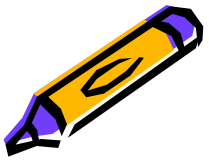
2025年度（令和7年）版

Ver. 2025-11-10a

Course number: CSC.T363

コンピュータアーキテクチャ
Computer Architecture

11. 仮想記憶 (2), 信頼性
Virtual Memory (2), dependability

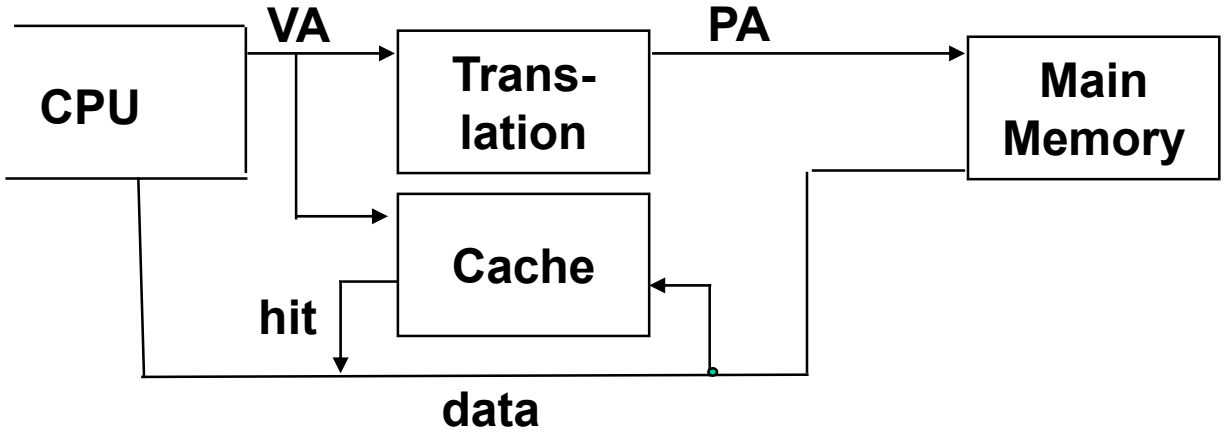


www.arch.cs.titech.ac.jp/lecture/CA/
Tue 13:30-15:10, 15:25-17:05
Fri 13:30-15:10

吉瀬 謙二　情報工学系
Kenji Kise, Department of Computer Science
kise _at_ c.titech.ac.jp

Why Not a **Virtually Addressed Cache?**

- A virtually addressed cache would only require address translation on cache misses



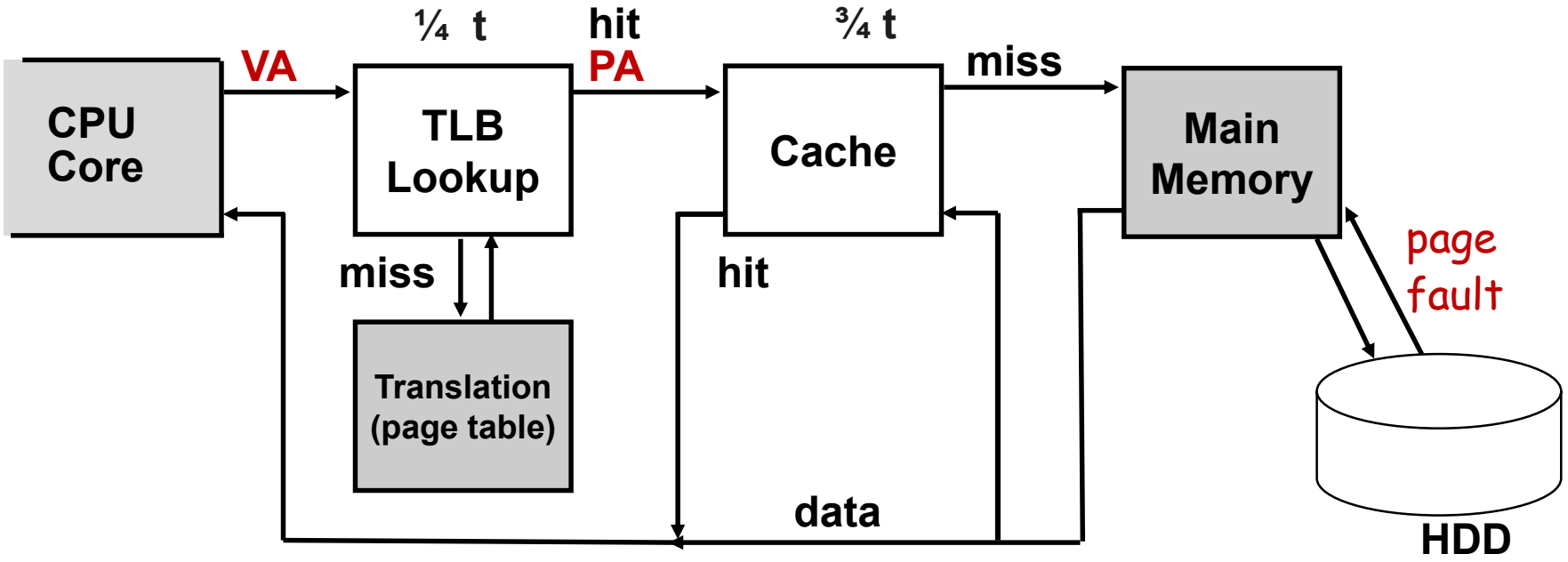
but

- Two different virtual addresses can map to the same physical address (when processes are sharing data),
- Two different cache entries hold data for the same physical address – **synonyms（別名）**
  - Must update all cache entries with the same physical address or the memory becomes inconsistent

The Hardware/Software Boundary

- What parts of the virtual to physical address translation is done by or assisted by the hardware?
  - Translation Lookaside Buffer (TLB) that caches the recent translations
    - TLB access time is part of the cache hit time
    - May cause an extra stage in the pipeline for TLB access
  - Page table storage, fault detection and updating
    - Page faults result in interrupts (precise) that are then handled by the OS
    - Hardware must support (i.e., update appropriately) Dirty and Reference bits (e.g., ~LRU) in the Page Tables

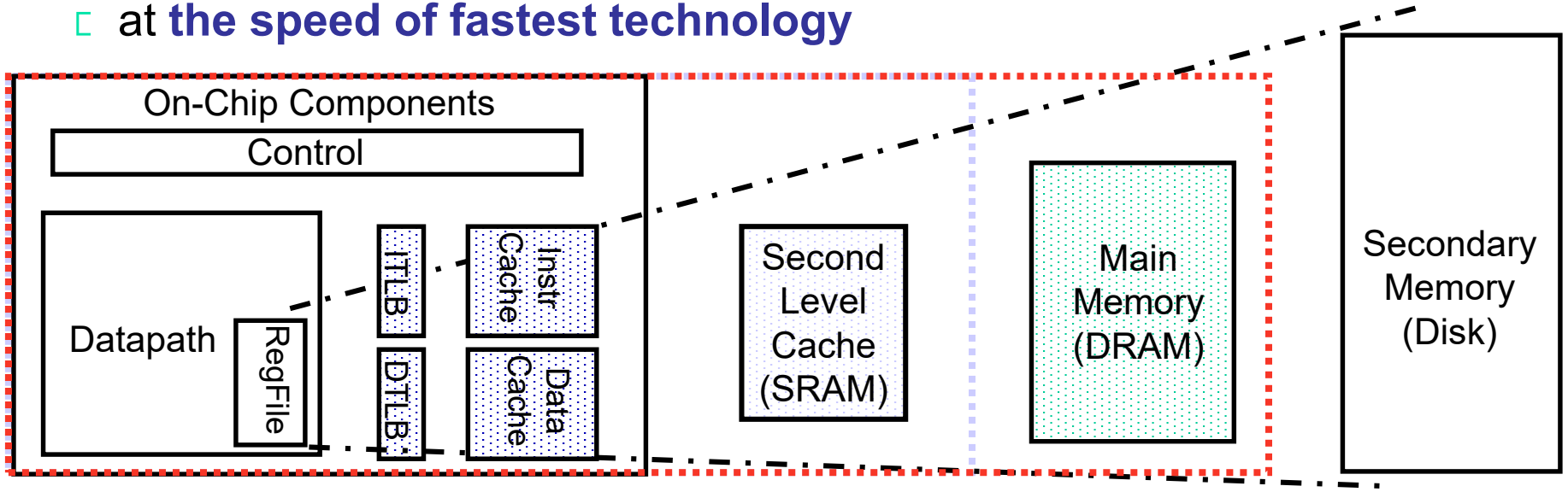
A TLB in the Memory Hierarchy



- **A TLB miss** – is it a TLB miss or a page fault ?
  - If the page is in main memory, then the TLB miss can be handled (in hardware or software) by loading the translation information from the page table into the TLB
    - Takes 100's of cycles to find and load the translation info into the TLB
  - If the page is not in main memory, then it's a true page fault
    - Takes 1,000,000's of cycles to service a page fault

A Typical Memory Hierarchy

- By taking advantage of **the principle of locality**
  - Present **much memory** in **the cheapest technology**
  - at **the speed of fastest technology**



Speed (%cycles):	½'s	1's	10's	100's	1,000's
Size (bytes):	100's	K's	10K's	M's	G's to T's
Cost:	highest				lowest

TLB: Translation Lookaside Buffer

Dependability（信頼性）

テレビ朝日、7月の障害の原因は「中性子線の衝突」 半導体の進化でソフトエラー発生率は上昇

2024年11月08日 21時59分 公開　[ITmedia]

印刷　見る　Share　B! 65　0

PR 国際的な規制強化で「待ったなし」　脱炭素とGX、未着手の企業はどうなるか

PR 生成AI活用は「試行モードから前進」　日立×パートナー3社の取り組み

テレビ朝日は11月8日、7月に放送機器の障害で地上波のCMやBSの番組、CMが放送できなくなった件について原因を特定したと発表した。中性子の衝突によりメモリーエラーが発生し、番組送出用のサーバが制御できなくなったという。

砂嵐が映るテレビのイラスト（いらすとや）※画像はイメージです

7月23日の午後10時過ぎ、局内のマスター設備内にあるネットワークスイッチの記憶装置で誤作動（メモリーエラー）が発生し、ネットワークに大量のデータが流れた。これにより番組やCMを送出する3系統のサーバが全て制御不能となったという。

この影響で、同日午後10時4分から午後11時59分の間、地上波ではCMが流れず、BSでは番組も放送できなかった。翌24日の早朝にも短時間ながら同じことが起きた。

エラーの原因については、設備メーカーから「中性子線の影響」と報告があったという。中性子は日常的に地上に降り注ぐ宇宙線が、大気中の原子に衝突することで発生し、これが半導体に衝突すると誤作動（ソフトエラー）を引き起こす場合がある。テレビ朝日の検証会議が大学や民間研究所などにヒアリングしたところ、見解はほぼ一致したという。

https://www.itmedia.co.jp/news/articles/2411/08/news203.html

Dependability, Fault, Error and Failure

- Fault（フォールト、故障）
  - 誤りの原因
- Error（エラー、誤り）
  - システム内の構成要素の正しくない出力
- Failure（障害）
  - システムが正常な動作をしない。コンポーネントやシステムが、期待した機能、サービス、結果から逸脱すること。

Error Detection and Correction of Main Memory

- Main memory stores a huge number of bits
  - Probability of bit flip becomes nontrivial
  - Bit flips (called soft errors) caused by
    - Slight manufacturing defects
    - Gamma rays and alpha particles
    - Electrical interference
    - Etc.
  - Getting worse with smaller feature sizes

- Reliable systems must be protected from soft errors via ECC (error correction codes)
  - Even PCs support ECC these days

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

Error Correcting Codes (ECC)

- Probabilities:

  P(1 word no errors) > P(single error) > P(two errors) >> P(>2 errors)

- Detection - signal a problem
- Correction - restore data to correct value
- Most common
  - Parity - single error detection
  - SECDED - Single Error Correction; Double Error Detection

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

ECC (Error Correcting Codes) for One Bit

Power	Correct	#bits	Comments
Nothing	0,1	1	
SED	00,11	2	01,10 detect errors
SEC	000,111	3	001,010,100 => 0 110,101,011 => 1
SECDED	0000,1111	4	One 1 => 0 Two 1's => error Three 1's => 1

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

ECC (Error Correcting Codes)

# 1's	0	1	2	3	4
Result	0	0	Err	1	1

- Hamming distance
  - No. of bit flips to convert one valid code to another
  - All legal SECDED codes are at Hamming distance of 4
    - I.e. in single-bit SECDED, all 4 bits flip to go from representation for '0' (0000) to representation for '1' (1111)

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

ECC (Error Correcting Codes)

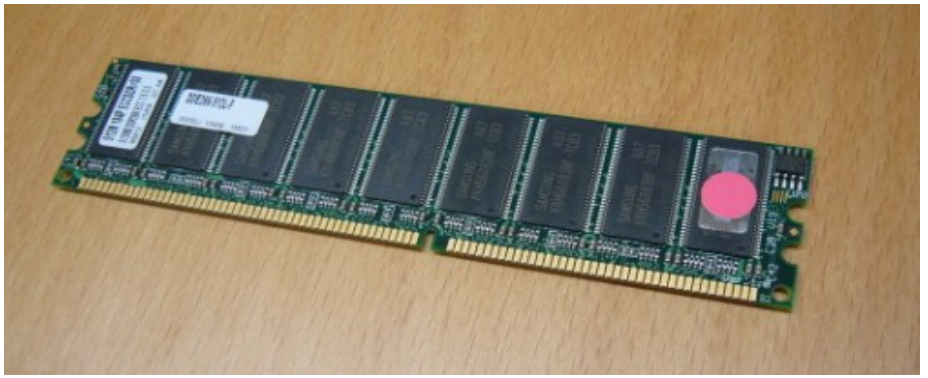
- Reduce overhead by applying codes to a word, not a bit
  - Larger word means higher p(>=2 errors)

# bits	SED overhead	SECDED overhead
1	1 (100%)	3 (300%)
32	1 (3%)	7 (22%)
64	1 (1.6%)	8 (13%)
n	1 (1/n)	$1 + \log_2 n$ + a little

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

64-bit ECC (Error Correcting Codes)

- 64 bits data with 8 check bits
  dddd…..d ccccccccc
- DIMM with 9x8-bit-wide DRAM chips = 72 bits



- Intuition
  - One check bit is parity
  - Other check bits point to
    - Error in data, or
    - Error in check bits, or
    - No error

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

ECC (Error Correcting Codes)

- To store (write)
  - Use $data_0$ to compute $check_0$
  - Store $data_0$ and $check_0$
- To load
  - Read $data_1$ and $check_1$
  - Use $data_1$ to compute $check_2$
  - Syndrome = $check_1$ xor $check_2$
    - I.e. make sure check bits are equal

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

ECC Syndrome

Syndrome	Parity	Implications
0	OK	$data_1$==$data_0$
n != 0	Not OK	Flip bit n of $data_1$ to get $data_0$
n != 0	OK	Signals uncorrectable error

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

4-bit SECDED Code

Bit Position	001	010	011	100	101	110	111	
Codeword	C_1	C_2	b_1	C_3	b_2	b_3	b_4	P
C_1	X		X		X		X	
C_2		X	X			X	X	
C_3				X	X	X	X	
P	X	X	X	X	X	X	X	X

$$C_1 = b_1 \oplus b_2 \oplus b_4$$
$$C_2 = b_1 \oplus b_3 \oplus b_4$$
$$C_3 = b_2 \oplus b_3 \oplus b_4$$
$$P = even\_parity$$

- C_n parity bits chosen specifically to:
  - Identify errors in bits where bit n of the index is 1
  - C_1 checks all odd bit positions (where LSB=1)
  - C_2 checks all positions where middle bit=1
  - C_3 checks all positions where MSB=1
- Hence, nonzero syndrome $\{C_1, C_2, C_3\}$ points to faulty bit

4-bit SECDED Example

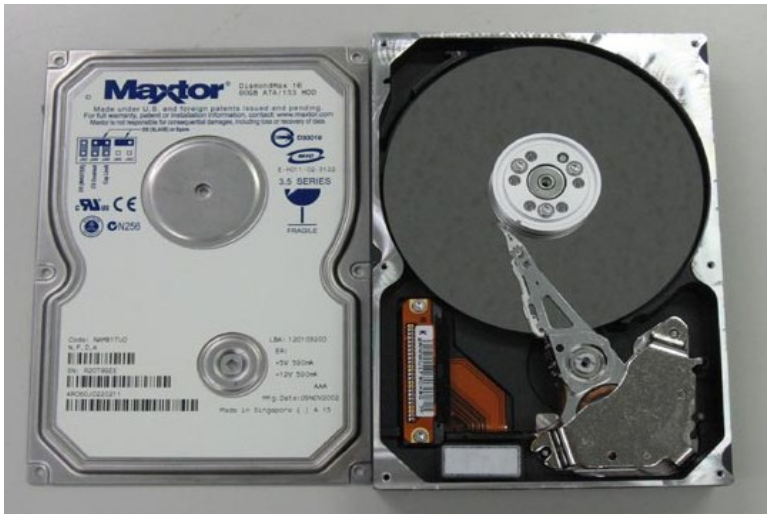
$$C_1 = b_1 \oplus b_2 \oplus b_4$$
$$C_2 = b_1 \oplus b_3 \oplus b_4$$
$$C_3 = b_2 \oplus b_3 \oplus b_4$$
$$P = even\_parity$$

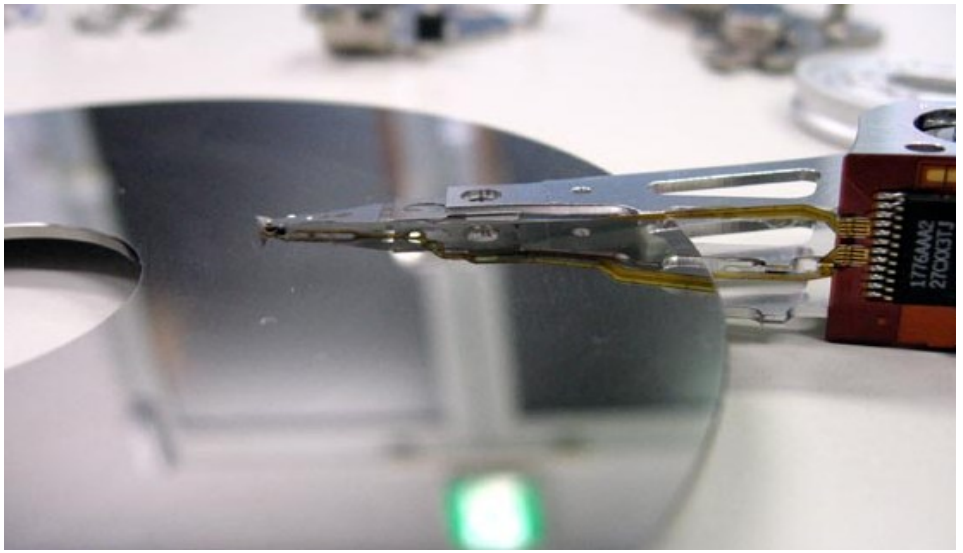
Bit Position	1	2	3	4	5	6	7		
Codeword	C_1	C_2	b_1	C_3	b_2	b_3	b_4	P	
Original data	1	0	1	1	0	1	0	0	Syndrome
No corruption	1	0	1	1	0	1	0	0	0 0 0, P ok
1 bit corrupted	1	0	0	1	0	1	0	0	0 1 1, P !ok
2 bits corrupted	1	0	0	1	1	1	0	0	1 1 0, P ok

- 4 data bits, 3 check bits, 1 parity bit
- Syndrome is $\{C_1, C_2, C_3\}$
  - If (syndrome==0) and (parity OK) => no error
  - If (syndrome != 0) and (parity !OK) => flip bit position pointed to by syndrome
  - If (syndrome != 0) and (parity OK) => double-bit error

From lecture slide of The University of Wisconsin Madison, ECE/CS 552

Magnetic Disk（磁気ディスク）





http://sougo057.aicomp.jp/0001.html

Q3 2022 Hard Drive Failure Rates

Annualized Failure Rate (AFR)

Backblaze SSD Quarterly Failure Rates for Q2 2022

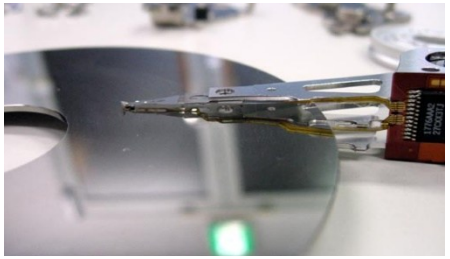
Reporting period: 4/1/22 thru 6/30/22 for drive models active as of 6/30/22

MFG	Model	Size (GB)	Drive Count	Drive Days	Drive Failures	AFR
Crucial	CT250MX500SSD1	250	272	20,002	0	-
Dell	DELLBOSS VD	480	351	29,066	0	-
Micron	MTFDDAV240TCB	240	89	8,084	1	4.52%
Seagate	ZA250CM10003	250	1,106	99,379	2	0.73%
Seagate	ZA500CM10003 (*)	500	3	42	0	-
Seagate	ZA2000CM10002	2000	3	271	0	-
Seagate	ZA250CM10002	250	559	50,477	4	2.89%
Seagate	ZA500CM10002	500	18	1,625	0	-
Seagate	ZA250NM1000 (*)	250	9	126	0	-
Seagate	SSD	300	106	9,541	0	-
WDC	WDS250G2B0A	250	42	3,781	0	-
			2,558	**222,394**	**7**	**1.15%**

(*) - New drive model in Q2 2022

Backblaze

https://www.backblaze.com/blog/ssd-drive-stats-mid-2022-review/



Backblaze Hard Drives Quarterly Failure Rates for Q3 2022

Reporting period: 7/1/2022 through 9/30/2022 for drive models active as of 9/30/2022

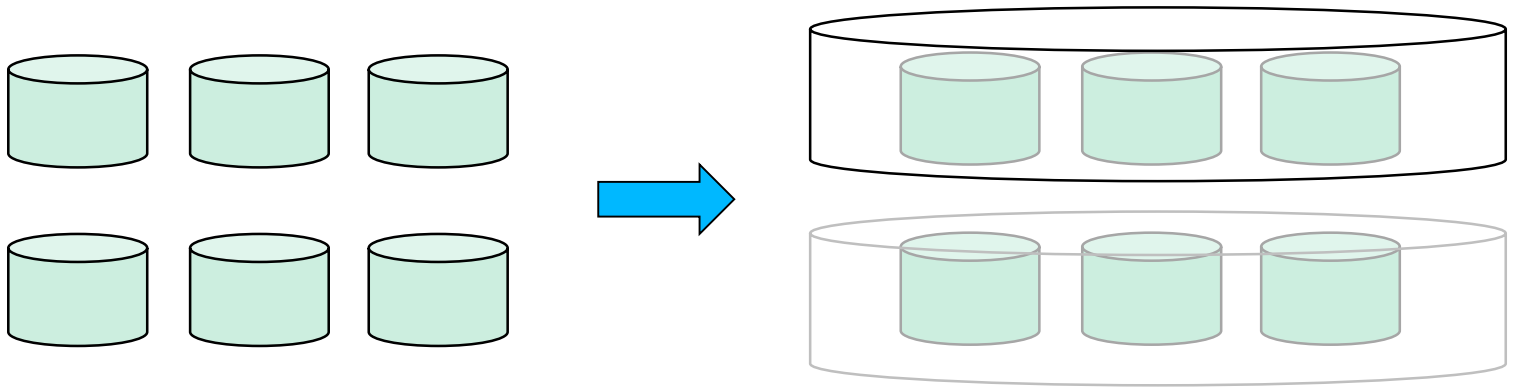
MFG	Model	Drive Size	Drive Count	Avg. Age (months)	Drive Days	Drive Failures	AFR
HGST	HMS5C4040ALE640	4TB	3,731	74.0	341,509	3	0.32%
HGST	HMS5C4040BLE640	4TB	12,730	71.1	1,170,925	14	0.44%
HGST	HUH728080ALE600	8TB	1,119	53.6	103,354	8	2.83%
HGST	HUH728080ALE604	8TB	95	62.6	7,637	-	0.00%
HGST	HUH721212ALE600	12TB	2,605	35.9	239,644	3	0.46%
HGST	HUH721212ALE604	12TB	13,157	18.3	1,209,798	19	0.57%
HGST	HUH721212ALN604	12TB	10,784	41.8	992,989	27	0.99%
Seagate	ST4000DM000	4TB	18,292	83.1	1,683,920	202	4.38%
Seagate	ST6000DX000	6TB	886	89.6	81,509	3	1.34%
Seagate	ST8000DM002	8TB	9,566	71.6	883,015	62	2.56%
Seagate	ST8000NM000A	8TB	79	11.2	26,974	-	0.00%
Seagate	ST8000NM0055	8TB	14,374	60.7	1,322,195	107	2.95%
Seagate	ST10000NM0086	10TB	1,174	58.6	108,372	9	3.03%
Seagate	ST12000NM0007	12TB	1,272	34.7	117,739	16	4.96%
Seagate	ST12000NM0008	12TB	19,910	30.1	1,837,021	124	2.46%
Seagate	ST12000NM001G	12TB	12,530	22.1	1,146,368	35	1.11%
Seagate	ST14000NM001G	14TB	10,737	19.9	987,184	40	1.48%
Seagate	ST14000NM0138	14TB	1,535	21.8	142,894	36	9.20%
Seagate	ST16000NM001G	16TB	20,402	10.7	1,696,759	29	0.62%
Seagate	ST16000NM002J	16TB	310	3.6	22,105	2	3.30%
Toshiba	MD04ABA400V	4TB	95	88.3	8,849	2	8.25%
Toshiba	MG07ACA14TA	14TB	38,203	23.1	3,514,384	117	1.22%
Toshiba	MG07ACA14TEY	14TB	537	18.4	47,742	2	1.53%
Toshiba	MG08ACA16TA	16TB	3,751	3.9	243,198	5	0.75%
Toshiba	MG08ACA16TE	16TB	5,942	11.7	546,805	22	1.47%
Toshiba	MG08ACA16TEY	16TB	4,244	11.9	385,715	12	1.14%
WDC	WUH721414ALE6L4	14TB	8,409	21.8	773,557	5	0.24%
WDC	WUH721816ALE6L0	16TB	2,702	11.8	248,428	-	0.00%
WDC	WUH721816ALE6L4	16TB	7,138	2.8	310,502	6	0.71%
			226,309		**20,201,091**	**910**	**1.64%**

https://www.backblaze.com/blog/backblaze-drive-stats-for-q3-2022/

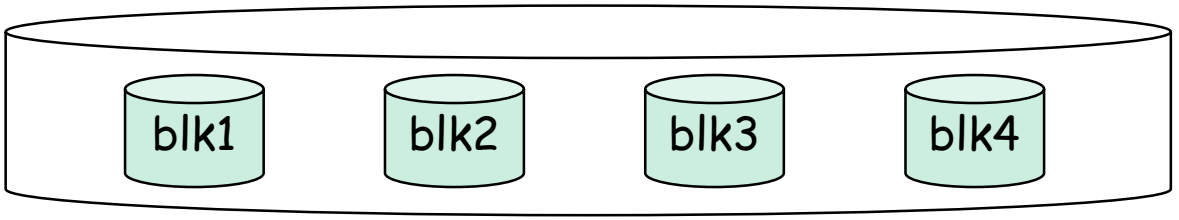
Backblaze

RAID: Redundant Array of Inexpensive Disks

- Arrays of small and inexpensive disks
  - Increase potential throughput by having many disk drives
  - Data is spread over multiple disk
  - Multiple accesses are made to several disks at a time
- **Reliability** is lower than a single disk
- But **availability** can be improved by adding redundant disks

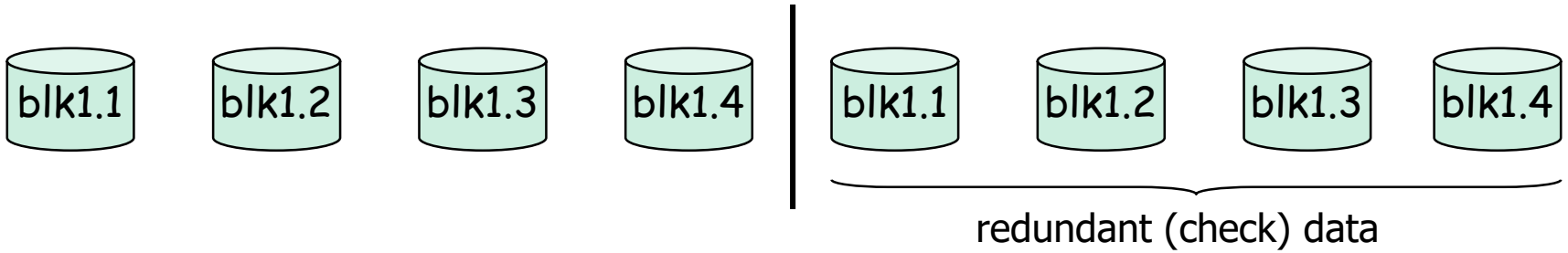


RAID: Level 0 (RAID 0, 冗長性なし，ストライピング)



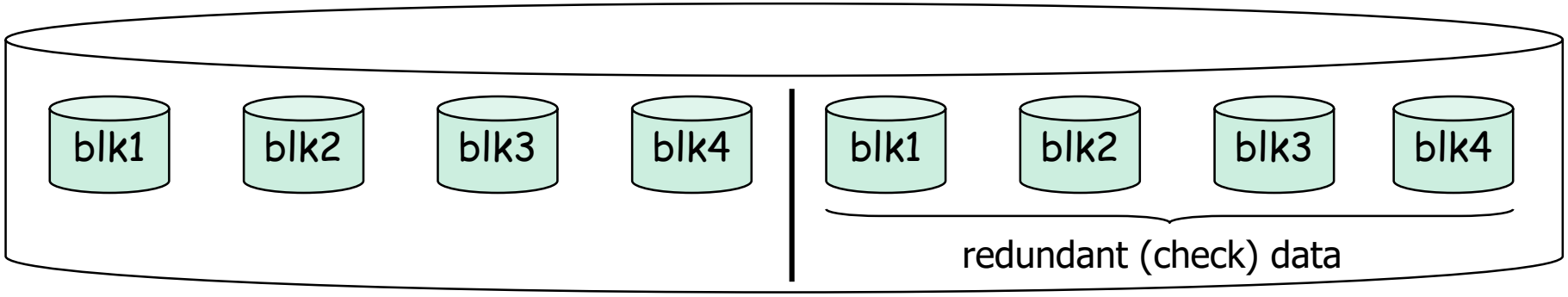
- Multiple smaller disks as opposed to **one big disk**
  - Spreading the blocks over multiple disks – **striping** – means that multiple blocks can be accessed in parallel increasing the performance
  - 4 disk system gives four times the throughput of a 1 disk system
  - Same cost as one *big* disk – assuming 4 small disks cost the same as one big disk
- No redundancy, so what if one disk fails?

RAID: Level 1 (Redundancy via Mirroring)



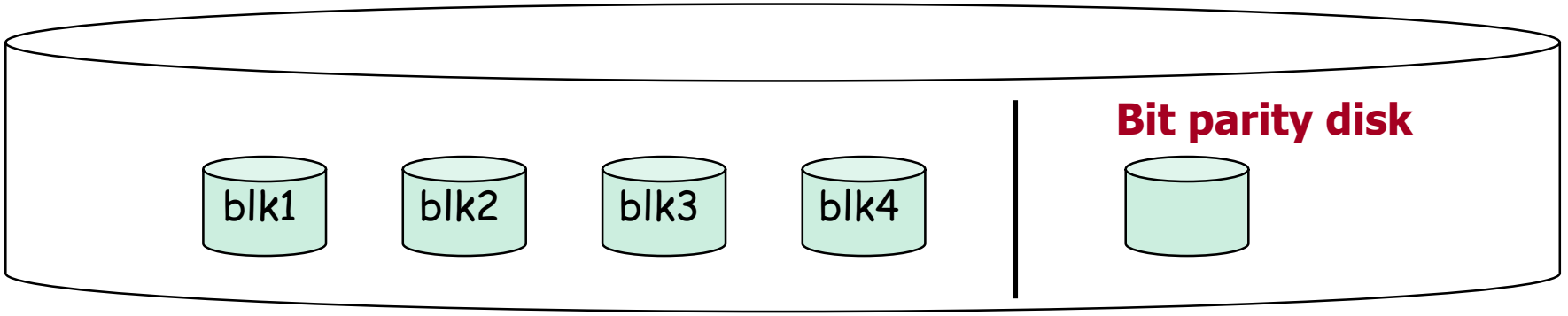
- Uses twice as many disks for redundancy so there are always two copies of the data
  - The number of redundant disks = the number of data disks so twice the cost of one big disk
    - writes have to be made to both sets of disks, so writes would be only 1/2 the performance of RAID 0
- What if one disk fails?
  - If a disk fails, the system just goes to the "**mirror**" for the data

RAID: Level 0+1 (RAID01, Striping with Mirroring)



- Combines the best of RAID 0 and RAID 1, data is striped across four disks and mirrored to four disks
  - Four times the throughput (due to striping)
  - # redundant disks = # of data disks so twice the cost of one big disk
    - writes have to be made to both sets of disks, so writes would be only 1/2 the performance of RAID 0
- What if one disk fails?
  - If a disk fails, the system just goes to the "**mirror**" for the data

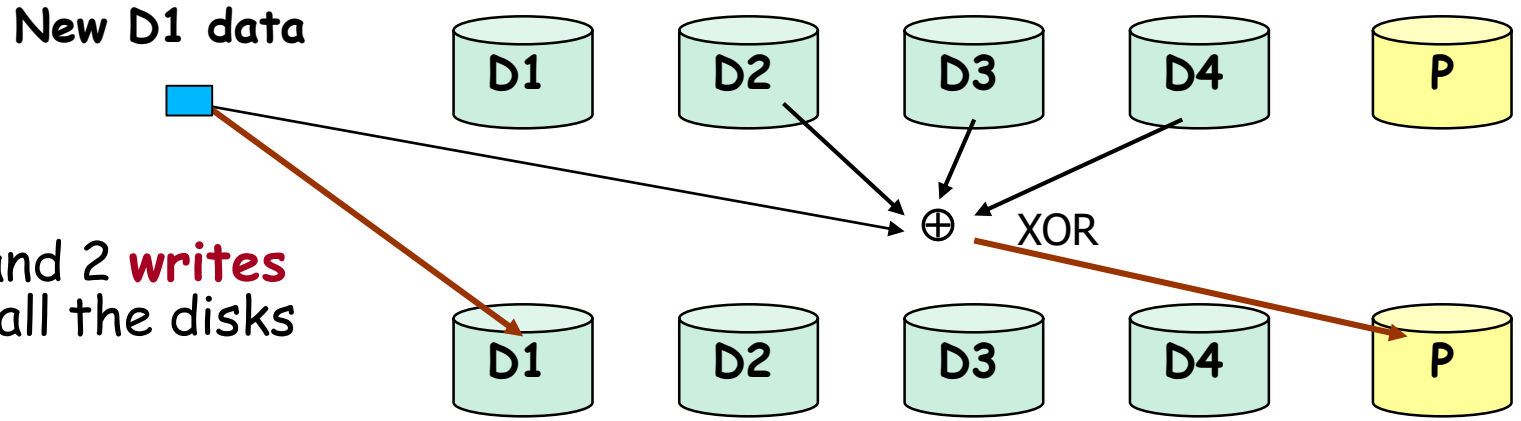
RAID: Level 3 (Bit/Byte-Interleaved **Parity**)



- Cost of higher availability is reduced to 1/N where N is the number of disks in a protection group
  - # redundant disks = 1 × # of protection groups
    - **writes** require writing the new data to the data disk as well as computing the parity, meaning reading the other disks, so that the parity disk can be updated
    - **reads** require reading all the operational data disks as well as the parity disk to calculate the missing data that was stored on the **failed disk**

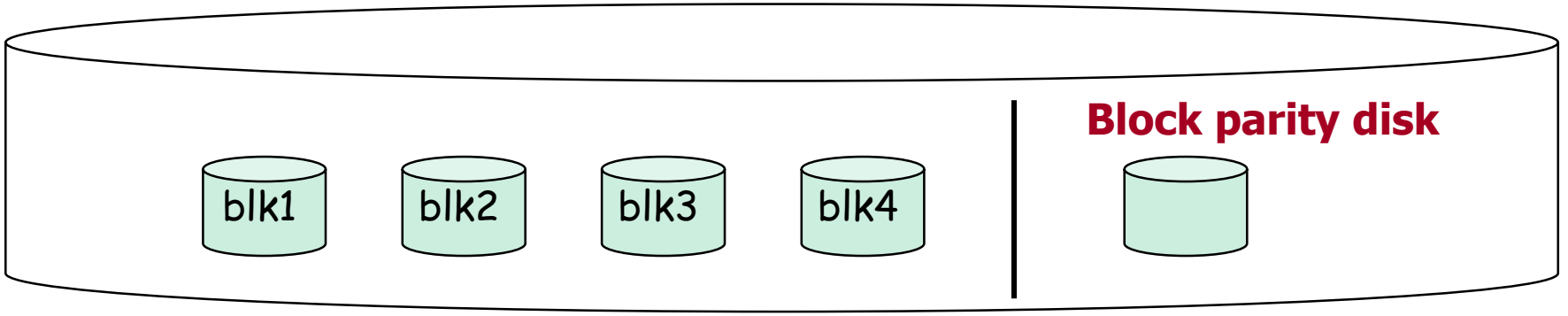
RAID 3 and parity

- RAID 3



3 **reads** and 2 **writes**
involving all the disks

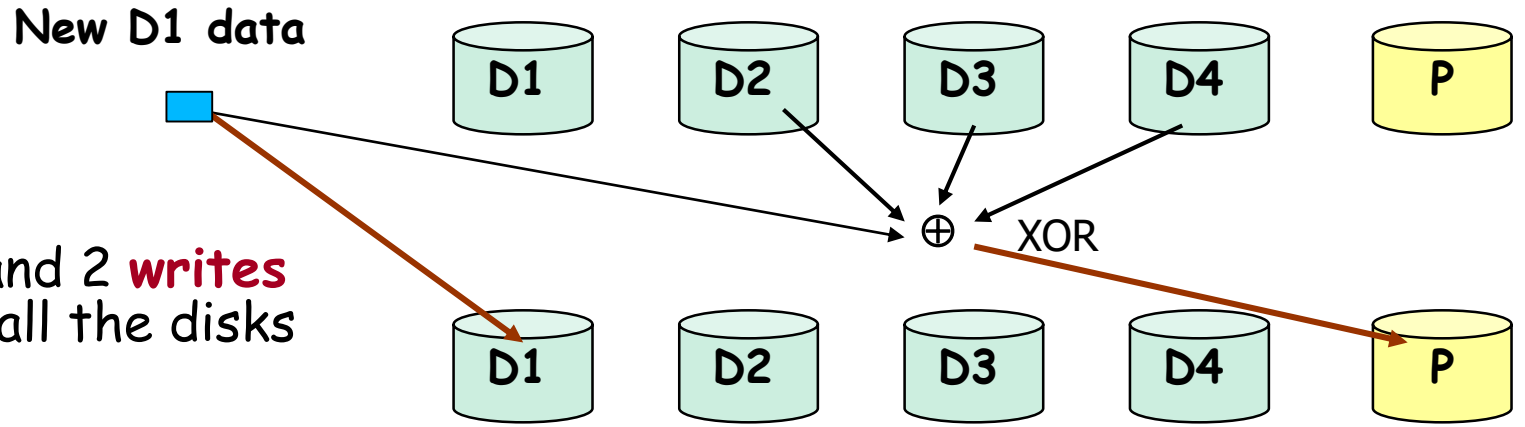
RAID: Level 4 (Block-Interleaved Parity)



- Cost of higher availability still only 1/N but the parity is stored as **blocks** associated with sets of data blocks
  - Four times the throughput (striping)
  - # redundant disks = 1 × # of protection groups
  - Supports "small reads" and "small writes" (reads and writes that go to just one (or a few) data disk in a protection group)

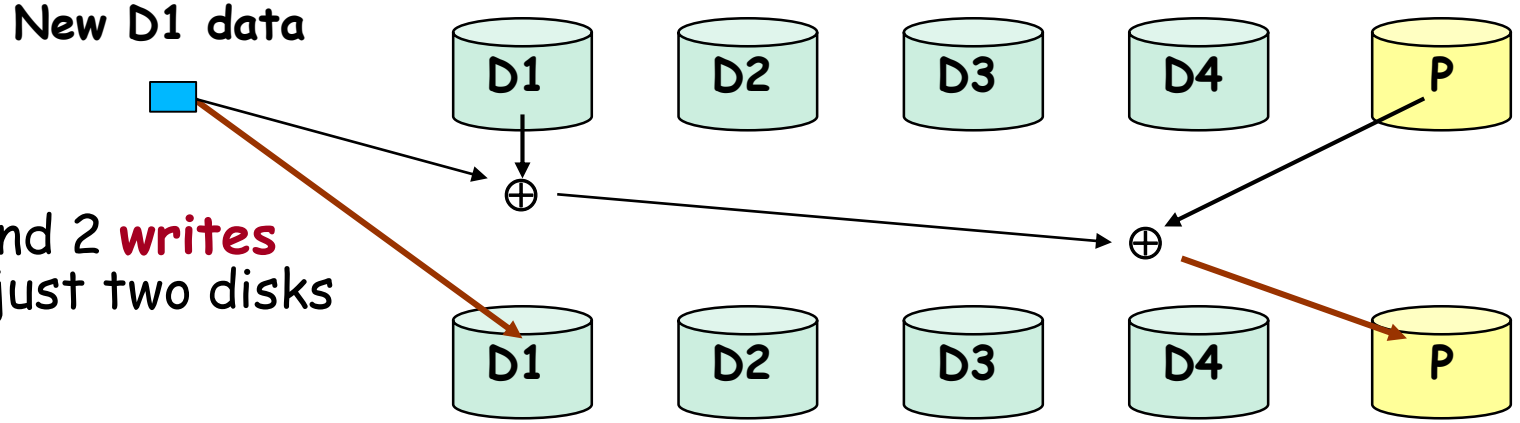
Small Reads and Small Writes

- RAID 3



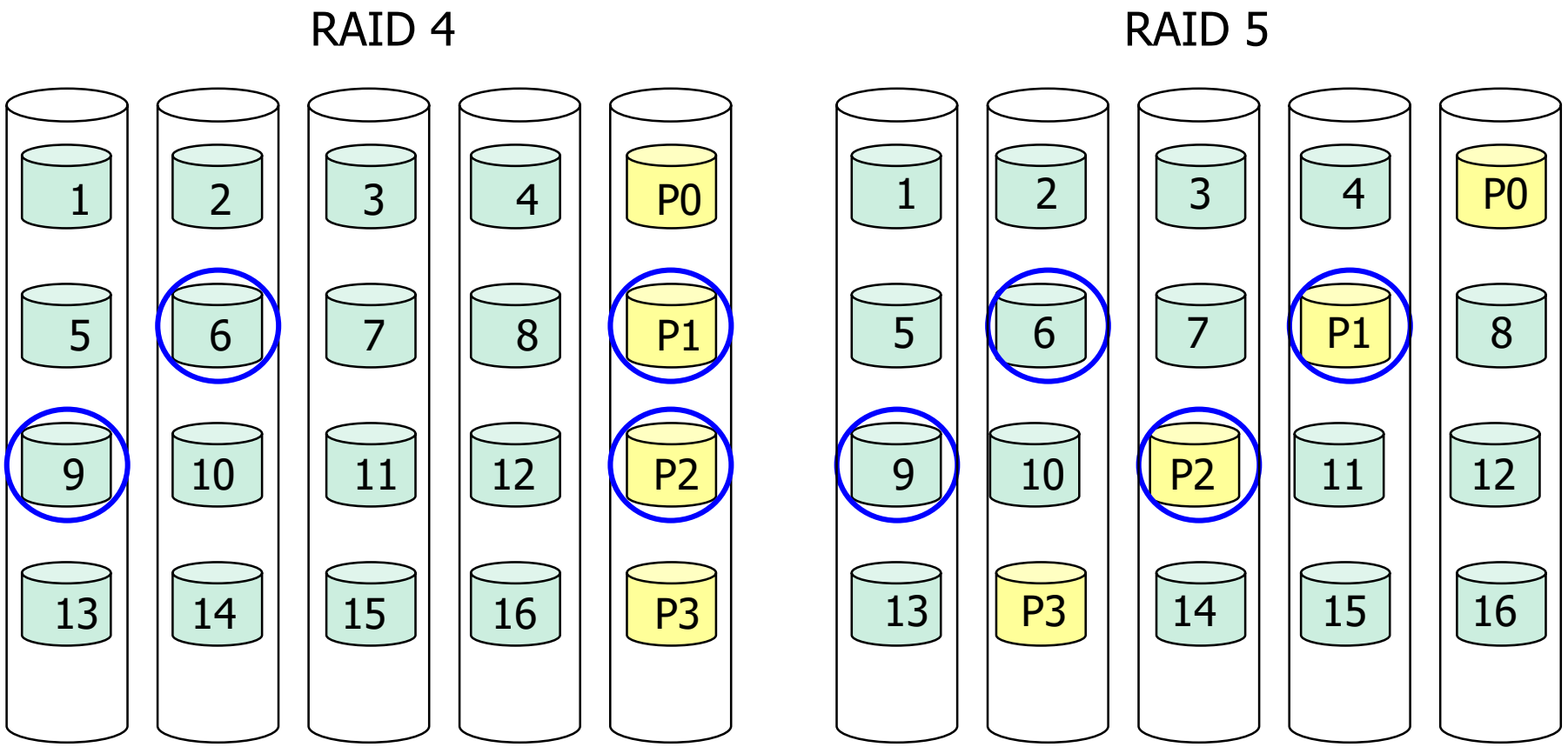
3 **reads** and 2 **writes**
involving all the disks

- RAID 4 small reads and small writes



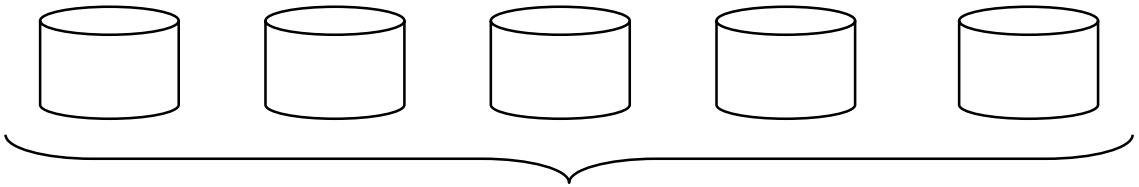
2 **reads** and 2 **writes**
involving just two disks

Distributing Parity Blocks



- By distributing parity blocks to all disks, some small writes can be performed **in parallel**

RAID: Level 5 (Distributed Block-Interleaved Parity)



one of these assigned as the block parity disk

- Cost of higher availability still only 1/N but the parity block can be located on any of the disks
  so there is no single bottleneck for writes
  - Still four times the throughput (striping)
  - # redundant disks = 1 × # of protection groups
  - Supports "**small reads**" and "**small writes**" (reads and writes that go to just one (or a few) data disk in a protection group)
  - Allows **multiple simultaneous writes**