537 : Persistence

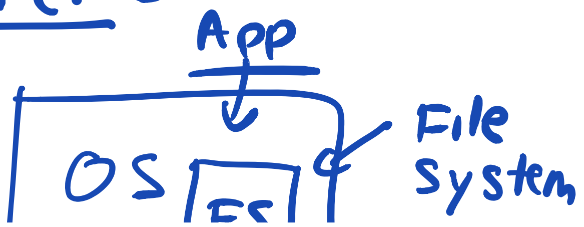
=) Hard Drive (last time)

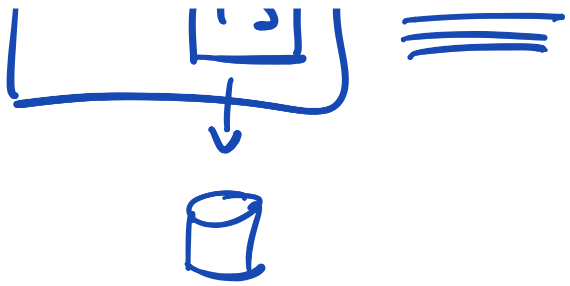
- =) cheap
- =) slow ops: milliseconds
  - =) large ops, sequential } fastest mode (~ 100 MB/s)
  - =) small ops, random } slow mode ~ 1 MB/s

Today:

=) RAID { Redundant Arrays of Independent Drives }

=) Intro to File Systems





RAID: many drives, not one

=> Why?

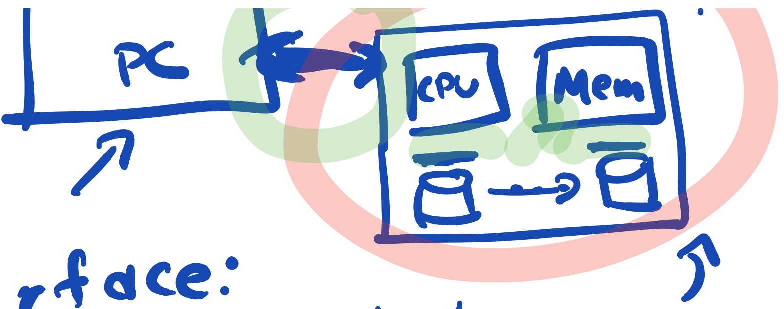
=> Performance (Bandwidth, IOPS) → I/Os per second

=> Reliability many copies: tolerate disk failure

=> Capacity

Interface:

RAID: "looks like" a disk



interface:
array of blocks,
read/write

Fault model: simple

How do hard drives fail?

⇒ drives
→ working
→ entirely not (broken)
easy to detect

RAID: {Levels}

Levels 0, 1, 4, 5 → parity-based

striping (no redundancy)

mirroring (n copies)

Metrics:

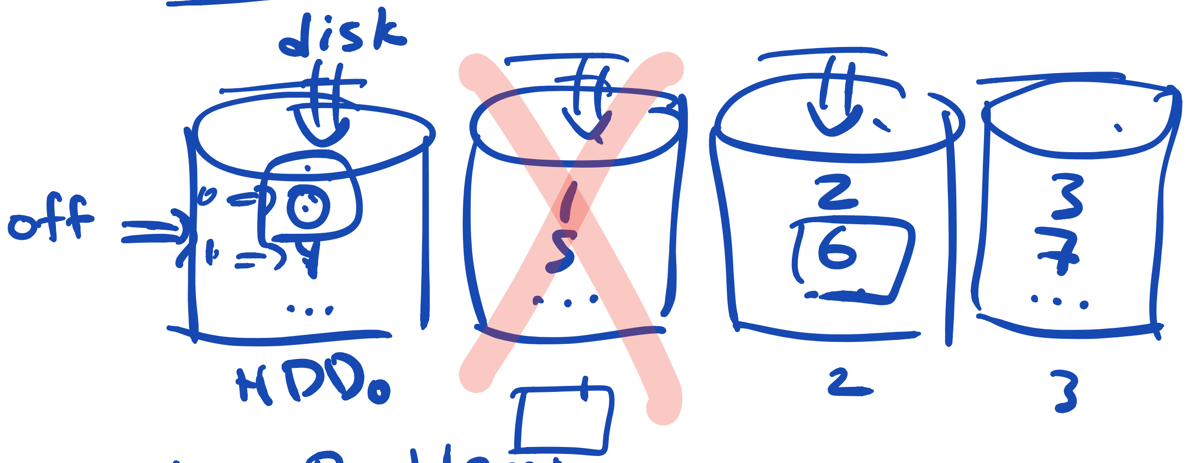
→ Perf
→ Reliability

⇒ Reliability
⇒ Capacity

RAID-0: Striping

0 .... N-1

array of read/write ⇒ blocks



Mapping Problem:

interface address ⇒ (disk, offset)

Calculation:
disk: address % num disks
offset: address / num disks

Striping:

Reliability: no redundancy ⇒ tolerate 0 failures ⇒ { " ~~bad~~ " } worse

Capacity : ⇒ "good"

N disks,
each have D bytes

[⇒ $N \cdot D$]

Performance : RAID-0

(Large)
Seq Read $N \cdot S$ MB/s } parallel
Seq. write $N \cdot S$ MB/s

N disks :

Single Disk:
⇒ S MB/s
(seq I/O bandwidth)

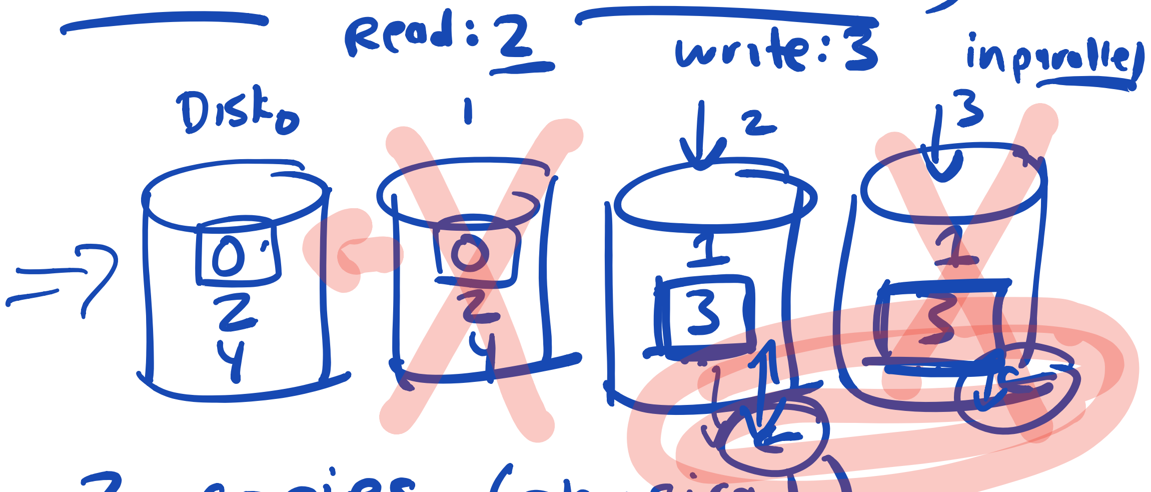
(Small)
Random Read: $N \cdot R$
Random writes: $N \cdot R$

⇒ R MB/s
(rand I/O Bandwidth)

$R \ll S$

(e.g. 1 MB/s << 100 MBs)

RAID: 1 (Mirroring)



2 copies (physical)

=> Capacity: N Drives, D bytes/drive

=> $\boxed{\frac{N \cdot D}{2}}$

Fault Model: whole drive failure

=> Reliability:

lucky: $\frac{N}{2}$

paranoid: $\boxed{1}$ (then replace)

=> Performance:

Random I/O: Bandwidth

Latency: writes => 2 a little longer than

Reads: N · R MB/s

Reads: N · R MB/s

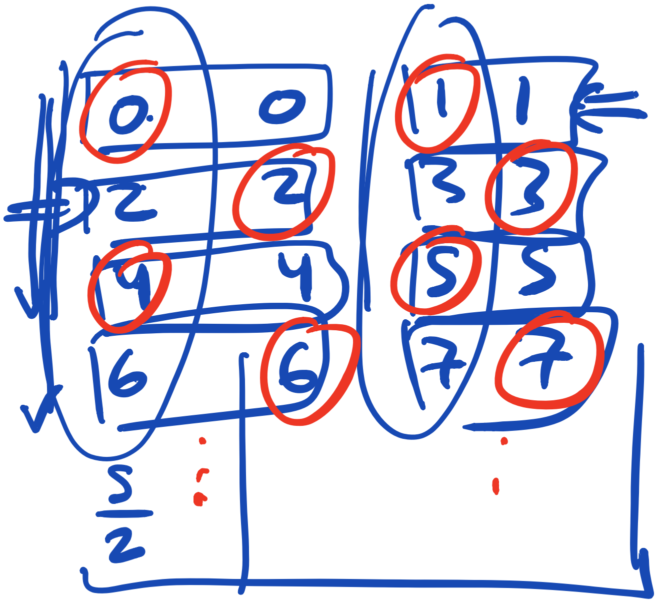
Writes: $\boxed{\frac{N}{2} \cdot R \text{ MB/s}}$ write: to 2
reads ⇒ 1

single disk: R MB/s
(~1 MB/s)

Seq R/W: $N \cdot {\sim}S \quad 2 \cdot S$

Reads: $\boxed{\frac{N}{2} \cdot S}$ MB/s ? S MB/s

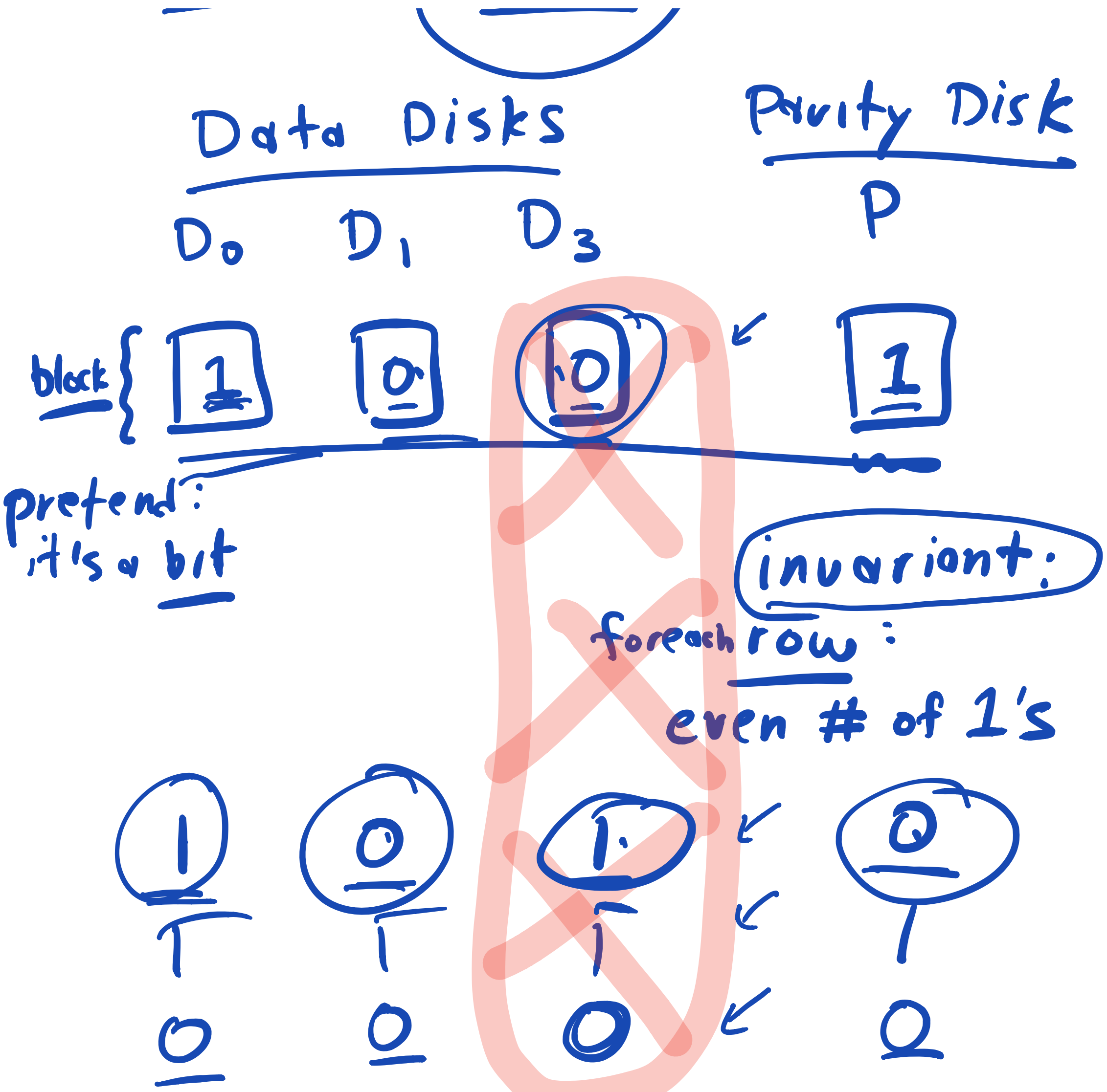
writes: $\boxed{\frac{N}{2} \cdot S}$ MB/s (~100MB/s)



N drives

RAID: Parity-based (4/5)

Parity: XOR

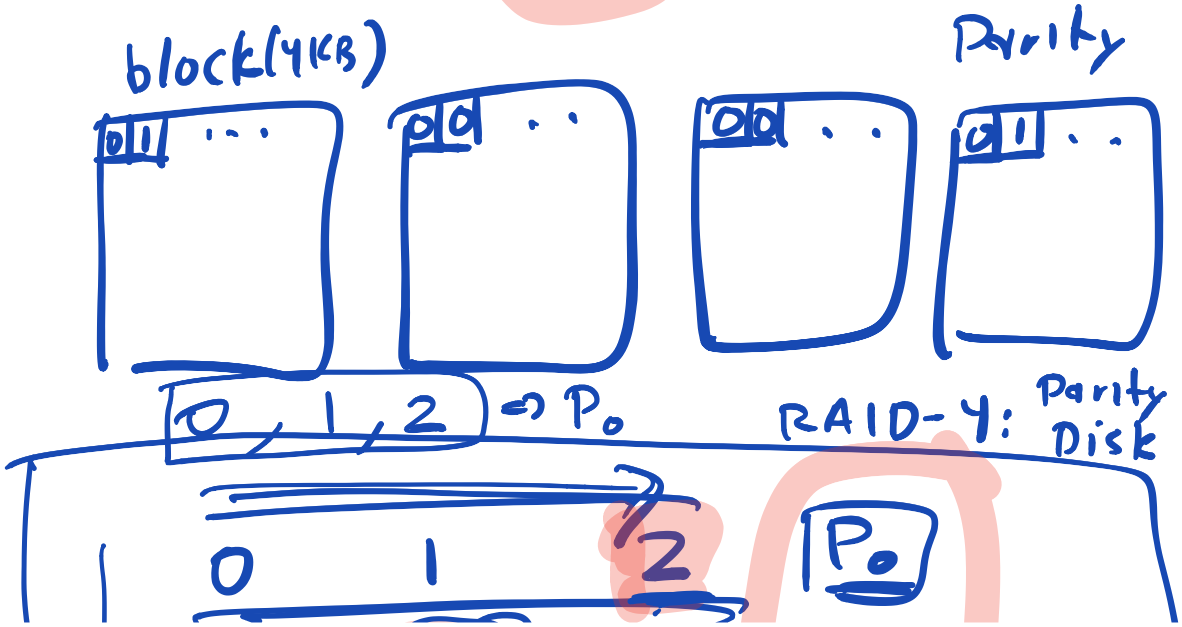


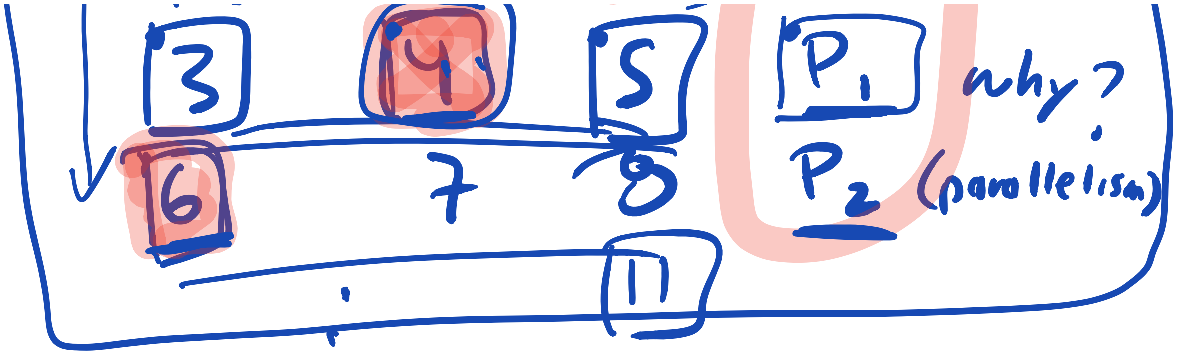
pretend: it's a bit

invariant:

foreach row:

even # of 1's





Reliability: 1 failure

Capacity: $(N-1) \cdot D$ MB

Performance:

Seq	Random
S MB/s	R MB/s

Random Reads: $(N-1) \cdot R$ MB/s

Random writes: $R/2$ MB/s

Seq Reads: $(N-1) \cdot S$ MB/s

Seq writes: $(N-1) \cdot S$ MB/s

Admin:

=> 4b due ~~next~~ monday

=> 4a ~~tomorrow~~ (concurrency) next next Monday*

=> 5 .... file systems { [Map Reduce] }

=> "final" <=

RAID-4 write: 4

write:

0	1	2	P_0
3	4	5	P_1
6	7	8	P_2
9	10	11	P_3

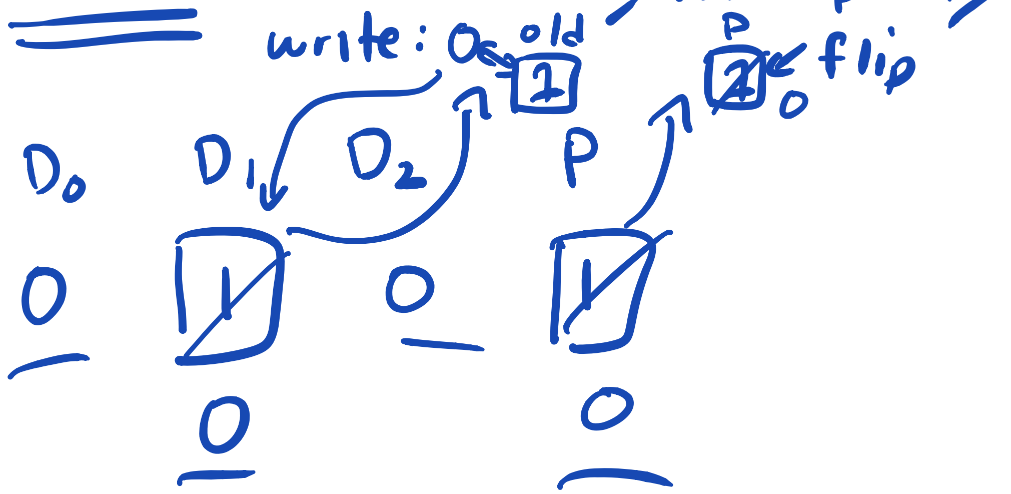
N-1

Read old data + Read old parity:

compare old data, new data => when they differ, flip in bit old parity

write

write new data, new parity



Latency: single RAID-4 write time?

latency of read or write: (T)

=> how long RAID-4 write?

I/O => (read old data, old parity)

compute => compute new parity (old data, old parity, new data)

=> write new data, new parity

⇒ ~2T RAID 4 single write

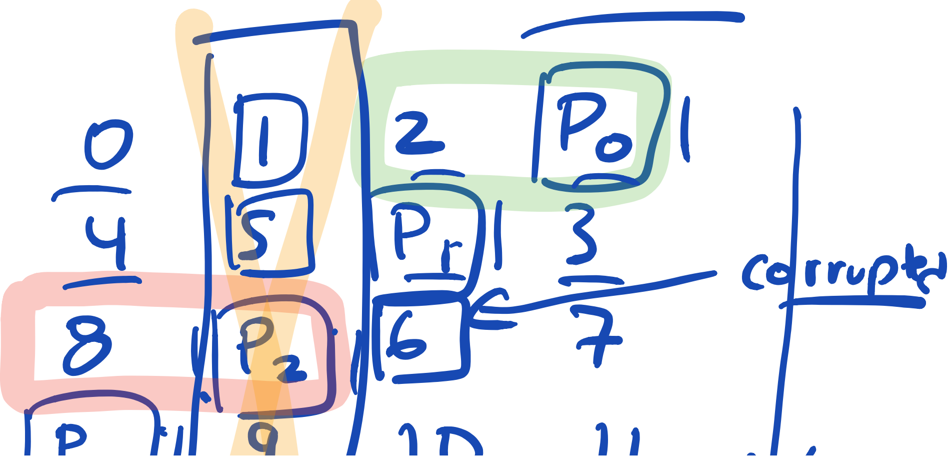
RAID-4: many small writes

⇒ Parity disk bottleneck

"small write problem"

R MB/s ⇒ random I/O bandwidth

RAID-5: Rotated Parity



Capacity, Reliability : Same as RAID-4

Performance:

Rand Read: $N \cdot R$ MB/s

Rand writes: $\frac{N}{4} \cdot R$ MB/s ← RAID-1: mirroring $\frac{N}{2} \cdot R$ MB/s

Seq Read : $(N-1) \cdot S$

Seq write : $(N-1) \cdot S$

capacity, Seq I/O, but

RAID-5: 4 I/Os / logical write

vs

Mirroring : small write perf (high capacity cost)

RAID:

Faster, larger, more reliable disk

=> 0, 1, 4, 5 6 => 2 parity disks

=> checksums : detect/ recover

from corruption

$\Rightarrow$