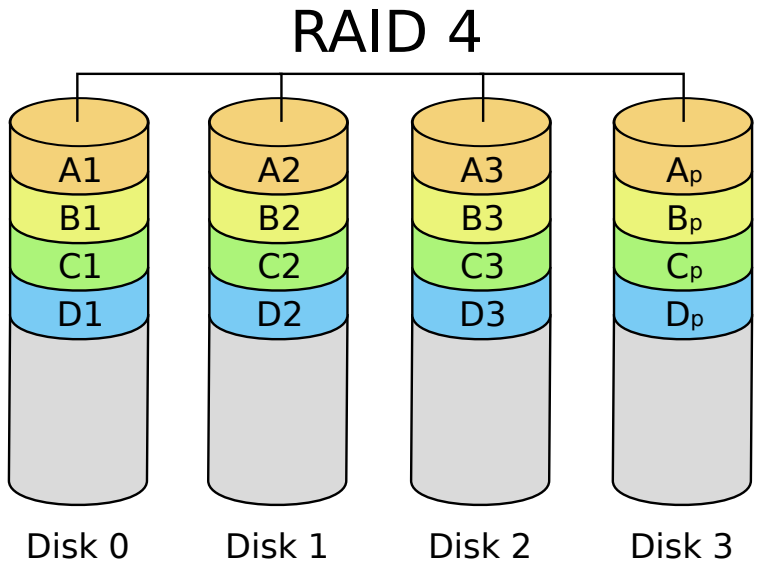
- Use **byte-level** striping (consecutive bytes $\rightarrow$ different disks).
- I/O needs **synchronized spindles**.
- parity needs far less storage space.

RAID 3: Byte level striping + parity ECC



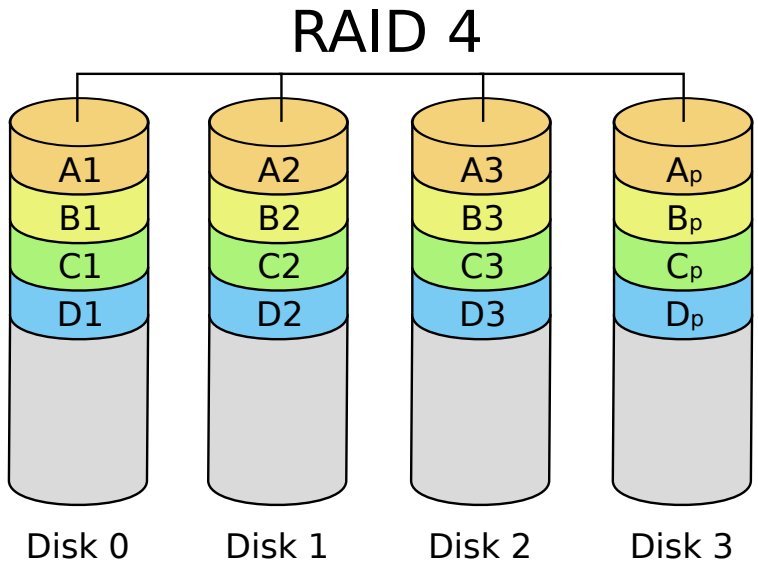
- Use **byte-level** striping (consecutive bytes $\rightarrow$ different disks).
- I/O needs **synchronized spindles**.
- parity needs far less storage space.
- Rarely used in practice.

RAID 4: Block level striping + parity ECC



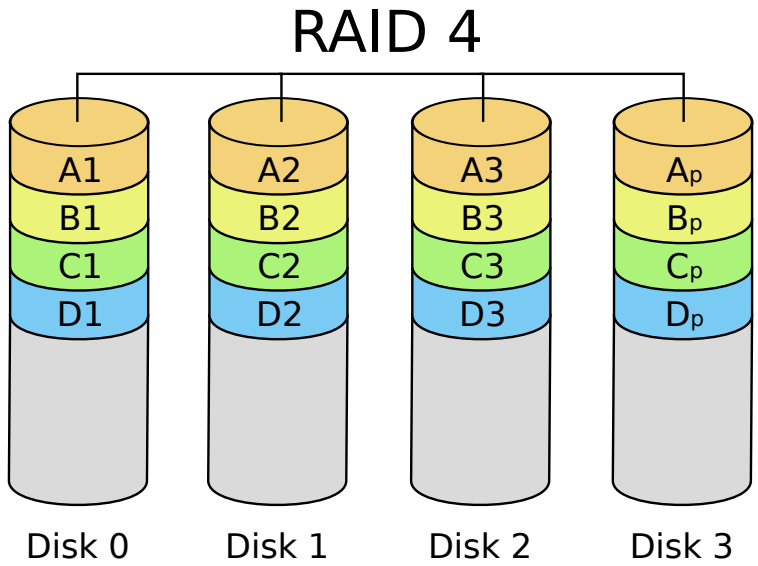
- Stripe data per-block (consecutive block $\rightarrow$ different disks).

RAID 4: Block level striping + parity ECC



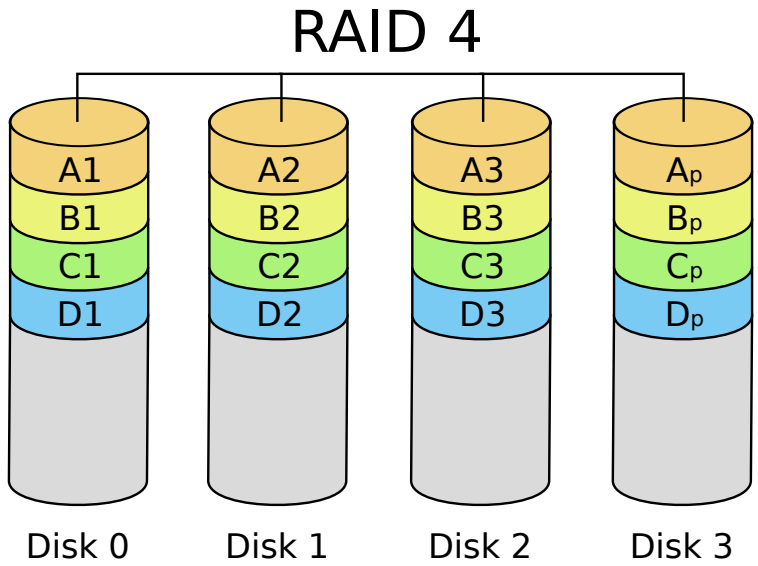
- Stripe data per-block (consecutive block $\rightarrow$ different disks).
- Block level parity needs far less storage space.

RAID 4: Block level striping + parity ECC



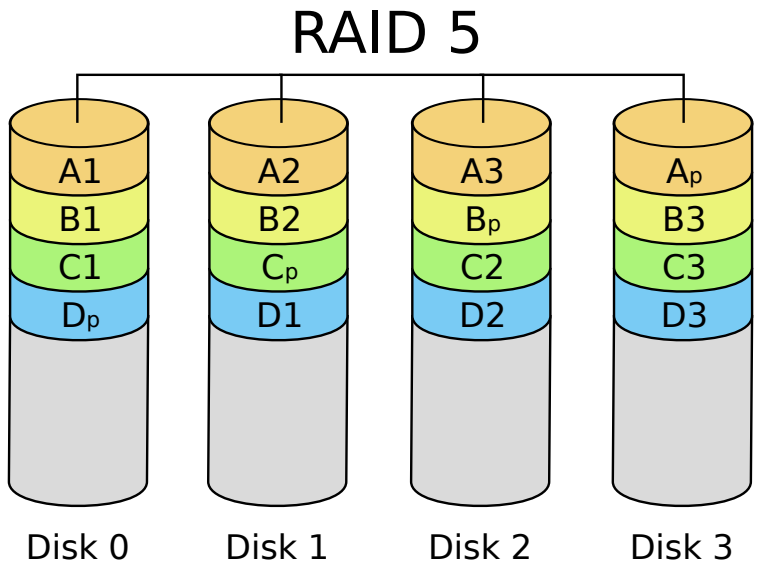
- Stripe data per-block (consecutive block $\rightarrow$ different disks).
- Block level parity needs far less storage space.
- Good random access read times.

RAID 4: Block level striping + parity ECC



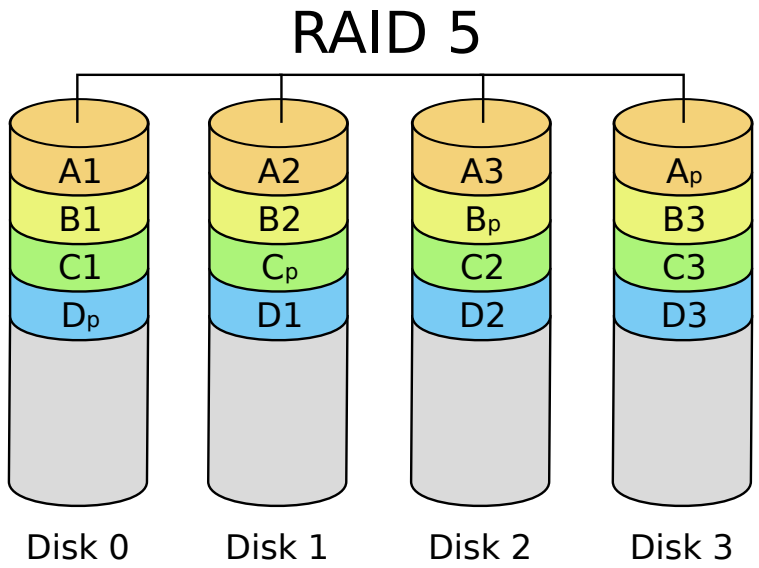
- Stripe data per-block (consecutive block $\rightarrow$ different disks).
- Block level parity needs far less storage space.
- Good random access read times.
- Bad write times: All parities in same disk.

RAID 5: Block level striping + distributed parity ECC



- Inherits the good from RAID 4.

RAID 5: Block level striping + distributed parity ECC



- Inherits the good from RAID 4.
- Good write times: parities distributed across disks.