CS 554: Advanced Database System

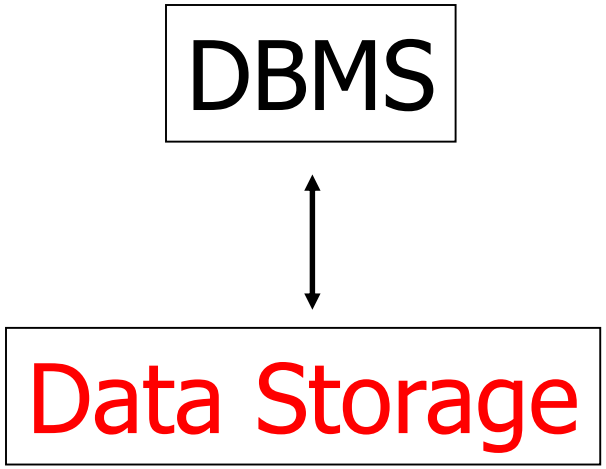
Notes 02: Hardware

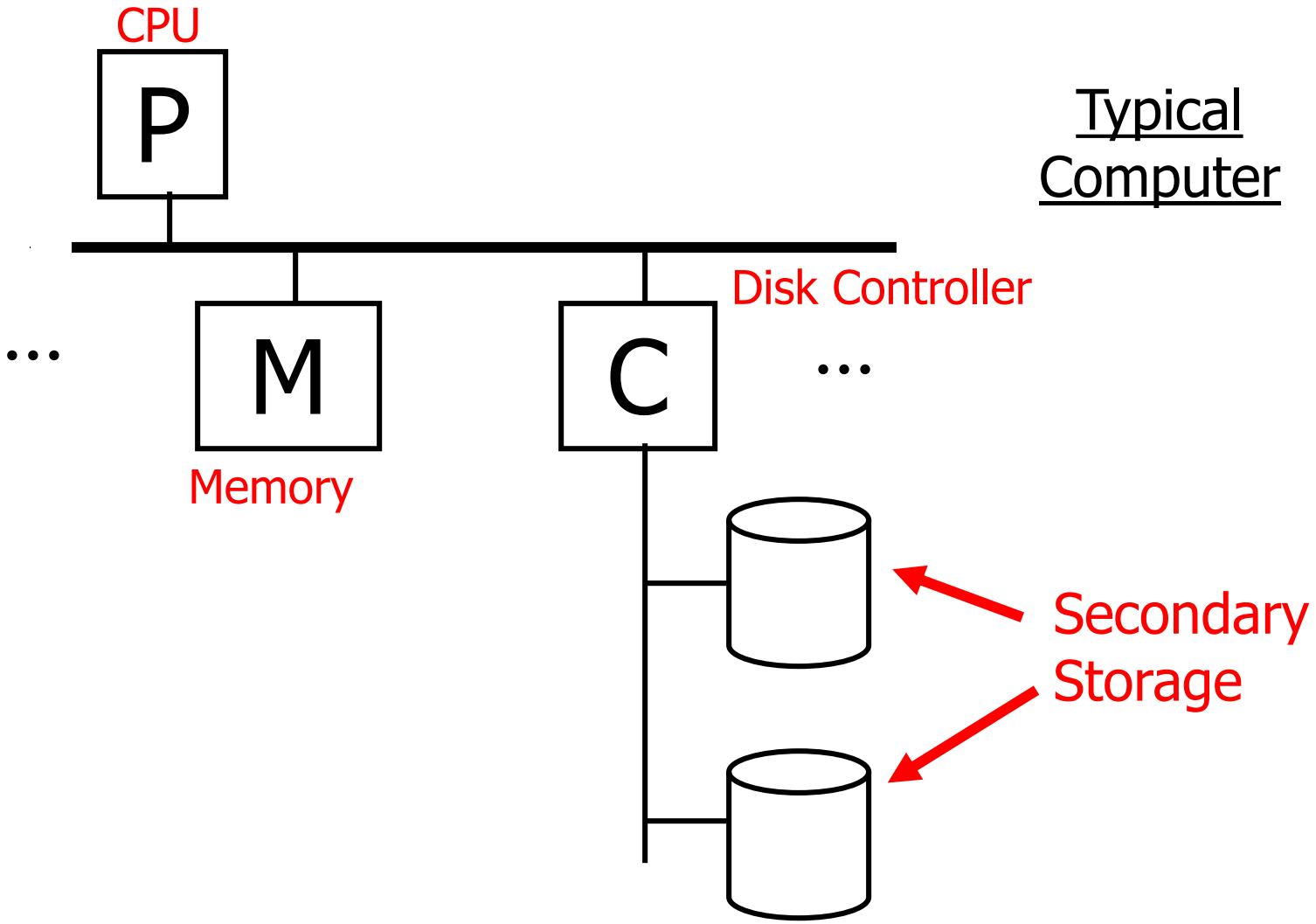
Hector Garcia-Molina

Outline

- Hardware: Disks
- Access Times (disk)
- Optimizations (disk access time)
- Other Topics:
  - Storage costs
  - Using secondary storage
  - Disk failures

Hardware



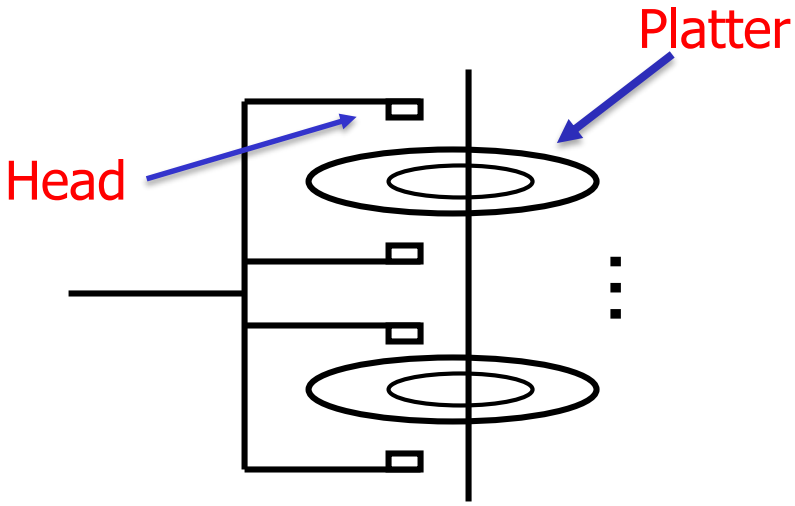

CPU
P
Typical
Computer
Disk Controller
M
C
Memory
Secondary
Storage

Secondary storage

Many flavors:

- Disk: Floppy (hard, soft)
  Removable Packs
  Winchester (most common)
  SSD disks
  Optical, CD-ROM…
  Arrays
- Tape: Reel, cartridge
  Robots

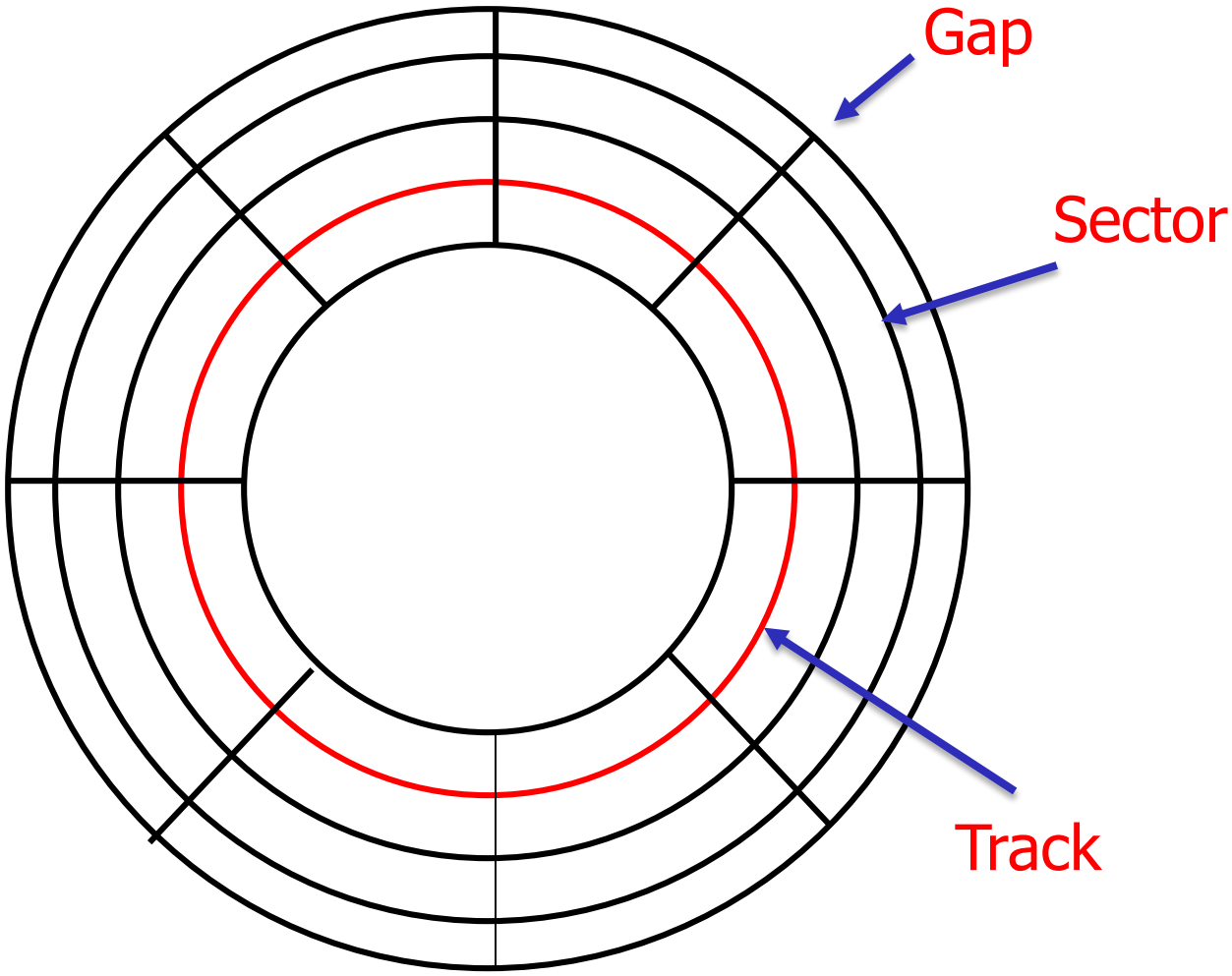
"Typical Disk:"



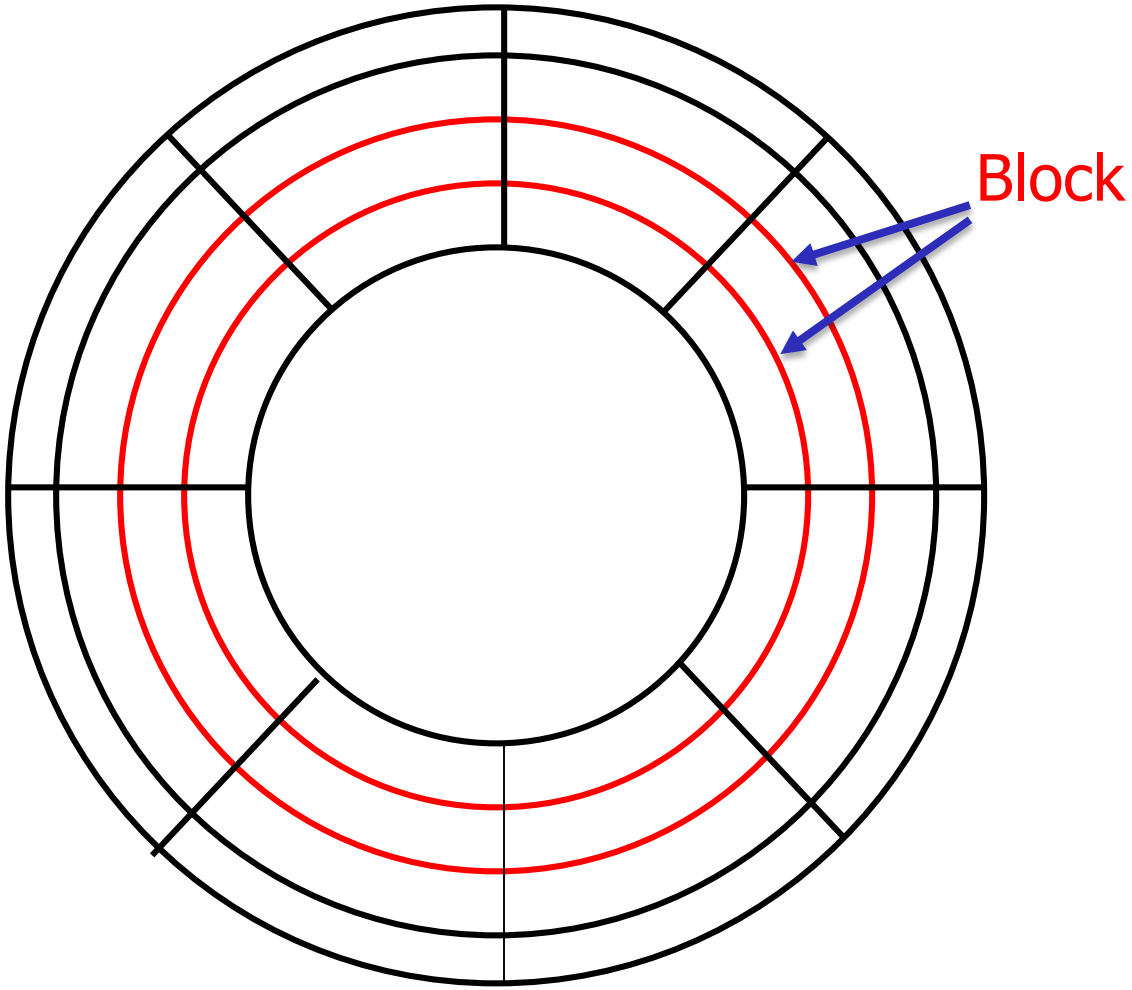
Terms: Platter, Head,
Cylinder, Track,

Sector (physical),
Block (logical), Gap

Top View



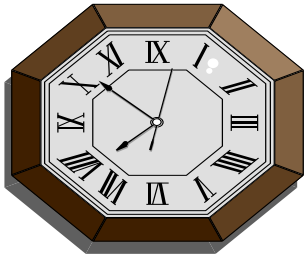
Block



Block = group of sectors that form a unit of access
One read/write operation will read/write one block

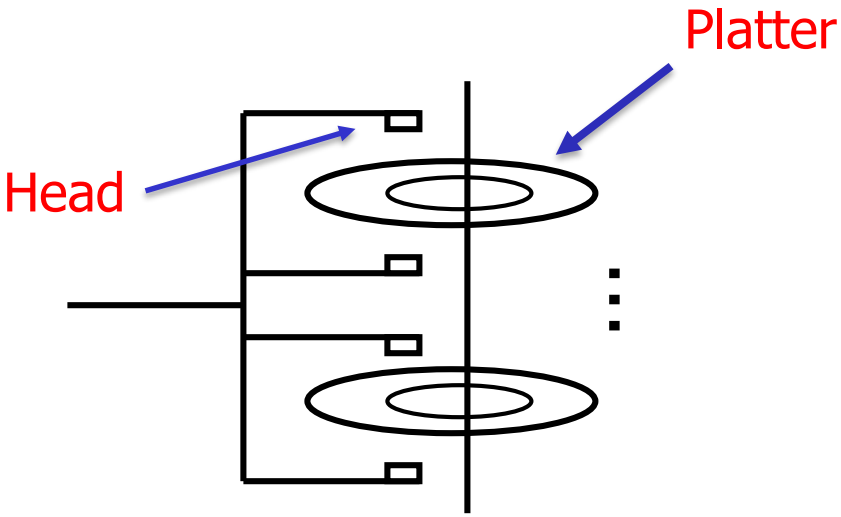
Disk Access Time

I want block X



block x in memory

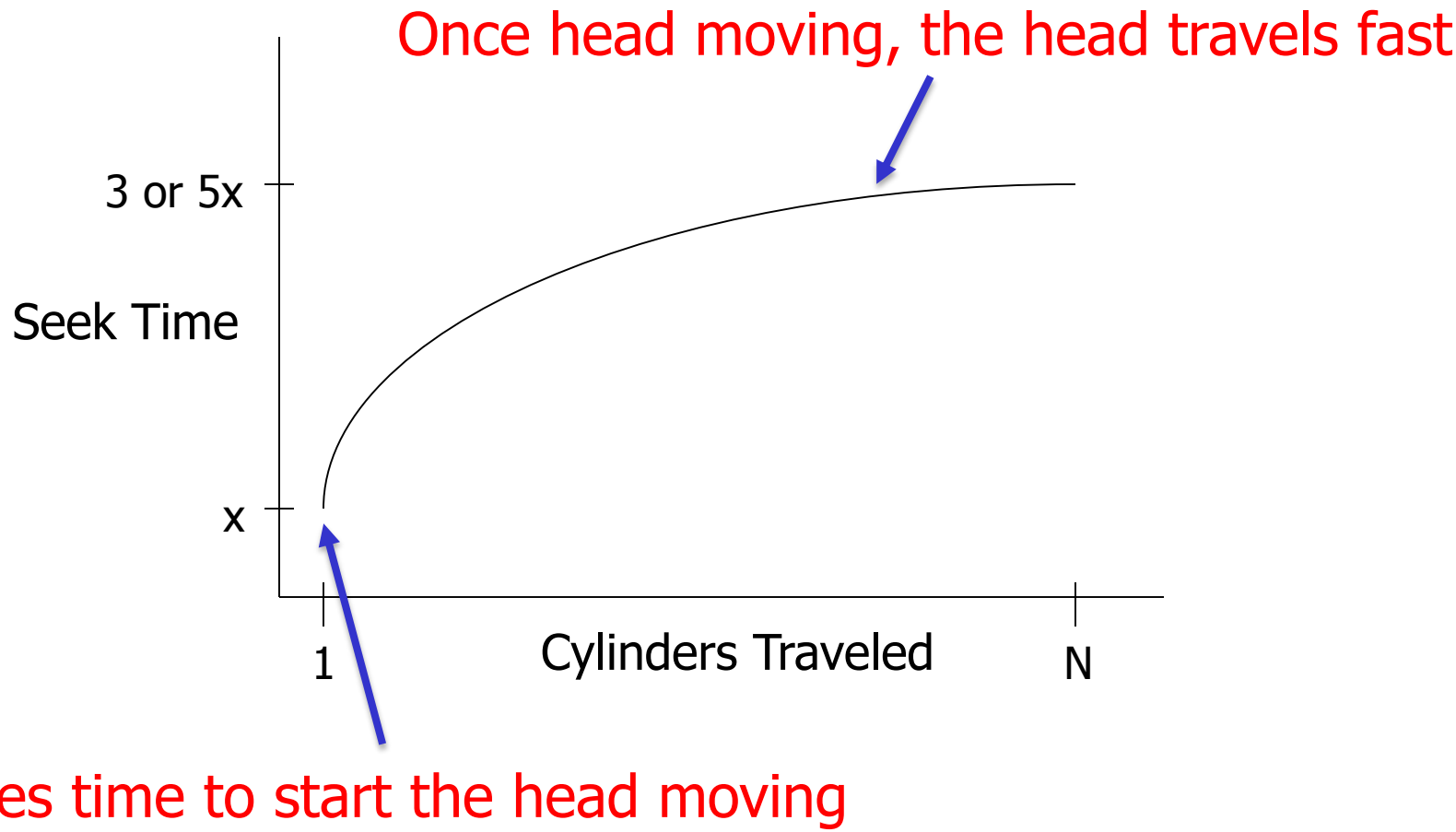
How long ?



Time = Seek Time +
Rotational Delay +
Transfer Time +
Other

Seek time: to move head to the desired cylinder (track)
Rotational delay: for waiting on the desired sector
Transfer time: to transfer data on sectors to memory

Seek Time


Once head moving, the head travels fast
3 or 5x
Seek Time
x
1
Cylinders Traveled
N
Takes time to start the head moving

Average Random Seek Time

Start at cylinder i → Go to cylinder j

$$S = \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{SEEKTIME}(i \rightarrow j)}{N(N-1)}$$

There are N starting cylinders and N-1 cylinders
Total: N(N-1) possible values

Average Random Seek Time

$$S = \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{SEEKTIME}(i \rightarrow j)}{N(N-1)}$$

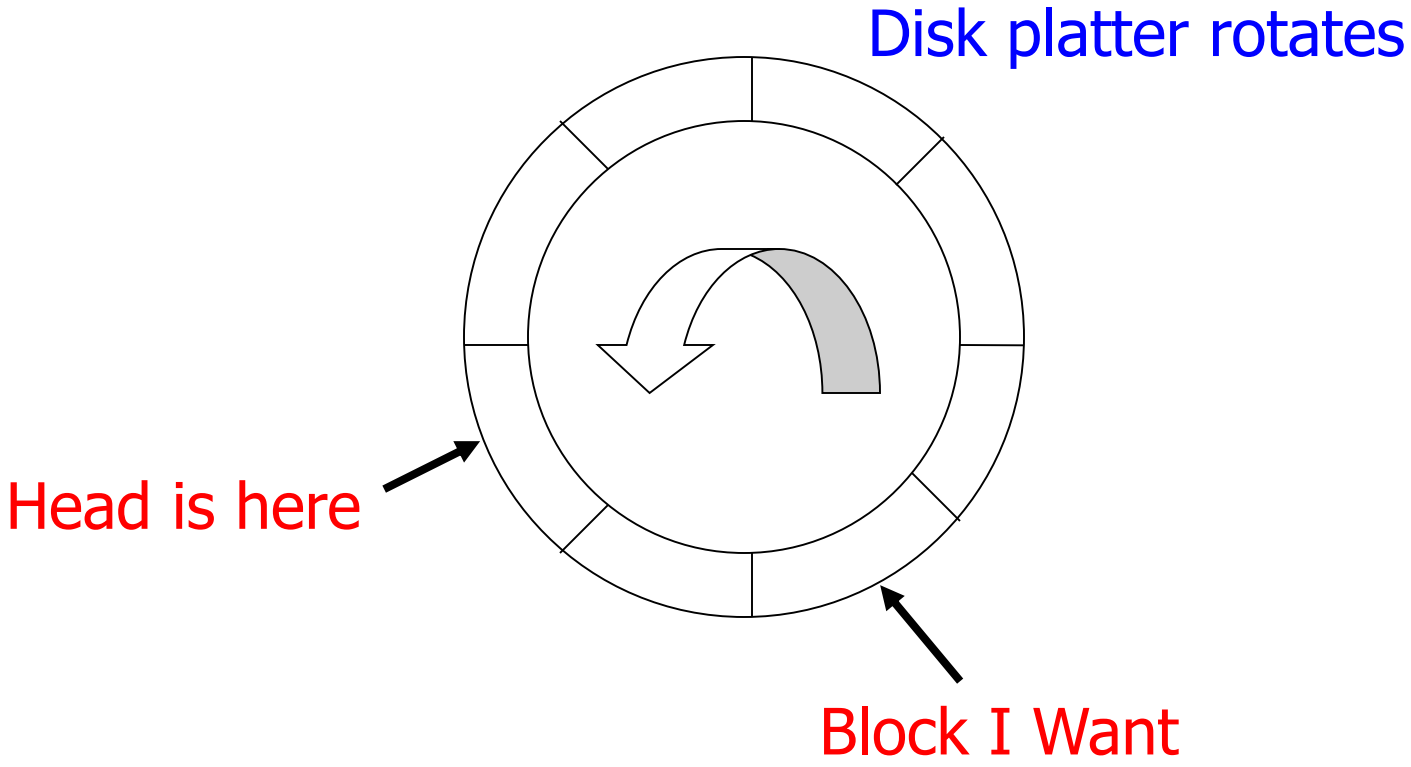
"Typical" S: 10 ms $\rightarrow$ 40 ms

Typical Seek Time

- Ranges from
  - 4ms for high end drives
  - 15ms for mobile devices
- Typical SSD (Solid State): ranges from
  - 0.08ms
  - 0.16ms

- Source: Wikipedia, "Hard disk drive performance characteristics"

Rotational Delay



Average Rotational Delay

R = 1/2 revolution

R=0 for SSDs

Typical HDD figures

HSpindle DD [rpm]	Average rotational latency [ms]
4,200	7.14
5,400	5.56
7,200	4.17
10,000	3.00
15,000	2.00

Source: Wikipedia, "Hard disk drive performance characteristics"

Transfer Rate: # bits transferred/sec

- Transfer rates:
  - HDD: up to 1000 Mbit/sec
  - 12x Blu-Ray: 432 Mbit/sec
  - 1xCD: 1.23 Mbits/sec
  - for SSDs, limited by interface
    e.g., SATA 3000 Mbit/s
- Transfer time: $\frac{\text{Amount data transferred}}{\text{Transfer rate}}$

Other Delays

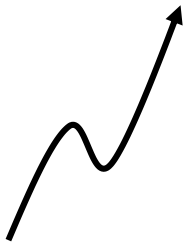
- CPU time to issue I/O
- Contention delay for disk controller
  - Different programs can be using the disk
- Contention delay for bus, memory
  - *Different programs* can be transferring data

These delays are negligible compared to
Seek time + rotational delay + transfer time

- So far: One (Random) Block Access
- What about: Reading "Next" block?

If we do things right (e.g., Double Buffer, Stagger Blocks…)

Time to get "next" block = Block Size / Transfer rate + Negligible



- skip gap
- switch track
- once in a while, next cylinder

Rule of Thumb	Random I/O: Expensive Sequential I/O: Much less

Cost for Writing similar to Reading

.... unless we want to verify:

need to add (full) rotation + Block size Transfer time

- To Modify a Block?

- To Modify a Block?

To Modify Block:

(a) Read Block into Memory

(b) Modify block in Memory

(c) Write Block

[(d) Verify?]

Random Access Time

- **Hand Drive:** Ranges from 2.9 msec (high end server drive) to 12 msec (laptop HDD)
- Due to the need to move the heads and wait for the data to rotate under the read/write head

Data Transfer Rate

- **Hard Disk:** Once the head is positioned, an enterprise HDD can transfer data at about 140 MBytes/sec.
- In practice, much lower speeds because….
- Data transfer rate depends also on rotational speed (of the platter) !

Reliability

- **Hard Disk:** According to a study performed by CMU for both consumer and enterprise-grade HDDs, their average failure rate is 6 years, and life expectancy is 9–11 years.

Cost and Capacity

- **Hard Drive:**
- In 2013: HDDs of up to 6 TB were available.
- In 2014: Cost: around $50 per TeraByte

Kibibytes

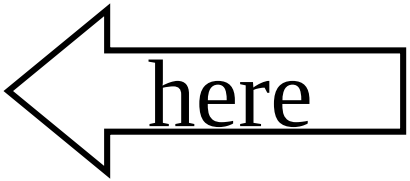
- 1 kibibyte = 2^{10} bytes = 1024 bytes.

Multiples of bytes				v · d · e
SI decimal prefixes		IEC binary prefixes		
Name (Symbol)	Value	Name (Symbol)	Value	
kilobyte (kB)	10^3	kibibyte (KiB)	$2^{10} = 1.024 \times 10^3$	
megabyte (MB)	10^6	mebibyte (MiB)	$2^{20} \approx 1.049 \times 10^6$	
gigabyte (GB)	10^9	gibibyte (GiB)	$2^{30} \approx 1.074 \times 10^9$	
terabyte (TB)	10^{12}	tebibyte (TiB)	$2^{40} \approx 1.100 \times 10^{12}$	
petabyte (PB)	10^{15}	pebibyte (PiB)	$2^{50} \approx 1.126 \times 10^{15}$	
exabyte (EB)	10^{18}	exbibyte (EiB)	$2^{60} \approx 1.153 \times 10^{18}$	
zettabyte (ZB)	10^{21}	**zebibyte** (ZiB)	$2^{70} \approx 1.181 \times 10^{21}$	
yottabyte (YB)	10^{24}	yobibyte (YiB)	$2^{80} \approx 1.209 \times 10^{24}$	
See also: Multiples of bits · Orders of magnitude of data				

from Wikipedia

Outline

- Hardware: Disks
- Access Times
- Optimizations
- Other Topics
  - Storage Costs
  - Using Secondary Storage
  - Disk Failures



Optimizations (in controller or O.S.)

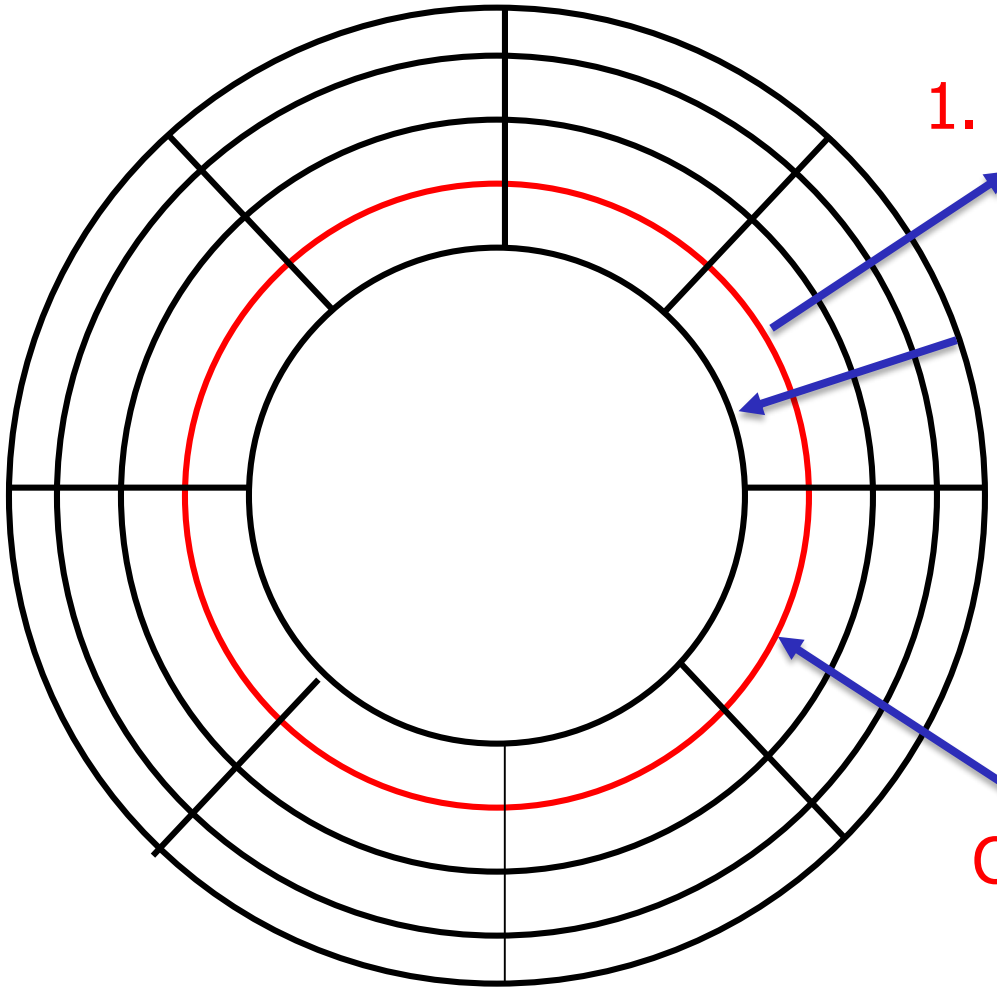
- Disk Scheduling Algorithms
  - e.g., elevator algorithm
- Pre-fetch (Double buffering)
- Arrays (RAID)
- Mirrored Disks

Disk Scheduling: Elevator Algorithm

Situation: Have many read/write requests

Question: In which order do you process the requests ?

Disk Scheduling: Elevator Algorithm



1. Process requests for these cylinders

2. Then process requests this way

Current cylinder

Double Buffering Algorithm

Problem: You have a File

» Sequence of Blocks B1, B2, …, Bn

You have a Program that:

» Process B1

» Process B2

» Process B3

⋮

Single Buffer Solution ("naïve" solution)

(1) Read B1 $\rightarrow$ Buffer

(2) Process Data in Buffer

(3) Read B2 $\rightarrow$ Buffer

(4) Process Data in Buffer ...

Say P = time to process/block

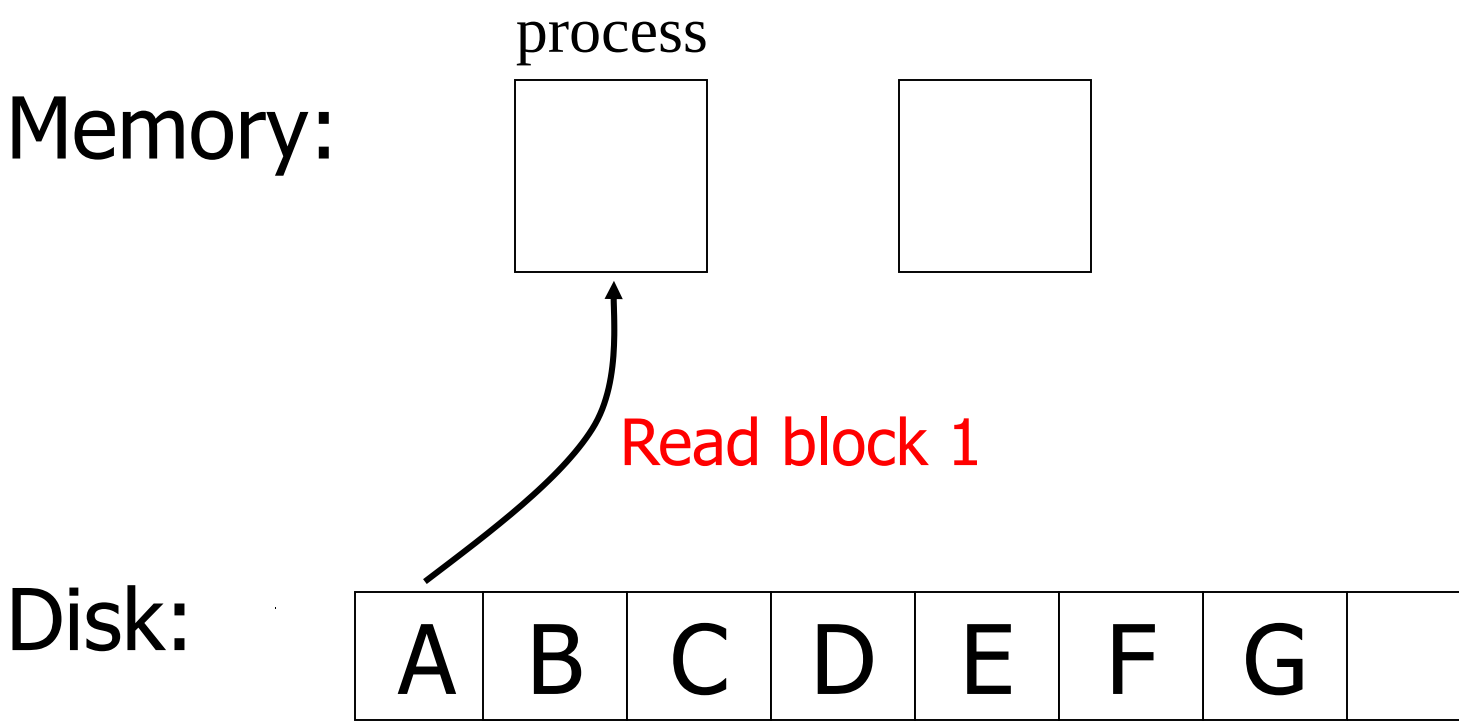
R = time to read in 1 block

n = # blocks

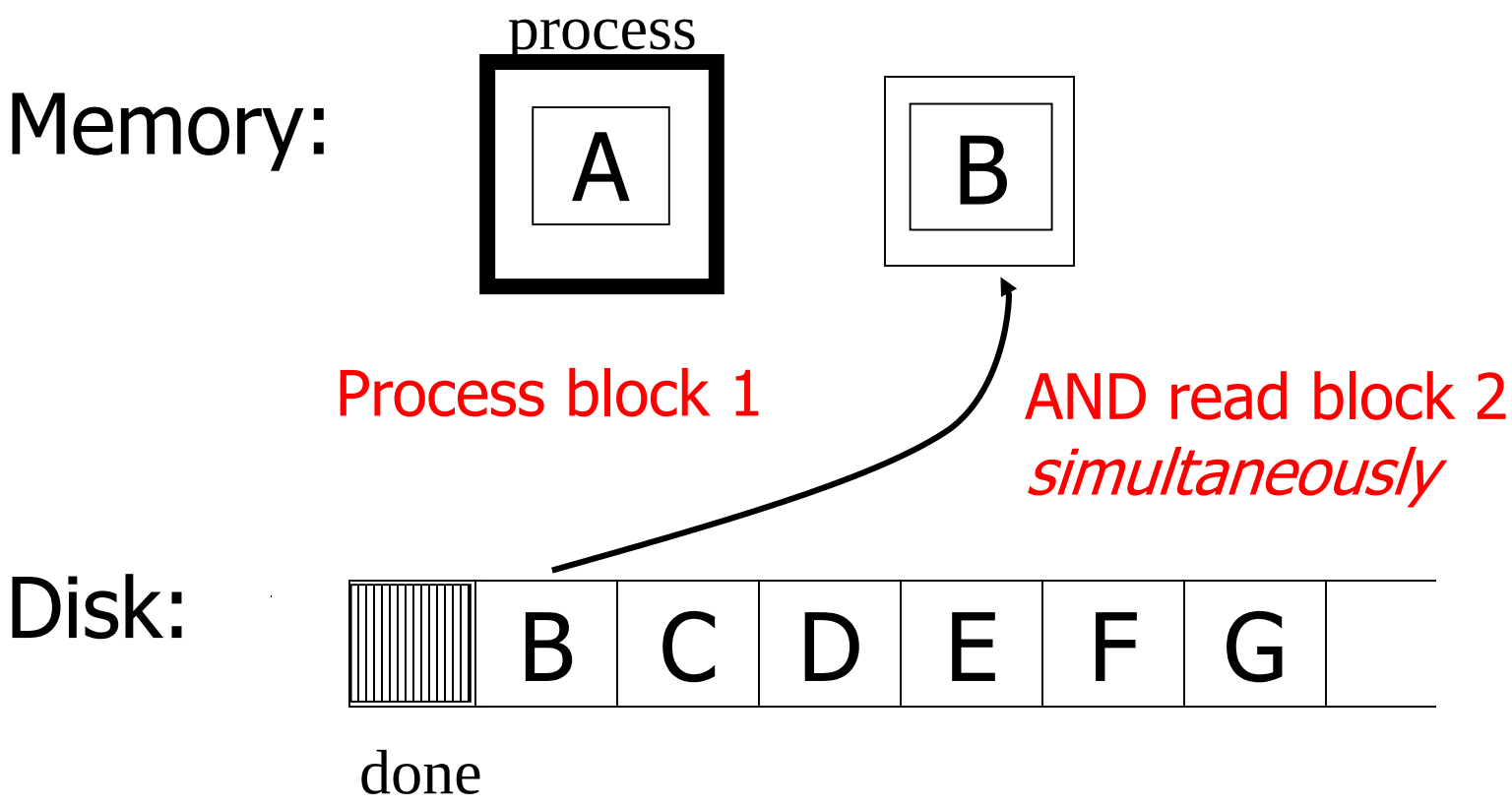
(1) Read B1 → Buffer → R
(2) Process Data in Buffer → P
(3) Read B2 → Buffer → R
(4) Process Data in Buffer ... → P

Time to process n block = n(P + R)

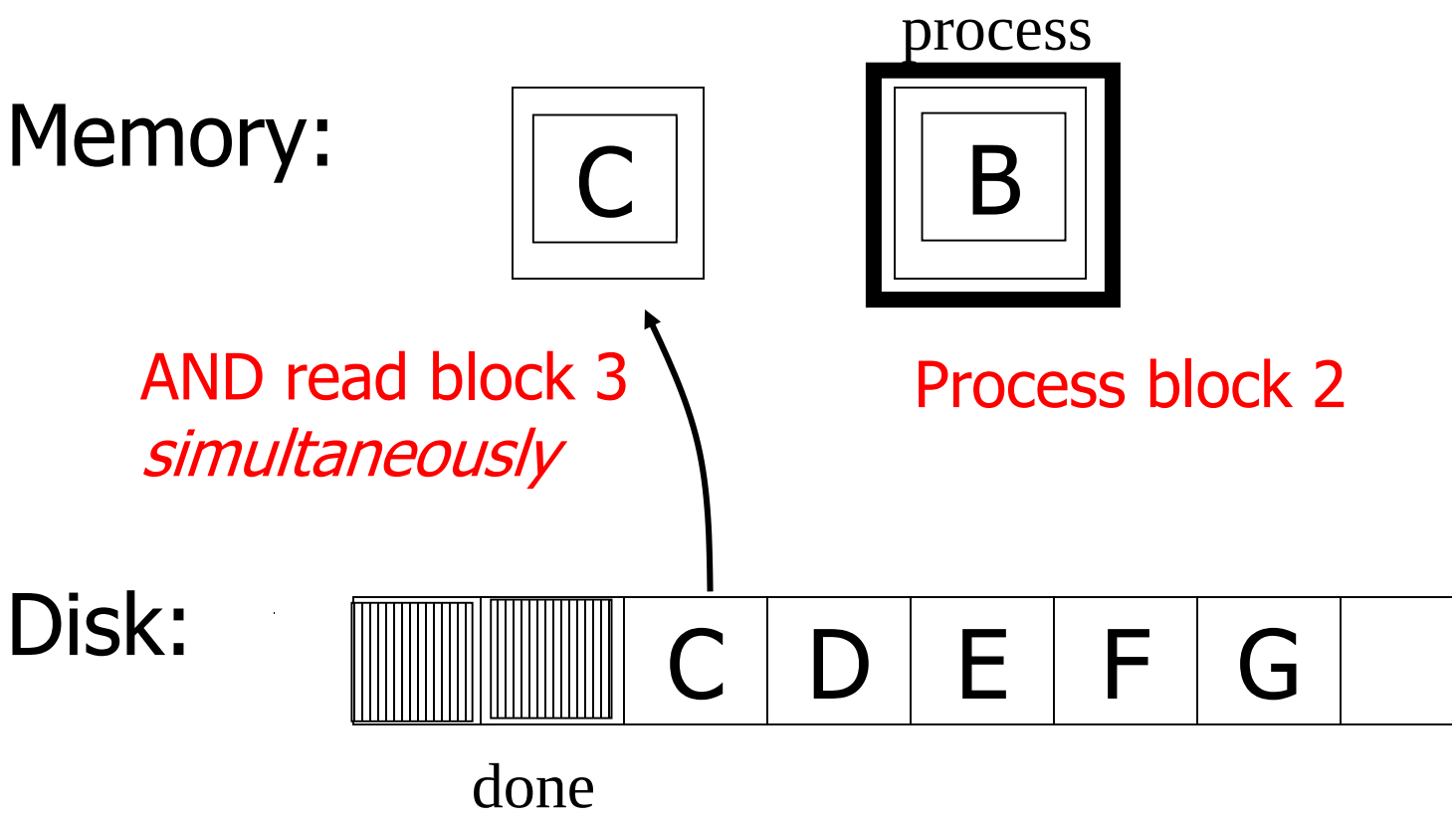
Double Buffering



Double Buffering



Double Buffering

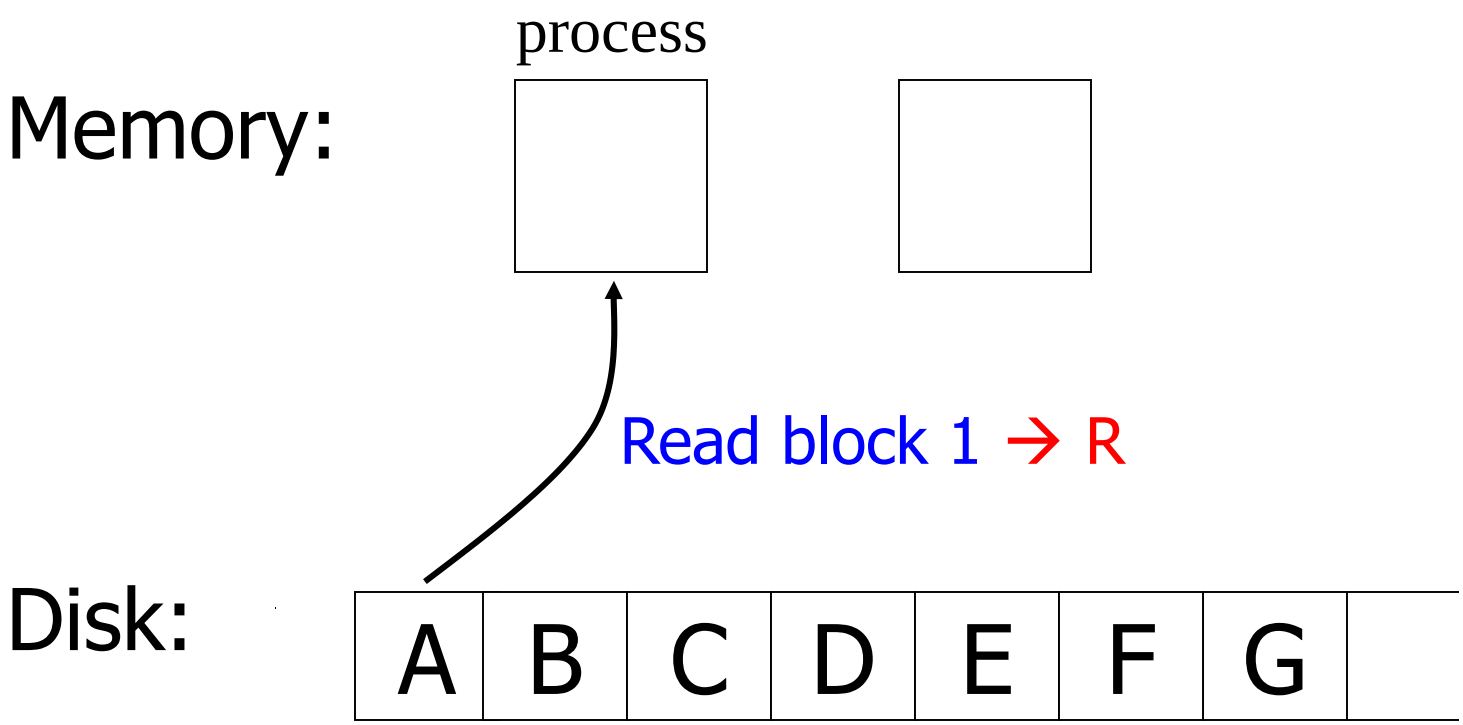


Say P > R

P = Processing time/block
R = IO time/block
n = # blocks

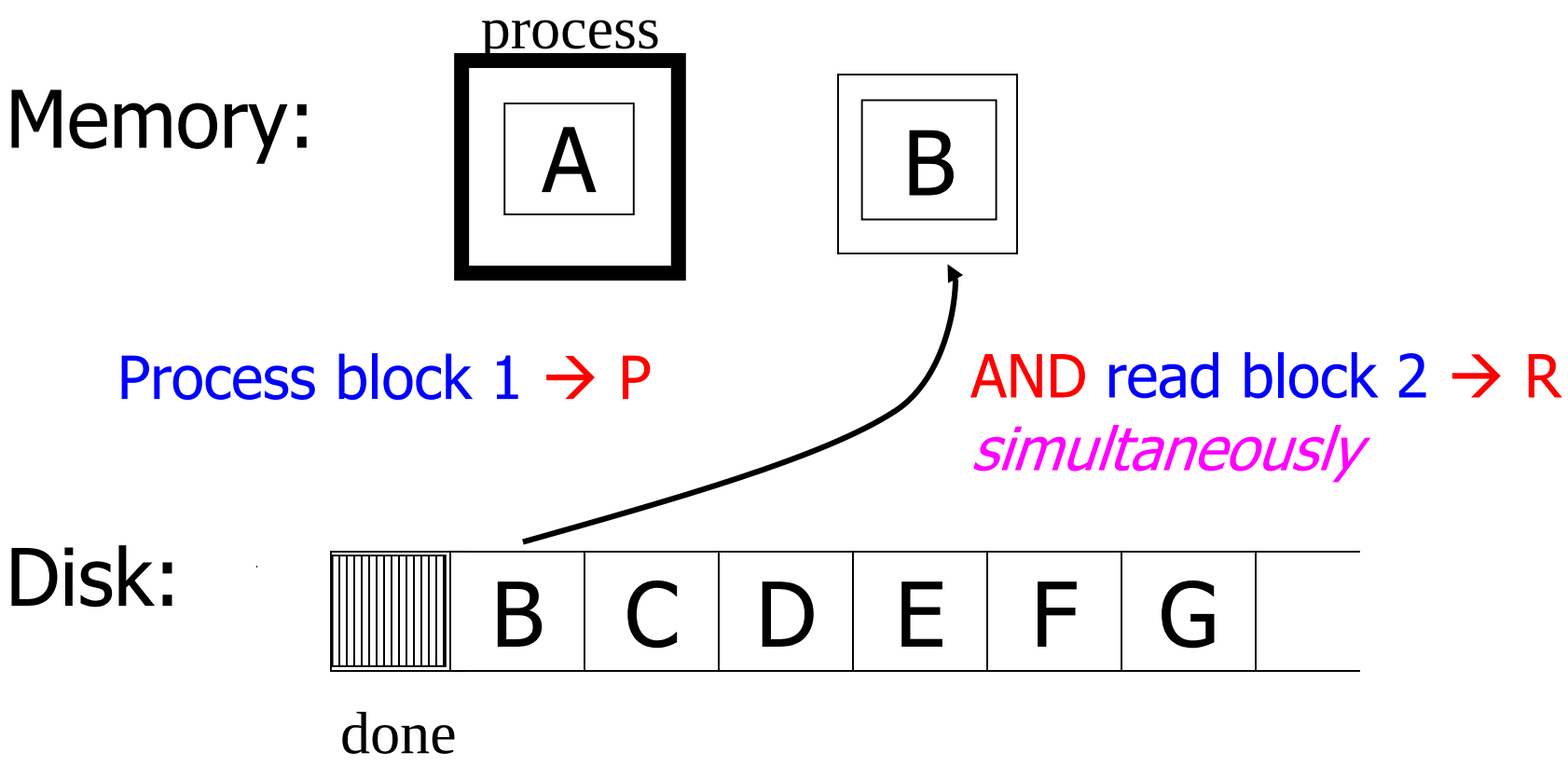
What is processing time?

Double Buffering



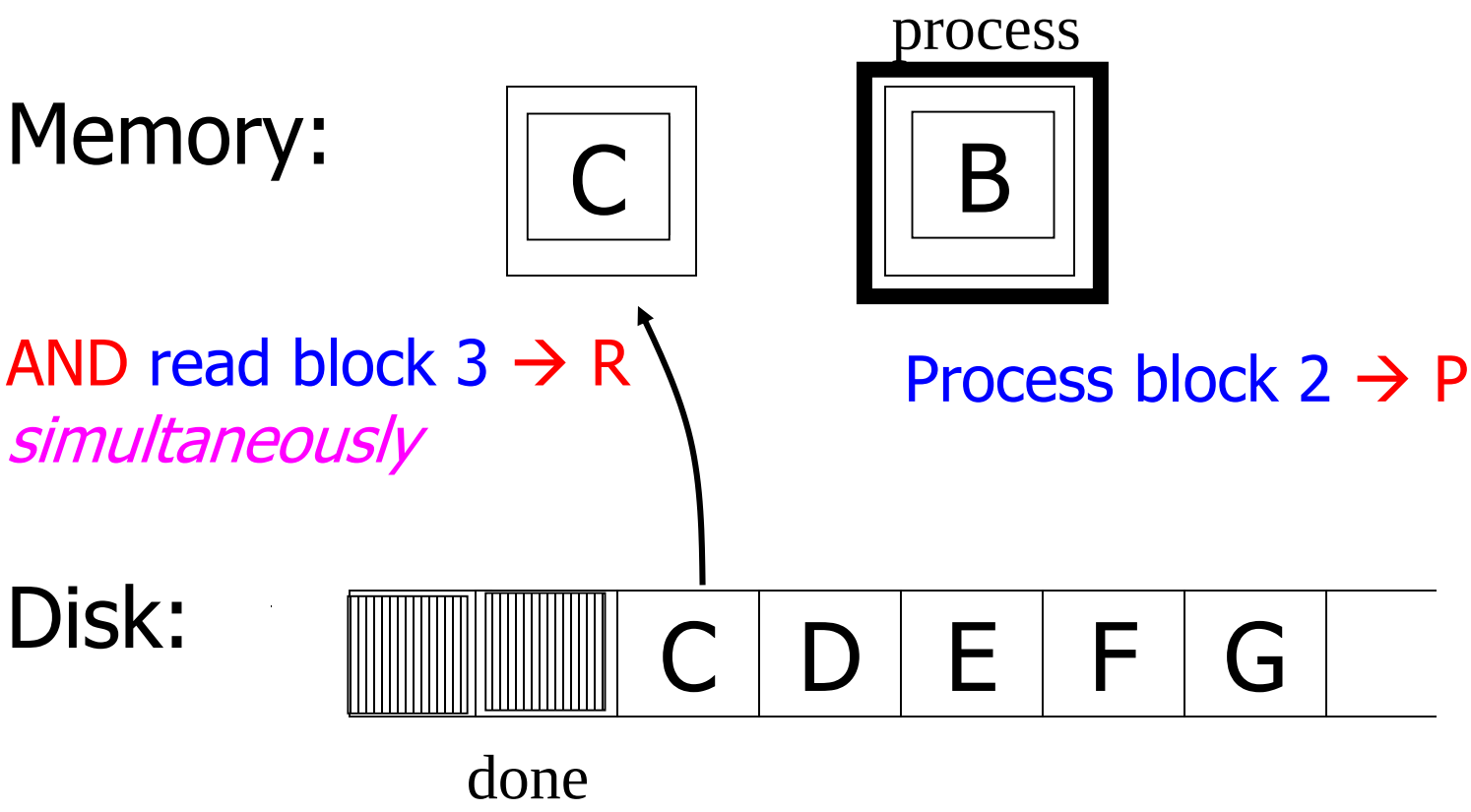
Double Buffering

Time needed = P (P > R)



Double Buffering

Time needed = P (P > R)



Say $P \geq R$

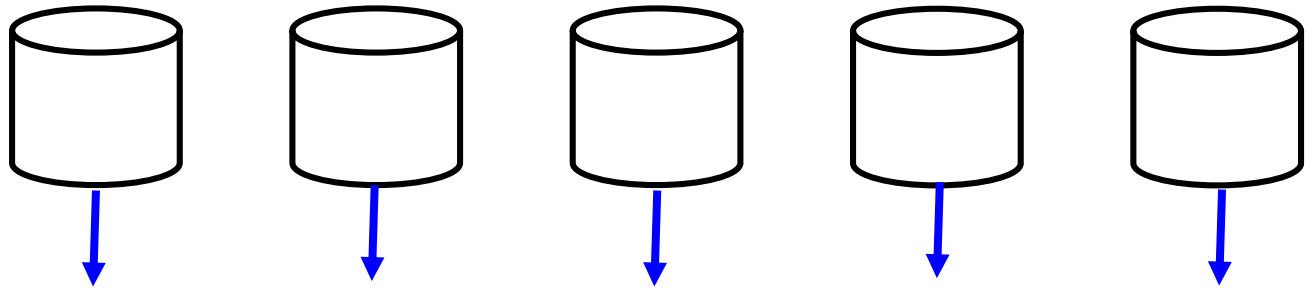
P = Processing time/block
R = IO time/block
n = # blocks

What is processing time?

- Double buffering time = $R + nP$
- Single buffering time = $n(R+P)$

Using disk array to accelerate disk access

- Why use multiple disks:
  - Multiple disks → multiple disk heads
  - Multiple outputs = Increased data rate

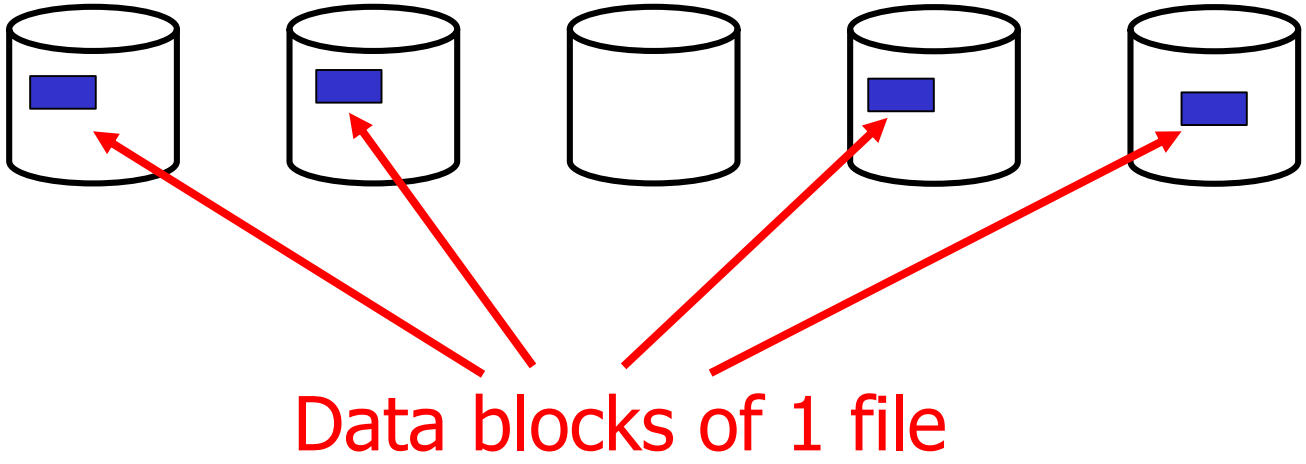


Techniques to deploit multiple disks

- Block Striping:
  - Store blocks of a file over multiple disks
  - (This technique uses multiple disks as point 2)
- Mirror disk:
  - Store the same data on multiple disks
- RAID:
  - Redundant Array of Independent (inexpensive) Disks

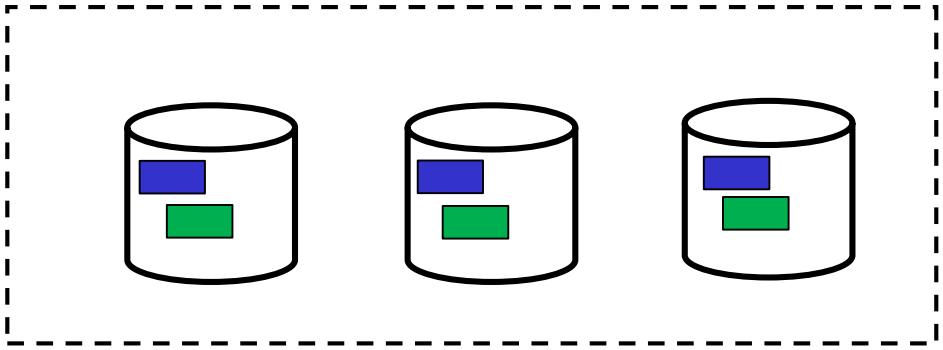
Block Striping

- Blocks of the same file stored on different disks



Disk Mirroring

- Mirrored disks contain identical content

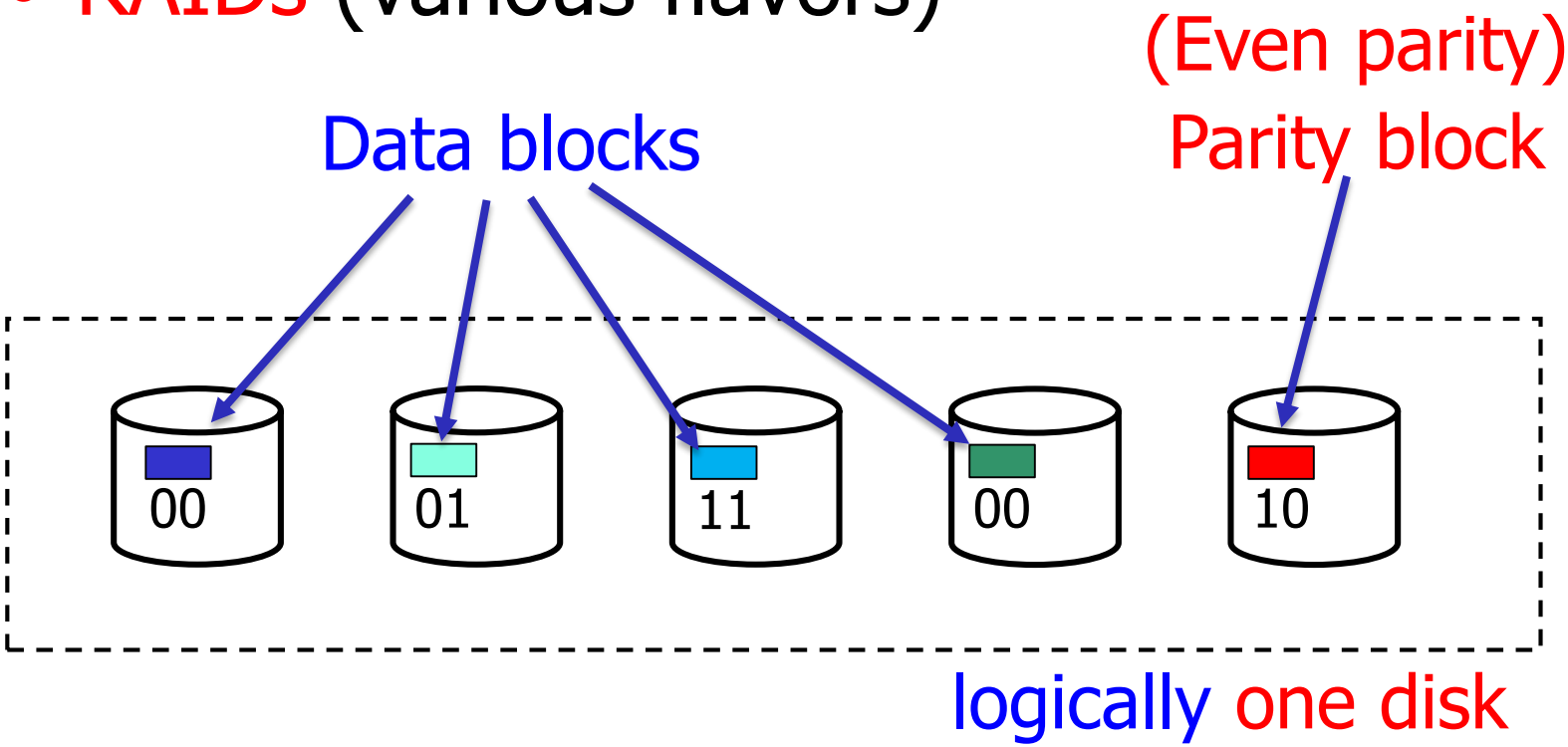


logically one disk

- Read operation: n times as fast
- Write operation: about the same as 1 disk

Disk Arrays

- RAIDs (various flavors)



Disk Failures

- Intermittent read failure
  - Cause: power fluctuations/failure
- Intermittent write failure
  - Cause: power fluctuation/failure
- Media decay ← discuss first
  - Disk surface worn out
- Permanent failure ← redundancy…
  - Disk crash

Coping with media decay

- Disk has a number of spare blocks
- When writing a block fails for n times:
  - Mark block as bad
  - Replace block with one of the spare blocks

Coping with Read/Write Failures

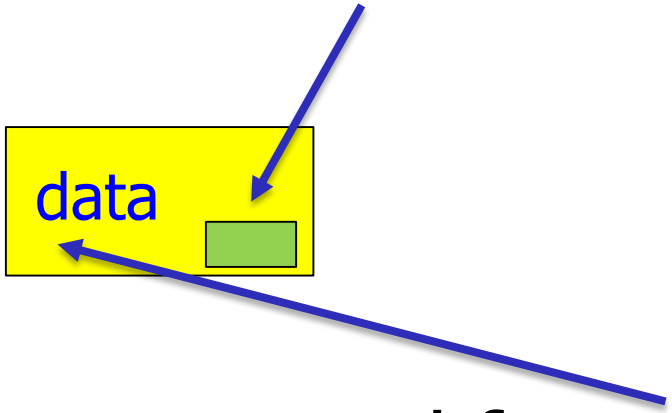
- Detection:
  - Read (verify) after writing data
  - Better: Use checksum

- Detect and Correct:

  $\Rightarrow$ Redundancy

Detecting read error:

- Block contains a check sum:

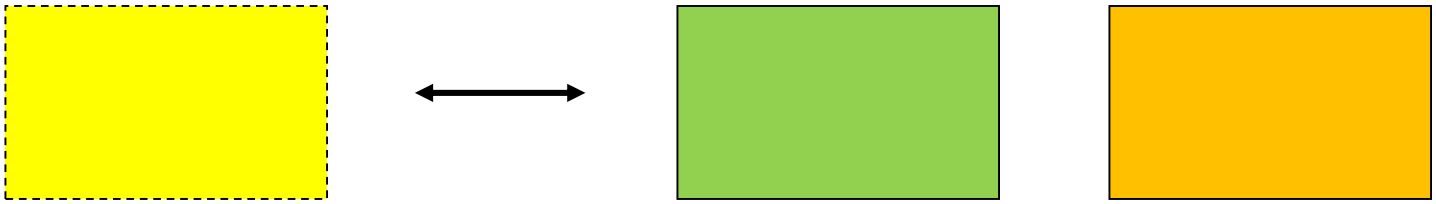


- Check sum computed from data in block
- Reading a data block:
  - Re-compute check sum with data and verify with recorded checksum

Power failure during a write operation

- Copy of data in memory will be lost
- Copy of data on disk may be corrupted
- Bottom line:
  - Power failure during a write operation can be catastrophic data lost
- Solution: stable storage update policy

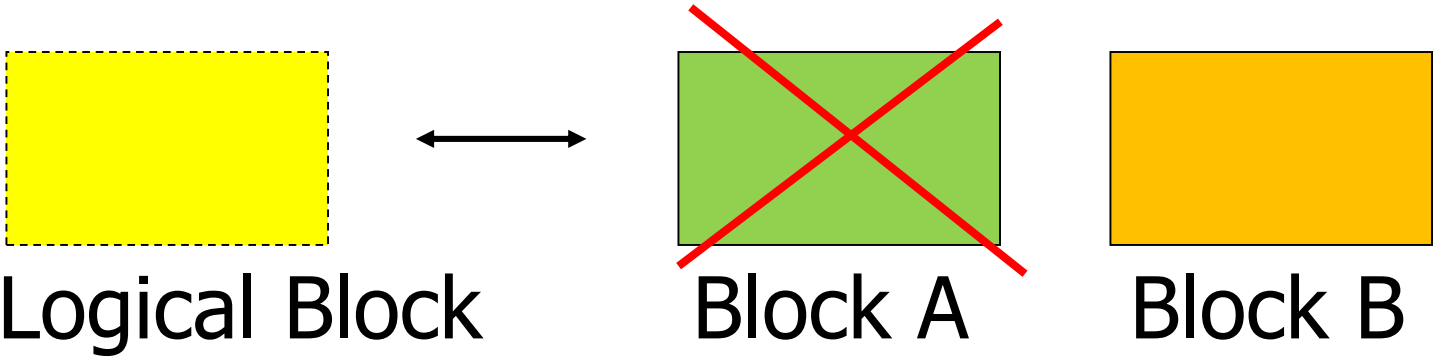
Stable Storage Update Policy



1 Logical Block Block A Block B

1. Write data to Block A
2. Read data back and verify (repeat if needed)
    1. If Block A write fail after n tries
        1. Mark Block A as bad
        2. Replace with spare block
        3. Repeat
3. Write data to Block B …

Stable Storage Update Policy



Power failure during Block A write:

We still have an older copy of data in Block B

Power failure during Block B write:

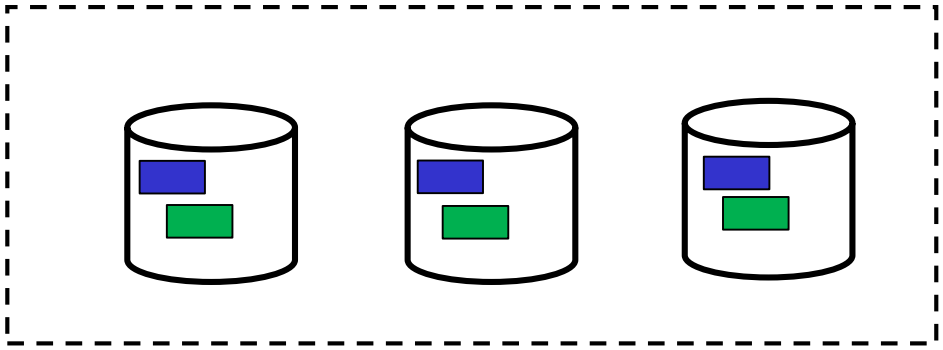
The new copy was written correctly

Coping with Disk Crash

- "3 ways":
  - Redundancy, redundancy, redundancy
- Different ways to achieve redundancy:
  - Exact copy (mirror)
  - RAID

Disk Mirroring

- Mirrored disks contain identical content

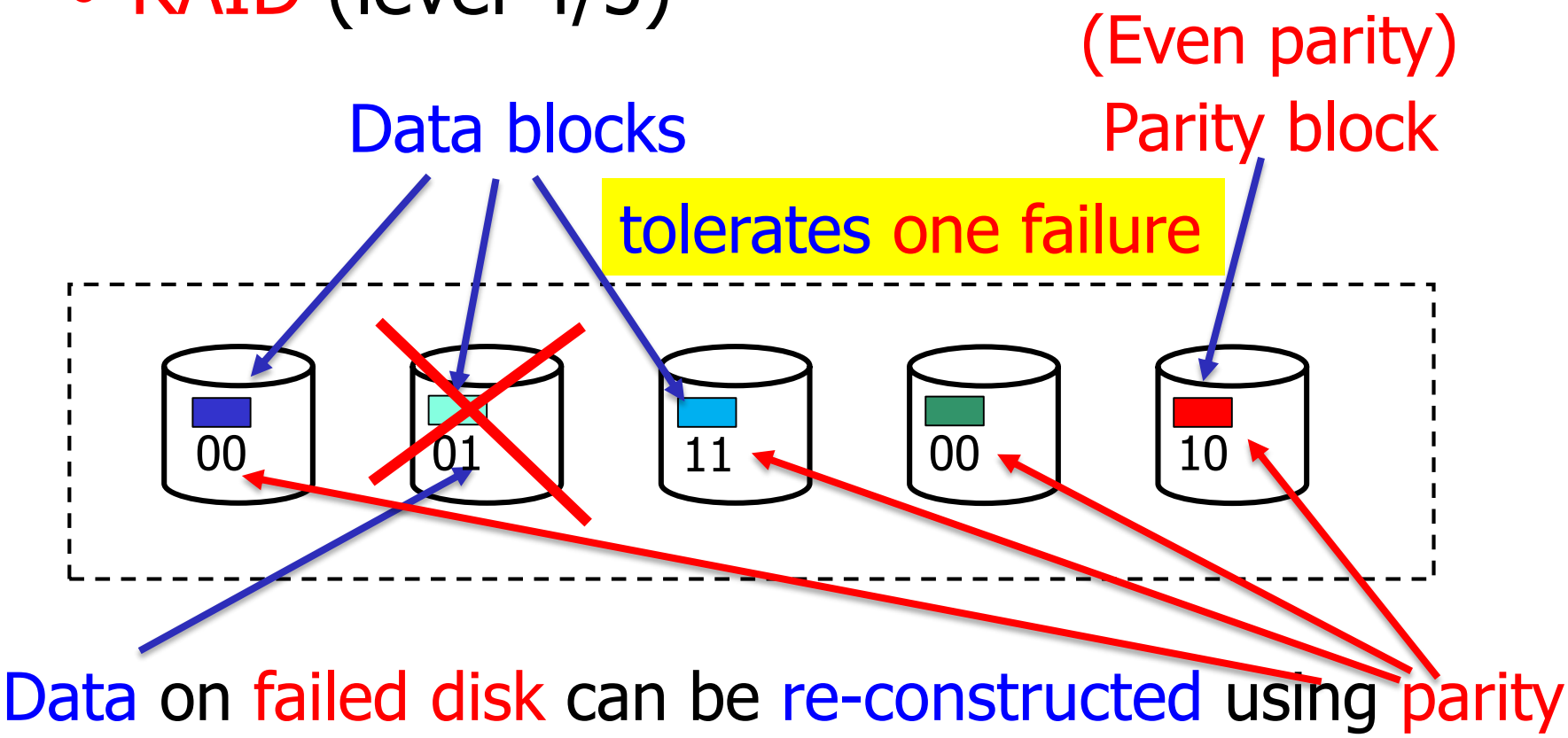


logically one disk

- Advantage: tolerates n-1 disk failures
- Disadvantage: expensive…

Disk Arrays

- RAID (level 4/5)



Summary

- Secondary storage, mainly disks
- I/O times
- I/Os should be avoided (if possible), especially random ones.....

Outline

- Hardware: Disks
- Access Times
- Optimizations
- Other Topics
  - Storage Costs
  - Using Secondary Storage
  - Disk Failures