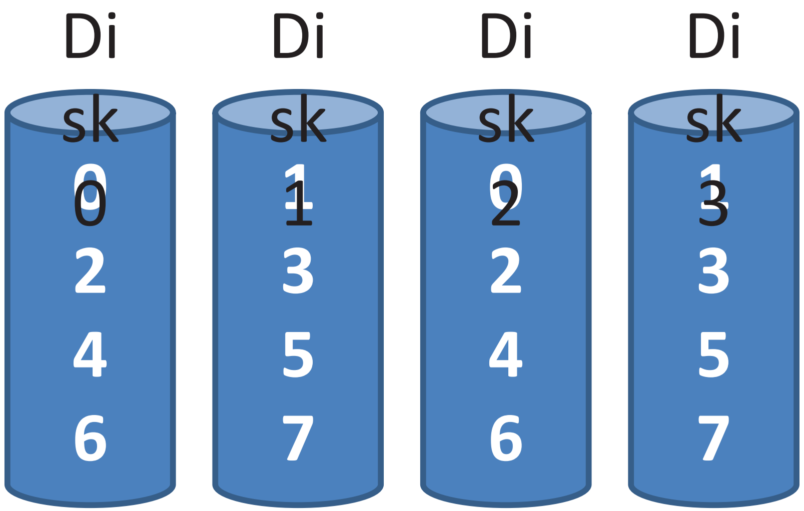
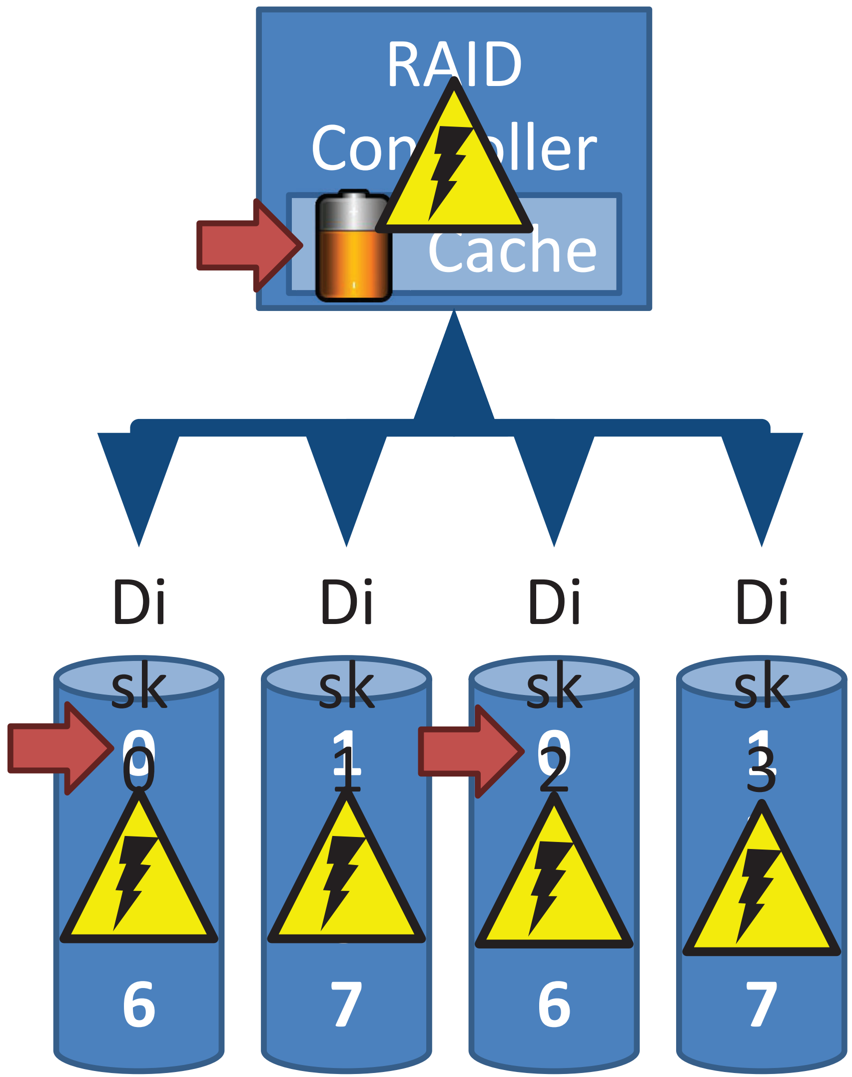
Analysis of RAID 1 (3)

- Random read: *N * R*
  - Best case scenario for RAID 1
  - Reads can parallelize across all disks
- Random write: *(N / 2) * R*
  - Two copies of all data, thus half throughput



The Consistent Update Problem

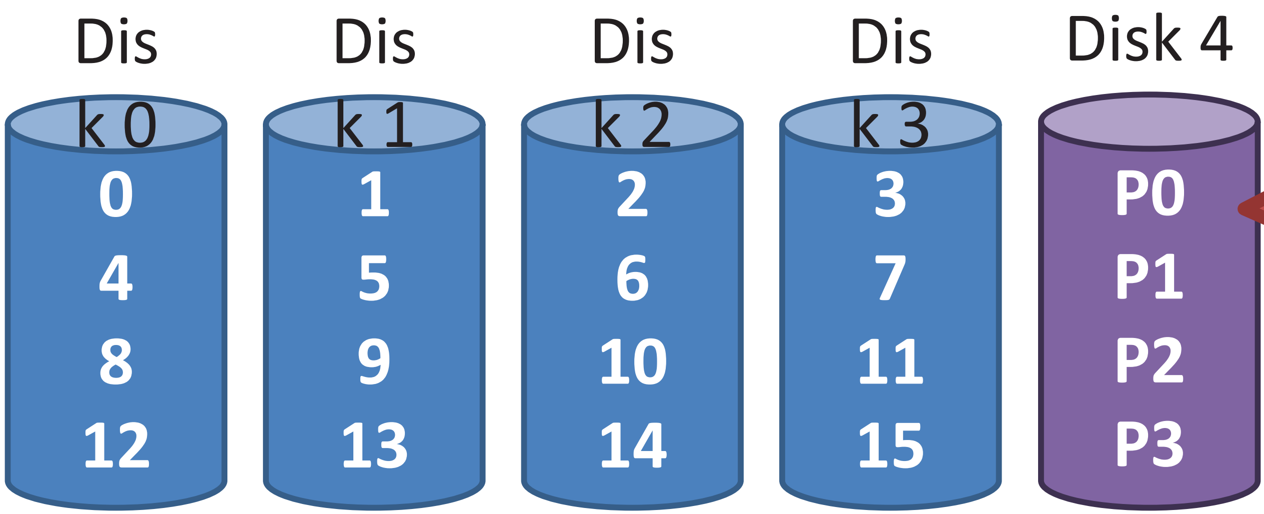
- Mirrored writes should be atomic
  - All copies are written, or none are written
- However, this is difficult to guarantee
  - Example: power failure
- Many RAID controllers include a write-ahead log
  - Battery backed, non-volatile storage of pending writes



Decreasing the Cost of Reliability

- RAID 1 offers highly reliable data storage
- But, it uses *N / 2* of the array capacity
- Can we achieve the same level of reliability without wasting so much capacity?
  - Yes!
  - Use information coding techniques to build light-weight error recovery mechanisms

RAID 4: Parity Drive



Disk *N* only stores parity information for the other *N-1* disks

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	0	1	1	0 ^ 0 ^ 1 ^ 1 = 0
0	1	0	0	0 ^ 1 ^ 0 ^ 0 = 1
1	1	1	1	1 ^ 1 ^ 1 ^ 1 = 0
0	1	1	1	0 ^ 1 ^ 1 ^ 1 = 1

Parity calculated using XOR

Updating Parity on Write

- How is parity updated when blocks are written?

1. Additive parity

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	0	0	1	0 ^ 0 ^ 0 ^ 1 = 1

Read other blocks

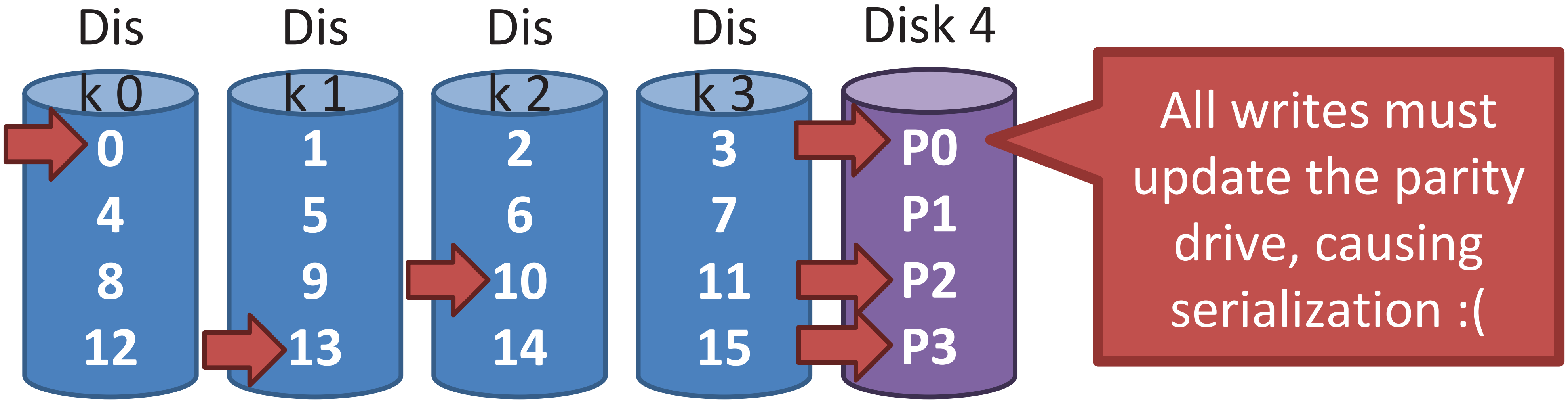
Update parity block

2. Subtractive parity

Disk 0	Disk 1	Disk 2		Disk 3	Disk 4	
0	0	1	0	1	0 ^ 0 ^ 1 ^ 1	1

$P_{new} = C_{old} \wedge C_{new} \wedge P_{old}$

Random Writes and RAID 4

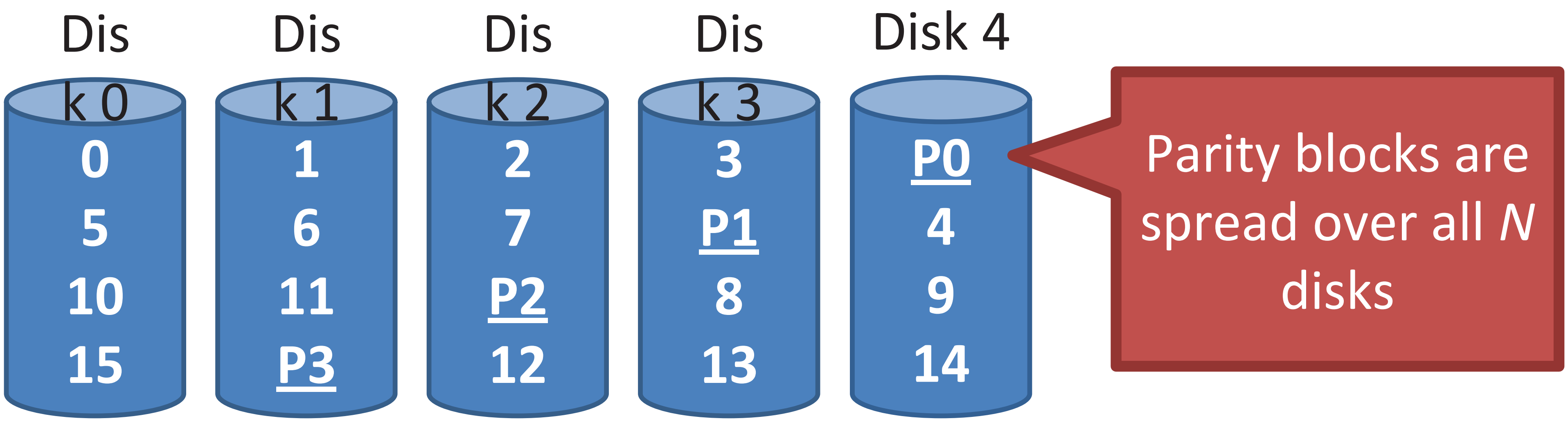


- Random writes in RAID 4
  1. Read the target block and the parity block
  2. Use subtraction to calculate the new parity block
  3. Write the target block and the parity block
- RAID 4 has terrible write performance
  - Bottlenecked by the parity drive

Analysis of RAID 4

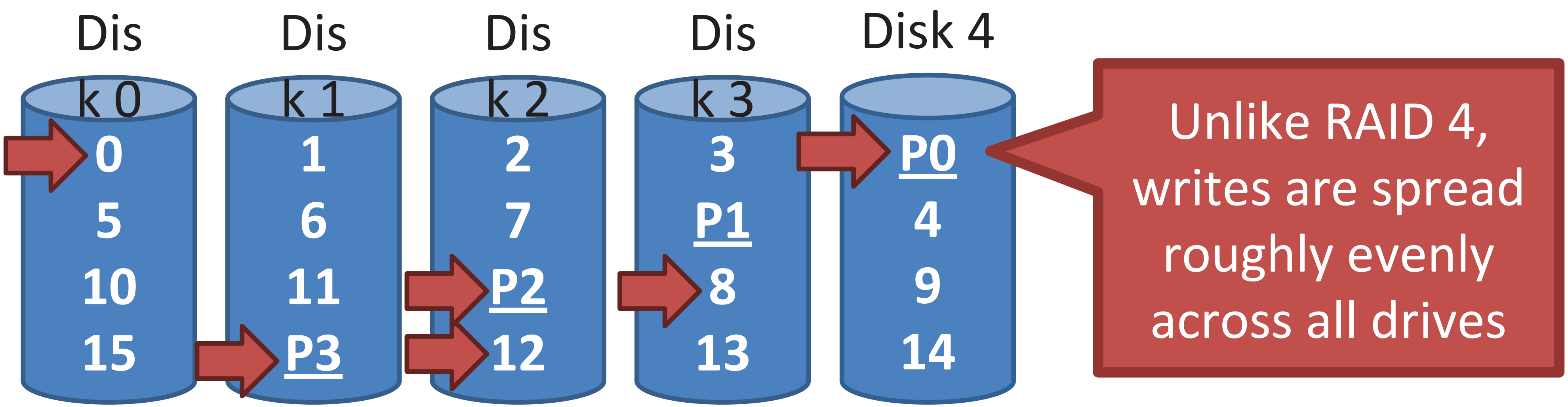
- Capacity: *N – 1*
  - Space on the parity drive is lost
- Reliability: 1 drive can fail
- Sequential Read and write: *(N – 1) * S*
  - Parallelization across all non-parity blocks
- Random Read: *(N – 1) * R*
  - Reads parallelize over all but the parity drive
- Random Write: *R / 2*
  - Writes serialize due to the parity drive
  - Each write requires 1 read and 1 write of the parity drive, thus *R / 2*

RAID 5: Rotating Parity



Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	0	1	1	0 ^ 0 ^ 1 ^ 1 = 0
1	0	0	0 ^ 1 ^ 0 ^ 0 = 1	0
1	1	1 ^ 1 ^ 1 ^ 1 = 0	1	1
1	0 ^ 1 ^ 1 ^ 1 = 1	0	1	1

Random Writes and RAID 5



- Random writes in RAID 5
  1. Read the target block and the parity block
  2. Use subtraction to calculate the new parity block
  3. Write the target block and the parity block
- Thus, 4 total operations (2 reads, 2 writes)
  – Distributed across all drives

Analysis of Raid 5

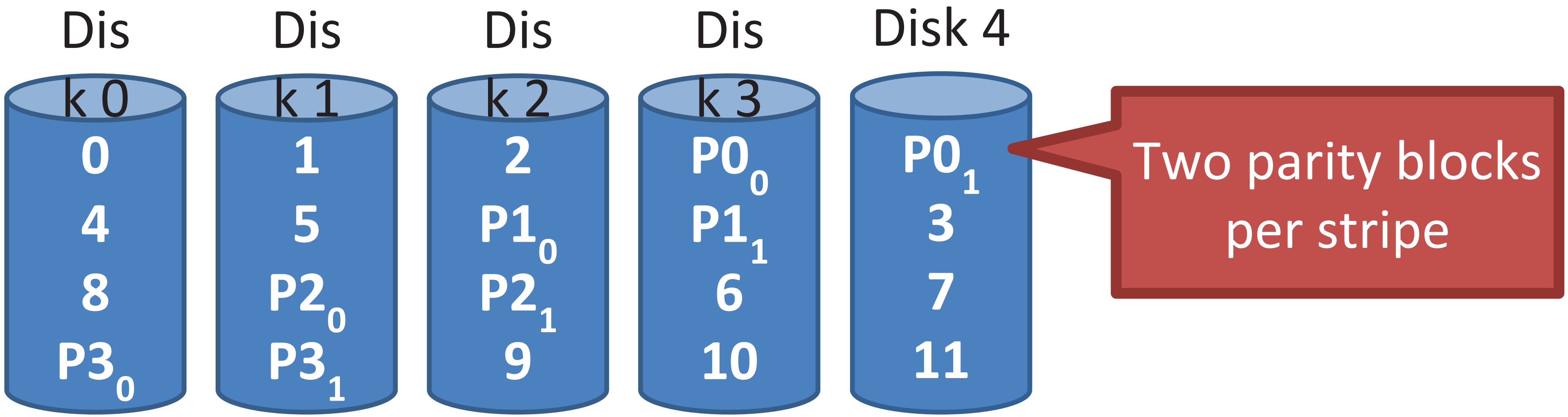
- Capacity: $N - 1$ [same as RAID 4]
- Reliability: 1 drive can fail [same as RAID 4]
- Sequential Read and write: $(N - 1) * S$ [same]
  - Parallelization across all non-parity blocks
- Random Read: $N * R$ [vs. $(N - 1) * R$]
  - Unlike RAID 4, reads parallelize over all drives
- Random Write: $N / 4 * R$ [vs. $R / 2$ for RAID 4]
  - Unlike RAID 4, writes parallelize over all drives
  - Each write requires 2 reads and 2 write, hence $N / 4$

Comparison of RAID Levels

- *N* – number of drives
- *S* – sequential access speed
- *R* – random access speed
- *D* – latency to access a single disk

		RAID 0	RAID 1	RAID 4	RAID 5
	Capacity	***N***	*N / 2*	*N – 1*	*N – 1*
	Reliability	*0*	***1* (maybe *N / 2*)**	*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*	*R / 2*	*(N / 4) * R*
Latency	Read	***D***	***D***	***D***	***D***
	Write	***D***	***D***	*2 * D*	*2 * D*

RAID 6



- Any two drives can fail
- *N – 2* usable capacity
- No overhead on read, significant overhead on write
- Typically implemented using Reed-Solomon codes

Choosing a RAID Level

- Best performance and most capacity?
  - RAID 0
- Greatest error recovery?
  - RAID 1 (1+0 or 0+1) or RAID 6
- Balance between space, performance, and recoverability?
  - RAID 5

Other Considerations

- Many RAID systems include a hot spare
  - An idle, unused disk installed in the system
  - If a drive fails, the array is immediately rebuilt using the hot spare
- RAID can be implemented in hardware or software
  - Hardware is faster and more reliable…
  - But, migrating a hardware RAID array to a different hardware controller almost never works
  - Software arrays are simpler to migrate and cheaper, but have worse performance and weaker reliability
    - Due to the consistent update problem

- Hard Drives
- RAID
- SSD