Persistence: RAID

Ram Alagappan

CS 537, Spring 2019

Administrivia

P2b, midterm, P3 grades posted

- Midterm solutions available
- for regrades, see Piazza

P4 due on 4th , Thursday

Learning Outcomes

Why more than one disk?

What are the different RAID levels?

(striping, mirroring, parity)

How do we compare the RAID levels? What metrics?

Only One Disk?

Typical scenario

Sometimes we may need many…Why?

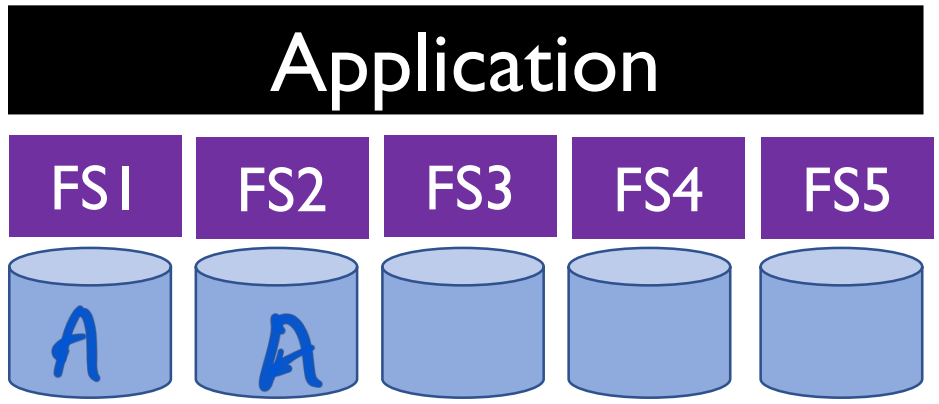
1. Capacity
2. Performance
3. reliability

Challenge: most file systems work with one disk

Solution-1: JBOD

Just a Bunch Of Disks

Applications store data on different FS

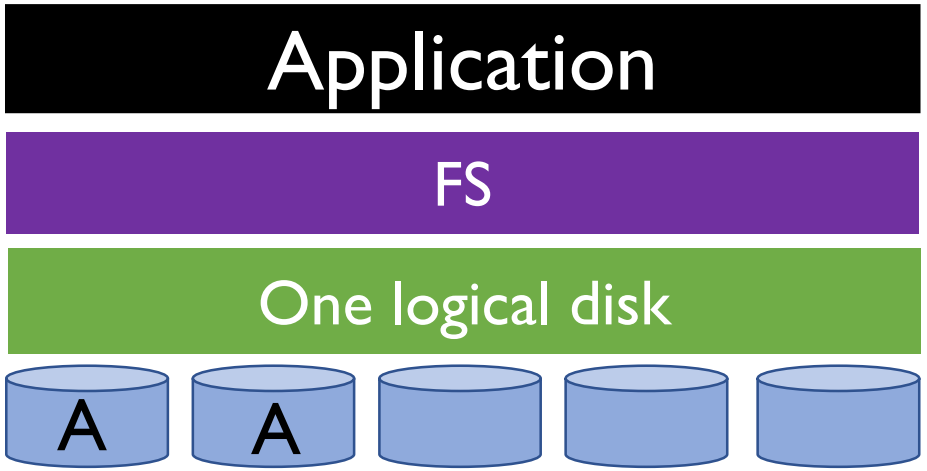


e.g., critical data: app decides to replicate

Downsides: need to know multiple devices, need to be rewritten, not deployable

Solution-2: RAID

Redundant Array of Inexpensive (Independent) Disks



one logical disk from

Many physical disks

Advantages: transparent to apps, deployable

Improved capacity, performance, and reliability!

Fault Model

Simple: fail-stop model

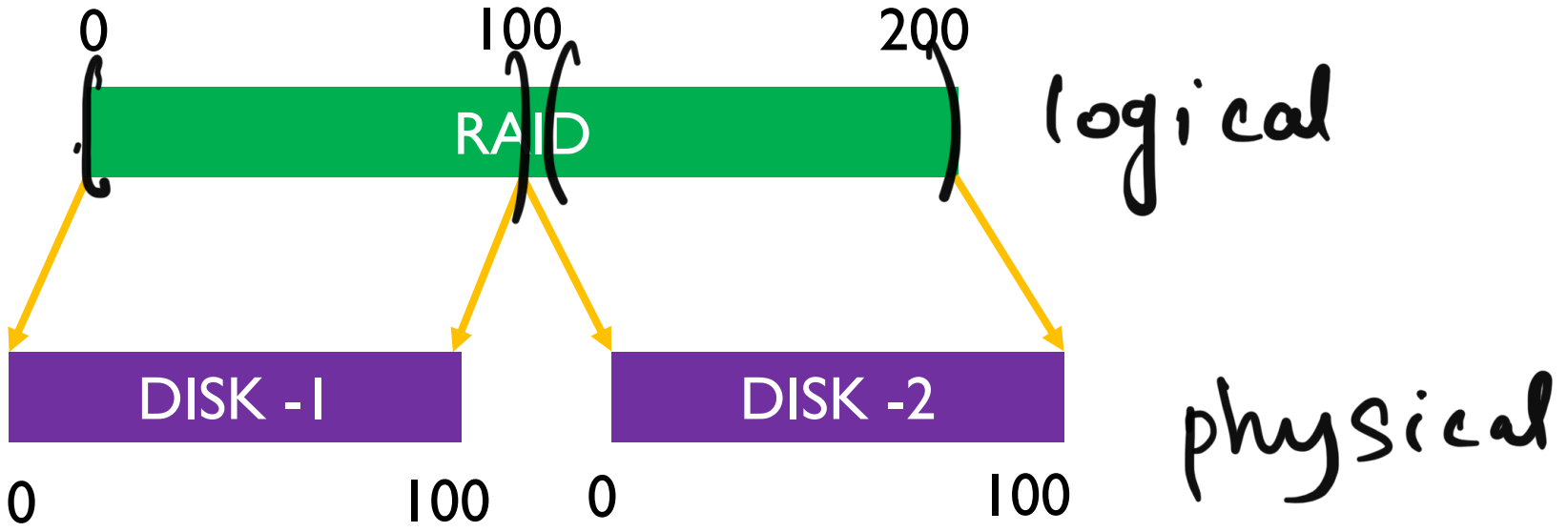
Works correctly or fails entirely

Easy to detect: system knows when not working

No silent failures: no data corruption etc.,

General Strategy: Mapping

Build fast, large disk from smaller ones



Mapping

How should we map logical block addresses to physical block addresses?

Is this similar to something?

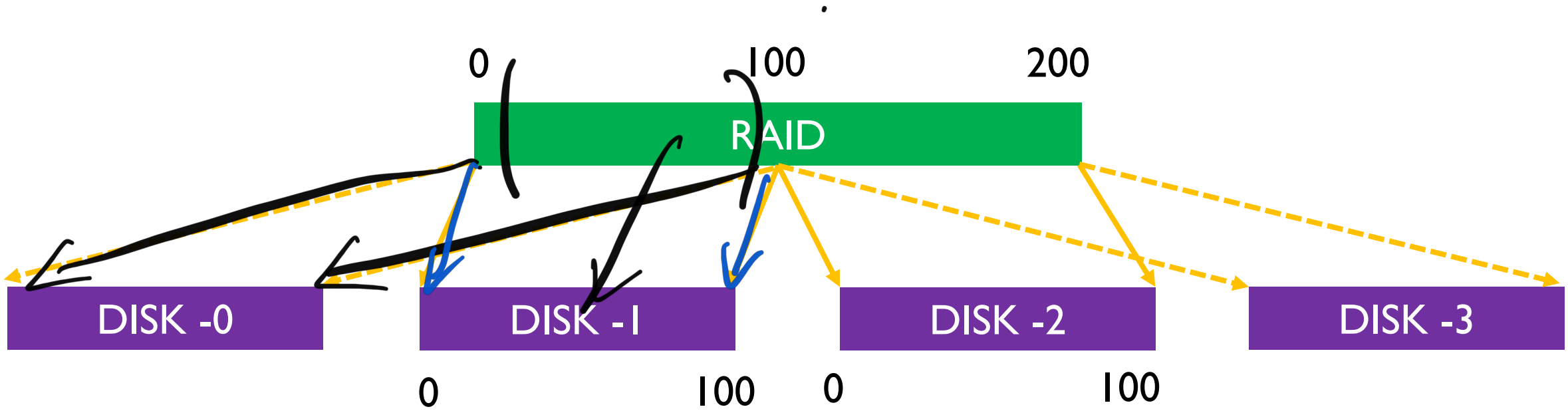
virtual memory

Virtual memory - dynamic mapping: page tables

RAID - Static mapping: simple calculation

General Strategy: Redundancy

Add even more disks for reliability



Redundancy

Trade-offs to amount of redundancy

Increase number of copies

- improves reliability (and maybe performance)

Decrease number of copies (deduplication)

- improves space efficiency

RAID Analysis

RAID: different levels

Workload: types of reads/writes issued by app

Metric: capacity, reliability, performance

Given Workload, Raid level, determine Metric

RAID Levels

Levels –

0 (striping)

1 (mirroring)

4 (parity)

5 (rotated parity)

2, 3, 6

We'll not discuss these

logical : 1 - 1000 blocks

Workload

Throughput / BW → MBs/s

Single request

- read
- write

e.g., read (51)

Write (5)

latency - t

Many requests

- random
  - reads
  - writes

read: 0, 51, 78, 767

roo...

- sequential
  - reads
  - writes

eg., read (100 - 500)

Metrics

Capacity: how much space can apps use?

Reliability: how many disks can we safely lose?

Performance: latency? throughput?

Metrics

$C \sim 500\,GB$

$S \sim 100\,MB/s$

$R \sim 1\text{-}7\,MB/s$

$S \gg R$

Normalize each to characteristics of one disk

N := number of disks

C := capacity of 1 disk

S := sequential throughput of 1 disk

R := random throughput of 1 disk

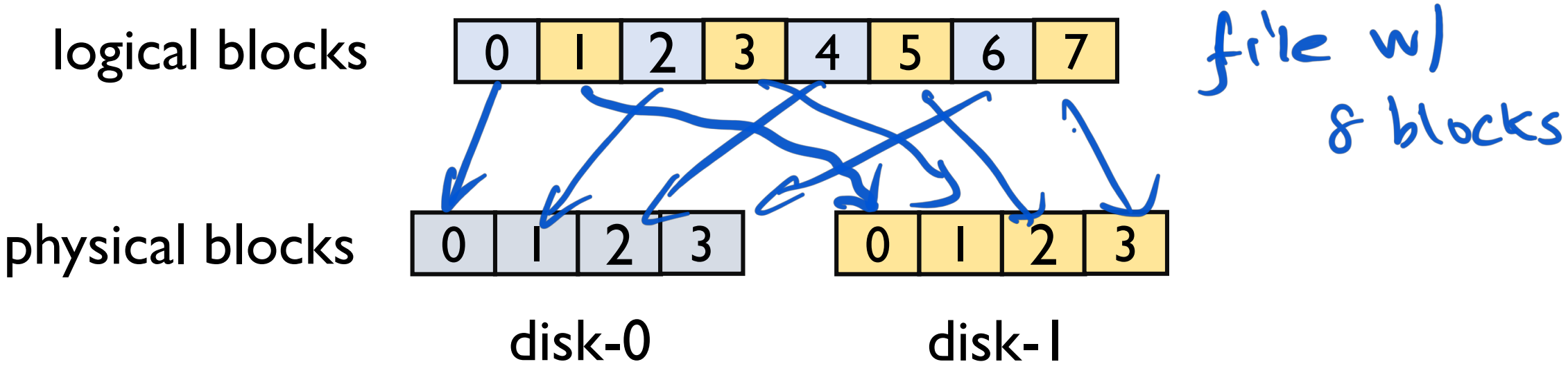
D := latency of one small I/O operation

Reasonable S? Reasonable R? How does S compare to R?

Assumptions. Symmetric perf. $r == w$

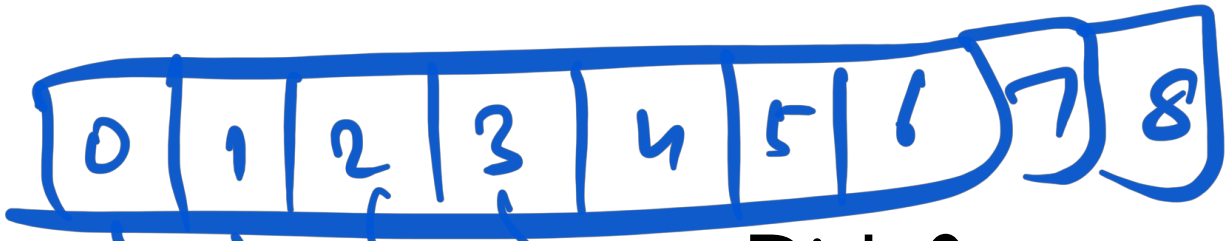
RAID-0: Striping

Optimize for capacity; no redundancy!



file w/ & blocks

RAID-0: Another View



Disk 0	Disk 1
0	1
2	3
4	5
6	7

4 Disks

	Disk 0	Disk 1	Disk 2	Disk 3
	0	1	2	3
Stripe	4	5	6	7
	8	9	10	11
	12	13	14	15

How to Map?

Given logical address A, find:

Disk = A % disk_count, Offset = A / disk_count

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ex. 10

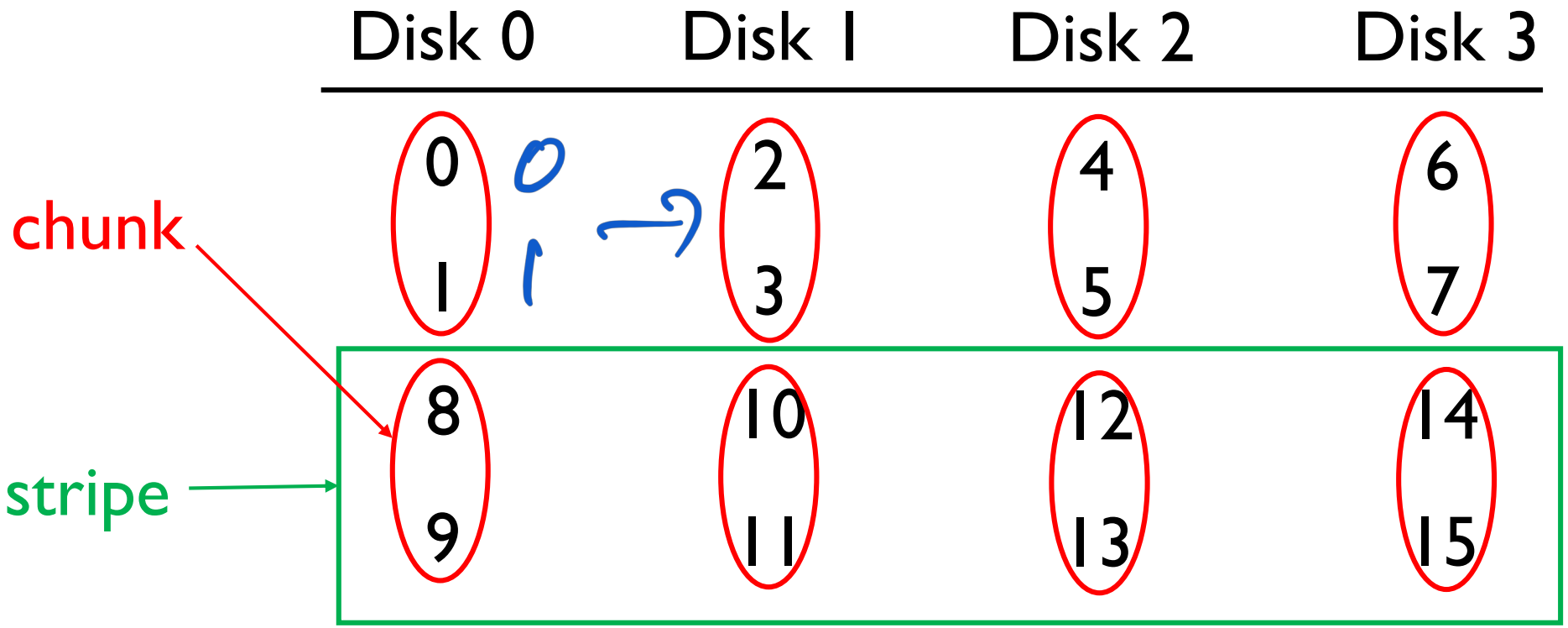
Disk# = 10 % 4
= 2

Offset = 10 / 4
= 2

Chunk Size = 1

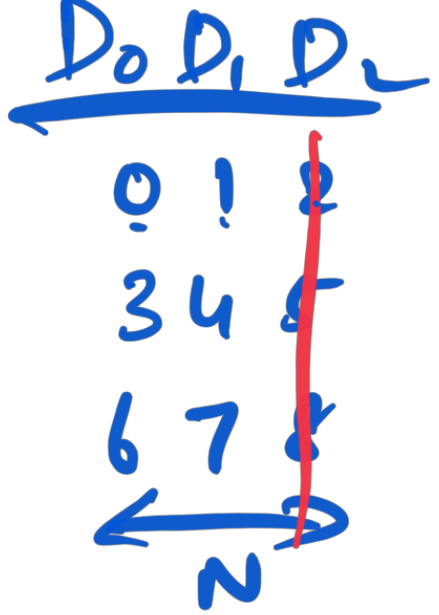
Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Chunk Size = 2



We'll assume chunk size of 1 for today. Sizes of 64KB are typical in deployment.

RAID-0: Analysis



What is the capacity? N · C

How many disks can fail? 0 - worse than single-disk in some cases?

Throughput (sequential, random)? Seq - read — N · S
- write — N · S

rand - r — N · R
- w — N · R

Latency — D

r : D, w : D

RAID-0: Analysis Results

What is the capacity?

N * C

upper bound

How many disks can fail?

0

bad

Throughput (sequential, random)?

N*S, N*R

upper bound

Latency

D

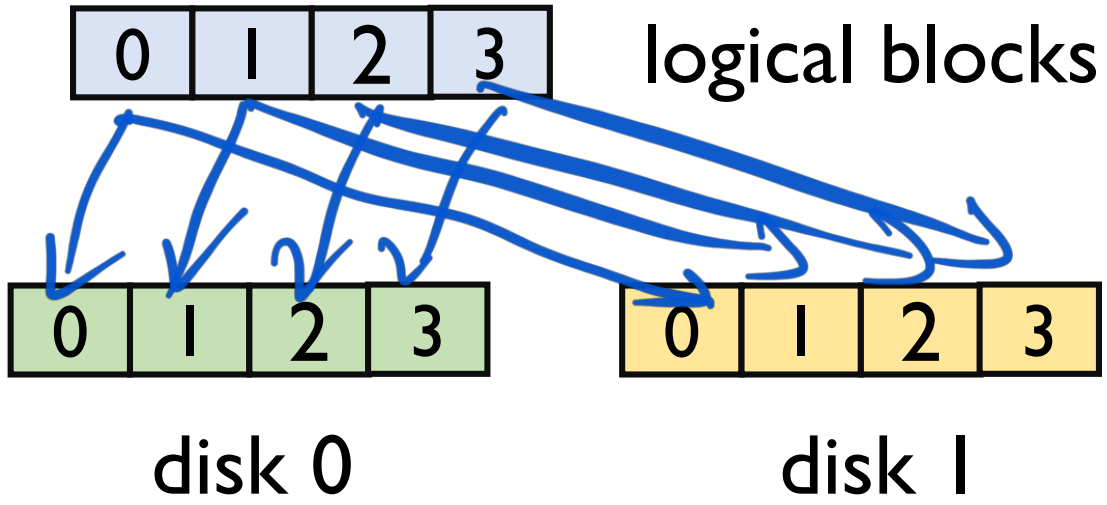
buying more disks improves throughput, but not latency!

RAID-1: Mirroring

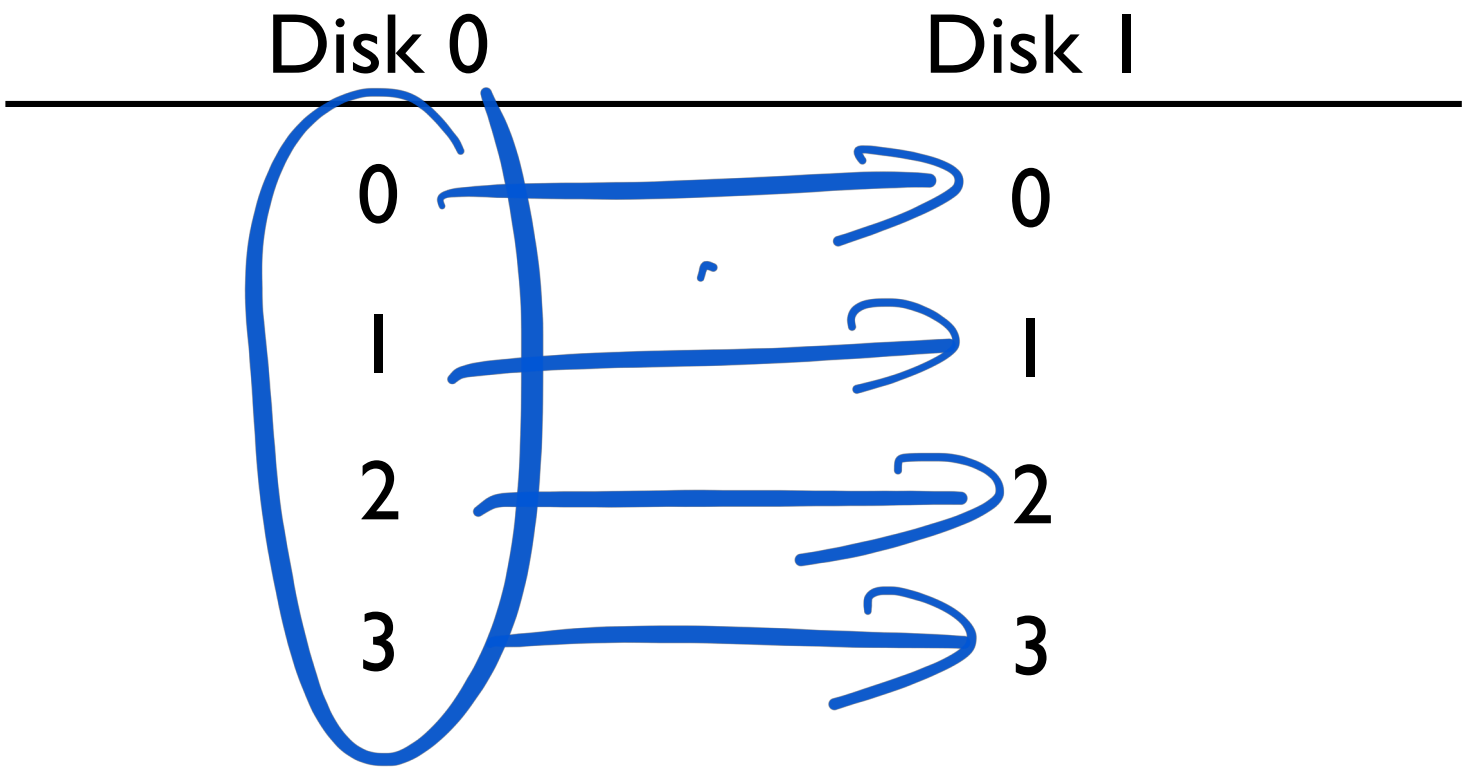
Problem w/ Raid-0: Poor reliabi-lity. Solution?

Keep two copies of all data

file w/ 4 blocks



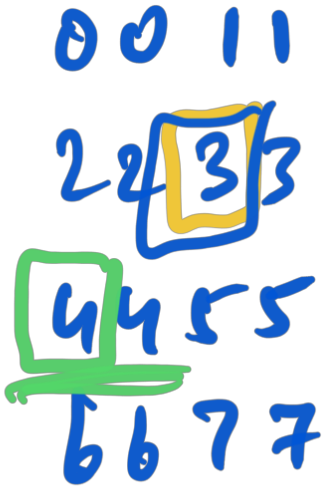
2 Disks



4 Disks

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

RAID-1: Analysis



What is the capacity? $\frac{N}{2} \cdot C$ [only half effective capacity]

How many disks can fail? 1 for sure, $\frac{N}{2}$ if lucky

Latency (read, write)?

are writes same as one disk? slower? faster?

r: D, w: D (2 writes but still only D, because we can do them in parallel)

RAID-1: Analysis Results

What is the capacity?

N/2 * C

How many disks can fail?

one (maybe N/2 if lucky)

Latency (read, write)?

read = D, write =D

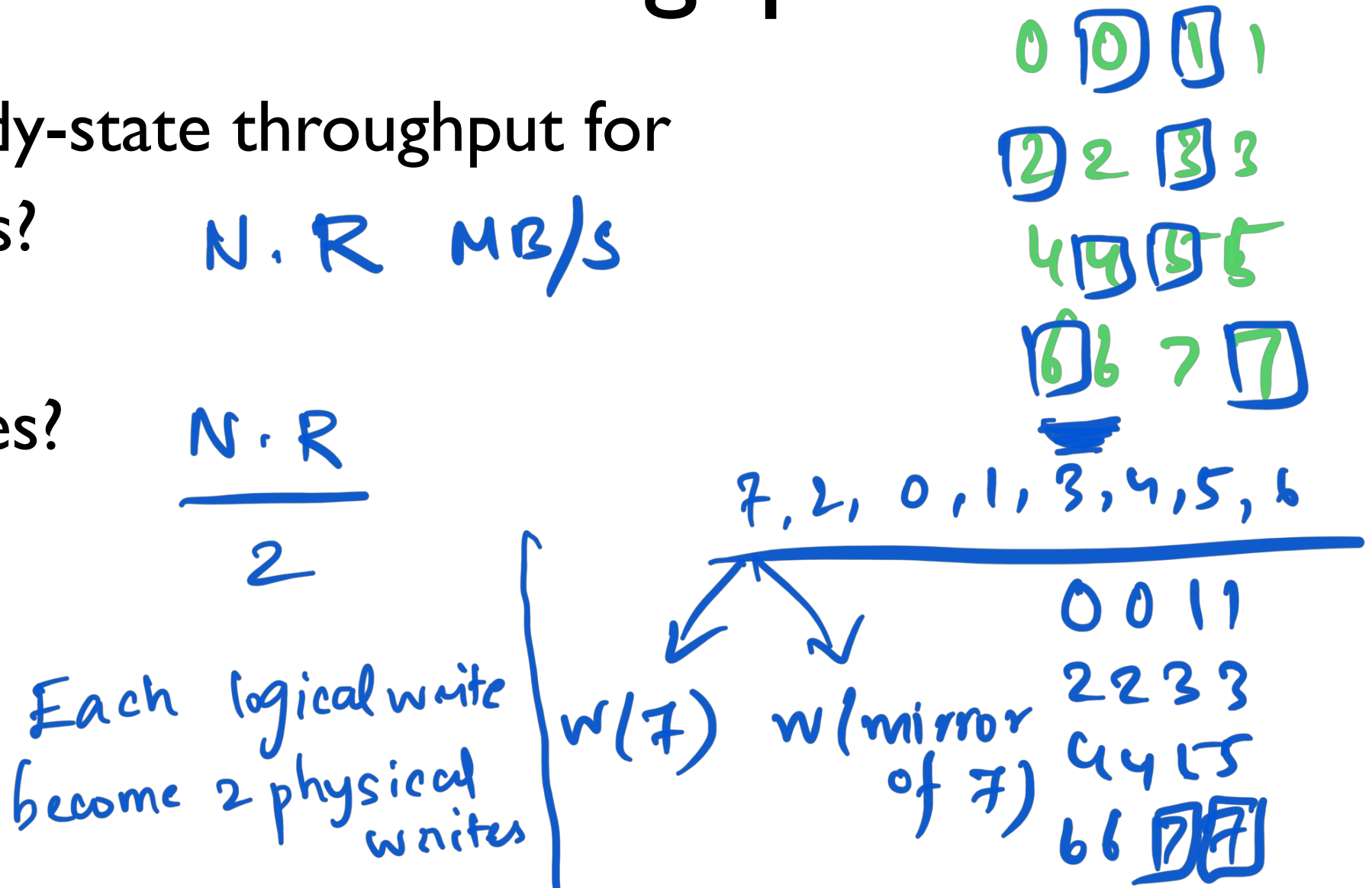
RAID-1 Throughput

What is steady-state throughput for random reads?

N · R MB/s

random writes?

$\frac{N \cdot R}{2}$



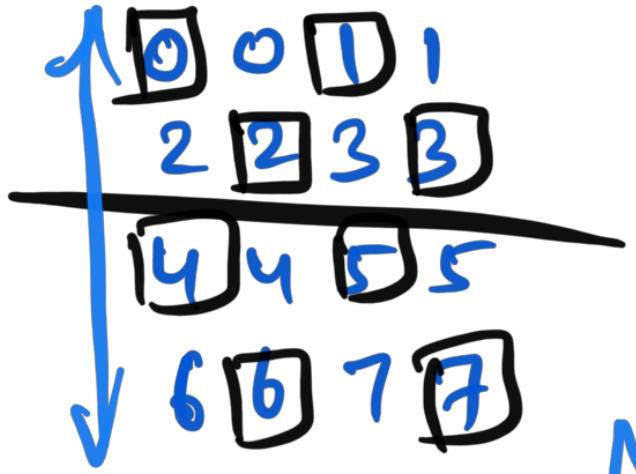
RAID-1 Throughput

What is steady-state throughput for sequential writes?

$\frac{N \cdot S}{2}$

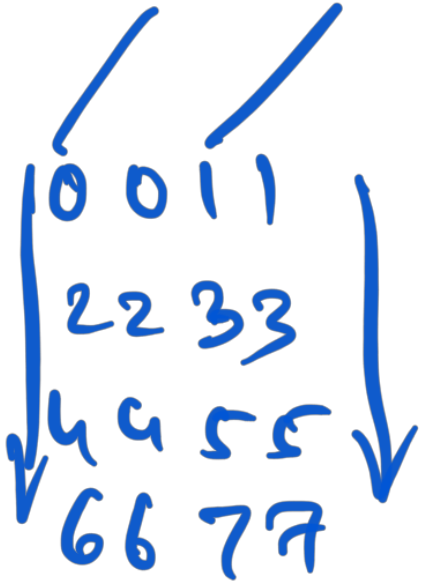
sequential reads?

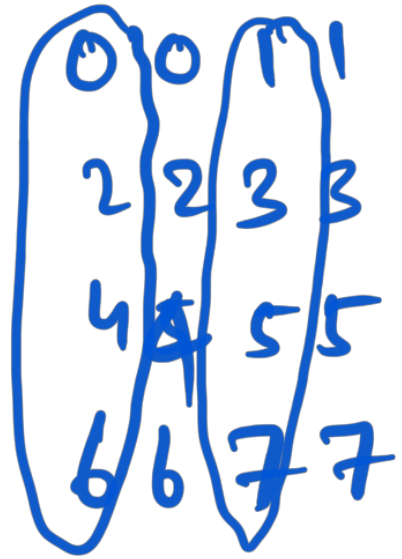
$N \cdot S$



$\frac{N \cdot S}{2}$

$N \cdot S / 2$





N. S.

0 - 7

0 - 7

$\frac{N}{2} \cdot S$

RAID-1 Throughput Results

What is steady-state throughput for

random reads?

N * R

same as raid-0

random writes?

N/2 * R

sequential writes?

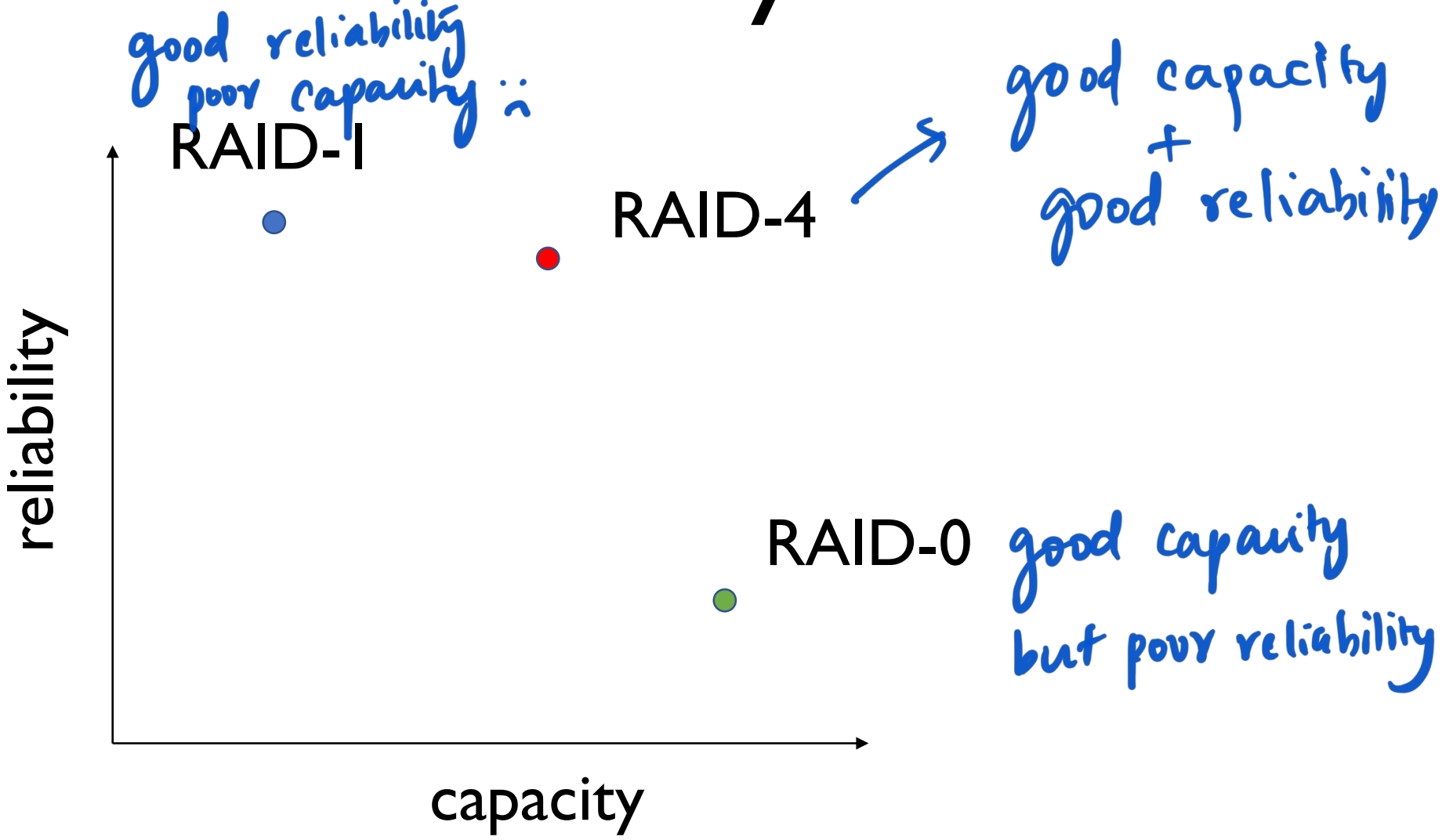
N/2 * S

sequential reads?

N/2 * S

only half of raid-0

RAID-4 Parity



RAID-4 Strategy

Stripe with parity

In algebra, if an equation has N variables, and N-1 are known, you can often solve for the unknown

Treat sectors across disks in a stripe as an equation

Data on bad disk is like an unknown in the equation

Saving Space with Parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

$P_3 = 0 \oplus 1 \oplus 2 \oplus 3$

XOR.

XOR Parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
4	5	6	7	P1
8	9	10	11	P2
12	13	14	15	P3

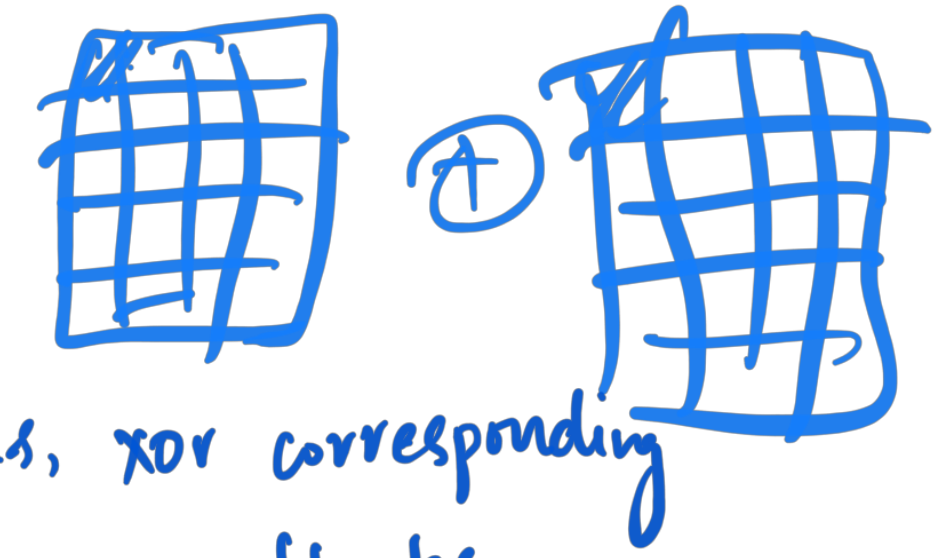
P0 = contents of 0 xor contents of 1 xor contents of 2 xor contents of 3

Parity Invariant

→ # 1's to be even

D_0	D_1	D_2	P
0	1	0	1
1	1	1	1
0	1	1	0
0	0	0	0

Pretend bits instead of blocks



for blocks, xor corresponding bits of the blocks

Data Reconstruction in Raid-4

Access 2nd block on D_2 but D_2 has failed

Reconstruct from other disks using parity invariant. We have three ones - so missing must be a 1 (not 0)

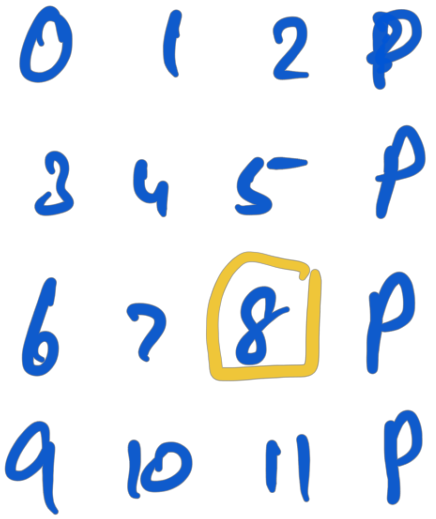
RAID-4 Analysis

What is the capacity? $(N-1) \cdot C$

How many disks can fail? 1

Latency (read, write)?

- read: D
- write: 2D (need to read old data, old parity; need to write new data, new parity)



Updating Parity on Writes

Ex:

0	1	2	P
3	4	5	P
6	7	8	P
9	10	11	P

Write (8) $8_{old} \rightarrow 8_{new}$

Approach 1: read 6, 7, we have 8_{new}

$P_{new} = 6 \oplus 7 \oplus 8_{new}$

Approach 2: read 8_{old}, P_{old}; calc P_{new}

if $8_{new} == 8_{old}$;

$P_{new} = P_{old}$

else

$P_{new} = (P_{old})'$

Latency (write)?

RAID-4 Analysis Results

What is the capacity? (N-1) * C

How many disks can fail? one

Latency (read, write)? D, 2D

RAID-4 Throughput

What is steady-state throughput for –

sequential reads? $(N-1) \cdot S$

sequential writes (full-stripe write)?

$P_0 = 0 \oplus 1 \oplus 2$

$(N-1) \cdot S$

0, 1, 2, P_0

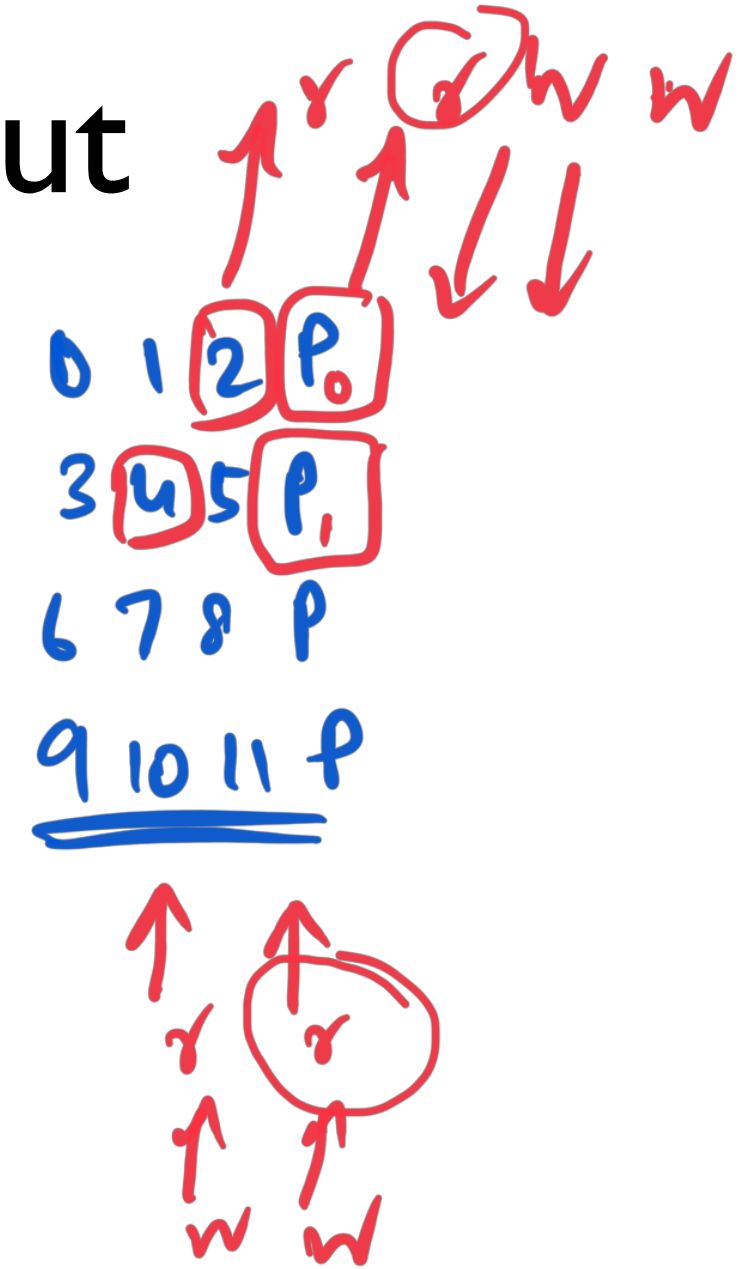
↳ calc. from memory

RAID-4 Throughput

What is steady-state throughput for –

random reads? $(N-1) \cdot R$

random writes? $\frac{R}{2}$ (this is very bad!)



random write perf. in RAID-4 is bottlenecked by the parity dist

RAID-4 Small-write Problem

Performance of random writes?

$R/2$ (notice: no $N!$)

RAID-4 Throughput Results

What is steady-state throughput for –

sequential reads? (N-1) * S

sequential writes? (N-1) * S

random reads? (N-1) * R

random writes? R/2

RAID-5: Rotated Parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
-	-	-	-	P0
-	-	-	P1	-
-	-	P2	-	-
-	P3	-	-	-
P4	-	-	-	-

RAID-5 Analysis

What is the capacity?

How many disks can fail?

Latency (read, write)?

RAID-5 Analysis Results

What is the capacity? (N-1) * C

How many disks can fail? one

Latency (read, write)? D, 2D

RAID-5 Throughput

What is steady-state throughput for –

sequential reads?

sequential writes?

random reads?

random writes?

RAID-5 Throughput Results

What is steady-state throughput for –

sequential reads? (N-1) * S

sequential writes? (N-1) * S

random reads? (N) * R

random writes? N/4 * R

RAID Summary

	Reliability	**Capacity**
RAID-0	0	C*N
RAID-1	1	C*N/2
RAID-4	1	C*(N-1)
RAID-5	1	C*(N-1)

RAID Summary

	Read latency	**Write latency**
RAID-0	D	D
RAID-1	D	D
RAID-4	D	2D
RAID-5	D	2D

RAID Summary

	Seq read	**Seq write**	**Rand read**	**Rand write**
RAID-0	N*S	N*S	N*R	N*R
RAID-1	N/2 * S	N/2 * S	N*R	N/2 * R
RAID-4	(N-1)*S	(N-1)*S	(N-1)*R	R/2
RAID-5	(N-1)*S	(N-1)*S	N*R	N/4*R

RAID-0 is always fastest and has best capacity (but at cost of reliability)

RAID-5 is strictly better than RAID-4

RAID-5 better than RAID-1 for sequential

RAID-1 better than RAID-4 for random write

Summary

RAID: a faster, larger, more reliable disk system

One logical disk built from many physical disk

Different mapping and redundancy schemes

Present different trade-offs