Hardware

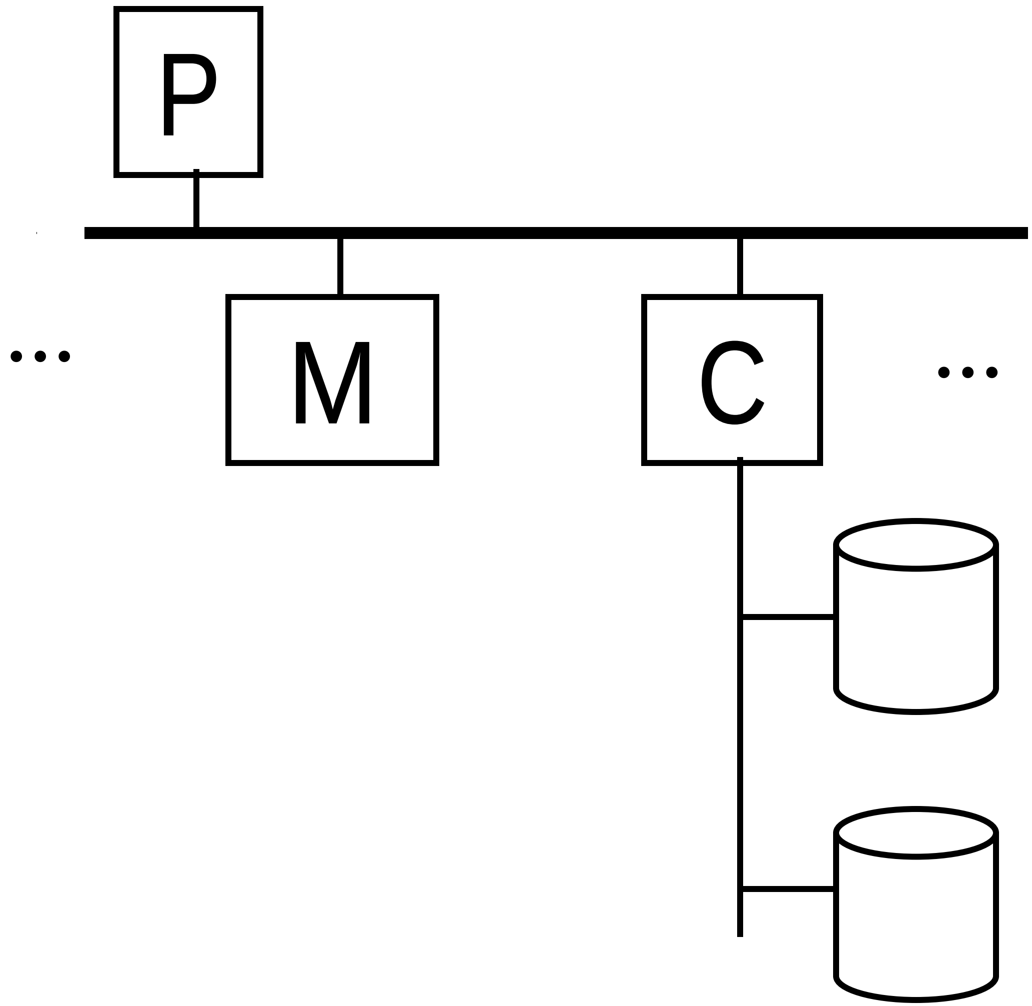
Read Chap. 4 Riguzzi et al. Sistemi Informativi

Slides derived from those by Hector Garcia-Molina
Some images by Wikipedia

Outline

- Hardware: Disks
- Access Times
- Reliability
- RAID

Typical Computer



Secondary Storage

Processor

Fast, slow, reduced instruction set, with cache, pipelined...

Speed: 10000 $\rightarrow$ 100000 MIPS

Memory

Fast, slow, non-volatile, read-only,...

Access time: $10^{-6} \rightarrow 10^{-9}$ sec.

$1\ \mu s \rightarrow 1$ ns

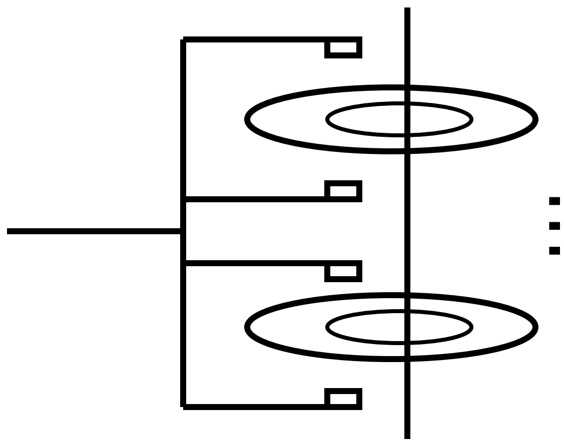
Secondary storage

- Hard Disks

Tertiary storage

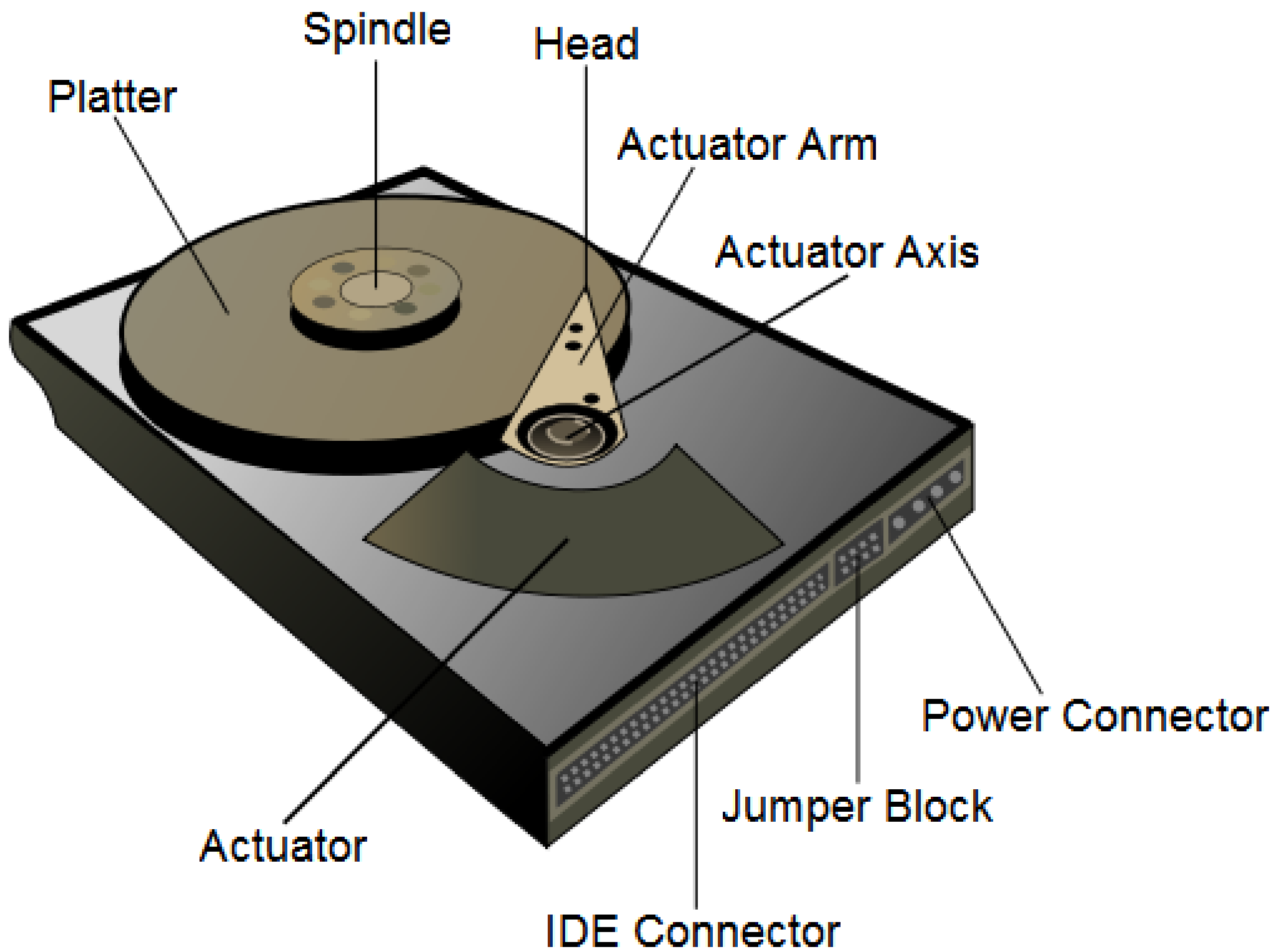
- Optical disks:
  - CD-ROM
  - DVD-ROM...
- Tape
  - Cartridges
- Robots

Focus on: "Typical Disk"

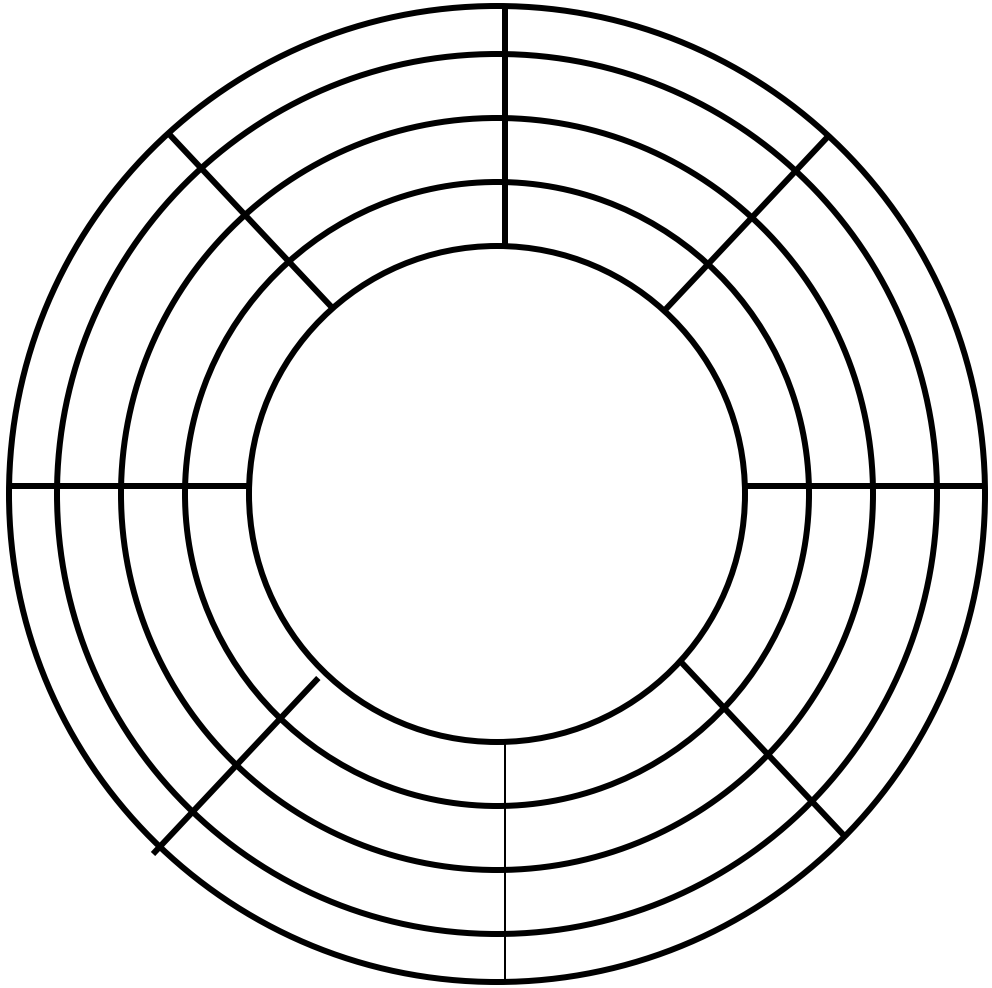


Terms: Platter, Surface, Head, Actuator
Cylinder, Track
Sector (physical),
Block (logical), Gap

Disk Architecture



Top View



“Typical” Numbers

Diameter: 1.8, 2.5 or 3.5 inches (1 inch=2.54 cm)

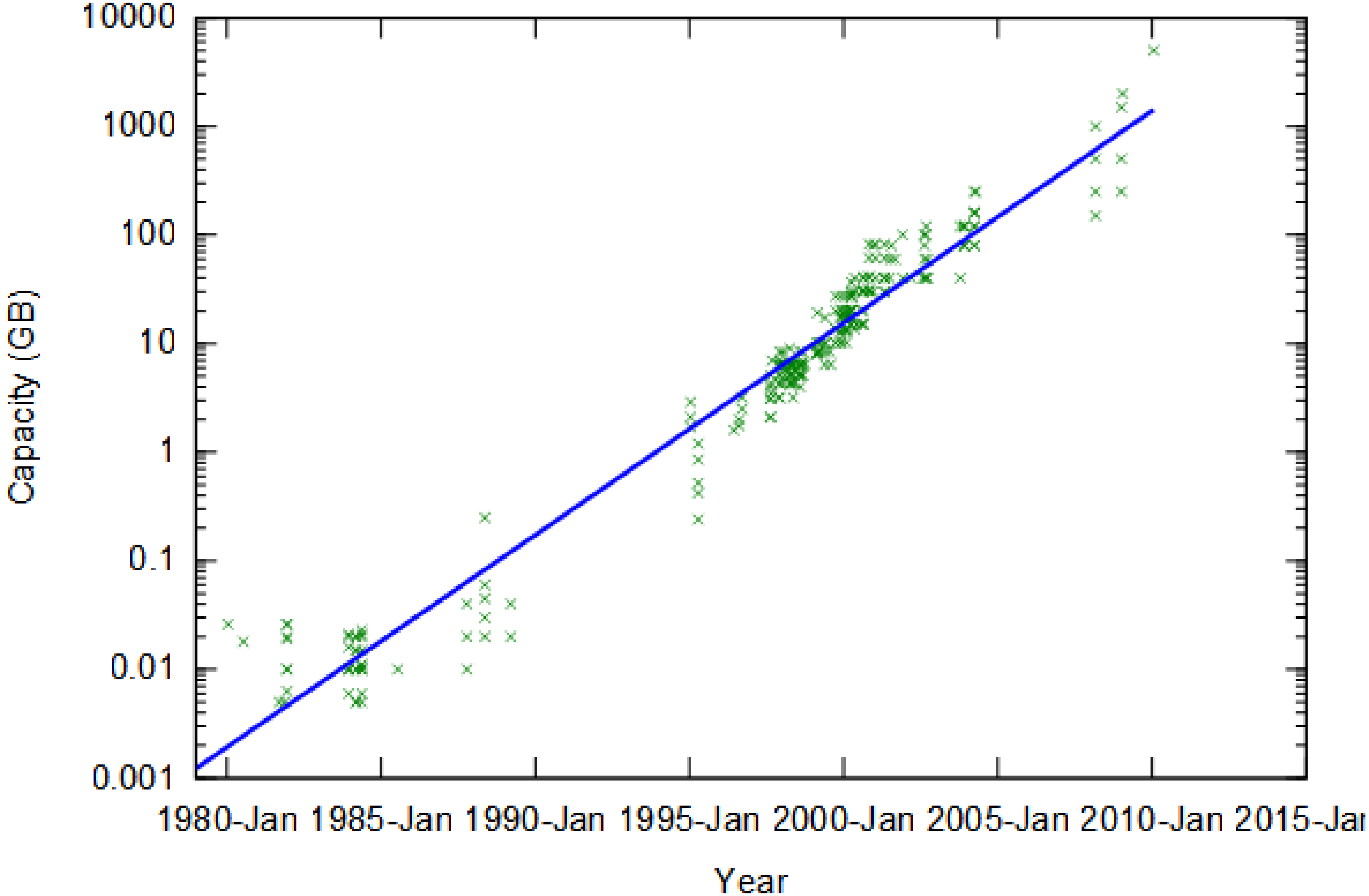
Cylinders: 10000 $\rightarrow$ 50000

Platters: 2 -> 7 (Tracks/cyl)

Sector Size: 512B $\rightarrow$ 50KB

Capacity: 72 GB $\rightarrow$ 6TB

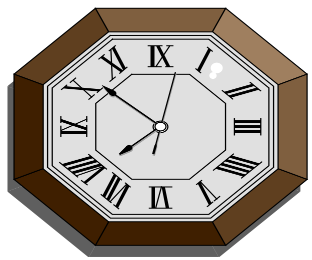
Capacity



10000
1000
100
10
1
0.1
0.01
0.001
Capacity (GB)
1980-Jan 1985-Jan 1990-Jan 1995-Jan 2000-Jan 2005-Jan 2010-Jan 2015-Jan
Year

Disk Access Time

I want
block X

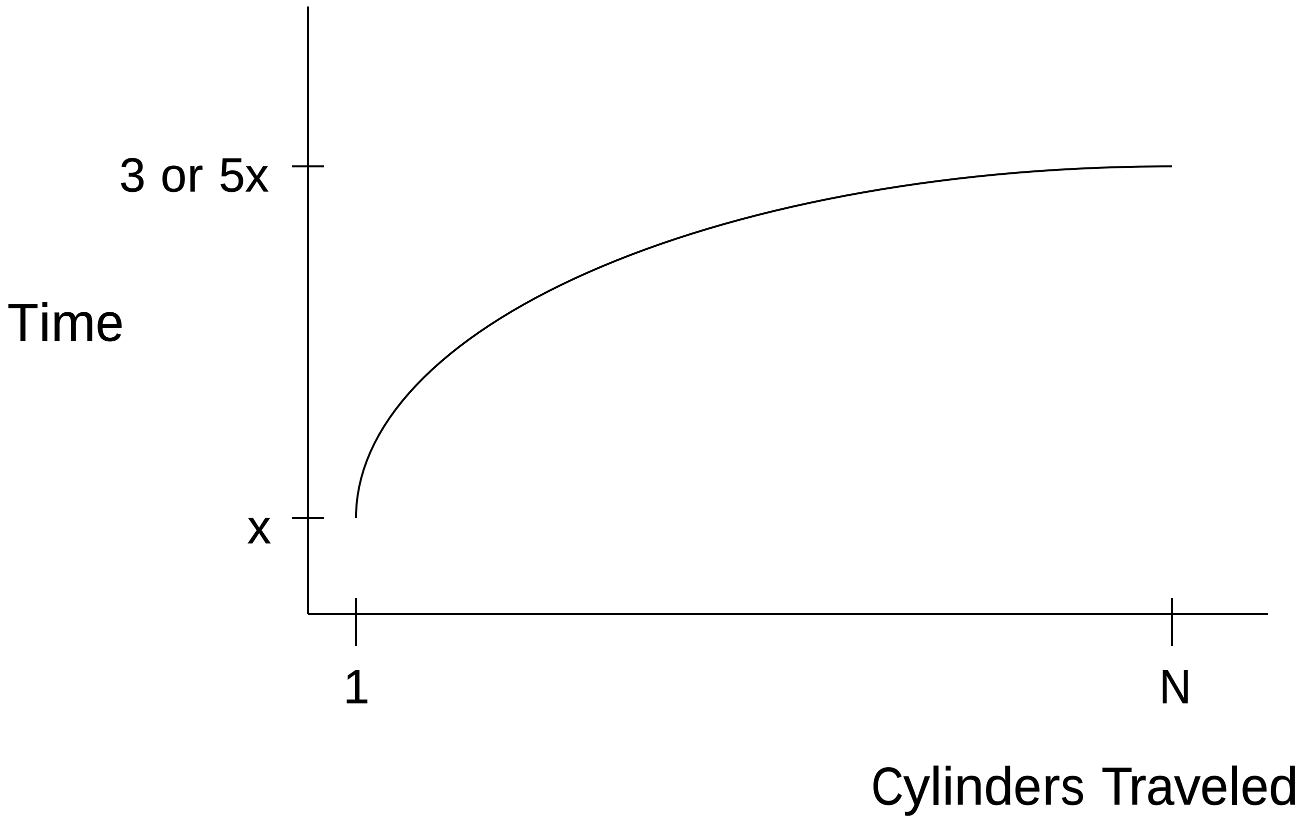


?

block x
in memory

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

Seek Time



Average Random Seek Time

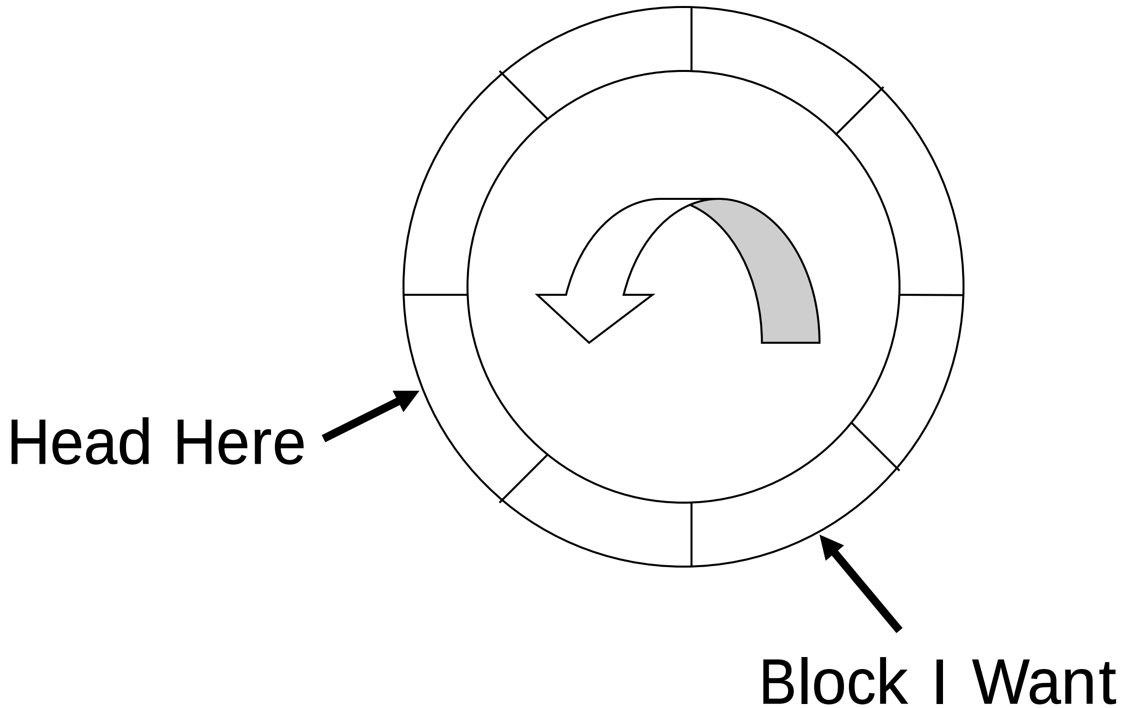
$$S = \frac{\sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \text{SEEKTIME}(i \rightarrow j)}{N(N-1)}$$

"Typical" S: 3 ms $\rightarrow$ 10 ms

Seek time

- Average seek time ranges from under 4 ms for high-end server drives to 15 ms for mobile drives
- The most common mobile drives at about 12 ms
- The most common desktop type typically being around 9 ms.

Rotational Delay



Average Rotational Delay

R = 1/2 revolution

"typical" R = 4.17 ms (7200 RPM)
R=3 ms (10000 RPM)
R=2 ms (15000 RPM)

Transfer Time

- transfer time: revolution/n. blocks per track

Other Delays

- CPU time to issue I/O
- Contention for controller
- Contention for bus, memory

“Typical” Value: 0

- So far: Random Block Access
- What about: Reading “Next” block?

Time to get block = revolution/blocks

Cost for <u>Writing</u> similar to <u>Reading</u>

.... unless we want to verify!

need to add (full) rotation + revolution/blocks

- To Modify a Block?

To Modify Block:

(a) Read Block

(b) Modify in Memory

(c) Write Block

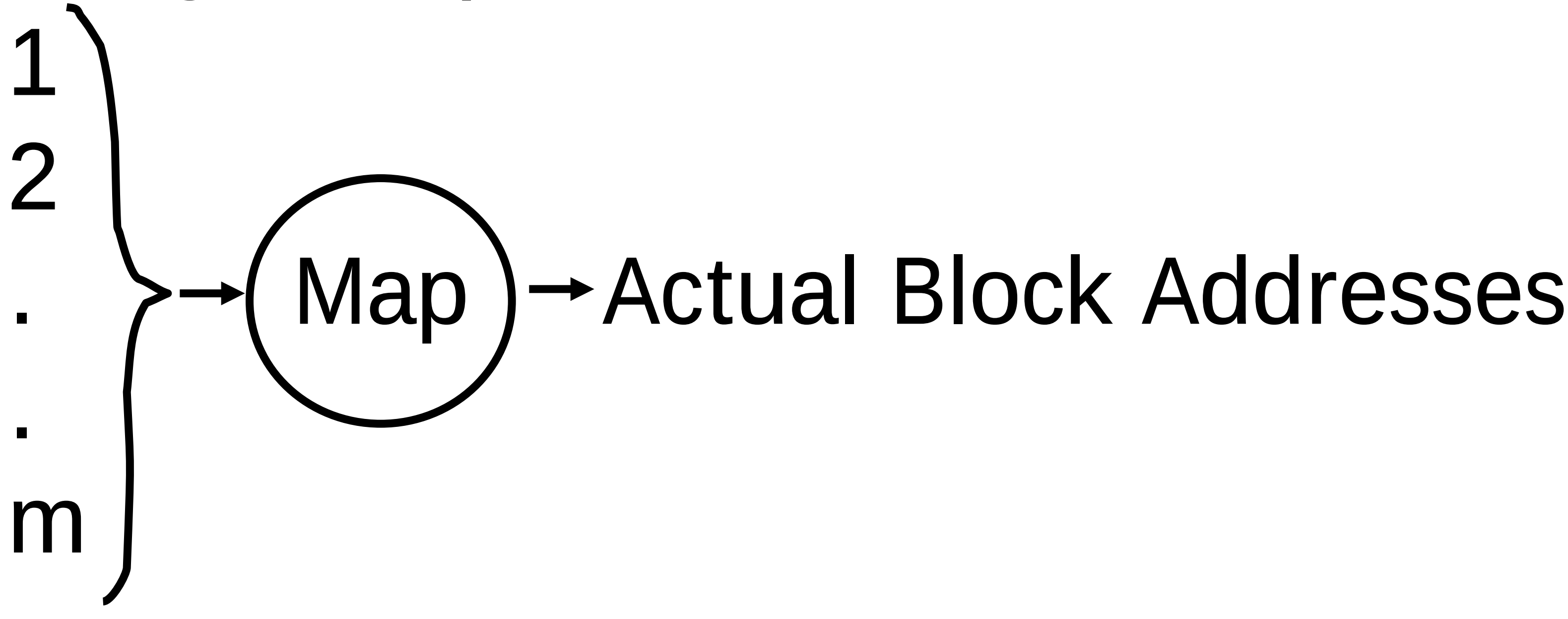
[(d) Verify?]

Block Address:

- Physical Device
- Cylinder (Track) #
- Surface #
- Sector

Complication: Bad Blocks

- Messy to handle
- May map via software to integer sequence



An Example

Megatron 747 Disk

- 3.5 in diameter
- 8 platters, 16 surfaces
- 2^{14}=16,384 tracks per surface (16,384 cylinders)
- 2^{7}=128 sectors per track
- 2^{12}=4096 bytes per sector

- Capacity
  - Disk=$2^{4}*2^{14}*2^{7}*2^{12}=2^{37}$=128GB
  - Single track=$2^{7}*2^{12}$=512KB

Megatron 747 Disk

- Rotation speed: 7200 RPM
- Average seek time: 8.5 ms

Layout

- Radius: 1.75 inches
- The tracks occupy the outer inch
- The inner 0.75 inch is unoccupied
- Track density in the radial direction: 16,384 tracks per inch
- 10% overhead between blocks

Density of bits

- Outermost track
  - Length=3.5π≈11 inches
  - One track = 512KB = 4Mbits
  - 90% of 11 inches holds 4Mbits
  - Density=420,000 bits per inch
- Innermost track
  - 90% of 4.71 inches holds 4Mbits
  - Density≈1Mbit per inch

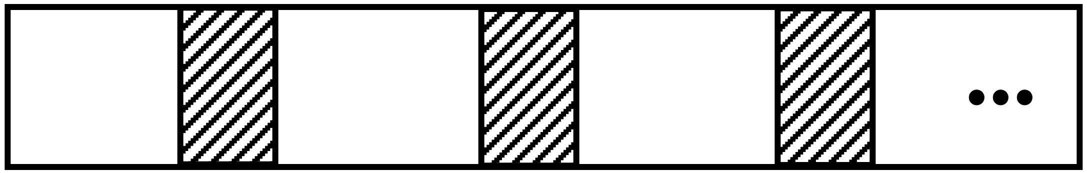
Density of bits

- To avoid such a high difference of density, the disk stores more sectors on the outer track than on the inner tracks
  - 96 sectors per track in the inner third
  - 128 in the middle third
  - 160 in the outer third
- The density varies from 742,000 bits per inch to 530,000 bits per inch

7200 RPM → 120 revolutions / sec

→ 1 rev. = 8.33 msec.

One track:



Time over useful data:(8.33)(0.9)=7.5 ms.

Time over gaps: (8.33)(0.1) = 0.833 ms.

Transfer time 1 sector = 7.5/128=0.059 ms.

Trans. time 1 sector+gap=8.33/128=0.065ms.

Burst Bandwith

4 KB in 0.059 ms.

BB = 4/0.059 = 68 KB/ms.

or

BB =68 KB/ms x 1000 ms/1sec

x 1MB/1024KB

= 68,000/1024 = 66.4 MB/sec

Sustained bandwith (over track)

512 KB in 8.33 ms.

SB = 512/8.33 = 61.5 KB/ms

or

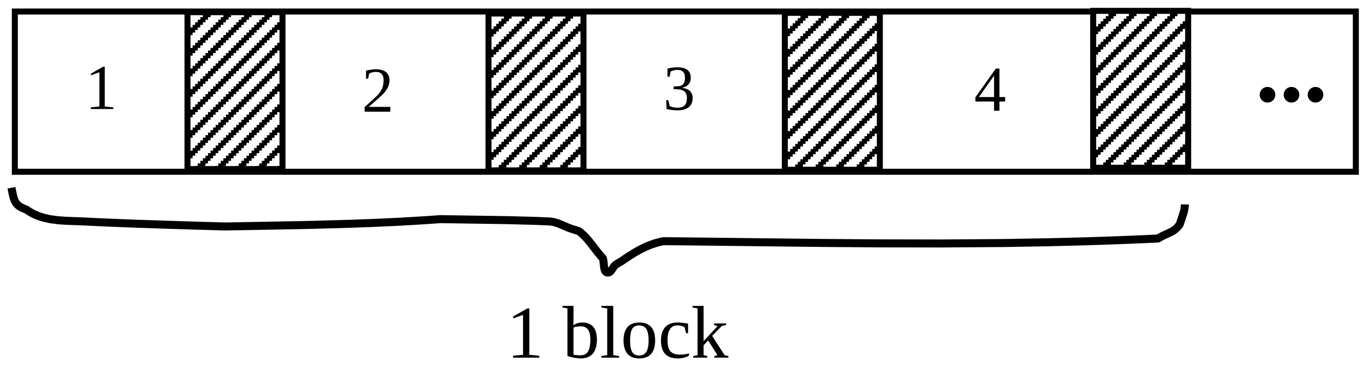
SB = 61.5 x 1000/1024 = 60 MB/sec.

T_1 = Time to read one random block

T_1 = seek + rotational delay + TT

= 8.5 + (8.33/2) + 0.059 = 12.72 ms.

Suppose OS deals with 16 KB blocks



T_4 = 8.5 + (8.33/2) + 0.059*1 + (0.065) * 3 = 12.92 ms

[Compare to T_1 = 12.72 ms]

T_T = Time to read a full track
(start at any block)

$T_T = 8.5 + (0.065/2) + 8.33^{*} = 16.86$ ms

to get to first block

* Actually, a bit less; do not have to read last gap.

Block Size Selection?

- Big Block $\rightarrow$ Amortize I/O Cost

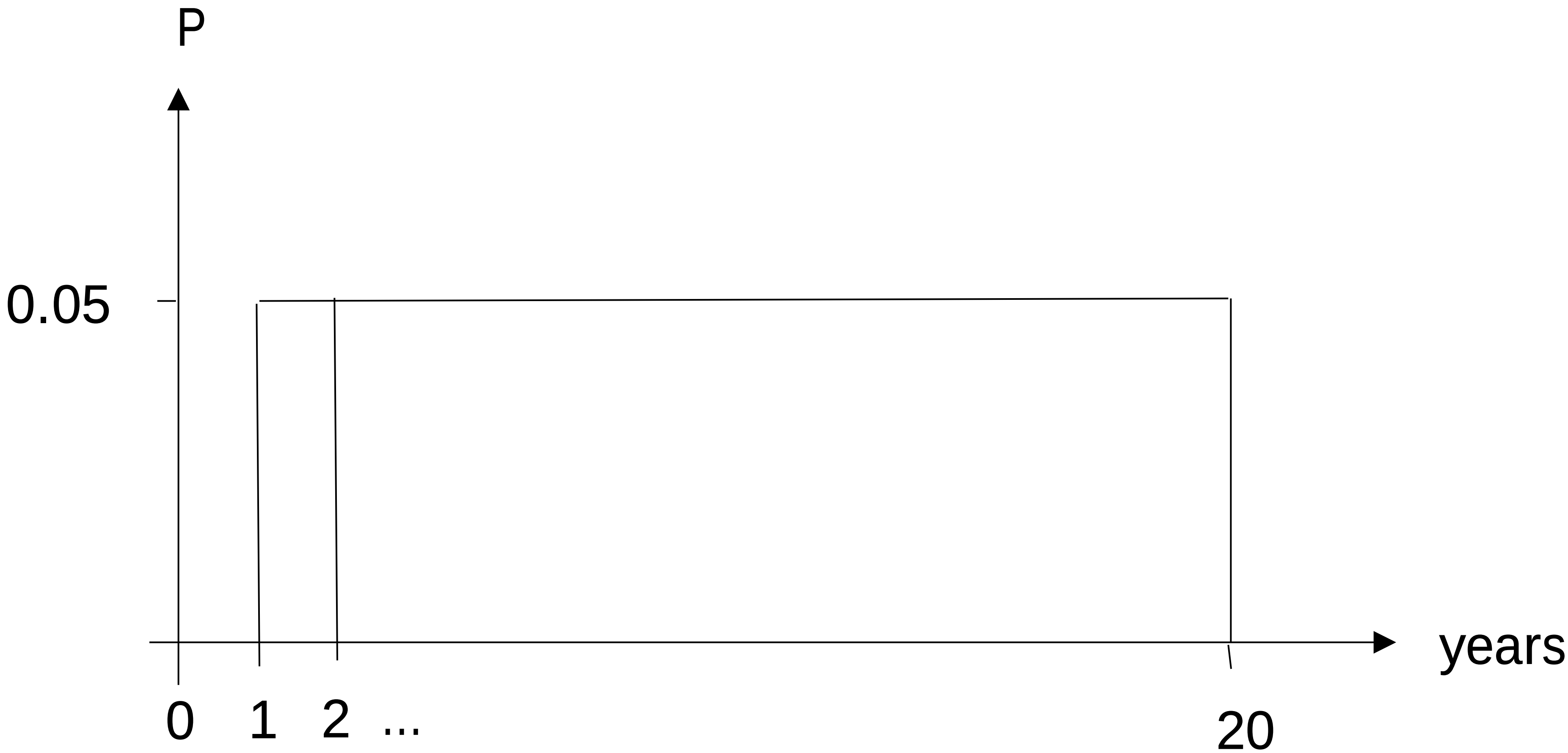
Unfortunately…

- Big Block $\Rightarrow$ Read in more useless stuff!
  and takes longer to read

Reliability

- Measured by the Mean Time to Failure (MTTF):
  - Length of time by which 50% of a population of disks will have failed catastrophically (head crash, no longer readable)
  - For modern disks, the MTTF is 10 years
  - This means that, on average, after 10 years it will crash
  - We can assume that every year 5% of the disks fail (uniform distribution assumption)
    - Probability that a disk fails in one year P_F=5%=1/20

Probability of Failure



MTTF

- Expected value of the failure year:
- MTTF=E(year)=
  =0.05*1+....+0.05*20=
  =0.05*20*(20+1)/2=21/2 $\approx$ 10

Disk Arrays

- Redundant Arrays of Inexpensive Disks (RAID)
- Two aims: increase speed and reliability

RAID 0

- Uses “block level striping”
  - Blocks that are consecutive for the OS are distributed evenly across different disks

RAID 0

A1	A2	consecutive blocks: A1-A8
A3	A4	
A5	A6	
A7	A8	

RAID 0

- Improves reading and writing speed
  - With two disks, two blocks can be read at the same time
  - A request for block "A1" would be serviced by disk 1. A simultaneous request for block A3 would have to wait, but a request for A2 could be serviced concurrently
- Reduces reliability: if one disk fails, the data is lost.

RAID 0

- P(data loss)=P(disk1 fails or disk2 fails)=

=P(disk1 fails)+P(disk2 fails)-P(disk1 fails and disk2 fails)=

$=P_F+P_F-P_F*P_F=2P_F-P_F^2=$

=2*0.05-0.0025=0.0975

RAID 0

- Number of years=1/0.0975 ≈10
- MTTF =E(year)=
  ≈ 0.0975*10*(10+1)/2 ≈ 11/2 ≈5.5

RAID 1

- Creates an exact copy (or **mirror**) of a set of data on two or more disks.
- Typically, a RAID 1 array contains two disks
- Improved
  - Reading speed: two blocks can be read at the same time
  - Reliability: if one disk crashes, we can use the other
- Writing speed remains the same

RAID 1

RAID 1

A1	A1
A2	A2
A3	A3
A4	A4

RAID 1 Reliability

- Two disks with MTTF of 10 years
- What is the MTTF resulting in data loss?
- Data loss happens when one disk fails and the other fails as well while we are replacing the first.
- Supposing it takes 3 hours to replace the first disk. This is 1/2920 of a year
- P(fails rep)=1/2920=3.42E-04

RAID 1 Reliability

- The probability that the second disk fails while replacing the first is

  P(fails1 and fails2 rep)=

  =5E-2*5E-2*3.42E-04=8.55E-07
- P(data loss)=P(fails1 and fails2 rep or fails2 and fails1 rep)=

RAID 1 Reliability

= P(fails1 and fails2 rep) + P(fails2 and fails1 rep)-P(fails2 and fails1 rep and fails1 and fails2 rep)=

≈2*8.55E-07

=1.71E-06

RAID 1 Reliability

- Number of years=1/1.71E-06 ≈ 584795
- MTTF=E(years)=
  =1.71E-06*584795*584796/2=
  =584796/2=292398

RAID 4

- Uses block-level striping with a dedicated parity disk.

RAID 4

A1	A2	A3	Ap
B1	B2	B3	Bp
C1	C2	C3	Cp
D1	D2	D3	Dp

Consecutive blocks

A1-A3,B1-B3,

C1-C3, D1-D3

Parity block

- Bit i of the block in position j on the parity disk is the parity bit of the bits in position i in the blocks in position j in the other disks
- Eg., blocks of one byte, blocks A1-A3

  Disk1 11110000

  Disk2 10101010

  <u>Disk3 00111000</u>

  Disk4 01100010 (parity disk)

RAID 4

- Improves reading time: multiple blocks can be read at the same time
- Improves reliability: if one disk fails, we can reconstruct its content (assuming the others are correct)

RAID 4

- Problem:
  - When writing a block, we need to read and write the parity disk's block
  - This creates a bottleneck

RAID 5

- Uses block-level striping with parity data distributed across all member disks.

RAID 5

A1	A2	A3	Ap
B1	B2	Bp	B3
C1	Cp	C2	C3
Dp	D1	D2	D3

RAID 5

- Reading and reliability as RAID 4
- Writing improved because the parity blocks are not all on one disk

RAID 6

- Uses block-level striping with dual parity data distributed across all member disks.

RAID 6

A1	A2	A3	Ap	Aq
B1	B2	Bp	Bq	B3
C1	Cp	Cq	C2	C3
Dp	Dq	D1	D2	D3

RAID 6

- p and q blocks are computed with two different algorithms, e.g.
  - parity and Reed-Solomon
  - orthogonal dual parity
  - diagonal parity

RAID 6

- It is able to recover from the loss of two disks
- Writing improved because the parity blocks are not all on one disk

Nested RAID Levels

- RAID 0+1:

```
               RAID 1
                 |
        /-----------------\
        |                 |
      RAID 0            RAID 0
  /-----------\     /-----------\
  |     |     |     |     |     |
120GB 120GB 120GB 120GB 120GB 120GB
```

RAID 0+1

- If a disk fails, it can be rebuilt from the corresponding disk in the other RAID 0 batch
- If two disk fails from the same stripe, no recovery

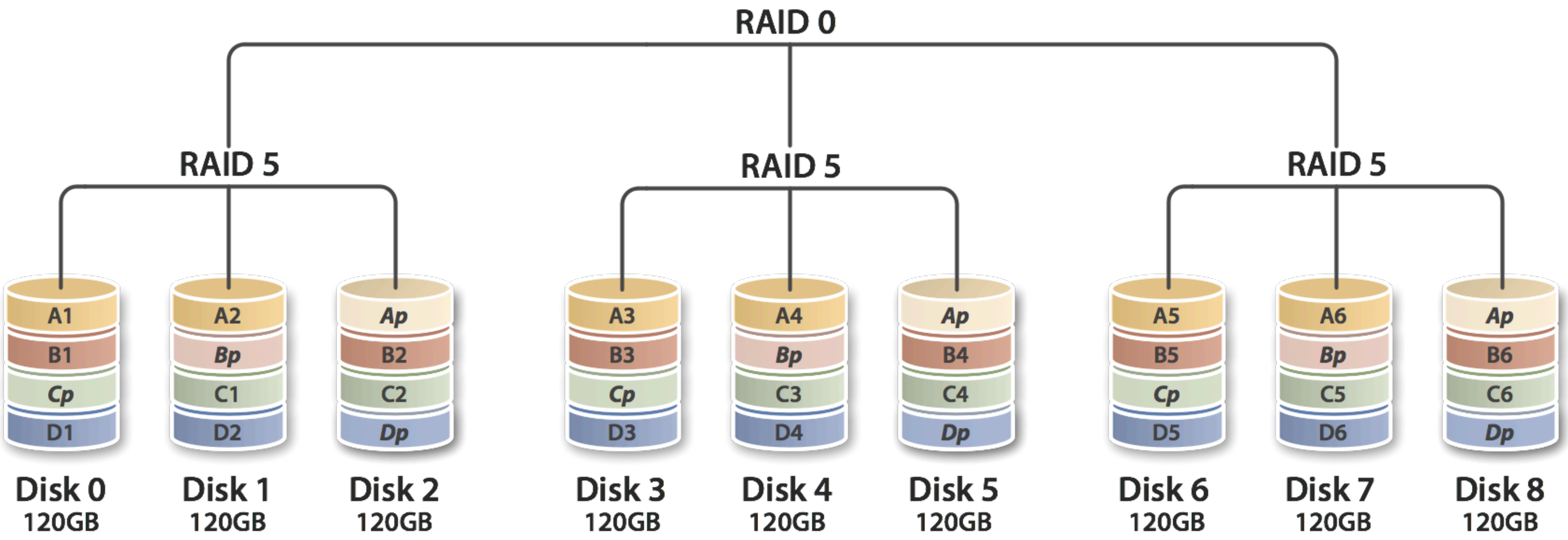
RAID 1+0 o RAID 10

```
                  RAID 0
                    |
     /-----------------------------\
     |              |              |
   RAID 1         RAID 1         RAID 1
  /------\       /------\       /------\
  |      |       |      |       |      |
120GB  120GB   120GB  120GB   120GB  120GB
```

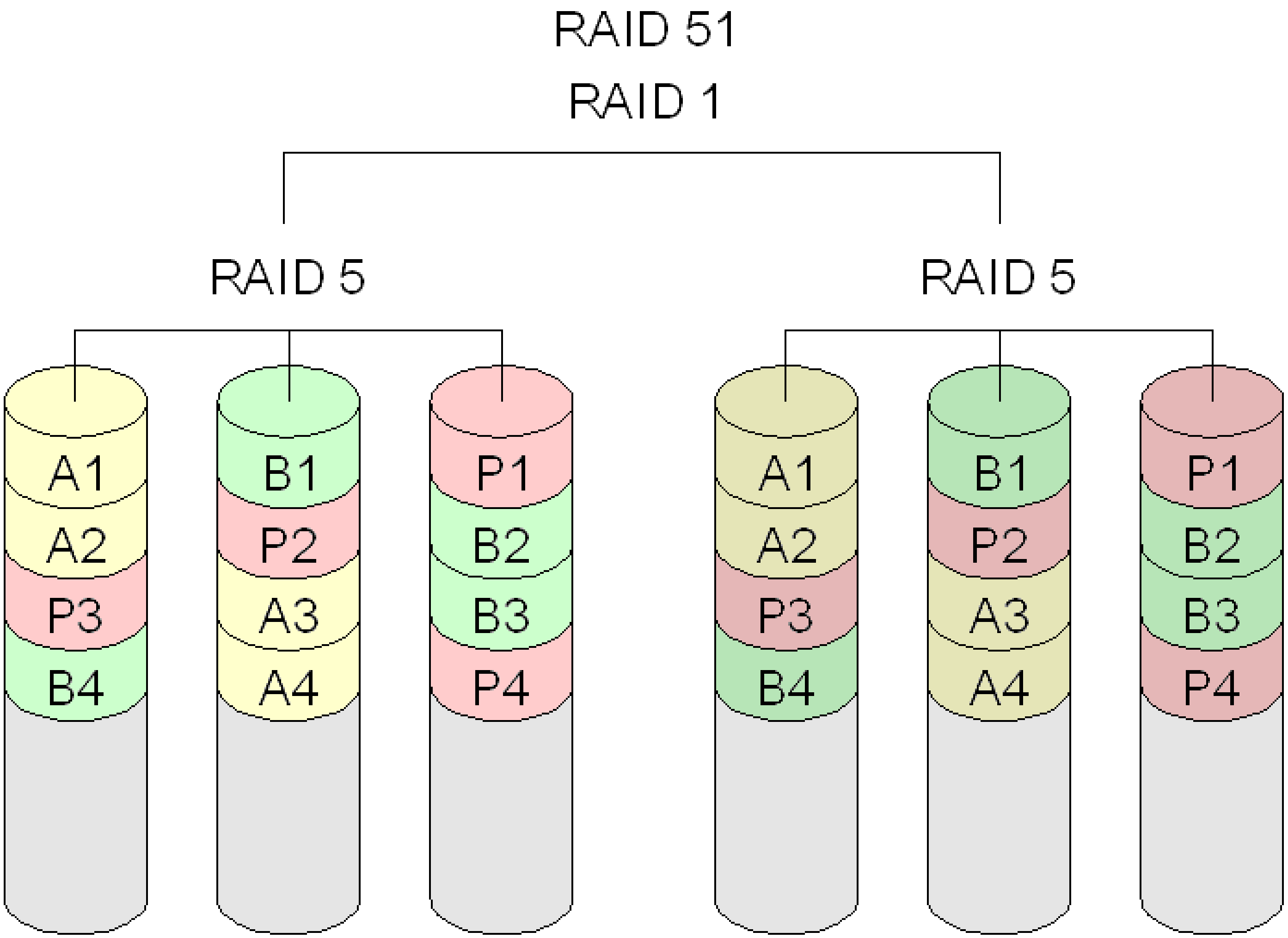
RAID 1+0 o RAID 10

- If a disk fails, it can be rebuilt from the corresponding disk in the other RAID 1 batches
- If two disk fails from the same RAID 1 batch, no recovery

RAID 5+0


RAID 0
RAID 5
RAID 5
RAID 5
A1
B1
Cp
D1
A2
Bp
C1
D2
Ap
B2
C2
Dp
A3
B3
Cp
D3
A4
Bp
C3
D4
Ap
B4
C4
Dp
A5
B5
Cp
D5
A6
Bp
C5
D6
Ap
B6
C6
Dp
Disk 0
120GB
Disk 1
120GB
Disk 2
120GB
Disk 3
120GB
Disk 4
120GB
Disk 5
120GB
Disk 6
120GB
Disk 7
120GB
Disk 8
120GB

RAID 5+1


RAID 51
RAID 1
RAID 5
RAID 5
A1
A2
P3
B4
B1
P2
A3
A4
P1
B2
B3
P4
A1
A2
P3
B4
B1
P2
A3
A4
P1
B2
B3
P4