Homework, etc.

- Reading: Chapter 12 (on website)
- Homework:
  - Exercises from Ch. 12
  - Due: 10/12 (later than usual due to break)
- Prelim I:
  - Friday, 10/12
  - Covers: Everything through Friday (10/5) lecture
  - Only up to today's (RAID) in-depth, though

Copyright © 2002–2018 UMaine School of Computing and Information Science – 1 / 35

COS 140: Foundations of Computer Science

RAID: Redundant Array of Independent Disks

Fall 2018

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

The problem

- How to store data:
  - Reliably
  - So that we can maximize a lot of *requests* by different processes
  - So that we can maximize the amount of data transferred/second to each process

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

The problem

- How to store data:
  - Reliably
  - So that we can maximize a lot of *requests* by different processes
  - So that we can maximize the amount of data transferred/second to each process
- These are conflicting, as we'll see!

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

The problem

- How to store data:
  - Reliably
  - So that we can maximize a lot of *requests* by different processes
  - So that we can maximize the amount of data transferred/second to each process
- These are conflicting, as we'll see!
- We'll concentrate on disk storage

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

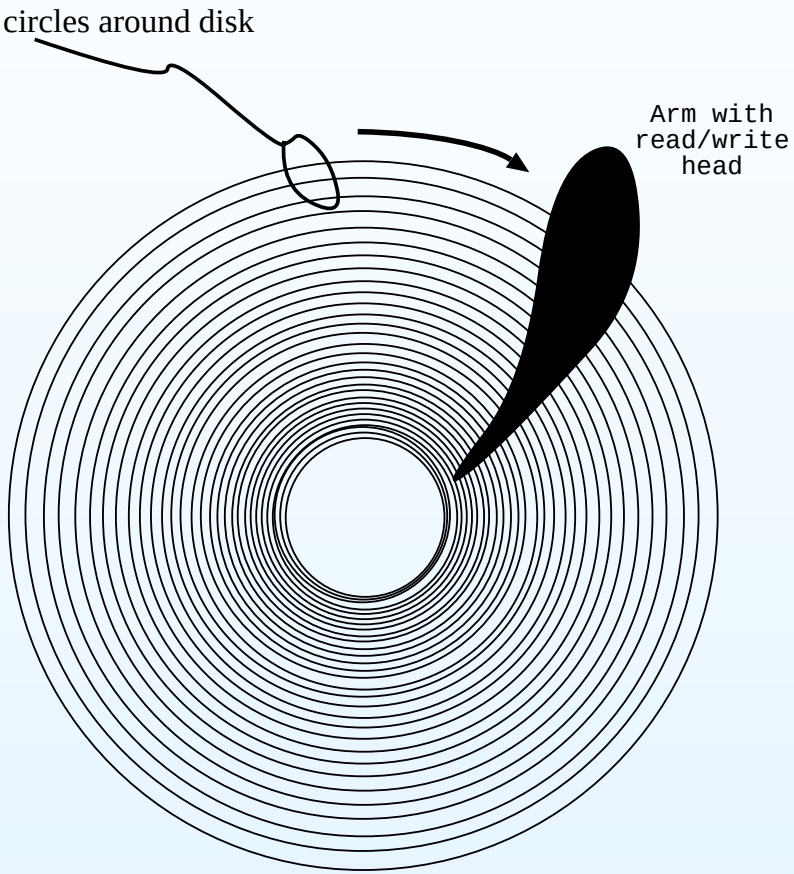
More About Magnetic Disks

- Type of external memory, like magnetic tape, flash, or optical disks (e.g., DVSs)
- Access method: *direct access*

Copyright © 2002–2018 UMaine School of Computing and Information Science – 4 / 35

Arrangement of Data on Mag Disks

(animation)

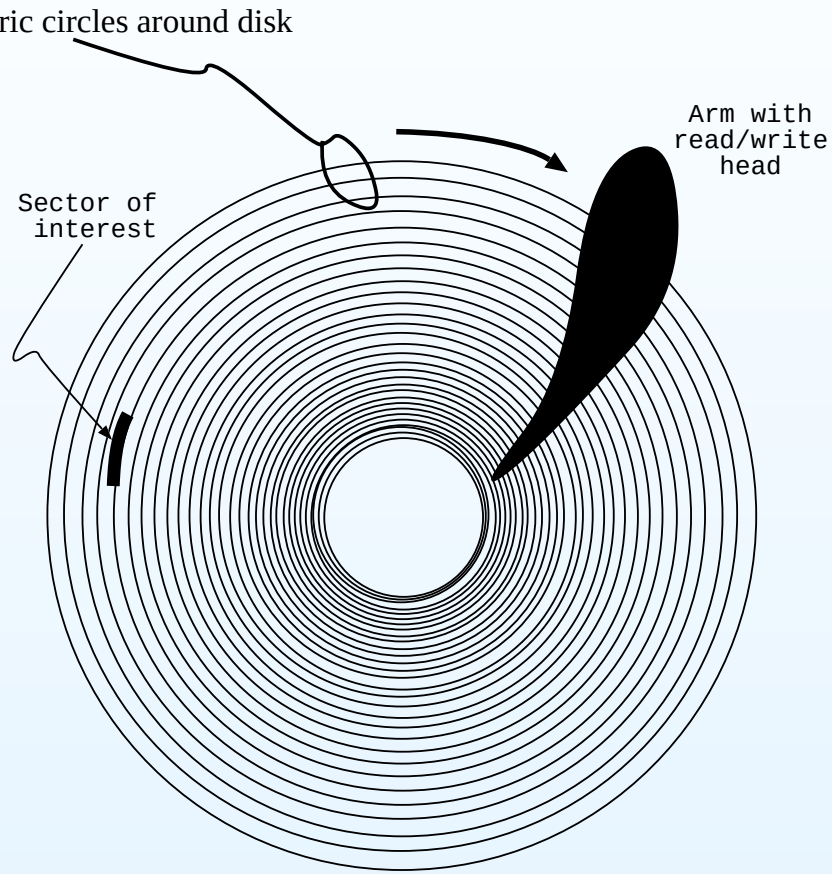


Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Arrangement of Data on Mag Disks

(animation)

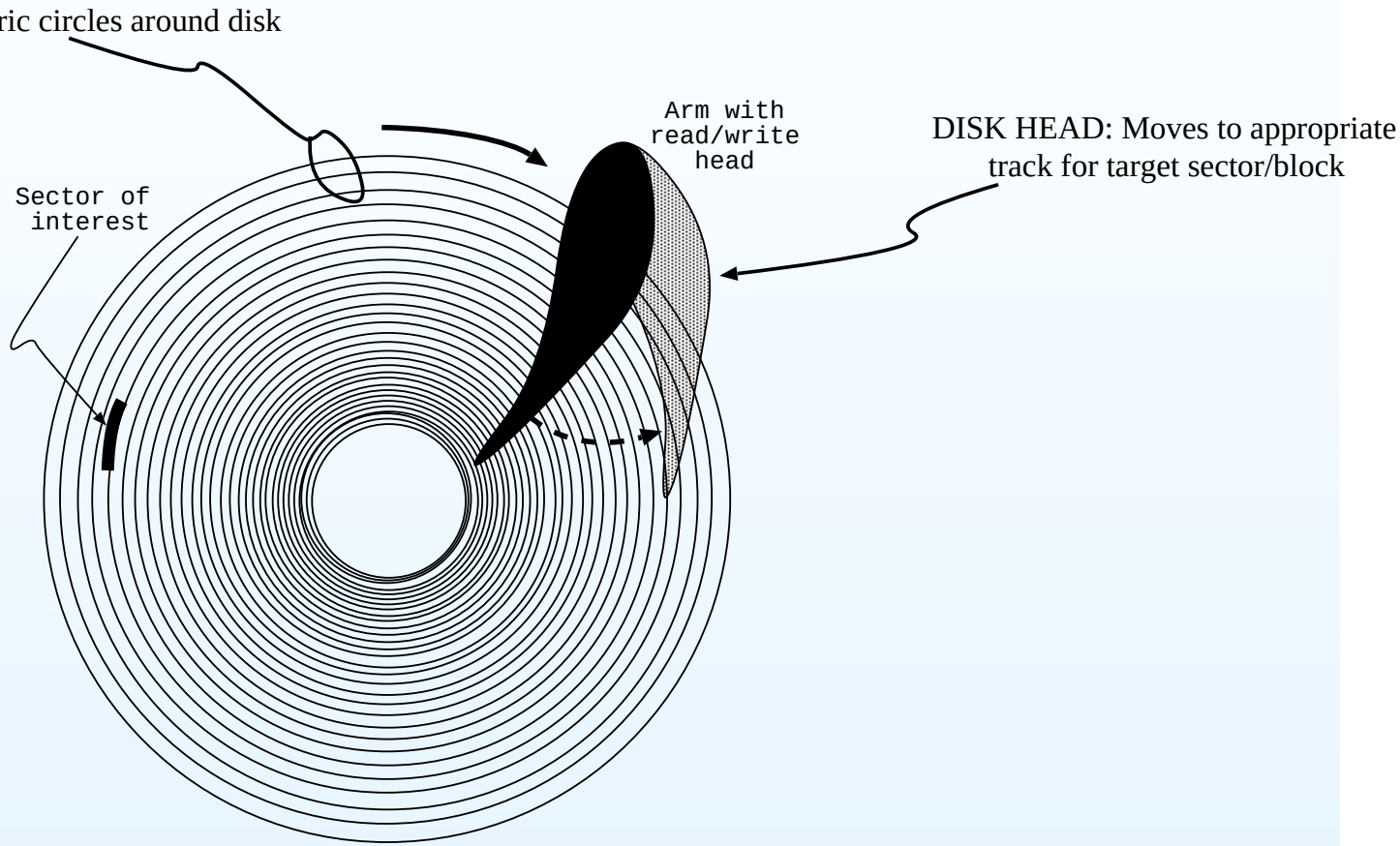


Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Arrangement of Data on Mag Disks

(animation)

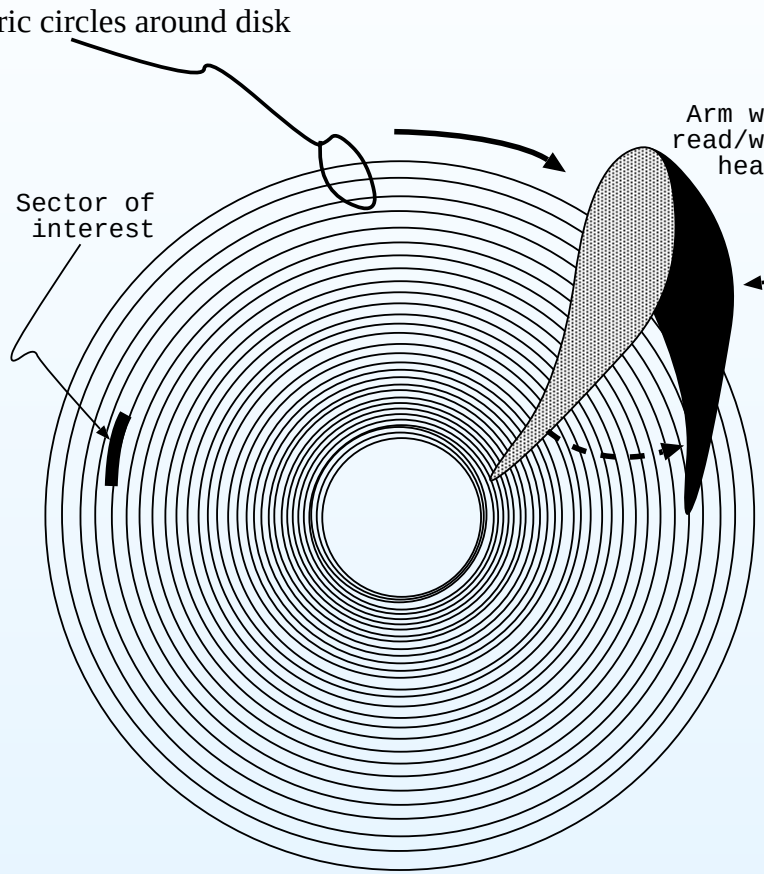


Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Arrangement of Data on Mag Disks

(animation)



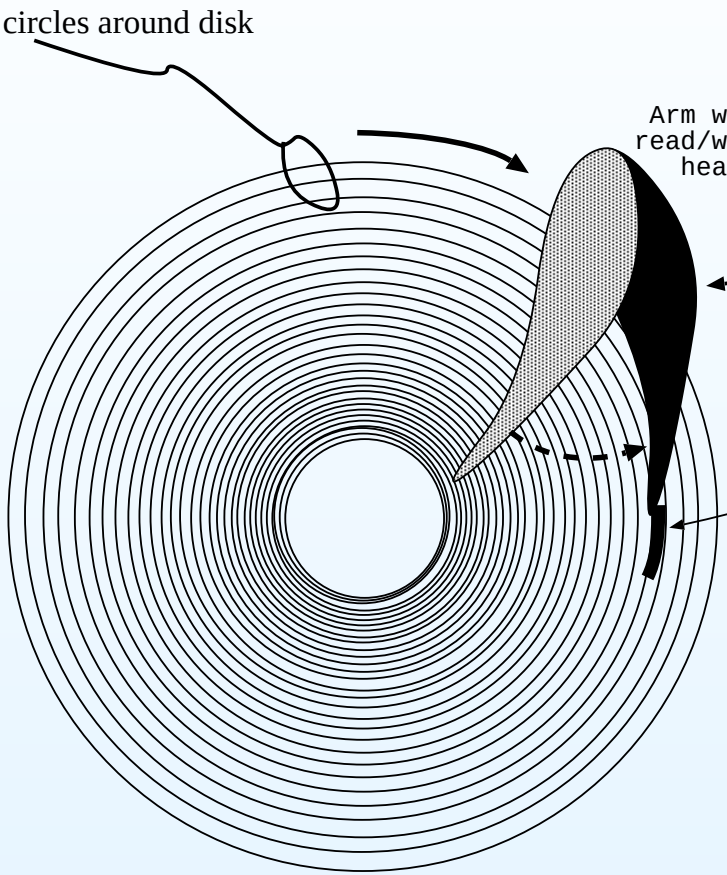
Once head is at right track,
have to wait until disk spins to bring sector under the head

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Arrangement of Data on Mag Disks

(animation)

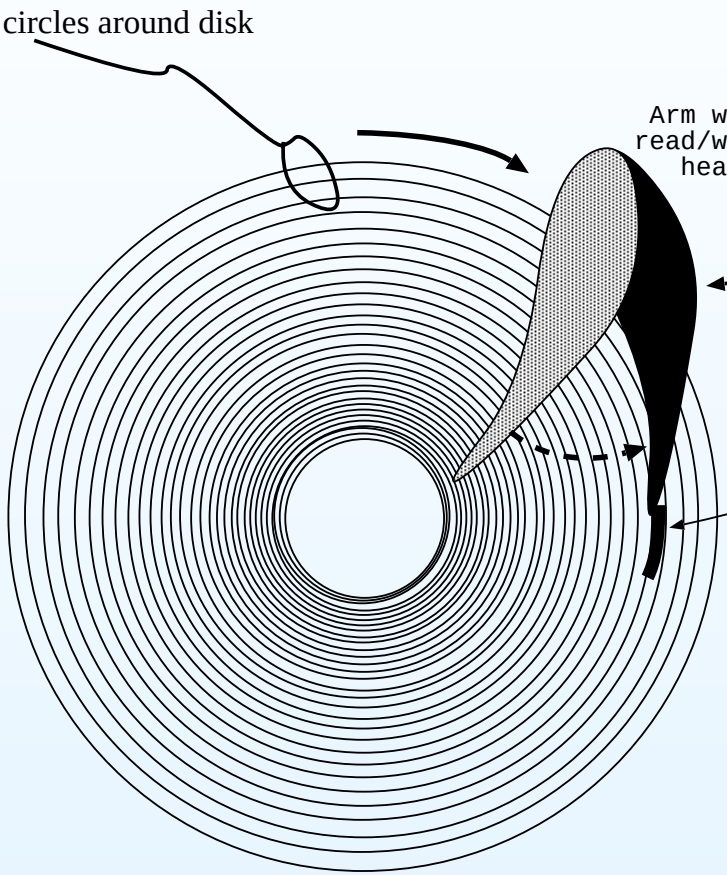


Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 5 / 35

Arrangement of Data on Mag Disks

(animation)



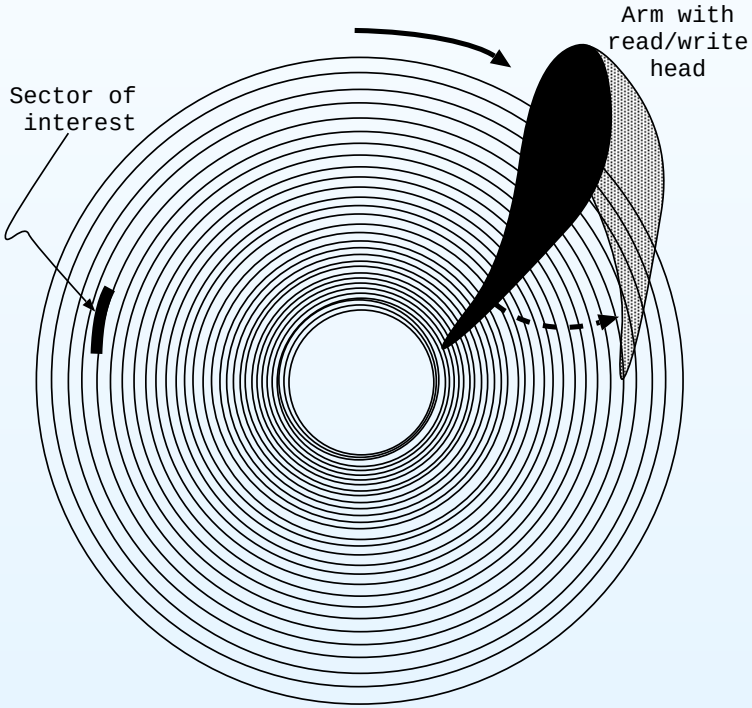
CONSTANT ANGULAR VELOCITY:
Disk spins at constant speed =>
data is more dense near center than at edges

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 5 / 35

Access Time for Disks

- *Seek time:* time till head on correct track
- *Rotational latency:* time till the correct sector under head
- Access time = Seek time + Rotational latency



Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency:

$$\frac{60s}{5400\text{rev}} \times 0.5\text{rev} \approx 6 \text{ ms average rotational latency}$$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency:

$$\frac{60s}{5400\text{rev}} \times 0.5\text{rev} \approx 6 \text{ ms average rotational latency}$$

  - Transfer time

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency:

$$\frac{60s}{5400\text{rev}} \times 0.5\text{rev} \approx 6 \text{ ms average rotational latency}$$

  - Transfer time:

$$4KB \times \frac{1GB}{2^{20}KB} \times \frac{8Gb}{1GB} \times \frac{1s}{3Gb} \approx 0.01\text{ms}$$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Example

- 2018 Western Digital 1TB laptop drive: 3 Gb/s max. transfer rate, 12 ms avg seek time, 5400 RPM, 512 B/sector
- Assume 4096 B wanted – also assume contiguous, sector-aligned:
  - Rotational latency:

$$\frac{60s}{5400\text{rev}} \times 0.5\text{rev} \approx 6 \text{ ms average rotational latency}$$

  - Transfer time:

$$4KB \times \frac{1GB}{2^{20}KB} \times \frac{8Gb}{1GB} \times \frac{1s}{3Gb} \approx 0.01\text{ms}$$

  - Total time $\approx 12 + 6 = 18$ ms

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 7 / 35

Types of Disks

- Type: depends on how close head gets to surface
- Closer the head $\Rightarrow$ narrower head can be $\Rightarrow$ narrower tracks $\Rightarrow$ more data
- Closer the head $\Rightarrow$ increased chance of errors due (e.g.) to impurities, dust, etc.
- Standard disks: head floats on a cushion of air – does not come in contact with the disk
- Floppy: head touches the disk when reading and writing
- Winchester: in a sealed unit so head can get closer to the disk because there are no contaminants

Copyright © 2002–2018 UMaine School of Computing and Information Science – 8 / 35

Example: Seagate 3.5 in. hard disk

(Eric Gaba Wikimedia Commons user: Sting)

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Performance Issues for External Memory

- Reliability
- Speed
  - *Transfer capacity* - how much data can be read from or written to the disk in a given amount of time
  - *I/O request rate* - how many reads or writes can be accomplished in a given amount of time
- Cost

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

How to Measure Speed: Transfer Capacity

- Amount of data that can be read from or written to the disk per second
- Important $\rightarrow$ large amount of data/request
- Depends on: buses, disk device, other factors

Copyright © 2002–2018 UMaine School of Computing and Information Science

How to Measure Speed: I/O Request Rate

- Number of requests/second that are serviced by disk (reads or writes)
- Important $\rightarrow$ many requests generated per second

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

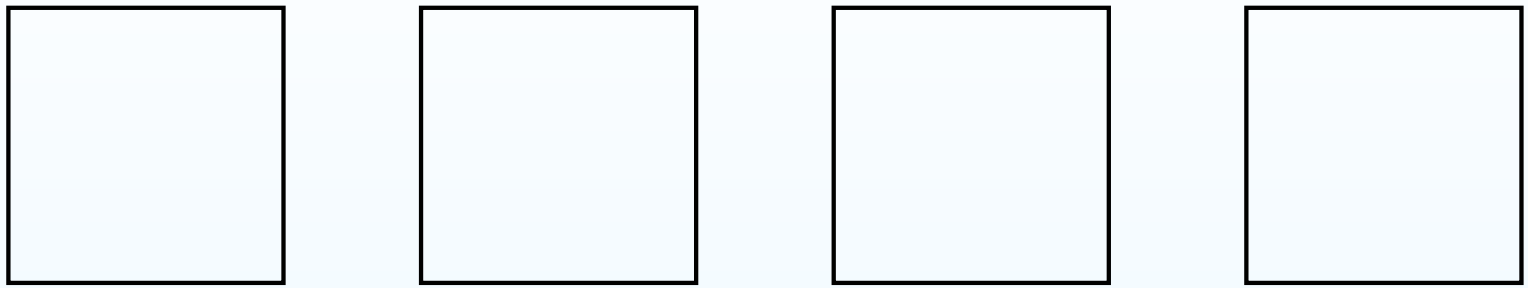
What RAID Hopes to Accomplish

- Improve performance through parallelism.
  - increase speed
  - increase reliability
- But: extra disks (for parallelism) $\Rightarrow$ higher cost.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Architecture of RAID



- Several disks in the *array*
- Different *RAID levels* specify how disks are used
  - Each level: addresses different issue(s)
  - Order of levels not significant

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

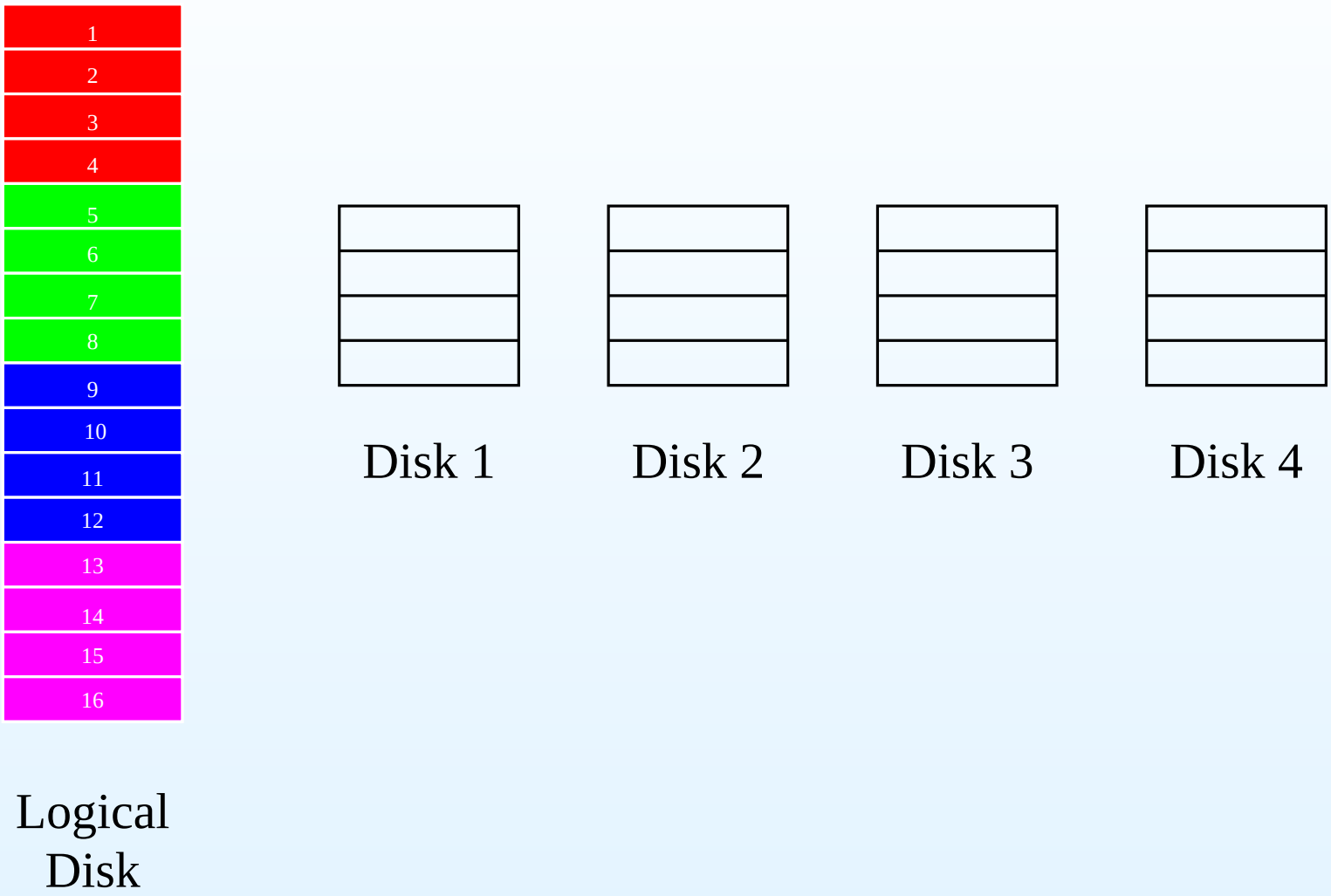
Distributing Data on the Disks

- *Logical disk:*
  - Abstraction of real disks
  - Think of single virtual disk on which data is stored
- Divide data unto equal-length chunks called *strips*
- Put strips on real disks in (e.g.) *round-robin* fashion
- *Stripe*: all the strips at correspondign locations on the disks

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

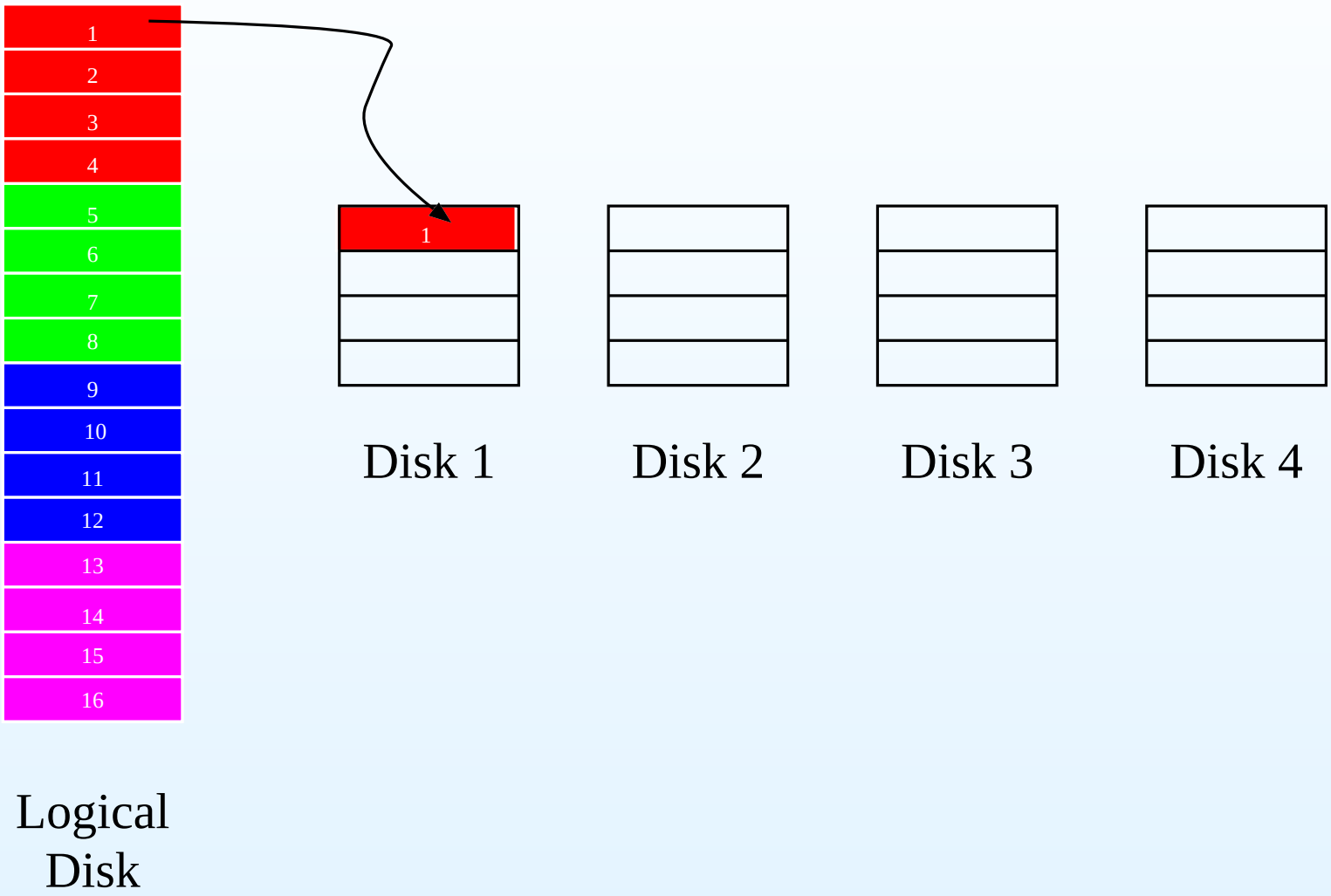
Distributing Data on the Disks



Computer Science Foundations

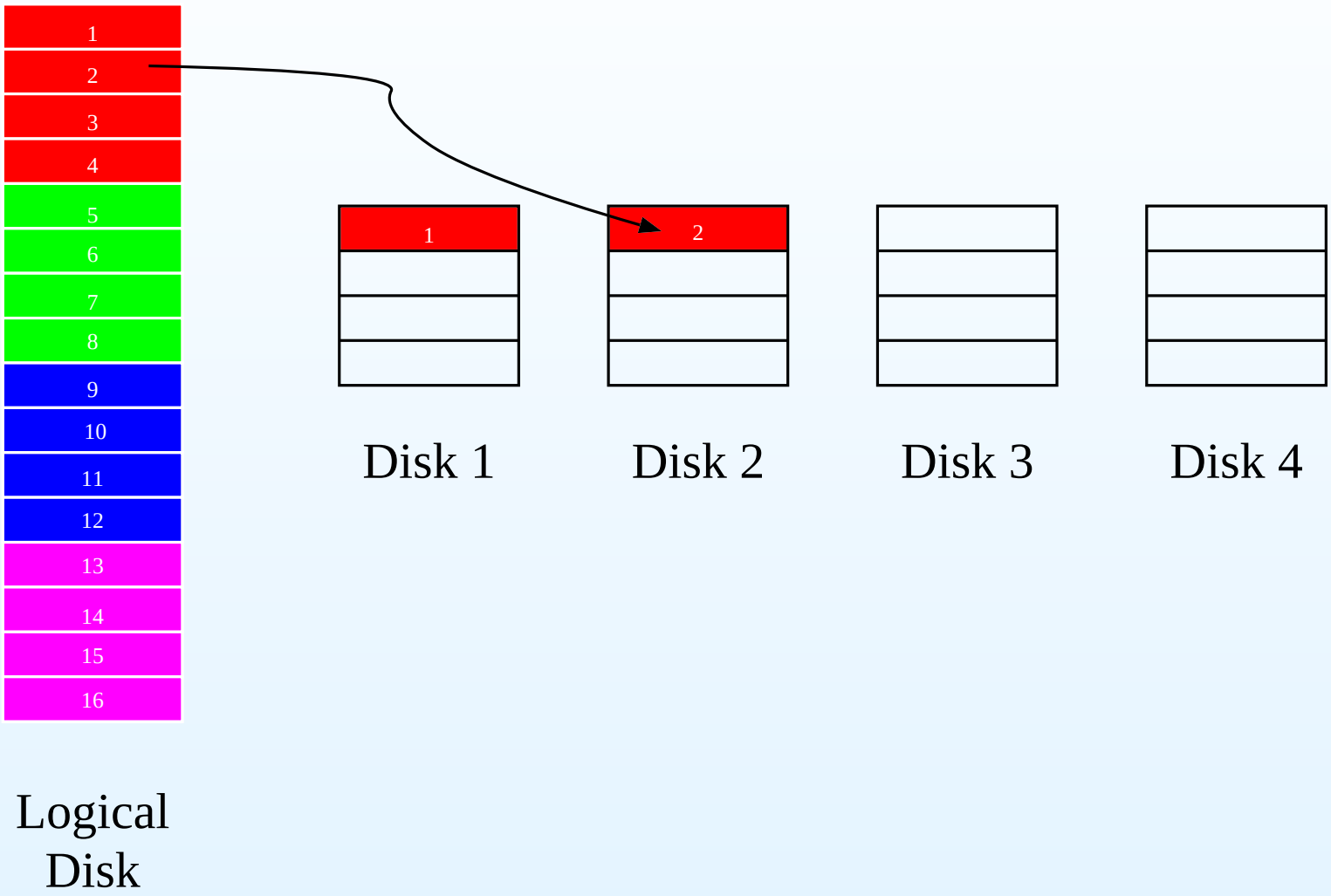
Copyright © 2002–2018 UMaine School of Computing and Information Science

Distributing Data on the Disks



Copyright © 2002–2018 UMaine School of Computing and Information Science – 16 / 35

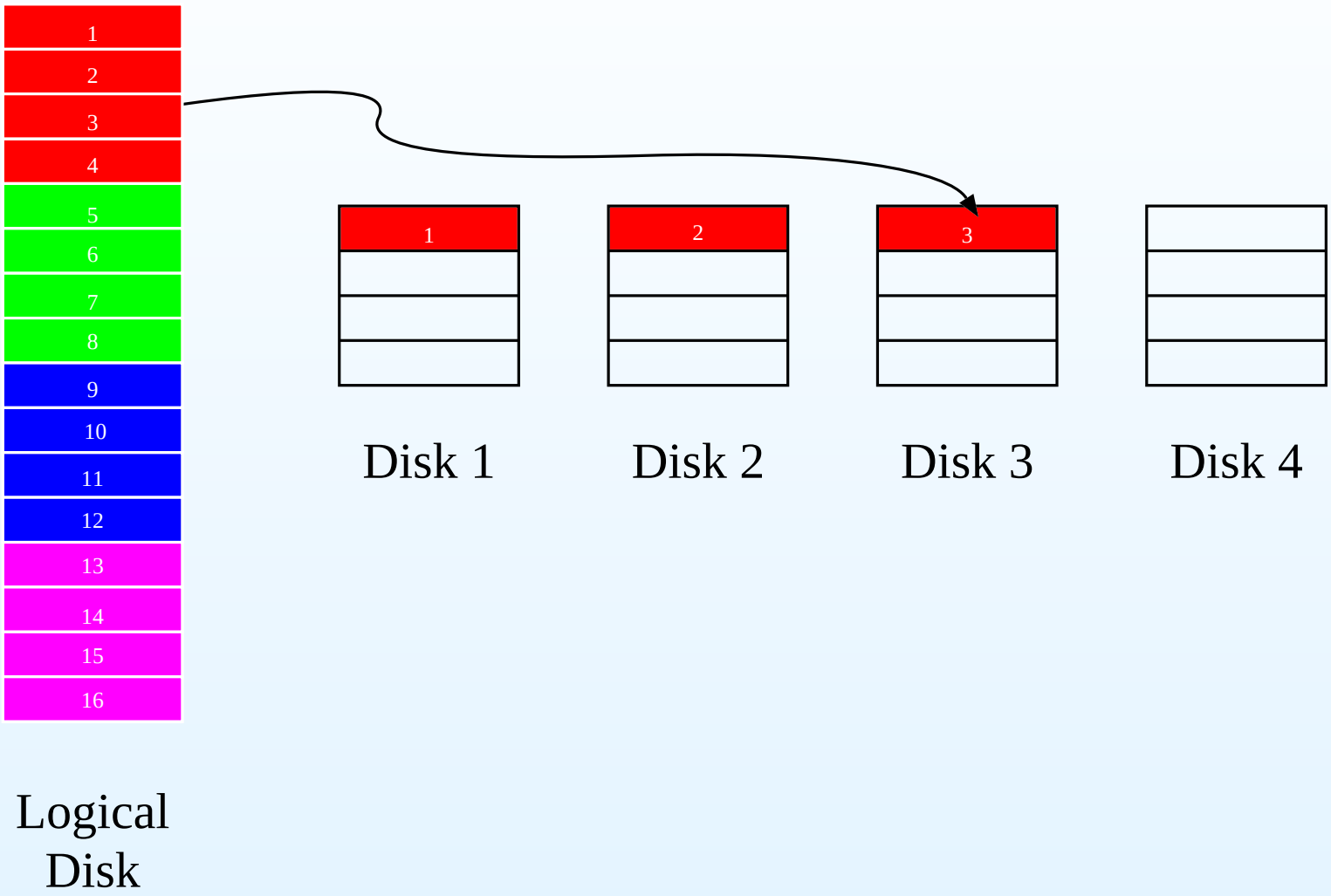
Distributing Data on the Disks



Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

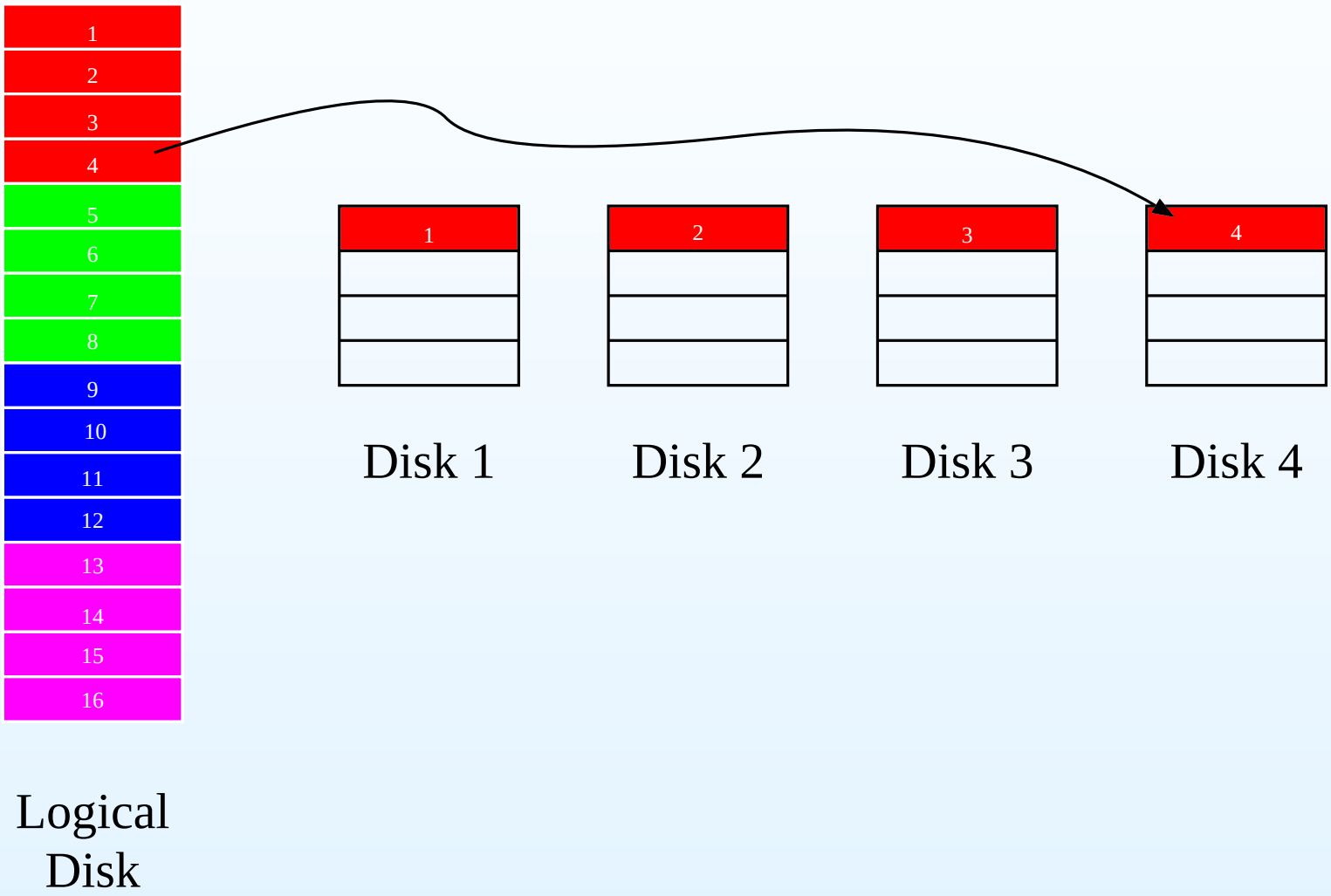
Distributing Data on the Disks



Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

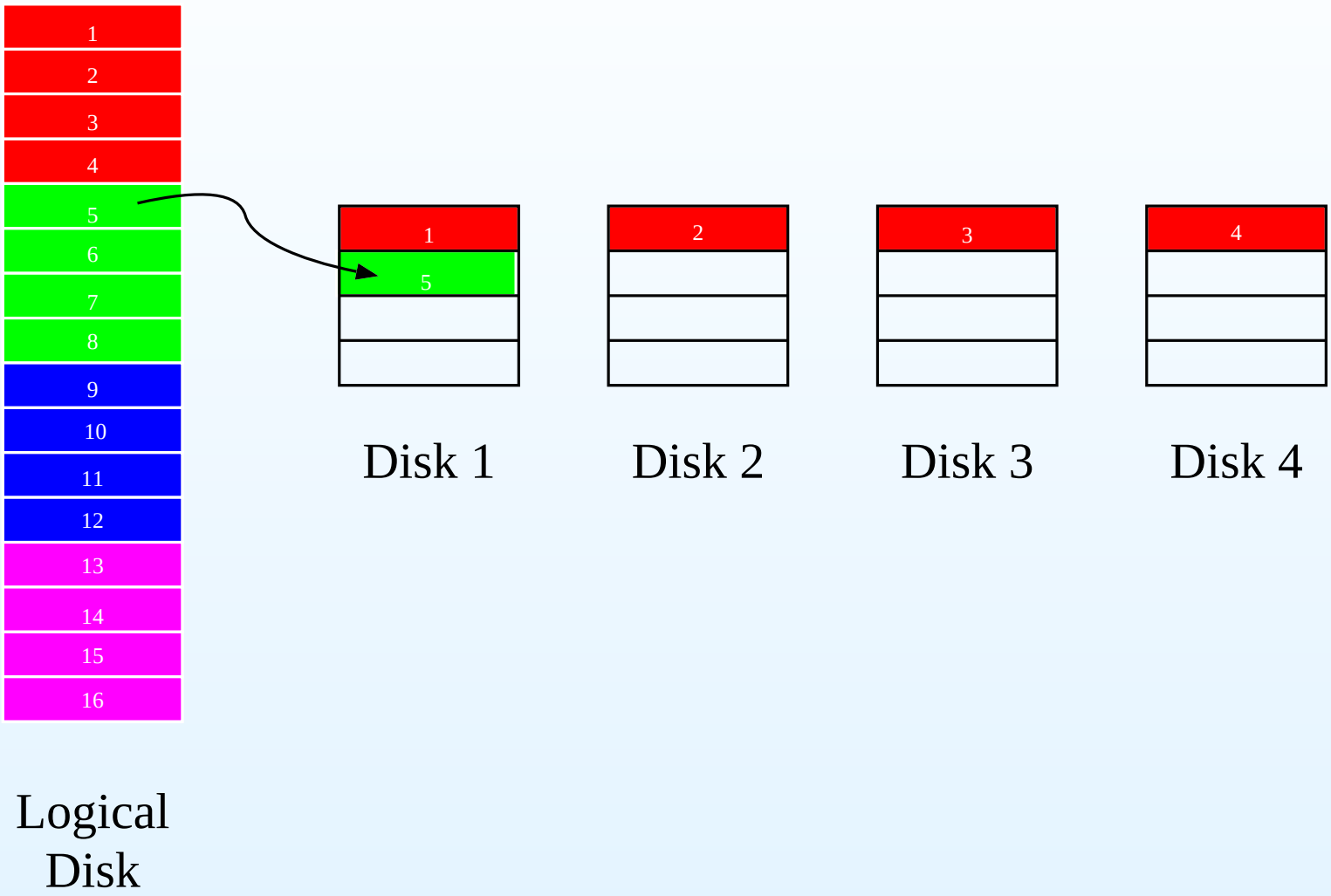
Distributing Data on the Disks



Computer Science Foundations

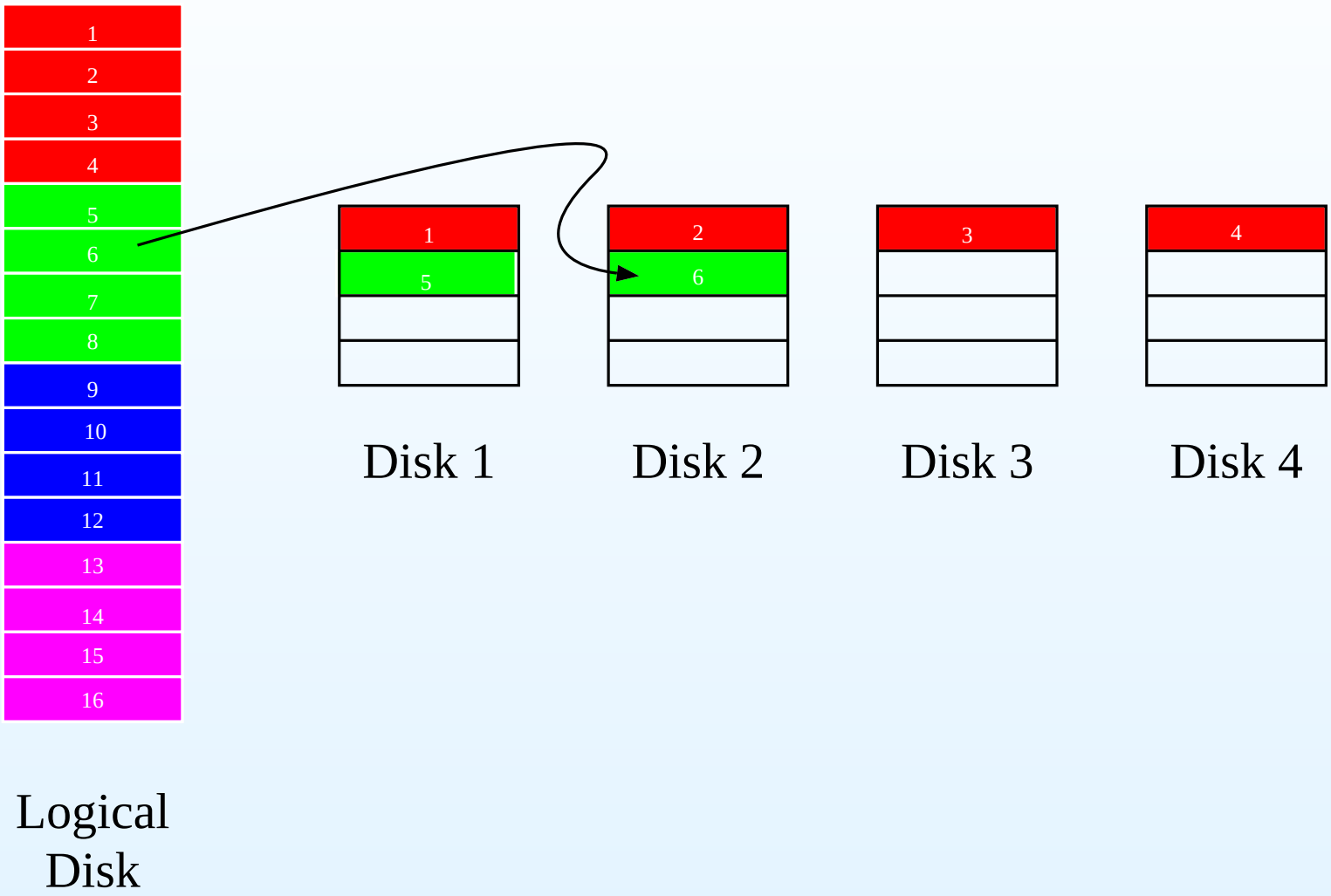
Copyright © 2002–2018 UMaine School of Computing and Information Science

Distributing Data on the Disks



Copyright © 2002–2018 UMaine School of Computing and Information Science – 16 / 35

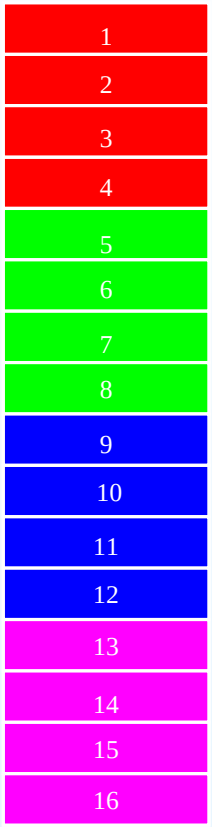
Distributing Data on the Disks



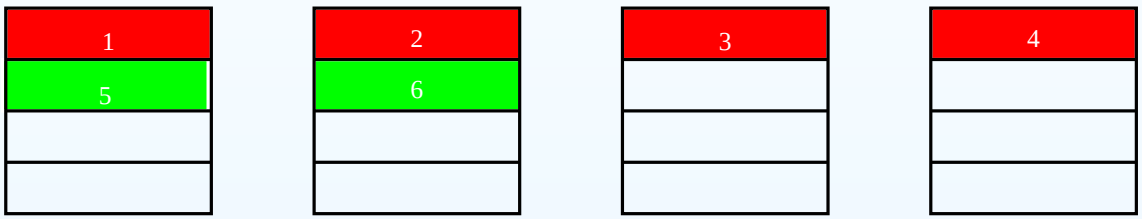
Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

Distributing Data on the Disks



Logical Disk

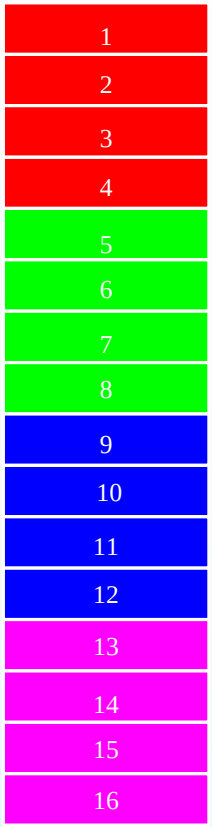


Disk 1 Disk 2 Disk 3 Disk 4

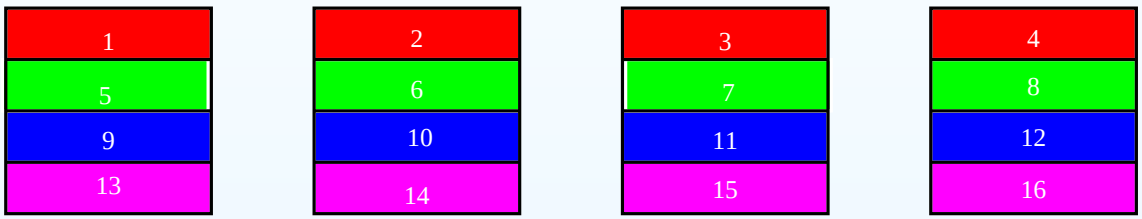
...and so on...

Copyright © 2002–2018 UMaine School of Computing and Information Science – 16 / 35

Distributing Data on the Disks



Logical Disk



Disk 1 Disk 2 Disk 3 Disk 4

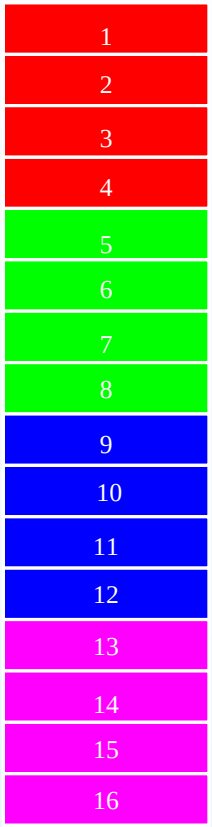
Each color shows a *stripe* of strips at the same location in each of the disks.

There are many more strips on the logical disk that will be distributed to disks in the same way. This small example is just for illustration.

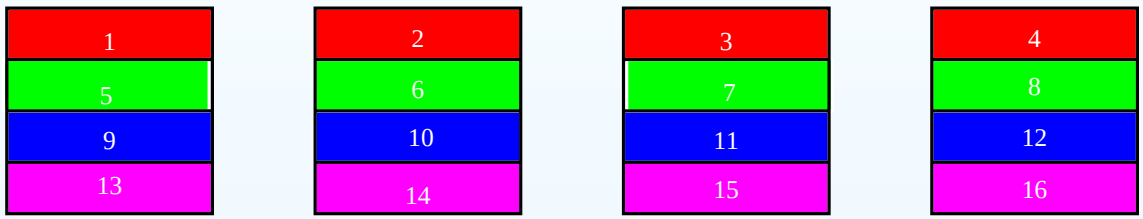
Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID Level 0



Logical Disk



Disk 1 Disk 2 Disk 3 Disk 4

- Simply stripe data onto multiple disks.
- Distribution of data puts system and user data on all strips.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 18 / 35

RAID 0 benefits

- No effect on reliability
- Let $r =$ average request size, $s =$ strip size, $n =$ # of disks, $S = ns =$ stripe size

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let $r =$ average request size, $s =$ strip size, $n =$ # of disks, $S = ns =$ stripe size
- Effect on transfer rate?

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times
- Effect on request rate?

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times
- Effect on request rate?
  - Suppose $r \leq s$
  - $\Rightarrow$ disk active per strip $\Rightarrow$ multiple requests handled at once
  - $\Rightarrow$ increased request rate
  - Ideally $r = s$: request rate increased up to n times

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times
- Effect on request rate?
  - Suppose $r \leq s$
  - $\Rightarrow$ disk active per strip $\Rightarrow$ multiple requests handled at once
  - $\Rightarrow$ increased request rate
  - Ideally $r = s$: request rate increased up to n times
- So which?

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 0 benefits

- No effect on reliability
- Let r = average request size, s = strip size, n = # of disks, $S = ns$ = stripe size
- Effect on transfer rate?
  - Suppose $s \leq r$
  - $\Rightarrow$ multiple disks $\Rightarrow$ strips for request
  - $\Rightarrow$ transfer rate increase
  - Ideally $r = S$: transfer rate increased up to n times
- Effect on request rate?
  - Suppose $r \leq s$
  - $\Rightarrow$ disk active per strip $\Rightarrow$ multiple requests handled at once
  - $\Rightarrow$ increased request rate
  - Ideally $r = s$: request rate increased up to n times
- So which? Depends on system characteristics, goals

Computer Science Foundations

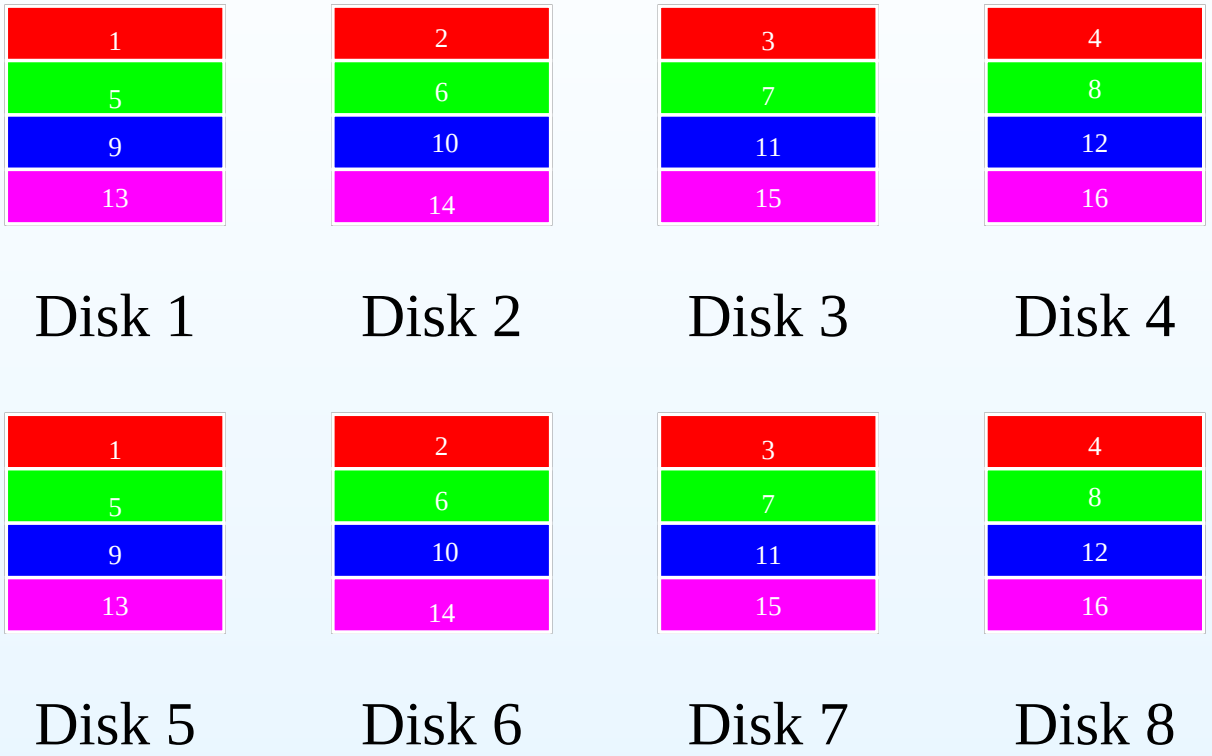
Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID level 0 cost

- If n disks were going to be used anyway $\Rightarrow$ no additional cost
- If not, then n small disks likely more expensive than 1 large disk

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID Level 1



- Two sets of disks, mirror images of each other.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 21 / 35

RAID level 1 benefits

- Reliability $\uparrow\uparrow$
  - Data completely redundantly stored
  - Disk fails: read from/write to copy
- Transfer rate, request rate
  - Same arguments as for RAID 0 re: strip size vs. request size
  - Additionally:
    - If disk busy, can read from duplicate $\Rightarrow \uparrow$ speeds
    - Could handle up to $2\times$ requests of RAID 0 if $s = r$
    - Could make $S = \frac{1}{2}r \Rightarrow$ all disks involved, up to $2\,times$ request rate of RAID 0
  - Writes must be done to both disks, but they can be done in parallel.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 1 cost

- If need n disks worth of data, need $2n$ disks
- I.e., doubles cost

Copyright © 2002–2018 UMaine School of Computing and Information Science

So what is RAID 1 good for?

- Critical data for which a failure cannot be tolerated and where the cost is not a problem.
- Additional ↑ transfer or request rates

Computer Science Foundations

RAID Level 2

- Idea: use some additional space to store an *error-correcting code*
- When an error occurs (on read or write), use that to fix it
- Uses a *Hamming code* (we'll study this later)
- For corresponding bit locations on each data disk, create Hamming code
  - Hamming code requires about $\log_2 n$ additional bits for n data bits
  - $\Rightarrow$ extra disks needed

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

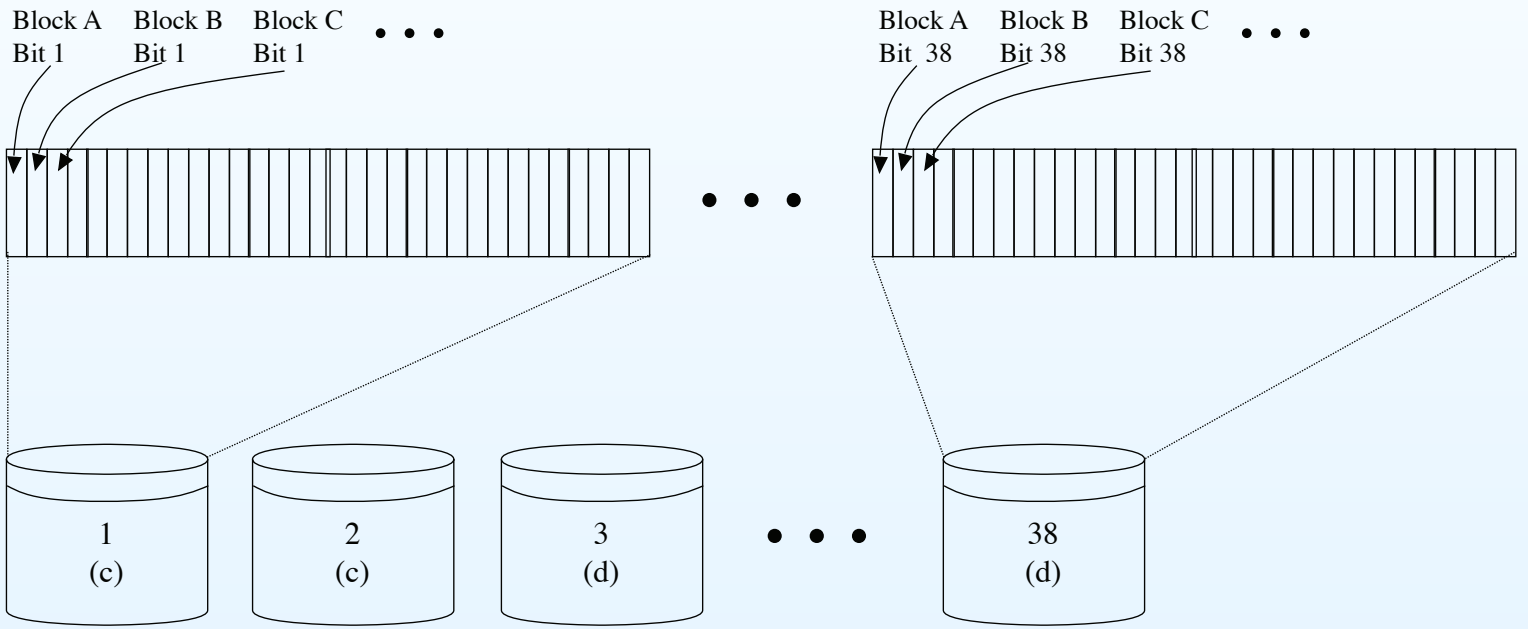
RAID 2 benefit

- Reliability
- Correction of 1-bit errors on read or write
- Reconstruct data if one disk fails.

Copyright © 2002–2018 UMaine School of Computing and Information Science – 26 / 35

RAID 2 costs

- Synchronized disks, write penalty (to compute ECC)
- May need a large number of disks – or a large number of reads/writes per disk



Disk #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38
Data bit #	c	c	1	c	2	3	4	c	5	6	7	8	9	10	11	c	12	13	14	15	16	17	18	19	20	21	22	23	24	c	25	26	27	28	29	30	31	32

RAID 2, 32-bit data blocks

Computer Science Foundations

What is RAID Level 2 Good For?

- Not commercially implemented
- Would be good if many single-disk errors...

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

What is RAID Level 2 Good For?

- Not commercially implemented
- Would be good if many single-disk errors...
- ...but unlikely and...

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

What is RAID Level 2 Good For?

- Not commercially implemented
- Would be good if many single-disk errors...
- ...but unlikely and...
- ...disks themselves use ECC!

Copyright © 2002–2018 UMaine School of Computing and Information Science – 28 / 35

What is RAID Level 2 Good For?

- Not commercially implemented
- Would be good if many single-disk errors...
- ...but unlikely and...
- ...disks themselves use ECC!
- Bit error rates ≈ 1 per 10^{14} bits read

Copyright © 2002–2018 UMaine School of Computing and Information Science – 28 / 35

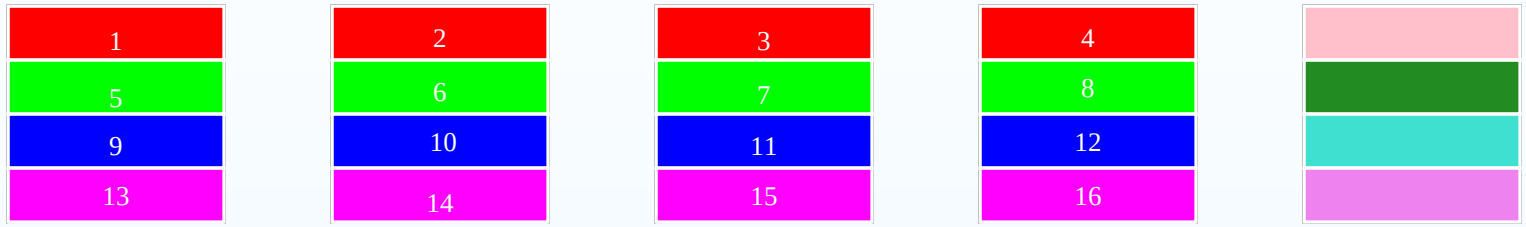
What is RAID Level 2 Good For?

- Not commercially implemented
- Would be good if many single-disk errors...
- ...but unlikely and...
- ...disks themselves use ECC!
- Bit error rates ≈ 1 per 10^{14} bits read
- If are really worried about data, use Level 1.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID Level 3



Disk 1 Disk 2 Disk 3 Disk 4 Disk 5

- Single extra disk stores a *parity bit*.
- Strip size: byte or word, access disks in parallel (synchronized)

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 29 / 35

What is a Parity Bit?

- Consider # 1s in data:
  - $0100\ 1011$
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even $\Rightarrow$ *even parity*
  - Set to make total # 1s odd $\Rightarrow$ *odd parity*
- E.g., even parity:
  - $0100\ 1011$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

What is a Parity Bit?

- Consider # 1s in data:
  - $0100\ 1011$
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even $\Rightarrow$ *even parity*
  - Set to make total # 1s odd $\Rightarrow$ *odd parity*
- E.g., even parity:
  - $0100\ 1011 \longrightarrow 0100\ 1011\ 0$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

What is a Parity Bit?

- Consider # 1s in data:
  - $0100\ 1011$
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even $\Rightarrow$ *even parity*
  - Set to make total # 1s odd $\Rightarrow$ *odd parity*
- E.g., even parity:
  - $0100\ 1011 \longrightarrow 0100\ 1011\ 0$
  - $0100\ 1111$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

What is a Parity Bit?

- Consider # 1s in data:
  - $0100\ 1011$
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even $\Rightarrow$ *even parity*
  - Set to make total # 1s odd $\Rightarrow$ *odd parity*
- E.g., even parity:
  - $0100\ 1011 \longrightarrow 0100\ 1011\ 0$
  - $0100\ 1111 \longrightarrow 0100\ 1111\ 1$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 30 / 35

What is a Parity Bit?

- Consider # 1s in data:
  - $0100\ 1011$
  - 4 1s
- Add an extra bit:
  - Set to make total # 1s even $\Rightarrow$ *even parity*
  - Set to make total # 1s odd $\Rightarrow$ *odd parity*
- E.g., even parity:
  - $0100\ 1011 \longrightarrow 0100\ 1011\ 0$
  - $0100\ 1111 \longrightarrow 0100\ 1111\ 1$
- Store both data and parity bit

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 30 / 35

How is the Parity Bit Used?

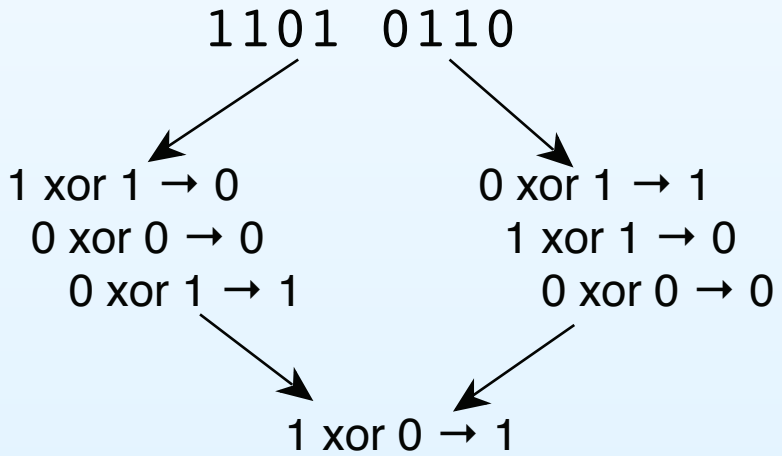
- When writing: compute, store parity bit
- When reading:
  - Know if parity *supposed* to be even or odd
  - Compute parity: if not correct $\Rightarrow$ error
- How to compute?
  - Remember homework asking about odd # of 1s in a number?
  - For even parity, can just XOR the bits

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 31 / 35

How is the Parity Bit Used?

- When writing: compute, store parity bit
- When reading:
  - Know if parity *supposed* to be even or odd
  - Compute parity: if not correct $\Rightarrow$ error
- How to compute?
  - Remember homework asking about odd # of 1s in a number?
  - For even parity, can just XOR the bits



Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 3 benefits

- Reliability ↑
  - Detect errors; can try re-reading
  - If disk drive fails ⟶ *reduced mode*
    - *Every* read will → parity error
    - But now know which bit is wrong!
- Transfer rate ↑ due to small strip size
- Request rate: no change

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 3 costs

- Need only 1 extra disk

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time

Copyright © 2002–2018 UMaine School of Computing and Information Science – 33 / 35

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?

Copyright © 2002–2018 UMaine School of Computing and Information Science – 33 / 35

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?
  - Modern disk drives are very good – maybe 1 error per 10^{14} bits read

Copyright © 2002–2018 UMaine School of Computing and Information Science – 33 / 35

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?
  - Modern disk drives are very good – maybe 1 error per 10^{14} bits read
  - P(error) in a 4 KB read: about 3×10^{-10}

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?
  - Modern disk drives are very good – maybe 1 error per 10^{14} bits read
  - P(error) in a 4 KB read: about 3×10^{-10}
  - P(2 errors) in a 4KB read $= (3 \times 10^{-10})^2 \approx 10^{-19}$

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 33 / 35

RAID 3 costs

- Need only 1 extra disk
- Have to access it every read, write...
- ...but small strip size, probably won't have multiple requests needing it at same time
- Can only catch single-bit errors – how bad is that?
  - Modern disk drives are very good – maybe 1 error per 10^{14} bits read
  - P(error) in a 4 KB read: about 3×10^{-10}
  - P(2 errors) in a 4KB read $= (3 \times 10^{-10})^2 \approx 10^{-19}$
  - I.e., 0.00000000000000000001, or expect 1 2-bit error in 10,000,000,000,000,000,000 4KB reads

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 33 / 35

What is RAID Level 3 Good For?

- When some error detection is needed
- High transfer rate and low number of outstanding requests.

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science

RAID Level 4, 5, 6, and beyond

- Same as RAID Level 3, but uses *independent access array*: disks in array operate independently
  - Uses larger strips ⇒ strip-level parity
  - Increases IO request rates at the expense of transfer capacity.
  - Write penalty: have to read old data strip, old parity, then write them.
  - Parity disk can become bottleneck.
- RAID Level 5 distributes the parity bits across the disks instead of having them all on one disk, to solve a potential problem with parity disk bottlenecks for Level 4.
- RAID Level 6: two parity blocks per stripe
- Others: some proprietary, some combinations of others (e.g., RAID 0+1)

Computer Science Foundations

Copyright © 2002–2018 UMaine School of Computing and Information Science – 35 / 35