OSTEP
Persistence:
RAID

Questions answered in this lecture:

Why more than one disk?

What are the different RAID levels? (striping, mirroring, parity)

Which RAID levels are best for reliability? for capacity?

Which are best for performance? (sequential vs. random reads and writ es)

Review Disks/Devices

Device Protocol Variants

Status checks: polling *vs.* interrupts

Data: PIO *vs.* DMA

Control: special instructions *vs.* memory-mapped I/O

Disks

Doing an I/O requires:
- seek
- rotate
- transfer

What is expensive

Schedulers

Strategy: reorder requests to meet some goal

- performance (by making I/O sequential)
- fairness
- consistent latency

Usually in both OS and H/W.

CFQ (Linux Default)

Completely Fair Queueing.

Queue for each process.

Do weighted round-robin between queues, with slice time proportional to priority.

Optimize order within queue.

Yield slice only if idle for a given time (anticipation).

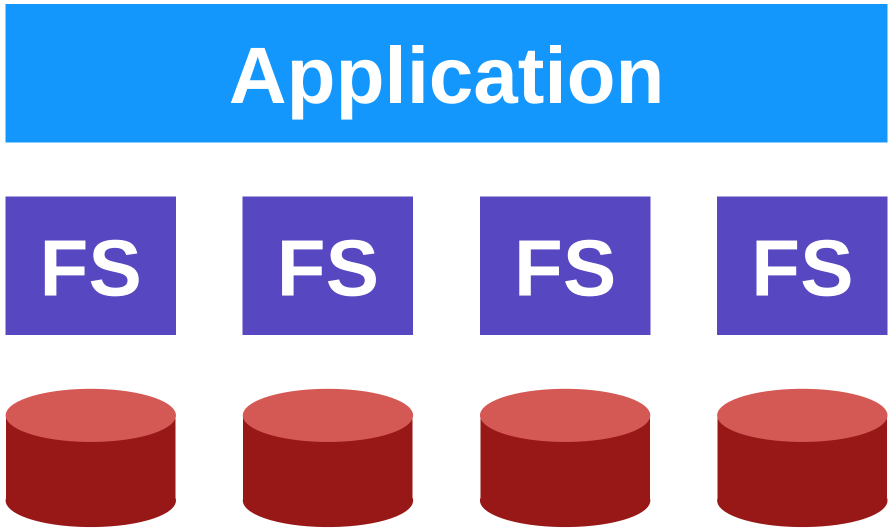
Only One Disk?

Sometimes we want many disks — why?

- capacity
- reliability
- Performance

Challenge: most file systems work on only one disk

Solution 1: JBOD



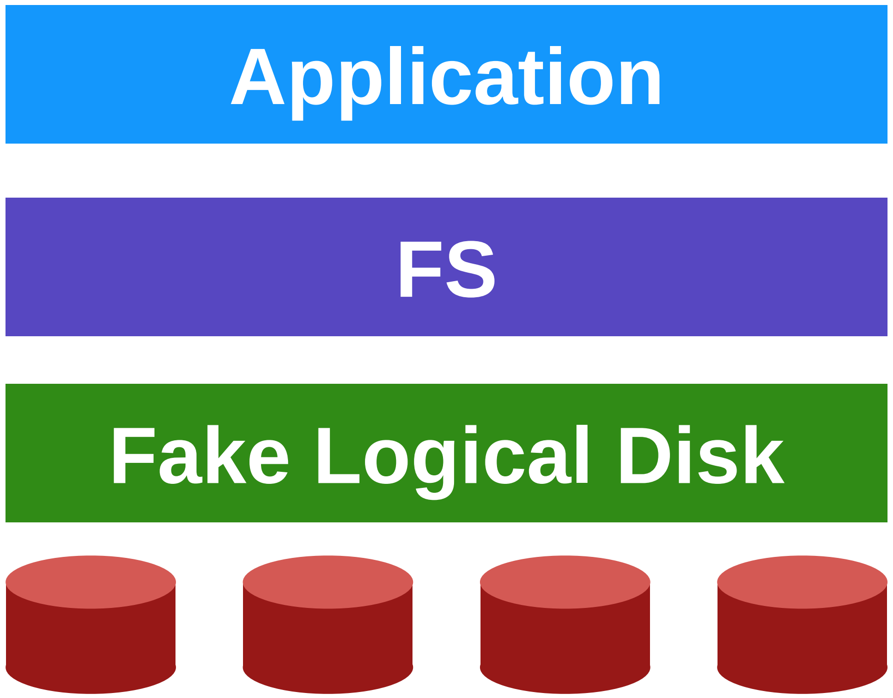
Application is smart, stores different files on different file systems.

JBOD: **J**ust a **B**unch **O**f **D**isks

Solution 2: RAID

RAID is:

- transparent
- deployable



Logical disk gives

- capacity
- performance
- reliability

Build logical disk from many physical disks.

RAID: **R**edundant **A**rray of **I**nexpensive **D**isks

RAID (Redundant Array of Inexpensive Disks)

- **Use multiple disks** in concert to build a **faster**, **bigger**, and more **reliable** disk system.
  - RAID just looks like a big disk to the host system.

- Advantage
  - **Performance** & **Capacity**: Using multiple disks in parallel
  - **Reliability**: RAID can tolerate the loss of a disk.

RAIDs provide these advantages transparently to systems that use them.

RAID Interface

- When a RAID receives I/O request,
  1. The RAID **calculates** which disk to access.
  2. The RAID **issue** one or more **physical I/Os** to do so.

- RAID example: A mirrored RAID system
  - Keep two copies of each block (each one on a separate disk)
  - Perform two physical I/Os for every one logical I/O it is issued.

RAID Internals

- A microcontroller
  - Run firmware to direct the operation of the RAID
- Volatile memory (such as DRAM)
  - Buffer data blocks
- Non-volatile memory
  - Buffer writes safely
- Specialized logic to perform parity calculation

Fault Model

- RAIDs are designed to **detect** and **recover** from certain kinds of disk faults.

- **Fail-stop** fault model
  - A disk can be in one of two states: *Working* or *Failed*.
    - Working: all blocks can be read or written.
    - Failed: the disk is permanently lost.
  - RAID controller can immediately observe when a disk has failed.

Why *Inexpensive* Disks?

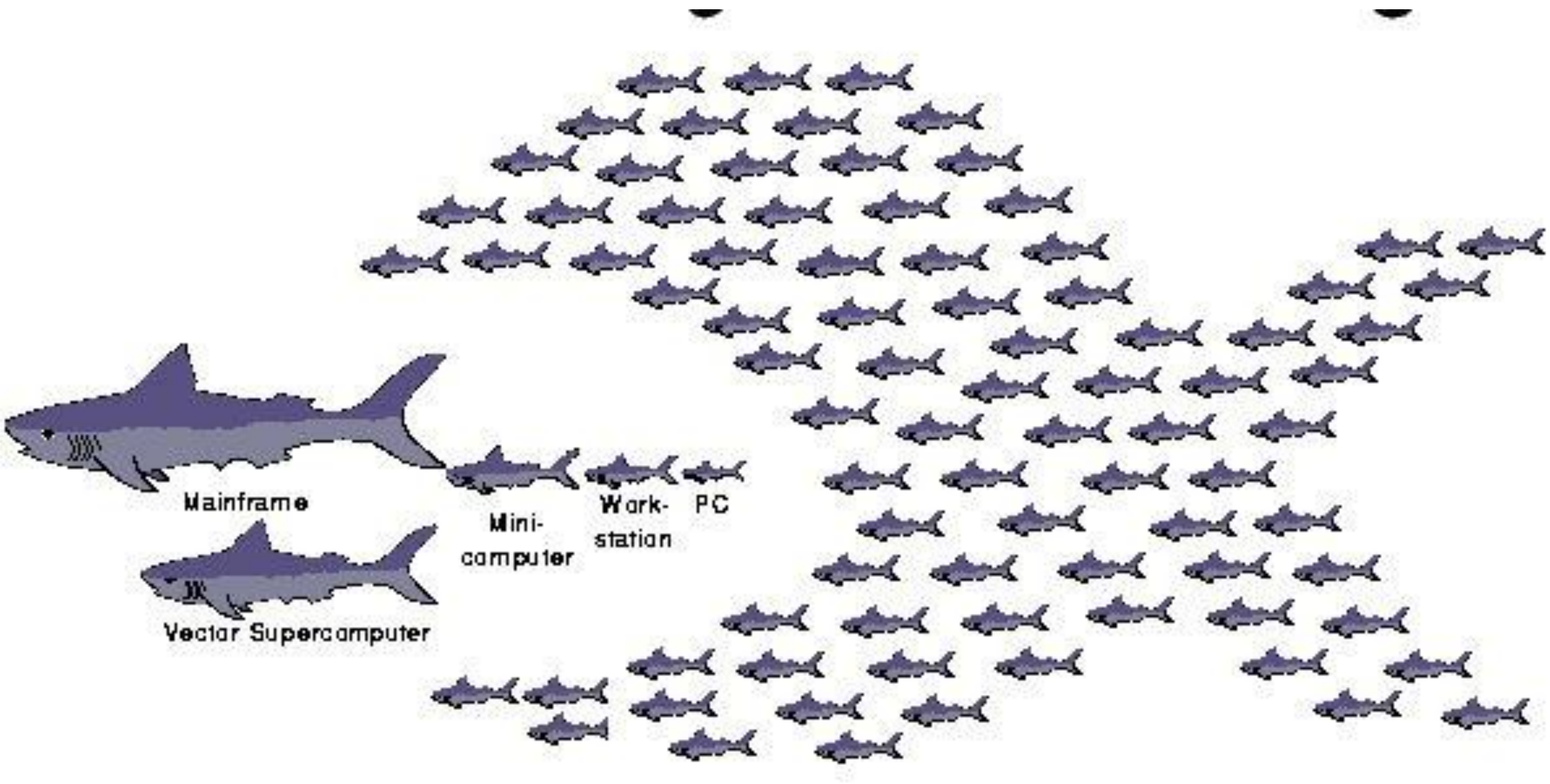
Economies of scale! Commodity disks cost less

Can buy many commodity H/W components for the same price as few high-end components

Strategy: write S/W to build high-quality logical devices from many cheap devices

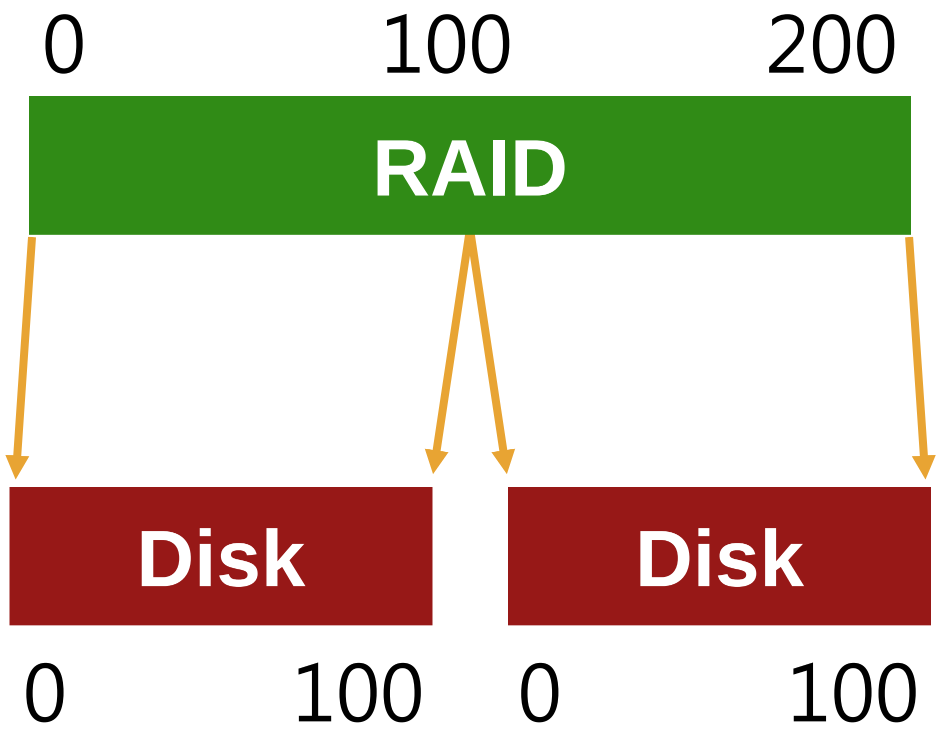
Alternative to RAID: buy an expensive, high-end disk

The Berkeley NoW Project



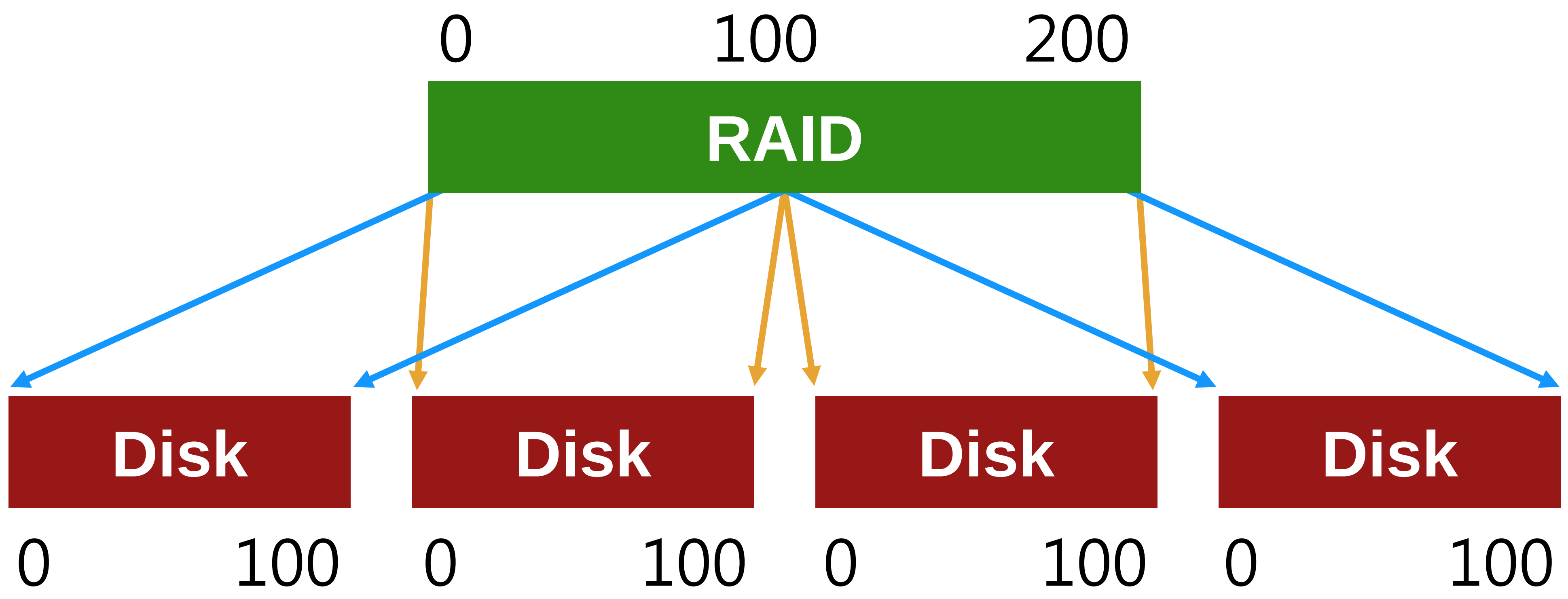
General Strategy: MAPPING

Build fast, large disk from smaller ones.



General Strategy: Redundancy

Add even more disks for reliability.



Mapping

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

- Some similarity to virtual memory

1) **Dynamic** mapping: use data structure (hash table, tree)
 - page tables

2) **Static** mapping: use simple math
 - RAID

RAID volume is fixed-sized, dense

Redundancy

Trade-offs to amount of redundancy

Increase number of copies: (e.g., RAID)

- improves reliability (and maybe performance)
- E.g., RAID

Decrease number of copies (deduplication) (e.g., code sha ring)

- improves space efficiency

One strategy: reduce redundancy as much is possible.
Then add back just the right amount.

RAID Analysis

Reasoning About RAID

RAID: system for mapping logical to physical blocks

Workload: types of reads/writes issued by applicatio ns (sequential vs. random)

Metric: capacity, reliability, performance

RAID Decisions

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
    - Random
- Writes
  - One operation
  - Steady-state I/O
    - Sequential
    - Random

Metrics

Capacity: how much space can apps use?

Reliability: how many disks can we safely lose? (assume fail stop!)

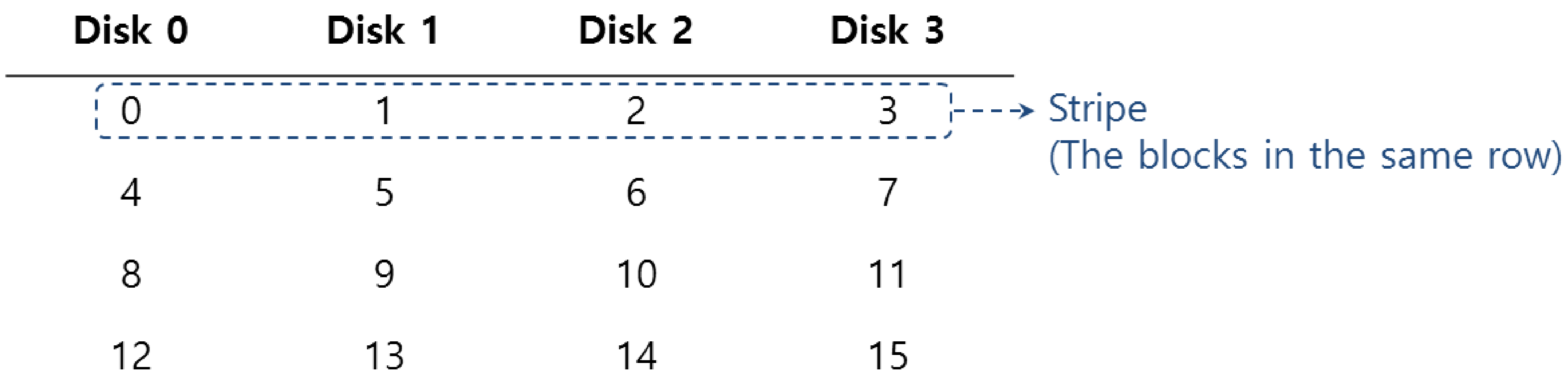
Performance: how long does each workload take?

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 Level 0: Striping

- RAID Level 0 is the simplest form as **striping** blocks.
  - **Spread the blocks** across the disks in a round-robin fashion.
  - No redundancy
  - Excellent performance and capacity



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

Stripe
(The blocks in the same row)

RAID-0: Simple Striping
(Assume here a 4-disk array)

RAID Level 0 (Cont.)

- Example) RAID-0 with a bigger chunk size
  - Chunk size : 2 blocks (8 KB)
  - A Stripe: 4 chunks (32 KB)

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

chunk size: 2blocks

Striping with a Bigger Chunk Size

Chunk Sizes

- Chunk size mostly affects performance of the array
  - **Small chunk size**
    - Increasing the parallelism
    - Increasing positioning time to access blocks
  - **Big chunk size**
    - Reducing intra-file parallelism
    - Reducing positioning time

Determining the "best" chunk size is hard to do.

Most arrays use larger chunk sizes (e.g., 64 KB)

RAID Level 0 Analysis

- **Capacity →** RAID-0 is perfect.
  - Striping delivers N disks worth of useful capacity.

- **Performance** of striping → RAID-0 is excellent.
  - All disks are utilized often in parallel.

- **Reliability →** RAID-0 is bad.
  - Any disk failure will lead to data loss.

Evaluating RAID Performance

- Consider two performance metrics
  - Single request latency
  - Steady-state throughput

- Workload
  - **Sequential**: access 1MB of data (block (B) ~ block (B + 1MB))
  - **Random**: access 4KB at random logical address

- A disk can transfer data at
  - `S` MB/s under a sequential workload
  - `R` MB/s under a random workload

Evaluating RAID Performance Example

- sequential (S) vs random (R)
  - **Sequential** : transfer 10 MB on average as continuous data.
  - **Random** : transfer 10 KB on average.
  - Average seek time: 7 ms
  - Average rotational delay: 3 ms
  - Transfer rate of disk: 50 MB/s

- Results:
  - S = $\frac{Amount\ of\ Data}{Time\ to\ access} = \frac{10\ MB}{210\ ms}$ = 47.62 MB /s
  - R = $\frac{Amount\ of\ Data}{Time\ to\ access} = \frac{10\ KB}{10.195\ ms}$ = 0.981 MB /s

Evaluating RAID-0 Performance

N : the number of disks

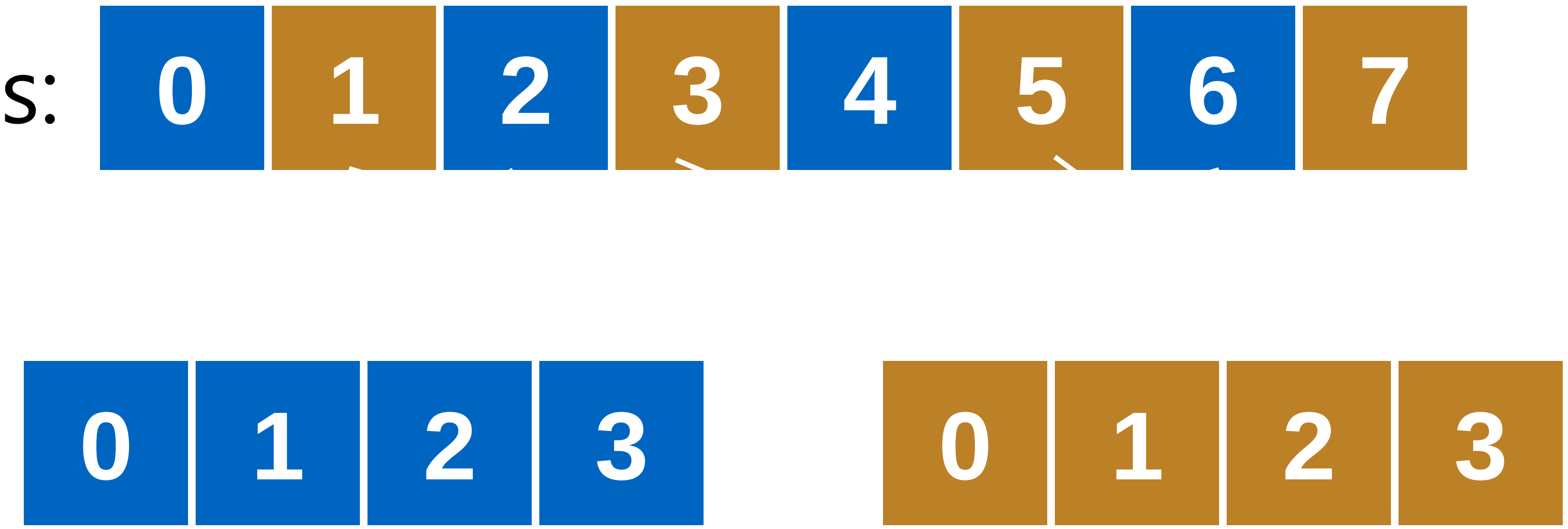
- Single request latency
  - Identical to that of a single disk.

- Steady-state throughput
  - **Sequential** workload : $N \cdot S$ MB/s
  - **Random** workload : $N \cdot S$ MB /s

RAID-0: Striping

Optimize for capacity. No redundancy

Logical Blocks:



Disk 0

Disk 1

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

4 disks

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

4 disks

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

Given logical address A, find:
Disk = …
Offset = …

Given logical address A, find:
Disk = A % disk_count
Offset = A / disk_count

Chunk Size

Chunk size = 1

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

Chunk size = 2

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

assume chunk size of 1

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 Level 1 : Mirroring

- RAID Level 1 tolerates **disk failures**.
  - **Copy** more than one of **each block** in the system.
  - Copy block places on a separate disk.

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

Simple RAID-1: Mirroring (Keep two physical copies)

- RAID-10 (RAID 1+0) : mirrored pairs and then stripe
- RAID-01 (RAID 0+1) : contain two large striping arrays, and then mirrors

RAID-1 Analysis

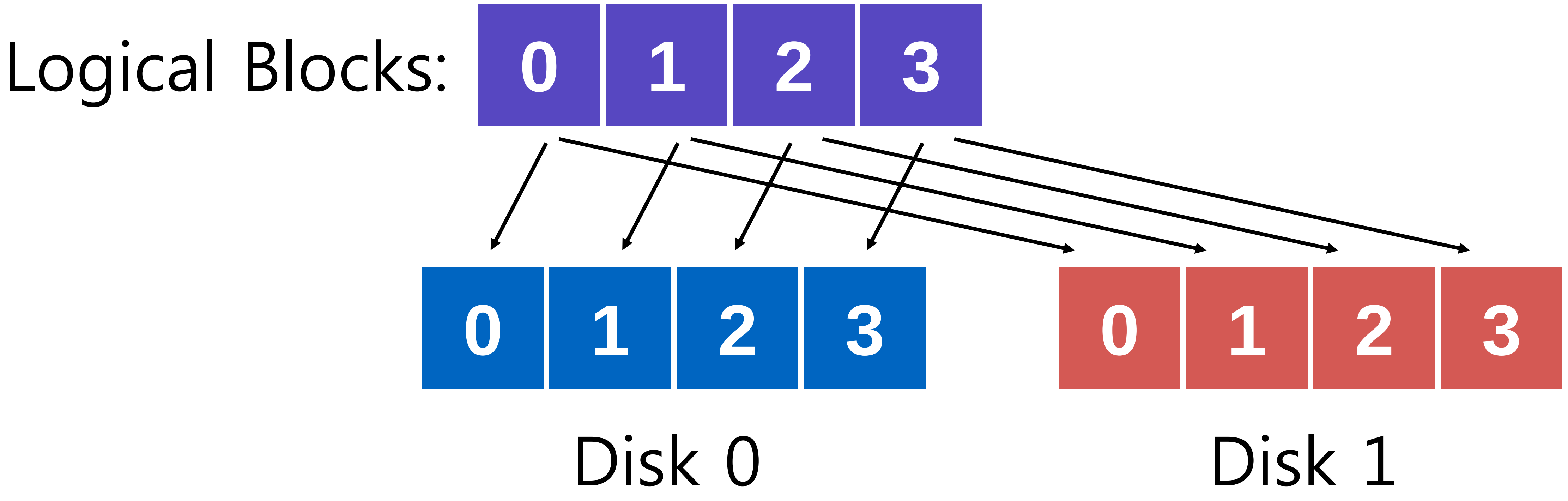
N : the number of disks

- **Capacity**: RAID-1 is Expensive
  - The useful capacity of RAID-1 is N/2.
- **Reliability**: RAID-1 does well.
  - It can tolerate the failure of any one disk (up to N/2 failures depending on which disk fail).

Performance of RAID-1

- Two physical writes to complete
  - It suffers the worst-case seek and rotational delay of the two request.
  - Steady-state throughput
    - **Sequential Write** : $\frac{N}{2} \cdot S$ MB/s
      - Each logical write must result in two physical writes.
    - **Sequential Read** : $\frac{N}{2} \cdot S$ MB/s
      - Each disk will only deliver half its peak bandwidth.
    - **Random Write** : $\frac{N}{2} \cdot R$ MB/s
      - Each logical write must turn into two physical writes.
    - **Random Read** : $N \cdot R$ MB/s
      - Distribute the reads across all the disks.

RAID-1: Mirroring



Keep two copies of all data.

Assumptions

Assume disks are **fail-stop**.
- they work or they don't
- we know when they don't

Tougher Errors:
- latent sector errors
- silent data corruption

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: 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

How many disks can fail?

Assume disks are **fail-stop**.
- each disk works or it doesn’t
- system knows when disk fails

Tougher Errors:
- latent sector errors
- silent data corruption

RAID-1: Analysis

What is capacity?	**N/2 * C**
How many disks can fail?	**1 (or maybe N / 2)**
Latency (read, write)?	**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

- sequential reads?
- sequential writes?
- random reads?
- random writes?

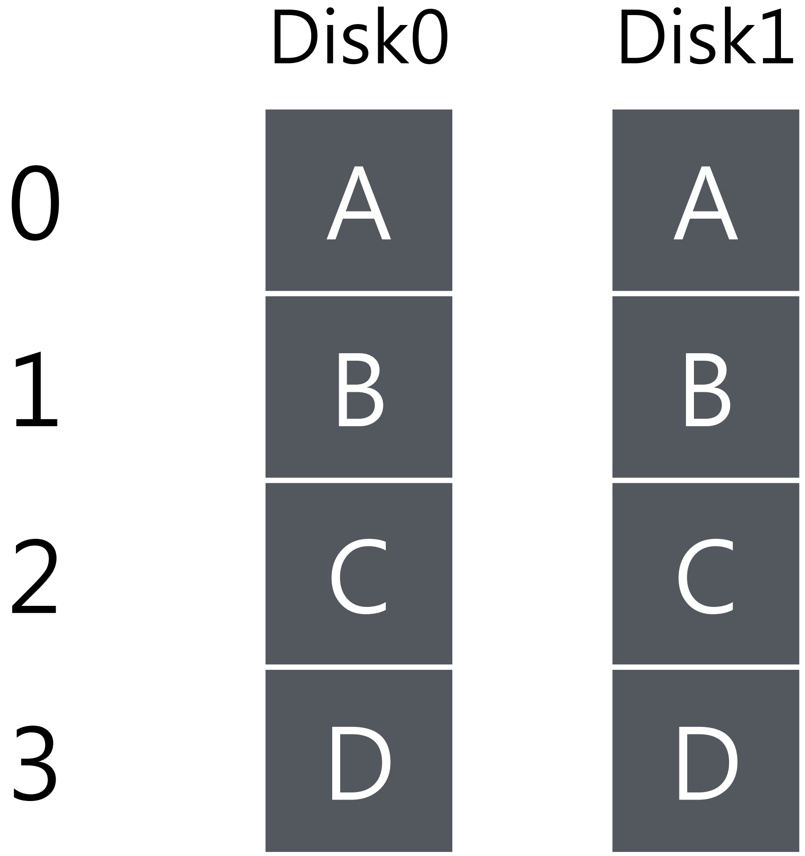
RAID-1: Throughput

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

Crashes

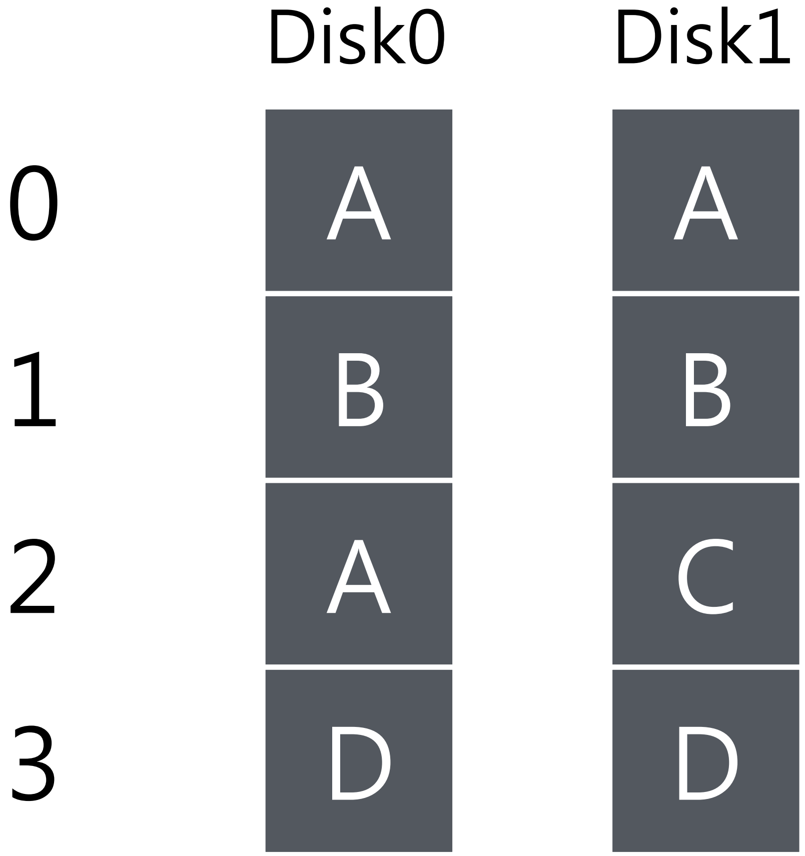


	Disk0	Disk1
0	A	A
1	B	B
2	C	C
3	D	D

Crashes

	Disk0	Disk1
0	A	A
1	B	B
2	C	C
3	D	D

Crashes



	Disk0	Disk1
0	A	A
1	B	B
2	A	C
3	D	D

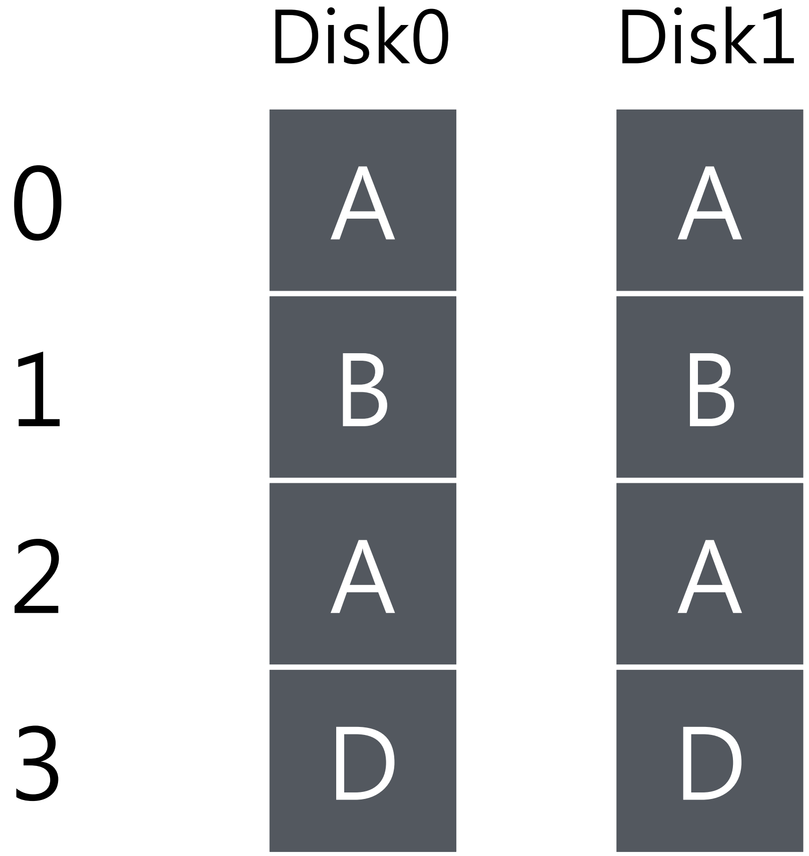
write(A) to 2

Crashes

	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	D

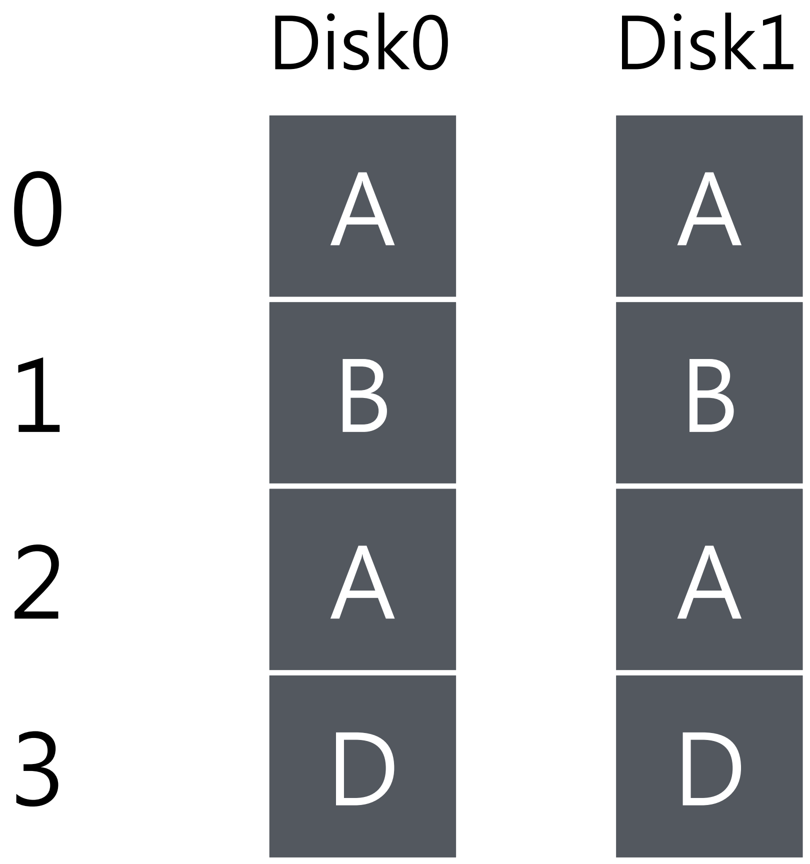
write(A) to 2

Crashes



	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	D

Crashes



	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	D

write(T) to 3

Crashes

	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	T

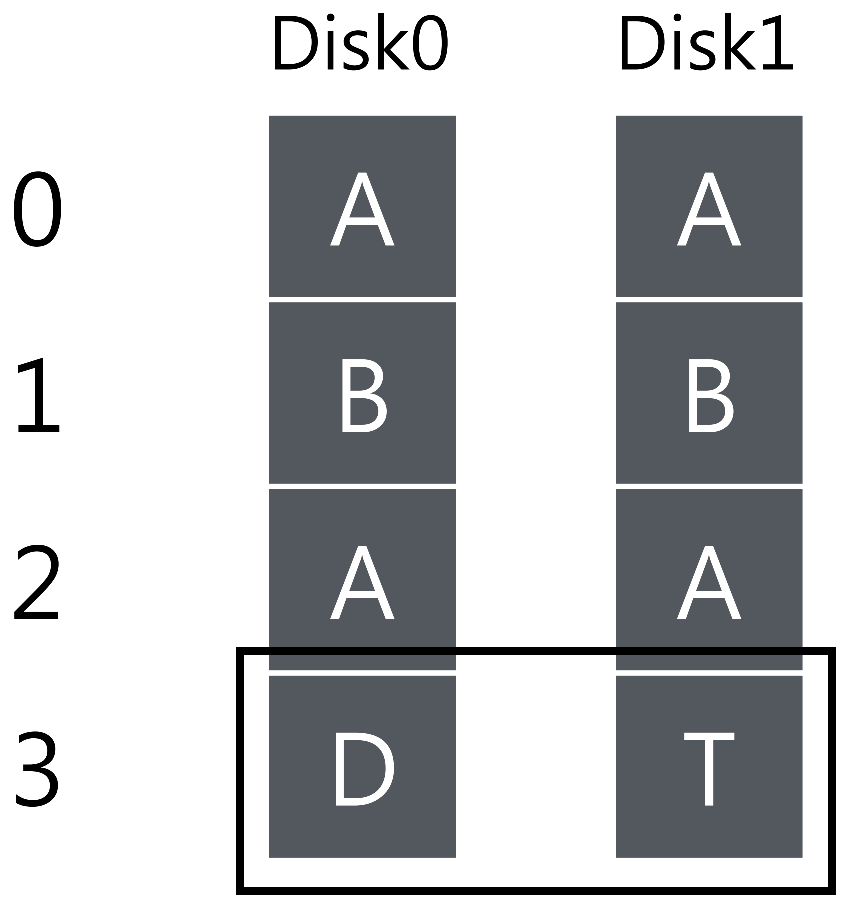
write(T) to 3

Crashes

	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	T

CRASH!!!

Crashes



	Disk0	Disk1
0	A	A
1	B	B
2	A	A
3	D	T

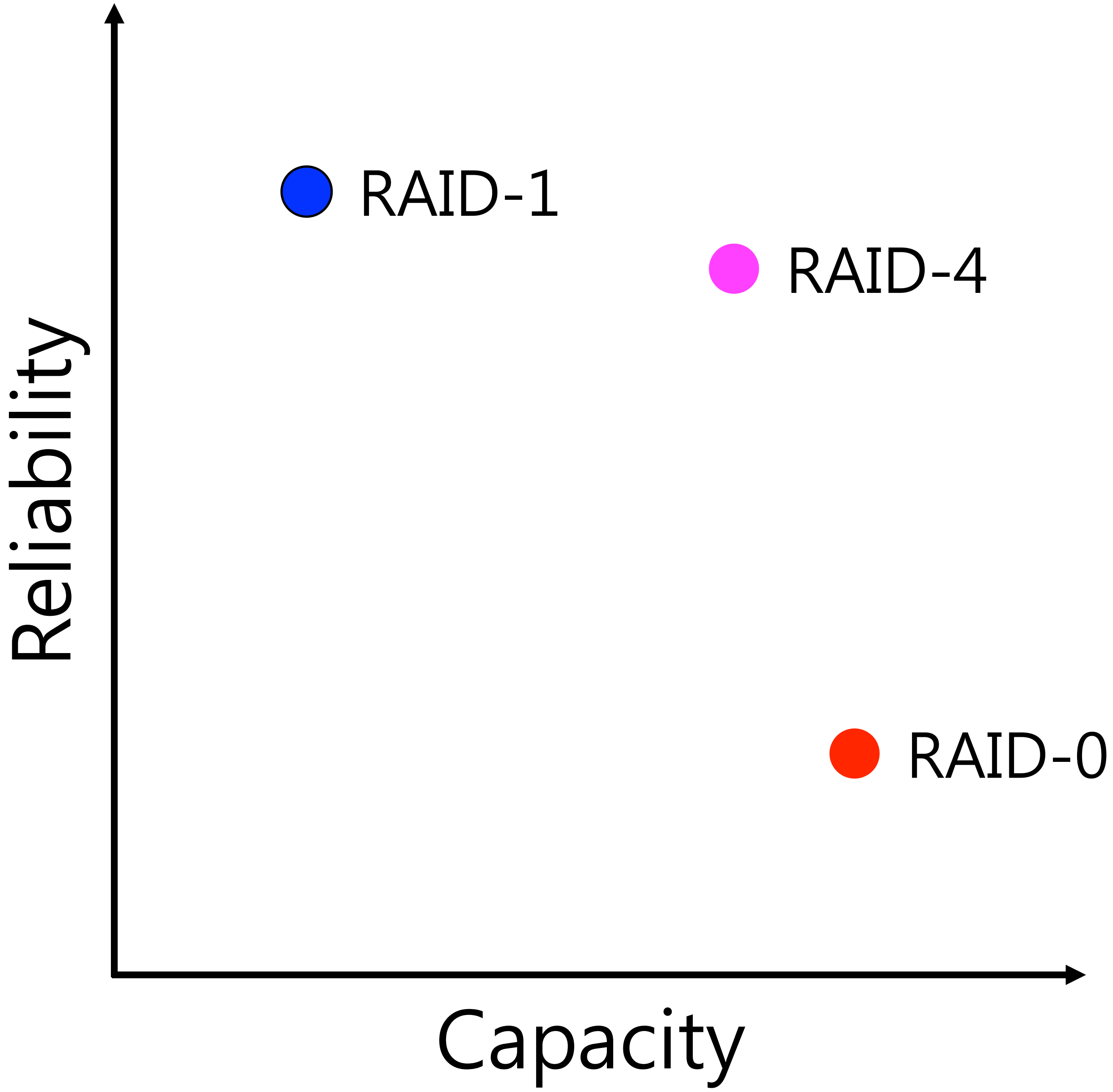
after reboot, how to tell which data is right?

H/W Solution

Problem: Consistent-Update Problem

Use non-volatile RAM in RAID controller.

Software RAID controllers (e.g., Linux md) don't have this option


RAID-1
RAID-4
RAID-0
Reliability
Capacity

RAID Level 4 : Saving Space With Parity

- Add **a single parity block**
  - **A Parity block** stores the *redundant information* for that stripe of blocks.

* P: Parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	0	1	1	P0
2	2	3	3	P1
4	4	5	5	P2
6	6	7	7	P3

Five-disk RAID-4 system layout

RAID Level 4 (Cont.)

- **Compute parity** : the XOR of all of bits

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

- **Recover from parity**
  - Imagine the bit of the C2 in the first row is lost.
    1. Reading the other values in that row : 0, 0, 1
    2. The parity bit is 0 → even number of 1's in the row
    3. What the missing data must be: a 1.

RAID-4 Analysis

N : the number of disks

- **Capacity**
  - The useful capacity is $(N-1)$.

- **Reliability**
  - RAID-4 tolerates 1 disk failure and no more.

RAID-4 Analysis (Cont.)

- **Performance**
  - Steady-state throughput
    - Sequential read: $(N-1) \cdot S$ MB/s
    - Sequential write: $(N-1) \cdot S$ MB/s

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

Full-stripe Writes In RAID-4

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

Random write performance for RAID-4

- Overwrite a block + update the parity
- **Method 1**: *additive parity*
  - Read in all of the other data blocks in the stripe
  - XOR those blocks with the new block (1)
  - **Problem**: the performance scales with the number of disks

Random write performance for RAID-4 (Cont.)

- **Method 2**: *subtractive parity*

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

- Update C2(old) → C2(new)
  1. Read in the old data at C2 (C2(old)=1) and the old parity (P(old)=0)
  2. Calculate P(new):
     - If C2(new)==C2(old) → P(new)==P(old)
     - If C2(new)!=C2(old) → Flip the old parity bit

$$P(new) = \left(C2(old)\ XOR\ C2(new)\right)\ XOR\ \ P(old)$$

Small-write problem

- The parity disk can be a **bottleneck.**
  - Example: update blocks 4 and 13 (marked with *)

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

Writes To 4, 13 And Respective Parity Blocks.

  - Disk 0 and Disk 1 can be accessed in parallel.
  - Disk 4 prevents any parallelism.

RAID-4 throughput under random small writes is $(\frac{R}{2})$ MB/s (*terrible*).

A I/O latency in RAID-4

- **A single read**
  - Equivalent to the latency of a single disk request.

- **A single write**
  - Two reads and then two writes
    - Data block + Parity block
    - The reads and writes can happen in parallel.
  - Total latency *is about twice* that of a single disk.

Raid-4 Strategy

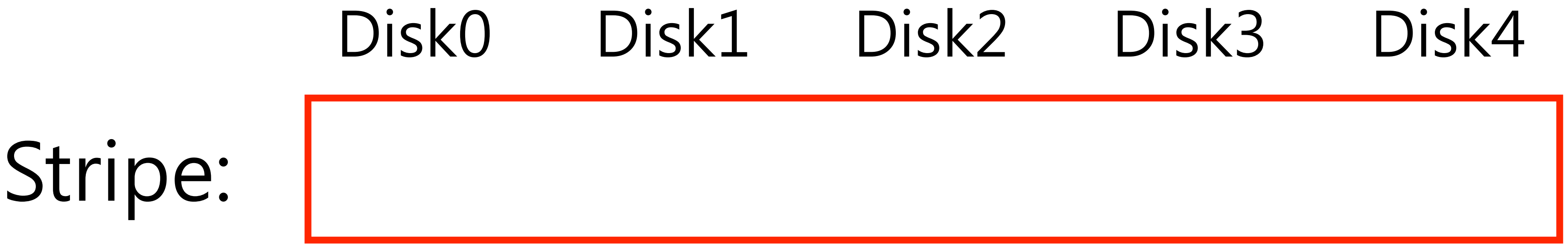
Use parity disk

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.

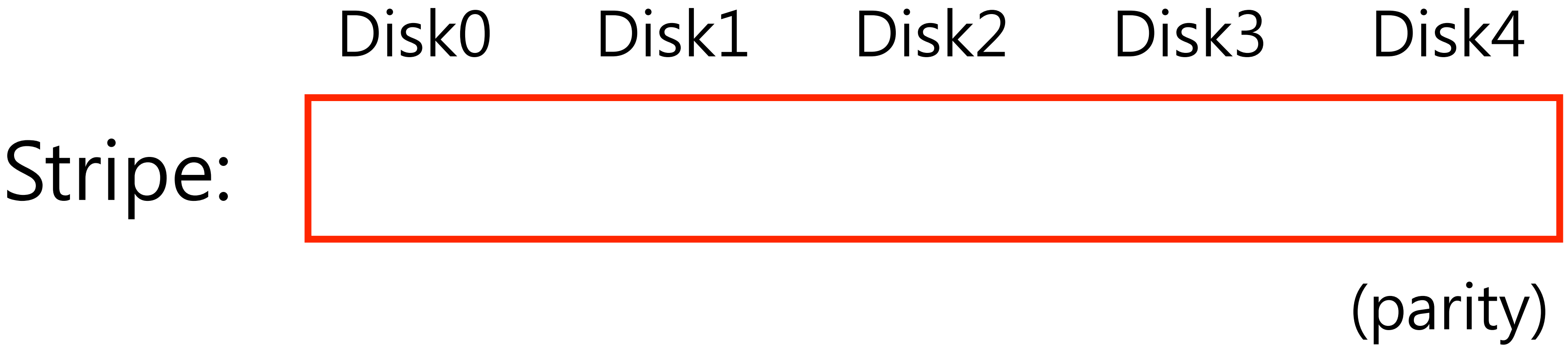
Example



Disk0 Disk1 Disk2 Disk3 Disk4

Stripe:

Example



Disk0 Disk1 Disk2 Disk3 Disk4

Stripe:

(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	5	3	0	1	
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	5	3	0	1	9
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	5	X	0	1	9
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	5	3	0	1	9
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	2	1	1	X	5
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	2	1	1	1	5
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	3	0	1	2	X
					(parity)

Example

	Disk0	Disk1	Disk2	Disk3	Disk4
Stripe:	3	0	1	2	6
					(parity)

Which functions are used to compute parity?

RAID-4: Analysis

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

How many disks can fail? **1**

Latency (read, write)? **D**, **2*D (read and write parity disk)**

Disk0	Disk1	Disk2	Disk3	Disk4
3	0	1	2	6
				(parity)

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-4: Throughput

What is steady-state throughput for

- sequential reads?
- sequential writes?
- random reads?
- random writes?

(N-1) * S

(N-1) * S

(N-1) * R

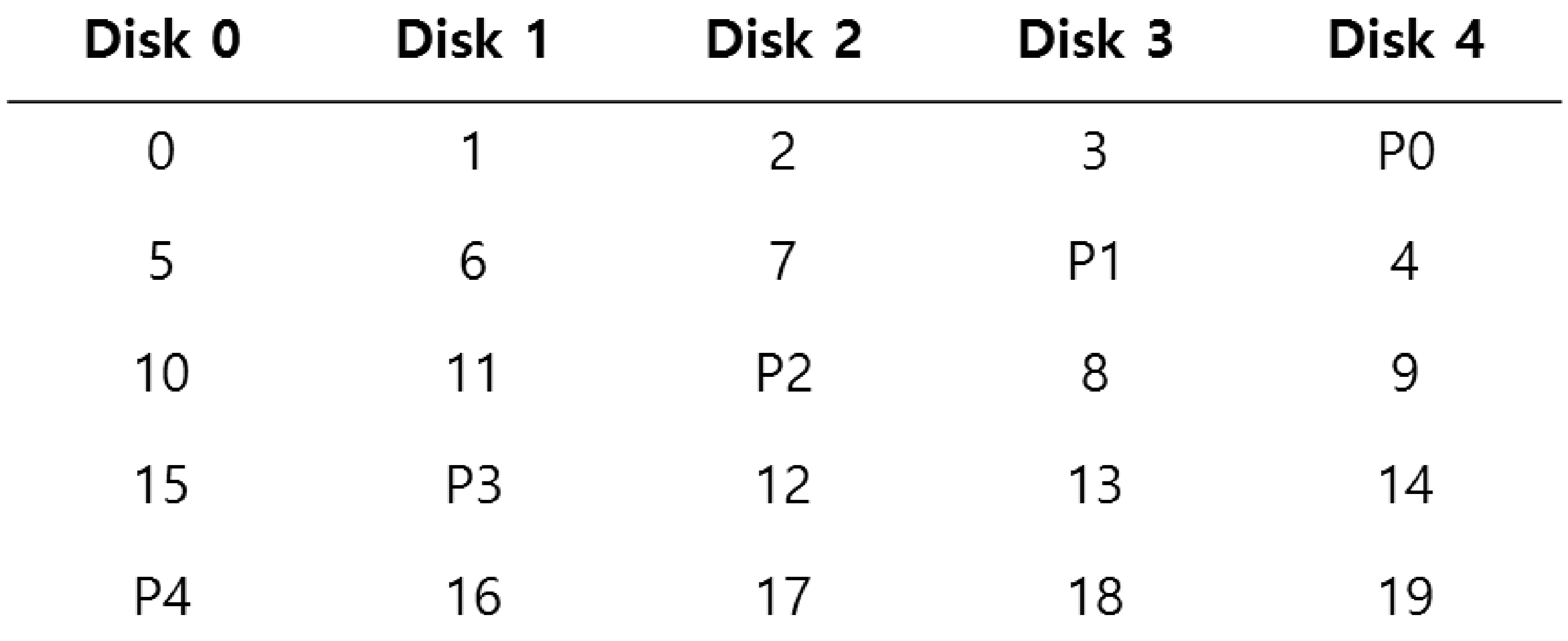
R/2 (read and write parity disk)

how to avoid parity bottleneck?

Disk0	Disk1	Disk2	Disk3	Disk4
3	0	1	2	6
				(parity)

RAID Level 5: Rotating Parity

- RAID-5 **is solution of** small write problem.
  - Rotate the parity blocks across drives.
  - Remove the parity-disk bottleneck for RAID-4



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

RAID-5 With Rotated Parity

RAID-5 Analysis

N : the number of disks

- **Capacity**
  - The useful capacity for a RAID group is $(N-1)$.

- **Reliability**
  - RAID-5 tolerates 1 disk failure and no more.

RAID-5 Analysis (Cont.)

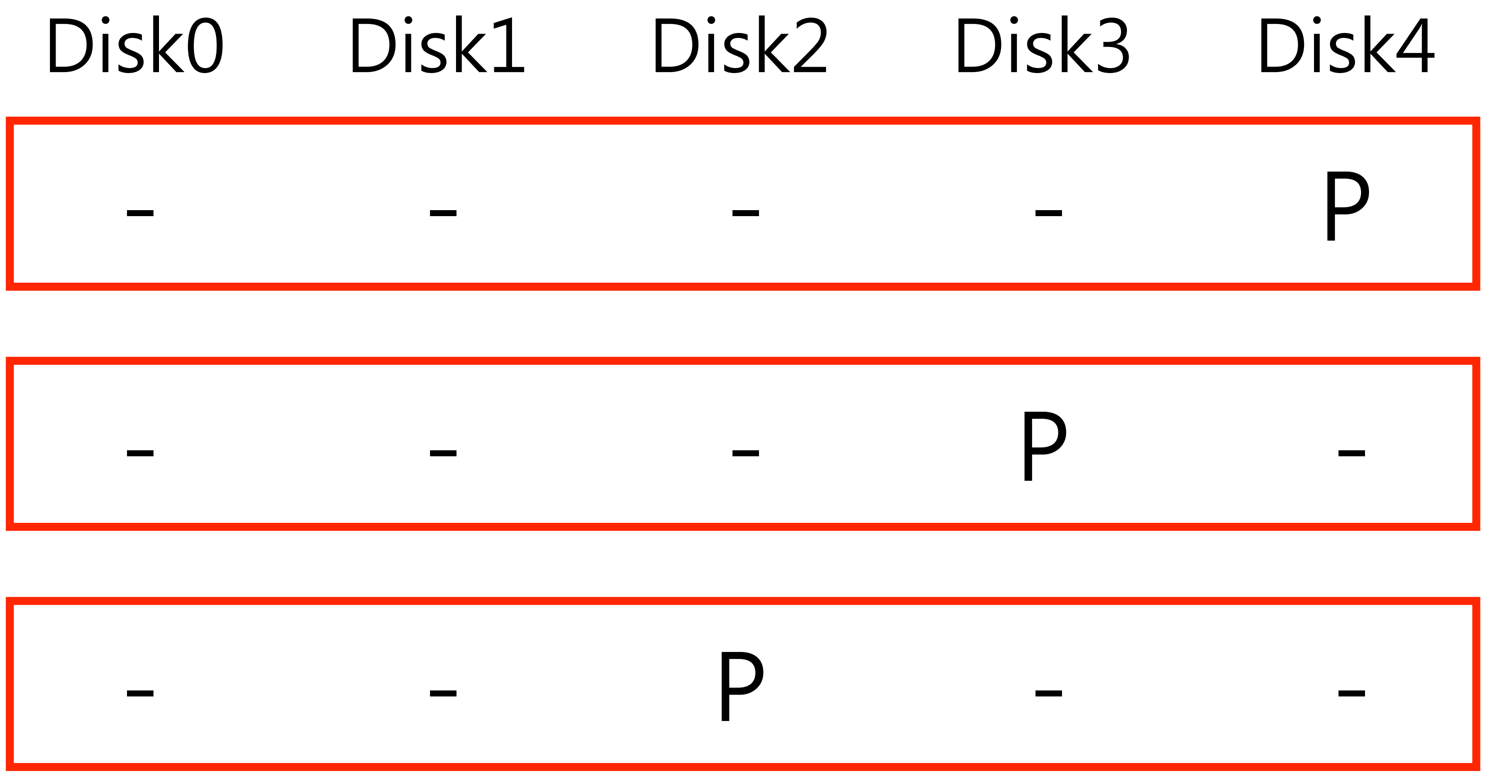
N : the number of disks

- **Performance**
  - Sequential read and write
  - A single read and write request

  Same as RAID-4

  - Random read : a little better than RAID-4
    - RAID-5 can utilize all of the disks.
  - Random write : $\frac{N}{4} \cdot R$ MB/s
    - The factor of four loss is cost of using parity-based RAID.

RAID-5



Disk0	Disk1	Disk2	Disk3	Disk4
-	-	-	-	P
-	-	-	P	-
-	-	P	-	-

…

Rotate parity across different disks

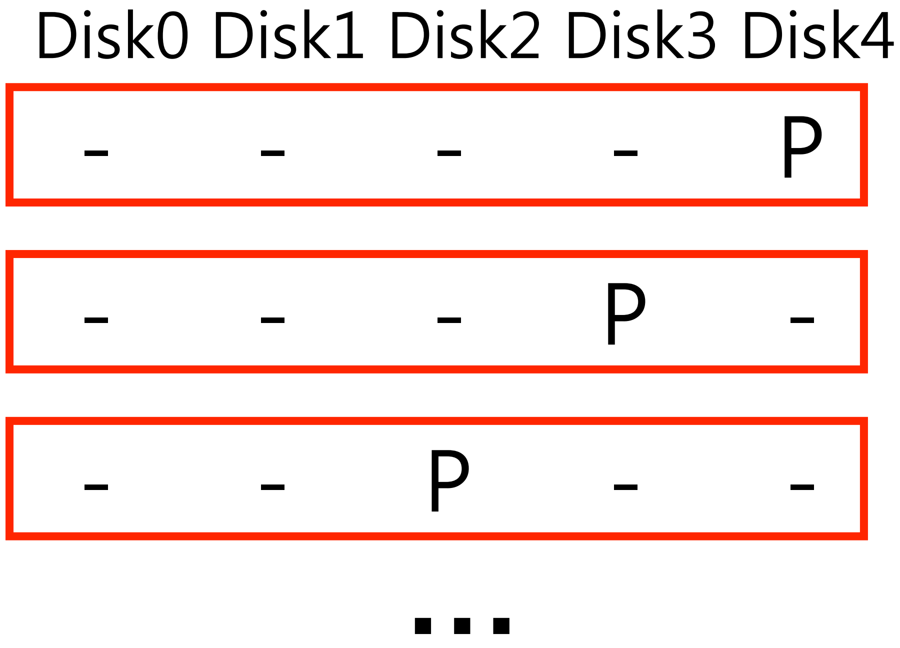
RAID-5: Analysis

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

How many disks can fail? 1

Latency (read, write)? **D, 2*D (read and write parity disk)**

Same as RAID-4…



Disk0	Disk1	Disk2	Disk3	Disk4
-	-	-	-	P
-	-	-	P	-
-	-	P	-	-

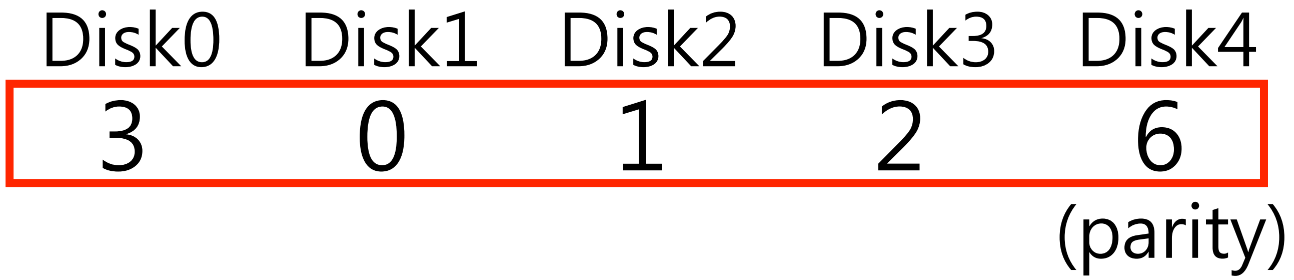
…

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-5: Throughput

Steady-state throughput for RAID-4:

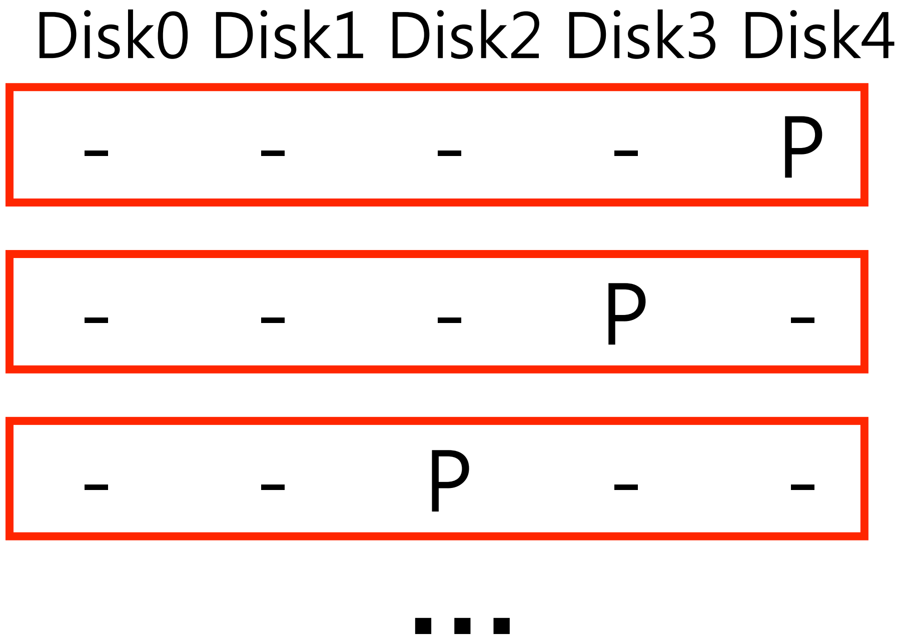
- sequential reads? **(N-1) * S**
- sequential writes? **(N-1) * S**
- random reads? **(N-1) * R**
- random writes? **R/2 (read and write parity disk)**



Disk0	Disk1	Disk2	Disk3	Disk4
3	0	1	2	6
				(parity)

What is steady-state throughput for RAID-5?

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



Disk0	Disk1	Disk2	Disk3	Disk4
-	-	-	-	P
-	-	-	P	-
-	-	P	-	-

...

RAID Level Comparisons

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

RAID LEVEL Comparisons

	Read Latency	Write Latency
RAID-0	D	D
RAID-1	D	D
RAID-4	D	2D
RAID-5	D	2D

but RAID-5 can do more in parallel

RAID Level Comparisons

	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-5 is strictly better than RAID-4

RAID Level Comparisons

	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-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 better than RAID-1 for sequential workloads

RAID-1 better than RAID-5 for random workloads

Summary

Many engineering tradeoffs with RAID

capacity, reliability, performance for different workloads

H/W controllers can handle crashes easier

Block-based interface:
Very deployable and popular storage solution due t
o transparency