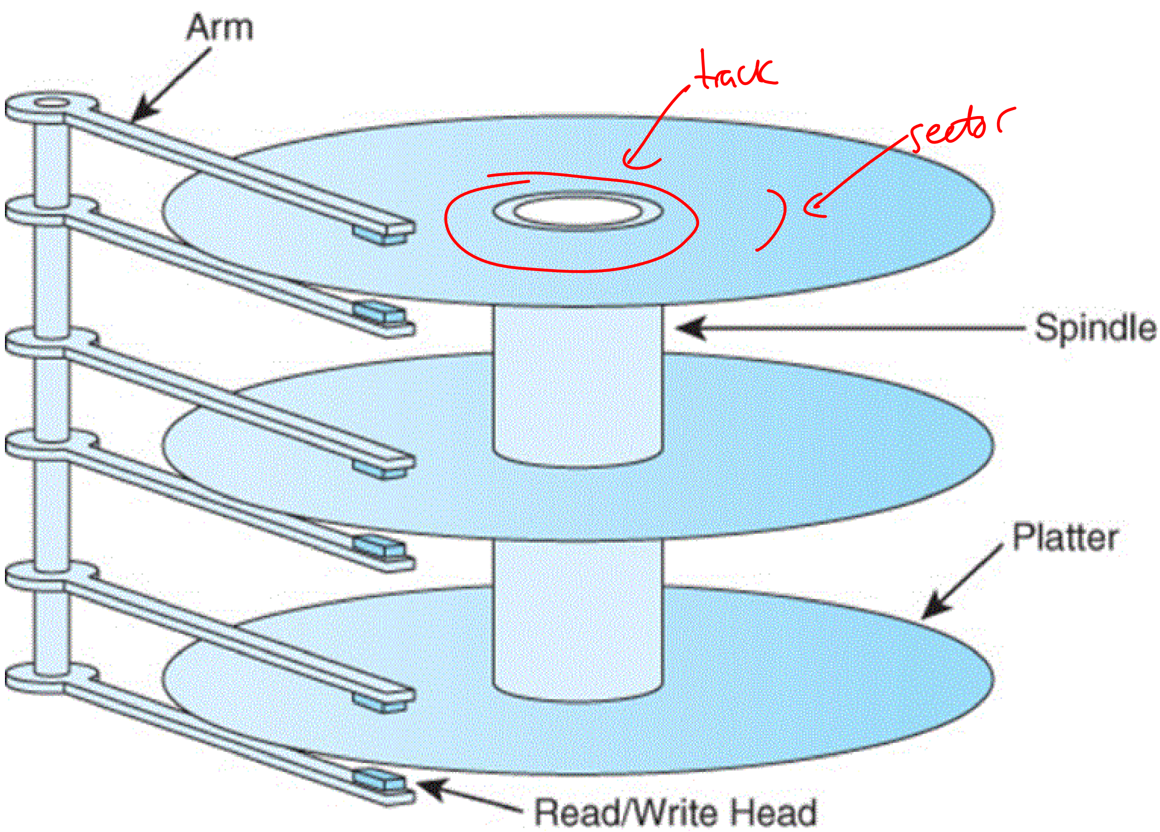
Disk systems

1. Data access and transfer times

Disk internals, from fig 7.14 of the textbook:



Important definitions:

- Seek time – time for disk arm to move to desired track (typically a few ms).
- Rotational latency – time for desired data to rotate to location under disk head.

e.g. for 7200 RPM disk,
seconds per rotation $= \frac{60}{7200} = 0.0083$
On average wait for ½ a rotation, i.e. 0.004 s.
So rotational latency ≈ 4 ms.

- Access time = seek time + rotational latency
  = (for example) 3ms + 4ms
  = 7 ms

- Transfer rate — rate at which sequential data can be transferred. (typically 100 MB/s as of 2010).

- Total time to read some sequential data

$$= \text{access time} + \frac{\text{data size}}{\text{transfer rate}}$$

Note: Caching is used everywhere in the storage heirarchy. e.g.
- the OS keeps a file system cache (100s of MB) in main memory
- the disk caches recently-accessed blocks in its controller

2. RAID 5

RAID ≡ "Redundant Array of Independent Disks"

- an array of disks with a controller that "looks like" one big disk, to the computer.
- employs two main ideas: <u>striping</u> and <u>parity</u>

(A) <u>Striping</u>

Striping means that logically contiguous parts of a file are stored on different disks. The chunks stored on each disk are called <u>stripes</u>. A Typical stripe size is 64kB

Simple example with 4 disks and stripe size of 3 bytes, storing the string "ABCDE...":

byte	disk 0	disk 1	disk 2	disk 3
0	A	D	G	J
1	B	E	H	K
2	C	F	I	L
3	M	P	S	V
4	N	Q	T	W
5	O	R	U	X
6	Y			
7	Z			
8				
⋮	⋮	⋮	⋮	⋮

(B) Parity

parity $\equiv$ XOR $\equiv$ binary addition without carries

Example: given binary data stripes 101, 111, 011, what is their parity?

Solution:

$$\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ \hline 0 & 0 & 1 \end{array}$$

parity is 001

Important property of parity: if we lose one of the data stripes (e.g. due to a disk failure), we can use the parity stripe to recover it!

e.g.

$$\begin{array}{cccc} & 1 & 0 & 1 \\ & 1 & 1 & 1 \\ \oplus & ? & ? & ? \\ \hline & 0 & 0 & 1 \end{array}$$

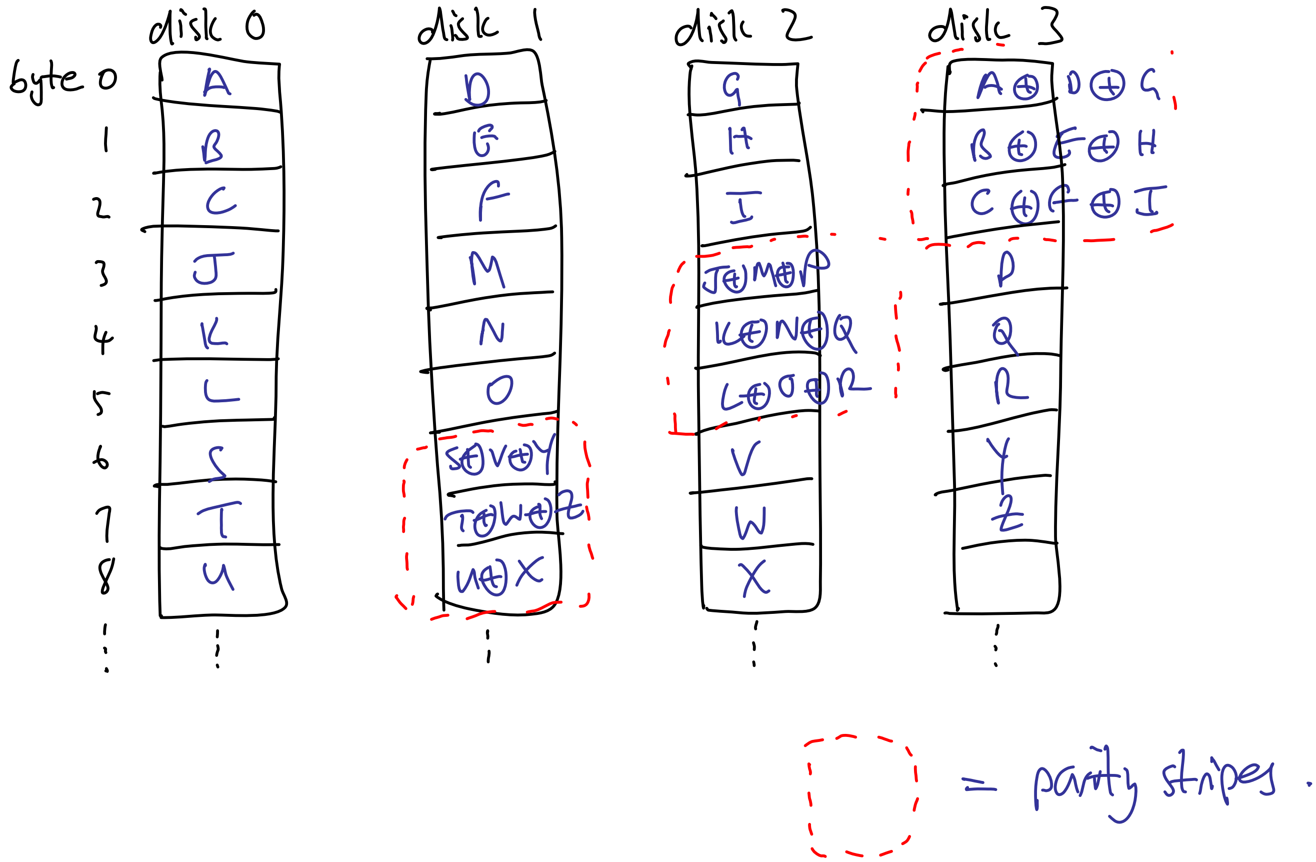
← recover by XORing everything else:

$$\begin{array}{cccc} & 1 & 0 & 1 \\ & 1 & 1 & 1 \\ \oplus & 0 & 0 & 1 \\ \hline & 0 & 1 & 1 \end{array}$$

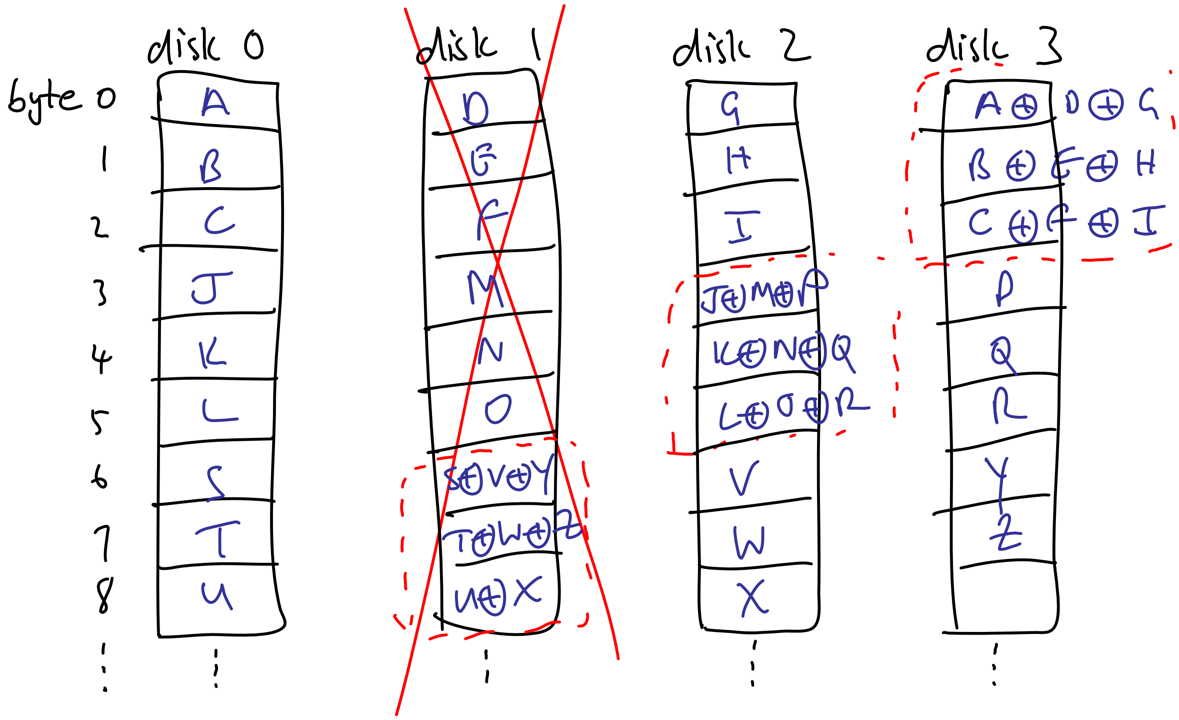
← recovered stripe

The RAID5 scheme stores parity stripes to enable recovery from failure of an entire disk.

Example, again storing "ABC..." with stripe size 3 bytes:



What happens if we lose disk 1? Recompute each stripe by XORing the appropriate stripes from the other disks.:



e.g. disk 1, byte 0 = $A \oplus G \oplus (A \oplus D \oplus G) = D$

byte 5 = $L \oplus (L \oplus O \oplus R) \oplus R = O$

byte 6 = $S \oplus V \oplus Y = (S \oplus V \oplus Y)$