Persistence: RAID

Sridhar Alagar

Many Disks

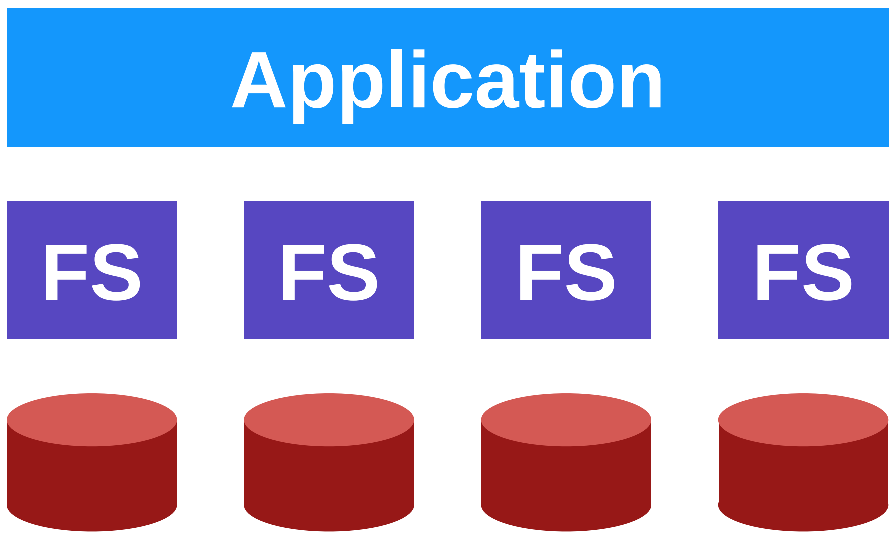
- We often want
  - More capacity
  - Better reliability
  - High performance

- Many inexpensive disks can be used together
  - Alternative: Buy a hig end expensive disk

- Challenge: FS work only on one disk

Solution 1: JBOD

- Just a Bunch of Disks together

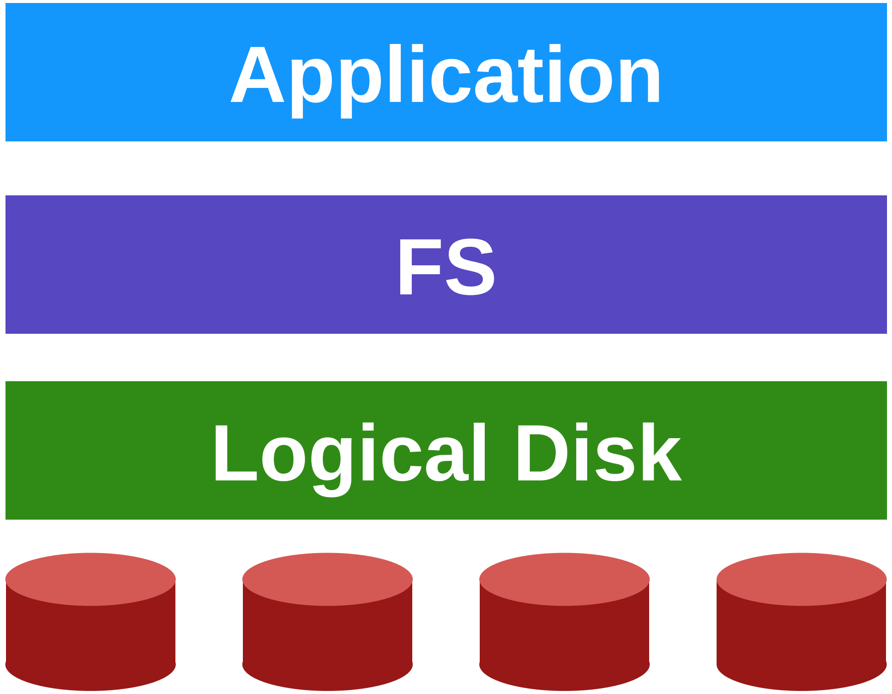


- Application needs to be smart to store files across different disks

Solution 2: RAID

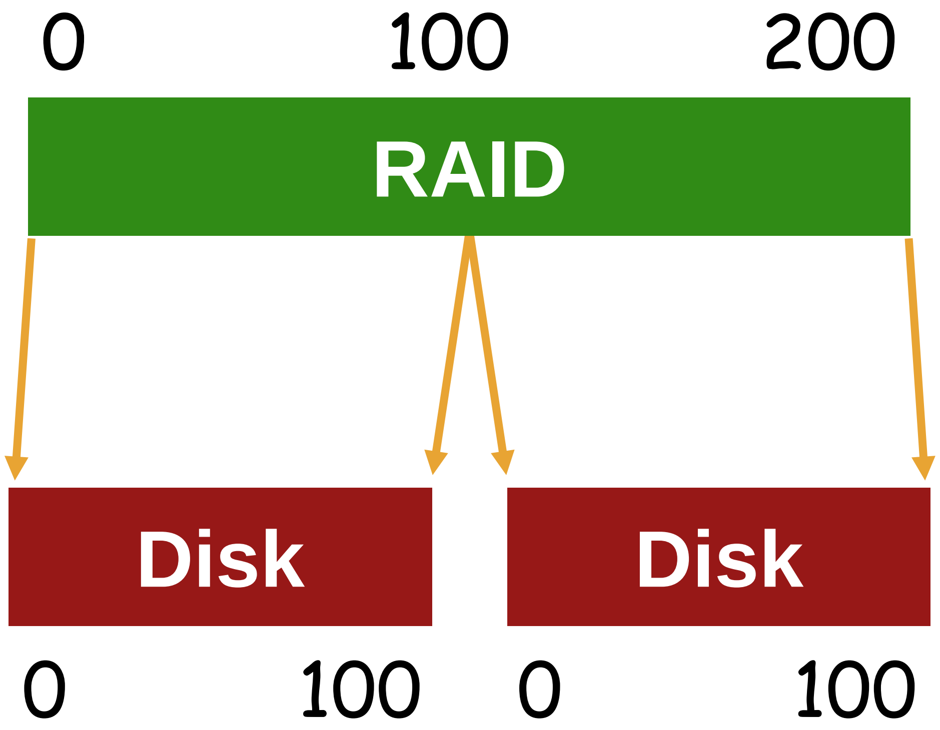
- Redundant Array of Inexpensive Disks
  - Build logical disks from many disks

- RAID provides
  - Capacity
  - Performance
  - Reliability
- RAID is
  - Transparent
  - Easily deployable



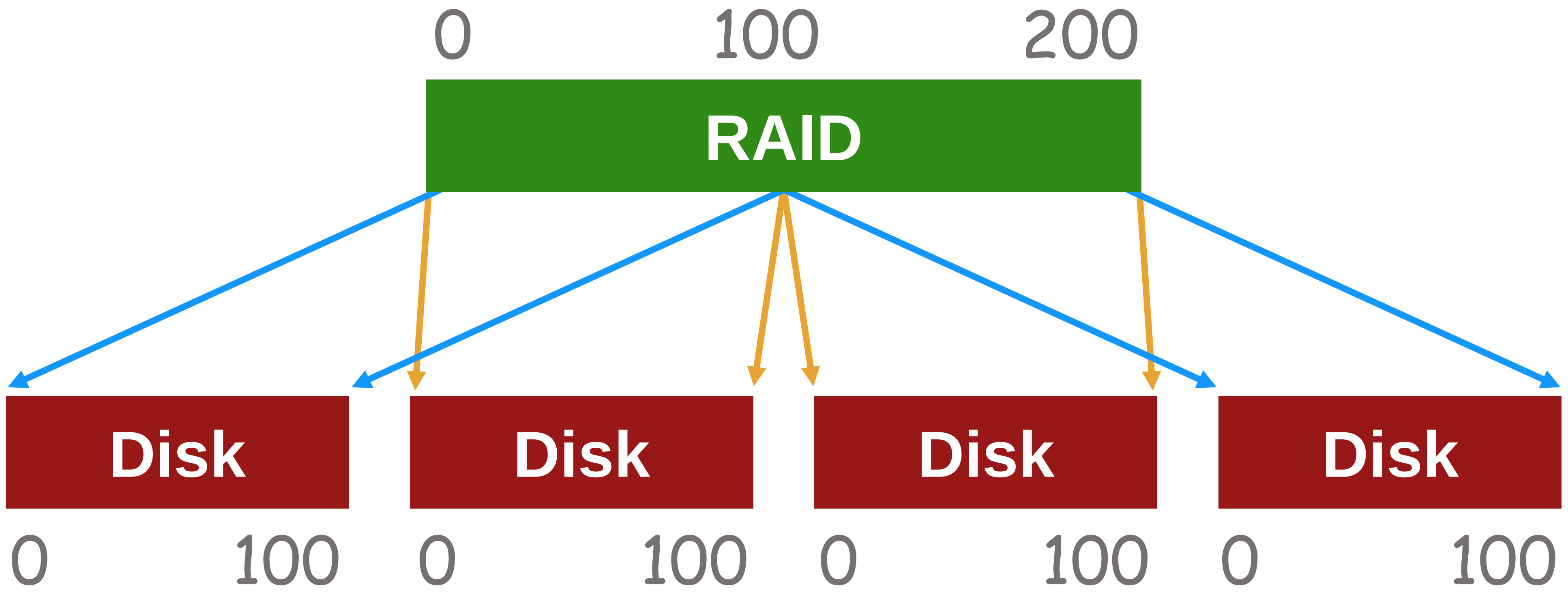
General Strategy: Mapping

- Build fast, large disk from many smaller ones



General Strategy: Redundancy

- Add more disks for reliability



RAID Levels

- Which logical blocks map to which physical blocks?
- How do we use extra physical blocks (if any)?
- Different RAID levels make different trade-offs

Workloads

- Reads
  - One operation
  - Steady-state I/O: Sequential and Random

- Writes
  - One operation
  - Steady-state I/: Sequential and Random

Metric

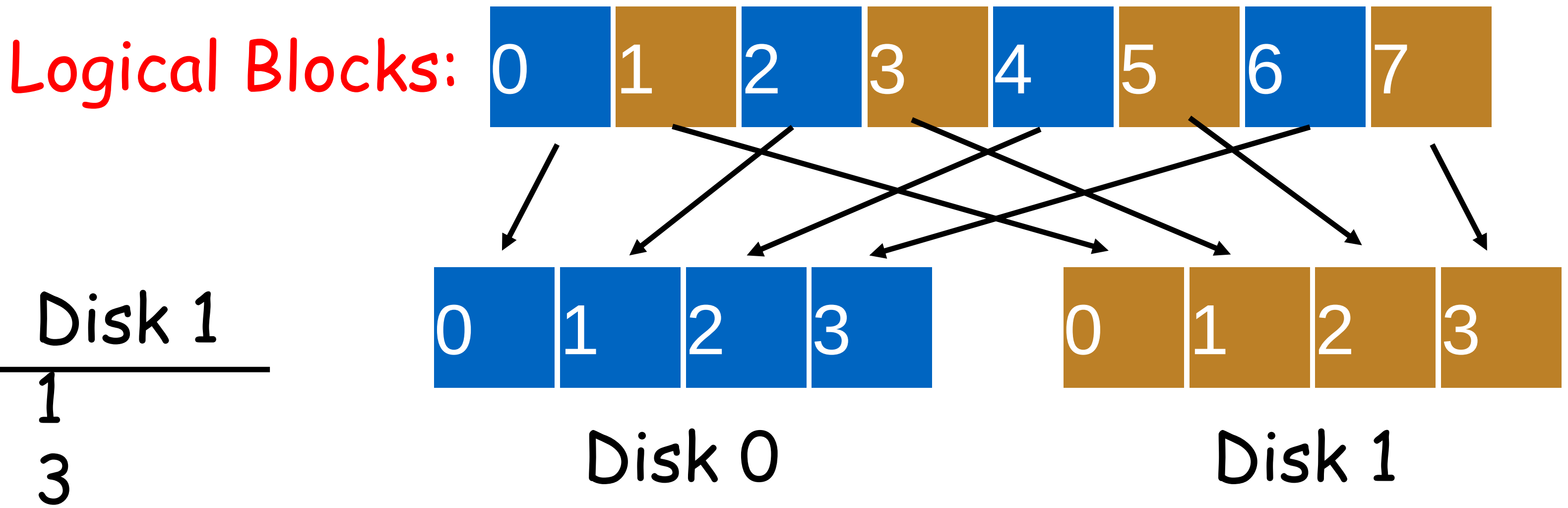
- Capacity: how much space can we use?
- Reliability: how many disks can we safely lose? (assume fail stop!)
- Performance: how long does each workload take?
- Normalize each to characteristics of one disk

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

RAID-0: Striping

Optimize for capacity. No redundancy

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



RAID - 0: Striping 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

Given logical address A, find:

Disk = ...

Offset = ...

Disk = A % N

Offset = A / N

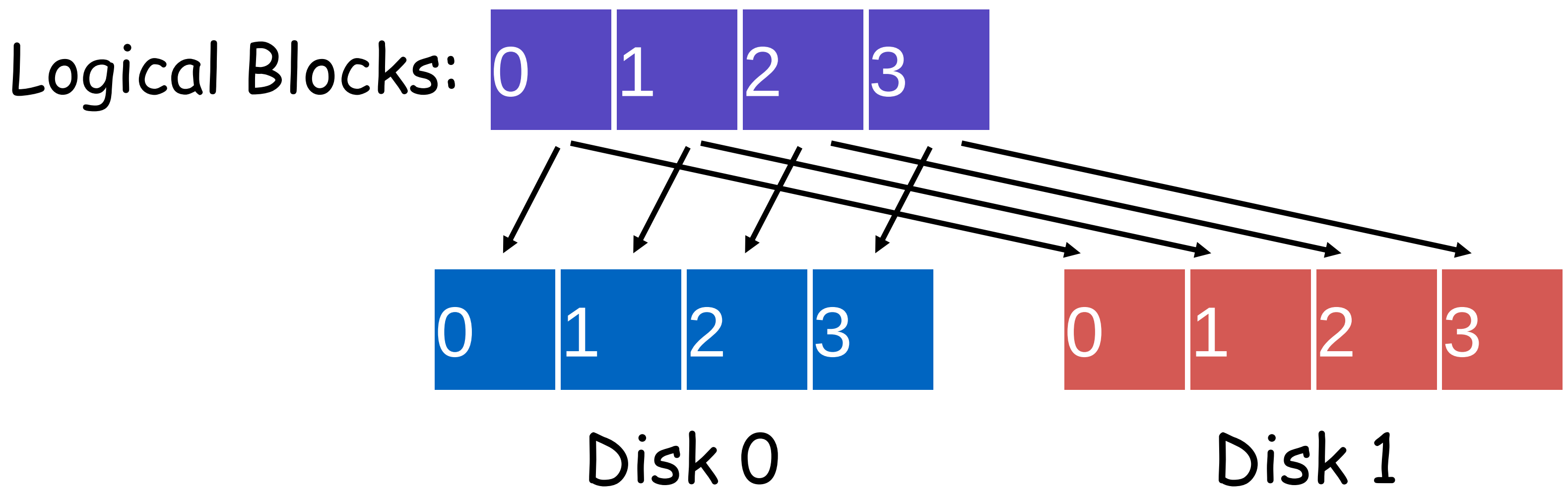
RAID-0: Analysis

What is capacity?	N * C
How many disks can fail?	0
Latency	D
Throughput (sequential, random)?	N*S , N*R

Buying more disks improves throughput, but not latency!

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

RAID-1: Mirroring



Keep two copies of all data.

Raid-1 Layout

2 disks

Disk 0	Disk 1
0	0
1	1
2	2
3	3

4 disks

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

RAID-1: Analysis

What is capacity?	N/2 * C
How many disks can fail?	1 (may be up to N/2)
Latency	D

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

RAID-1: Throughput

What is steady-state throughput for

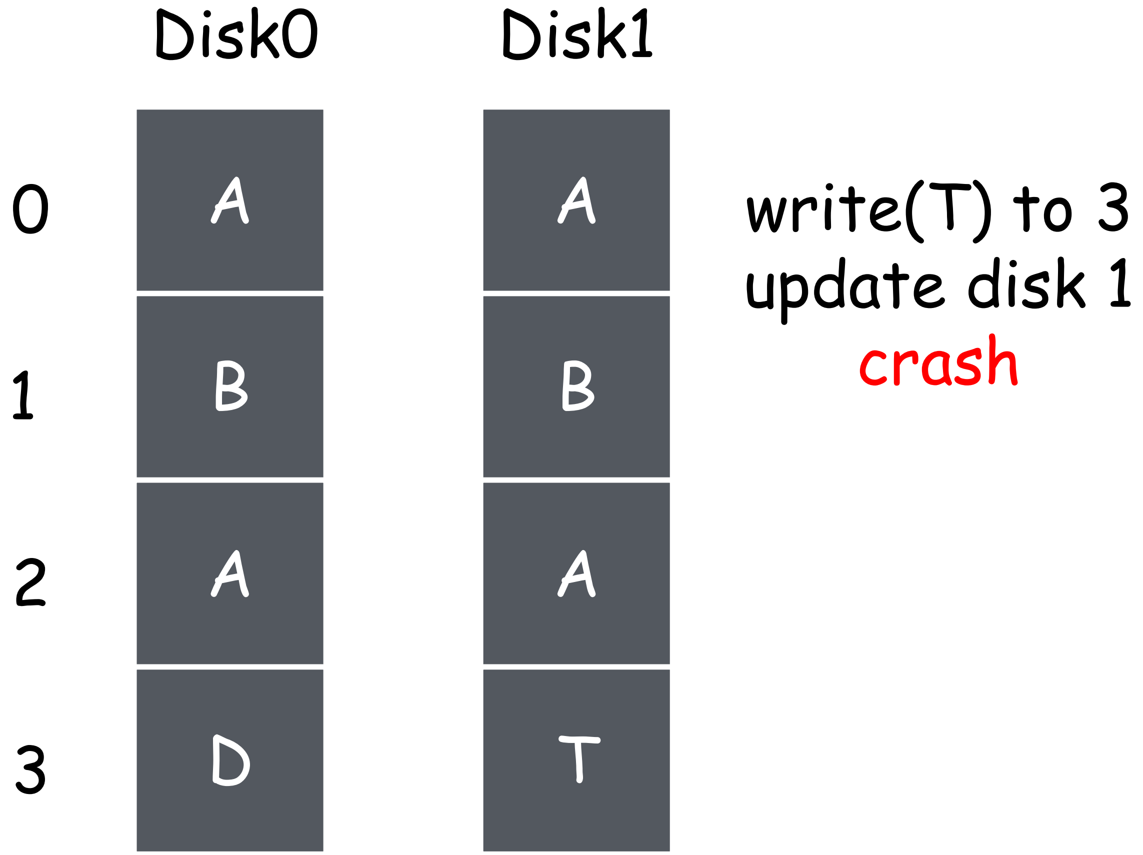
- random reads? N * R
- random writes? N/2 * R
- sequential writes? N/2 * S
- sequential reads? Book: N/2 * S (other models: N * S)

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

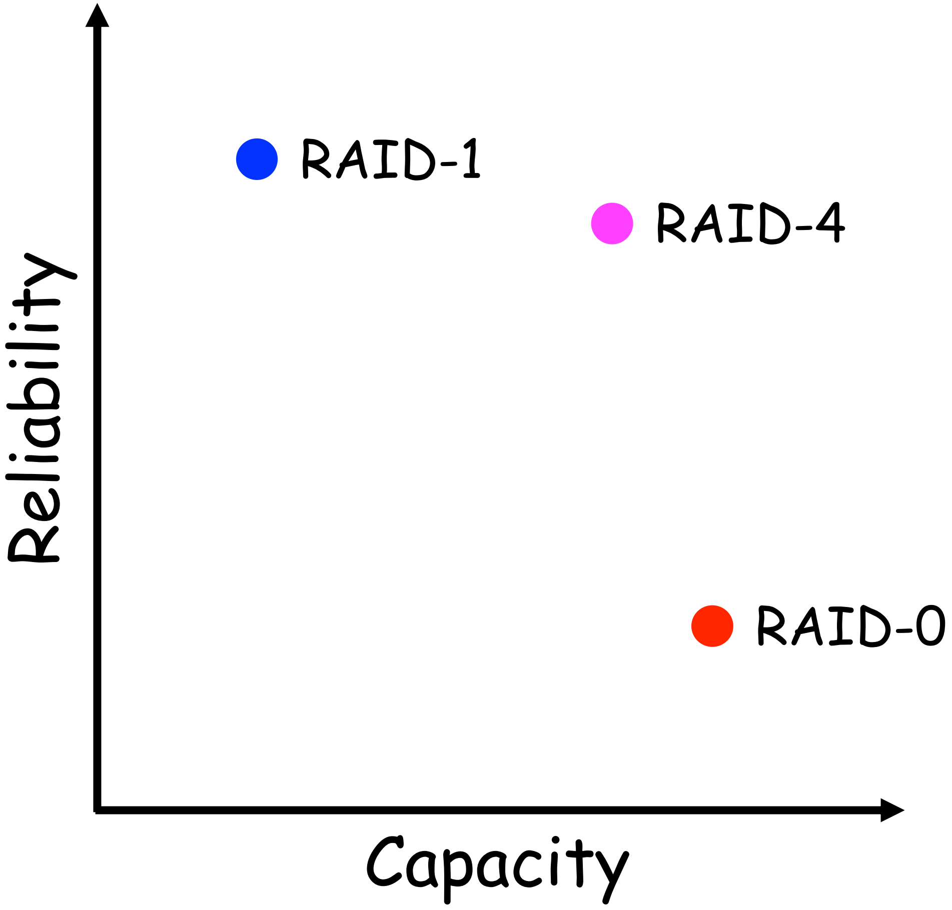
Mirroring Issues

- Disk 0 and Disk 1 are inconsistent

- Seen this problem before
  - Write-ahead logging



Capacity vs Reliability



RAID 4: Strategy

- Use Parity Disk

C0	C1	C2	C3	P
0	0	1	1	XOR(0,0,1,1) = 0
0	1	0	0	XOR(0,1,0,0) = 1

- Using parity bit, can reconstruct the lost column (disk)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	1	0	1	1	1
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	1	0	1	1	1
					(parity)

Disk 2 failed. Reconstruct the data based on the other data in the stripe

Parity Block: How to construct?

Block0	Block1	Block2	Block3	Parity
00	10	11	10	11
10	01	00	01	10

- i^{th} bit in the parity block is the parity bit of i^{th} bit of all the other blocks in the stripe

RAID-4: Analysis

What is capacity?	(N-1) * C
How many disks can fail?	1
Latency (read, write)	D, 2D (read and write parity 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

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

RAID-4: Throughput

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 (read and write parity disk sequentially)

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

RAID-4: Small Write Problem

- Writes to block 4 and 13 cannot happen in parallel
  - P1 and P3 can only be updated sequentially
- Parity disk is the bottleneck

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

RAID-5: Rotating Parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P0
5	6	7	P1	4
10	11	P2	8	9
15	P3	12	13	14
P4	16	17	18	19

- Parity blocks are rotated across disks drive

RAID-5: Throughput

What is steady-state throughput for

- sequential reads?	(N-1) * S
- sequential writes?	(N-1) * S
- random reads?	N * R
- random writes?	N*R/4

RAID LEVEL COMPARISONS

	RAID-0	RAID-1	RAID-4	RAID-5
Capacity	N	N/2	N-1	N-1
Reliability	0	1 (for sure) $\frac{N}{2}$ (if lucky)	1	1
Throughput				
Sequential Read	N · S	(N/2) · S	(N-1) · S	(N-1) · S
Sequential Write	N · S	(N/2) · S	(N-1) · S	(N-1) · S
Random Read	N · R	N · R	(N-1) · R	N · R
Random Write	N · R	(N/2) · R	$\frac{1}{2}$R	$\frac{N}{4}$ R
Latency				
Read	D	D	D	D
Write	D	D	2D	2D

Summary

- Capacity, reliability, and performance can be increased using RAID
- Transparent and easily deployable
- Offers many trade-offs

Disclaimer

- Some of the materials in this lecture slides are from the lecture slides by Prof. Andrea, Prof. Youjip, and other educators. Thanks to all of them.