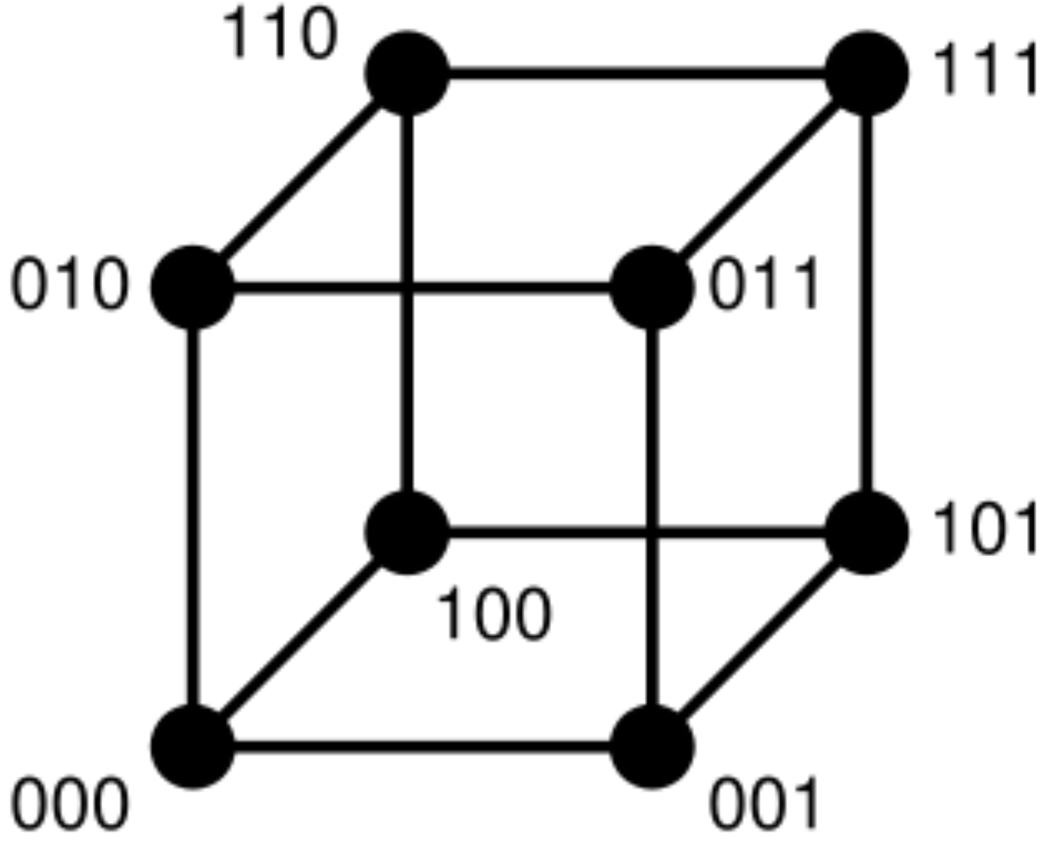
CS 61C:
Great Ideas in Computer Architecture

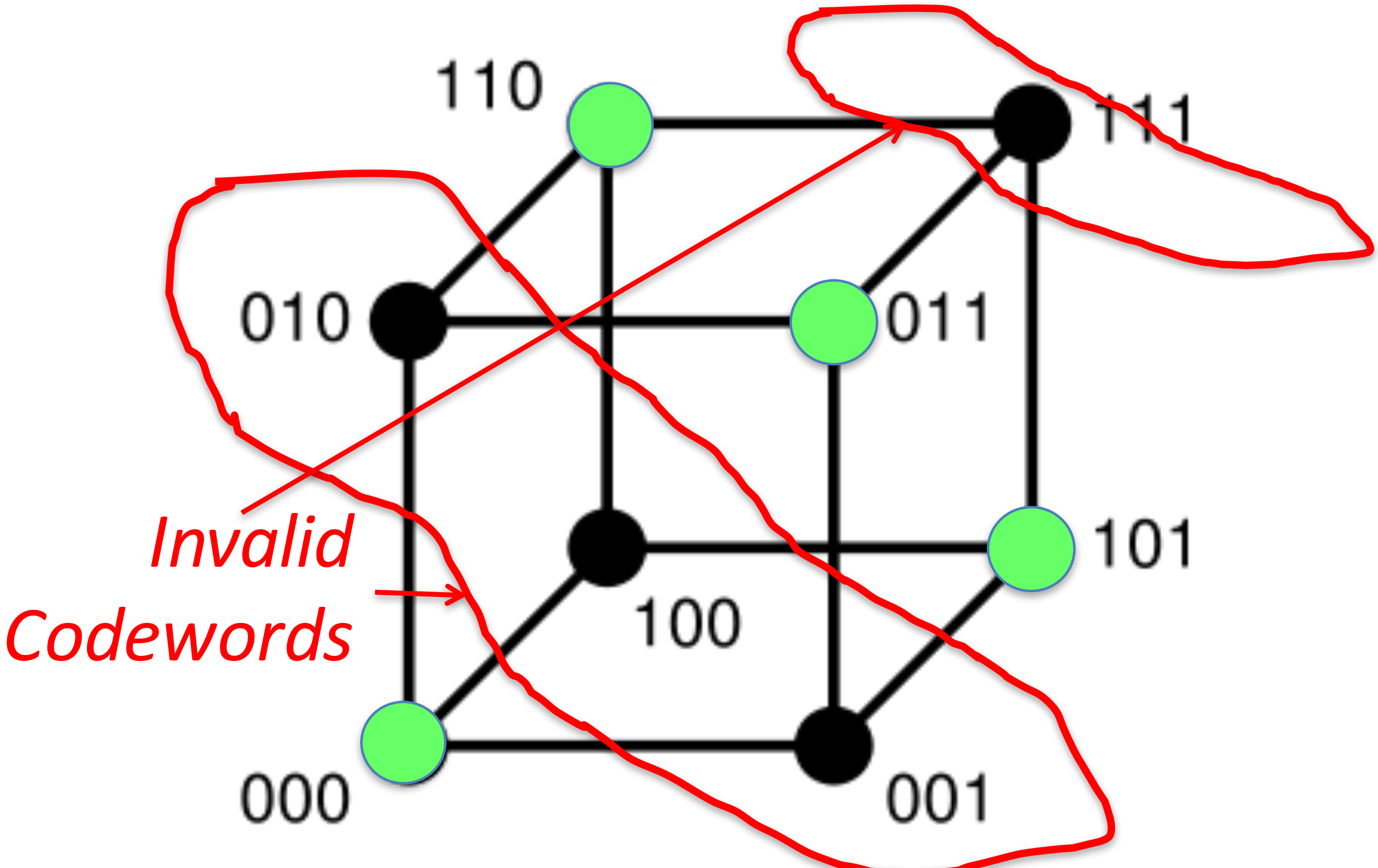
Dependability – More on ECC, RAID

Vladimir Stojanovic & Nicholas Weaver
http://inst.eecs.berkeley.edu/~cs61c/

Hamming Distance: 8 code words



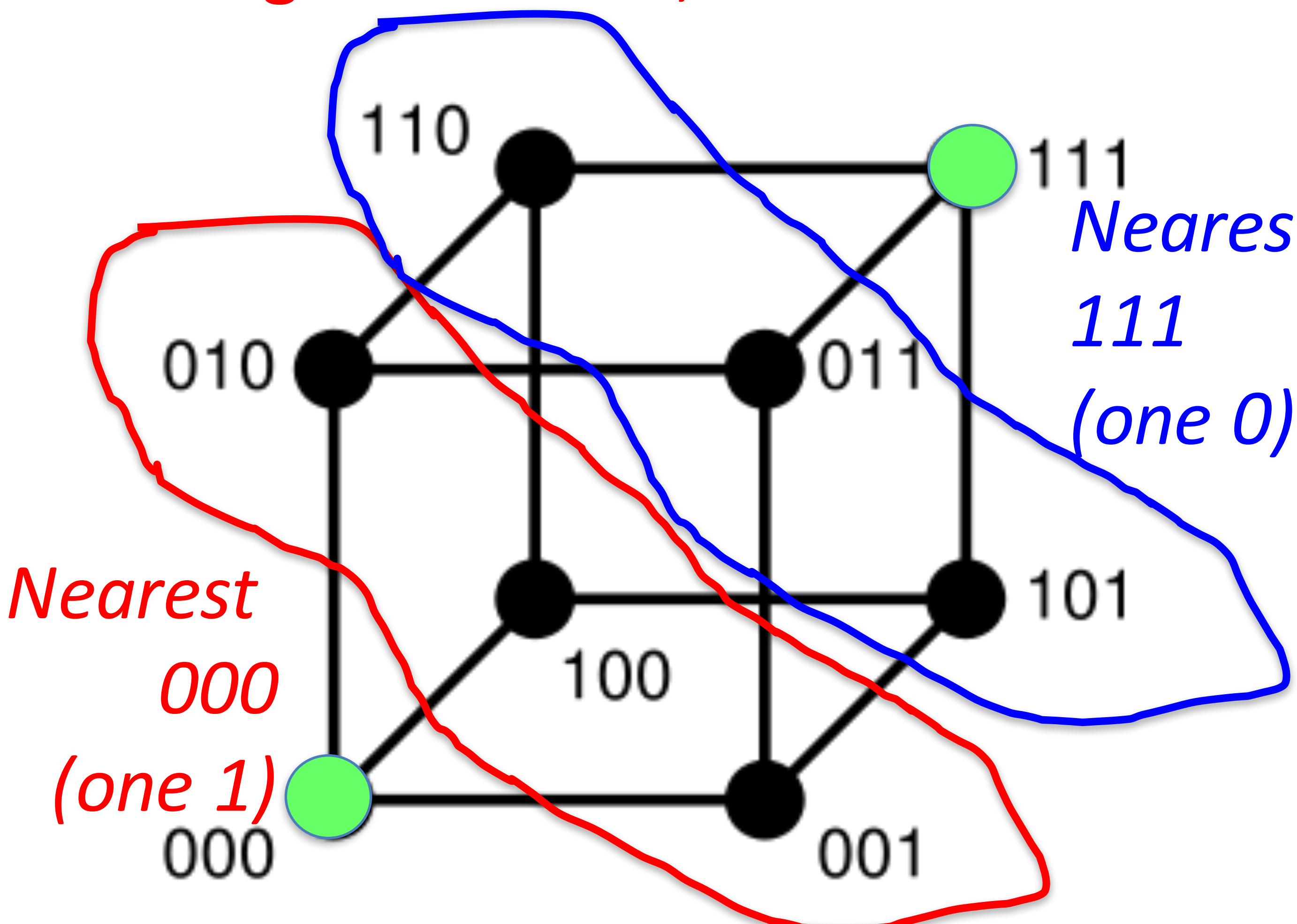
Hamming Distance 2: Detection
Detect Single Bit Errors



- No 1 bit error goes to another valid codeword
- ½ codewords are valid

Hamming Distance 3: Correction

Correct Single Bit Errors, Detect Double Bit Errors



- No 2 bit error goes to another valid codeword; 1 bit error near
- 1/4 codewords are valid

Hamming Error Correcting Code

- Overhead involved in single error-correction code
- Let p be total number of parity bits and d number of data bits in $p + d$ bit word
- If p error correction bits are to point to error bit ($p + d$ cases) + indicate that no error exists (1 case), we need:

  $$2^p >= p + d + 1,$$

  thus $p >= \log_2(p + d + 1)$

  for large d, p approaches $\log_2(d)$
- *8 bits data => $d = 8$, $2^p >= p + 8 + 1 => p >= 4$*
- *16b data => 5b parity,*
  32b data => 6b parity,
  64b data => 7b parity

Hamming Single-Error Correction, Double-Error Detection (SEC/DED)

- Adding extra parity bit covering the entire word provides double error detection as well as single error correction

  1 2 3 4 5 6 7 8

  p_1 p_2 d_1 p_3 d_2 d_3 d_4 p_4

- Hamming parity bits H (p_1 p_2 p_3) are computed (even parity as usual) plus the even parity over the entire word, p_4:

  H=0 p_4=0, no error

  H≠0 p_4=1, correctable single error (odd parity if 1 error => p_4=1)

  H≠0 p_4=0, double error occurred (even parity if 2 errors=> p_4=0)

  H=0 p_4=1, single error occurred in p_4 bit, not in rest of word

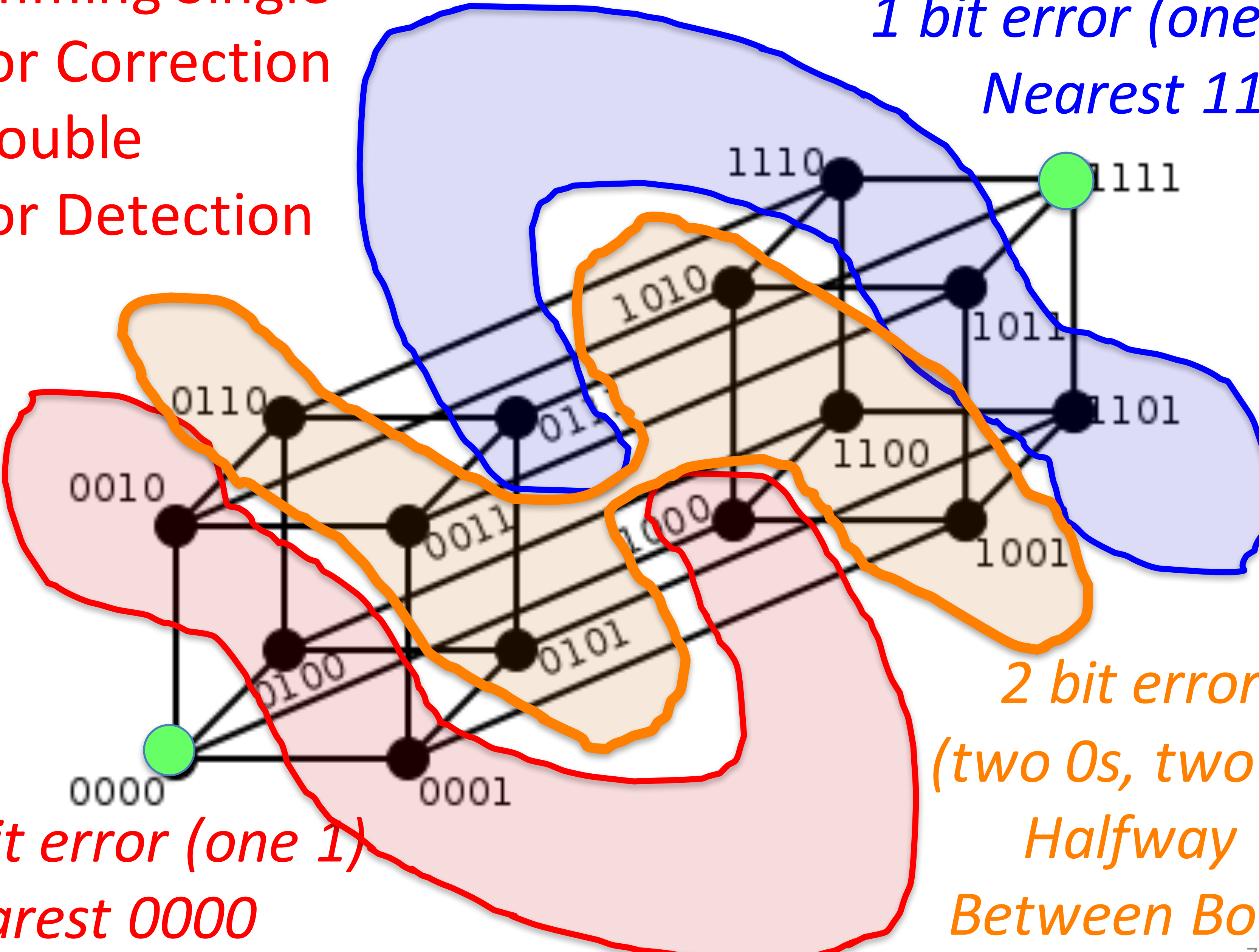
Typical modern codes in DRAM memory systems:

64-bit data blocks (8 bytes) with 72-bit code words (9 bytes).

Hamming Distance = 4

Hamming Single Error Correction + Double Error Detection

1 bit error (one 0) Nearest 1111



2 bit error (two 0s, two 1s) Halfway Between Both

1 bit error (one 1) Nearest 0000

iClicker Question

The following word is received, encoded with Hamming code:

0 1 1 0 0 0 1

What is the corrected data bit sequence?

A. 1 1 1 1
B. 0 0 0 1
C. 1 1 0 1
D. 1 0 1 1
E. 1 0 0 0

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11
Parity bit coverage	p1	X		X		X		X		X		X		X		X
	p2		X	X			X	X			X	X			X	X
	p4				X	X	X	X					X	X	X	X
	p8								X	X	X	X	X	X	X	X

What if More Than 2-Bit Errors?

- Network transmissions, disks, distributed storage common failure mode is bursts of bit errors, not just one or two bit errors
  - Contiguous sequence of *B* bits in which first, last and any number of intermediate bits are in error
  - Caused by impulse noise or by fading in wireless
  - Effect is greater at higher data rates
- Solve with Cyclic Redundancy Check (CRC), interleaving or other more advanced codes

iClicker Question

The following word is received, encoded with Hamming code:

0 1 1 0 0 0 1

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11
Parity bit coverage	p1	X		X		X		X		X		X		X		X
	p2		X	X			X	X			X	X			X	X
	p4				X	X	X	X					X	X	X	X
	p8								X	X	X	X	X	X	X	X

check p1: 0 x 1 x 0 x 1 – o.k.

check p2: x 1 1 x x 0 1 – error in p2

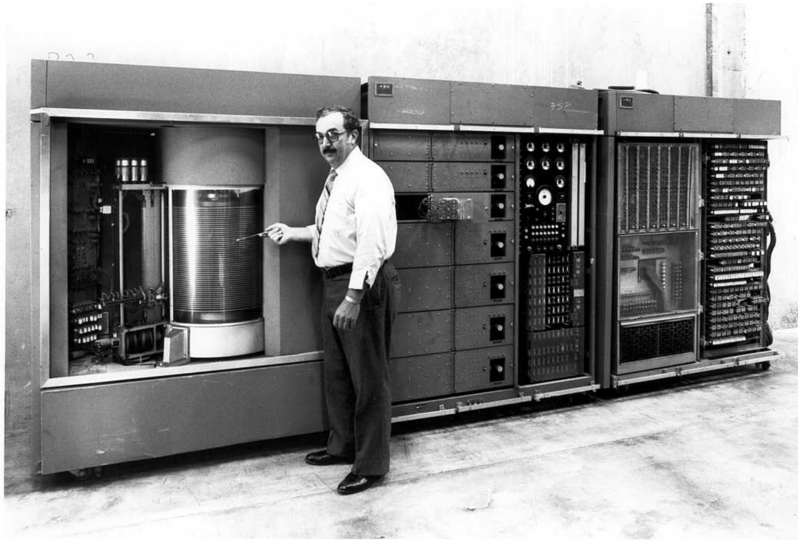
check p4: x x x 0 0 0 1 – error in p4

Error in location 2+4 =6

Correct data: 1 0 1 1 (answer D)

Evolution of the Disk Drive

IBM 3390K, 1986



IBM RAMAC 305, 1956

Apple SCSI, 1986

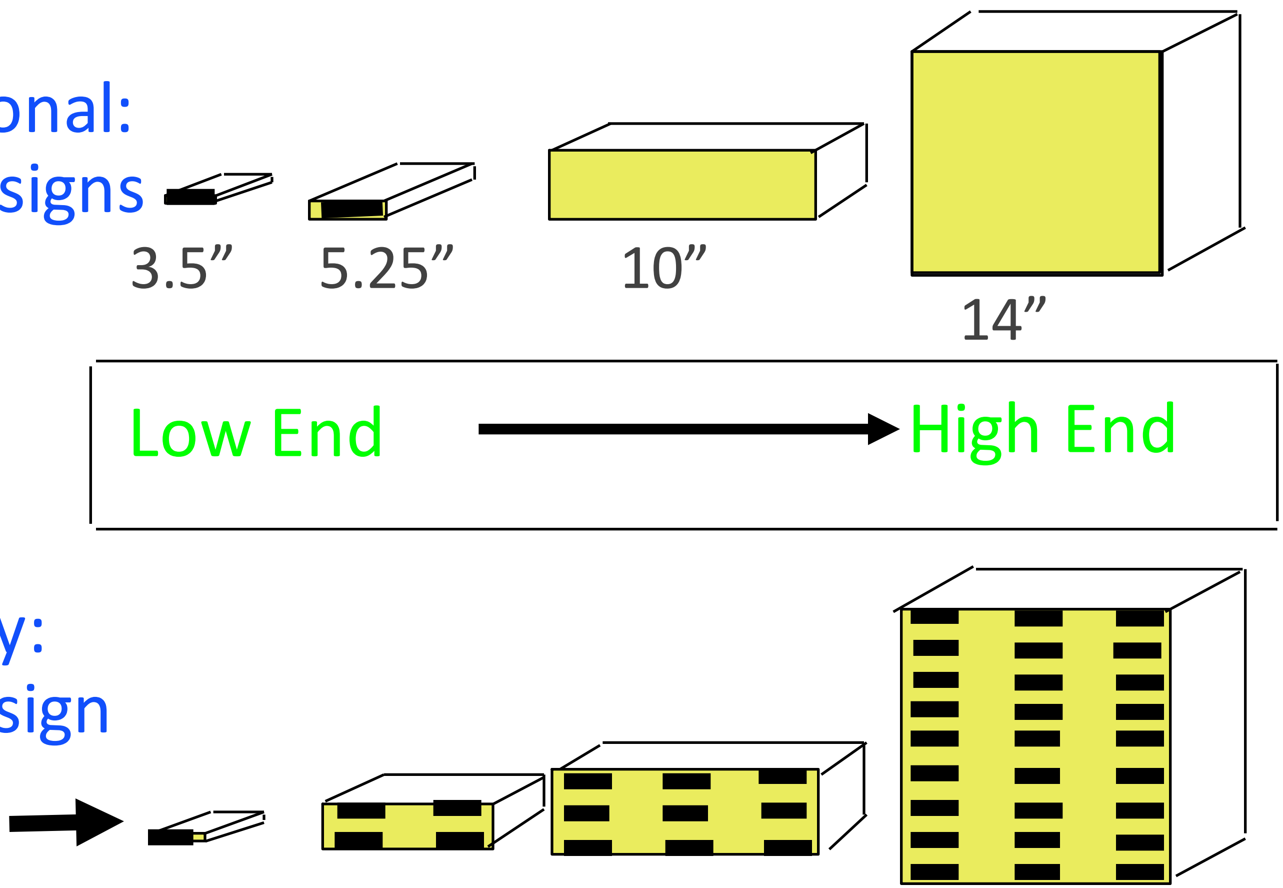
Arrays of Small Disks

Can smaller disks be used to close gap in performance between disks and CPUs?

Conventional:
4 disk designs

Disk Array:
1 disk design

3.5”



Replace Small Number of Large Disks with Large Number of Small Disks! (1988 Disks)

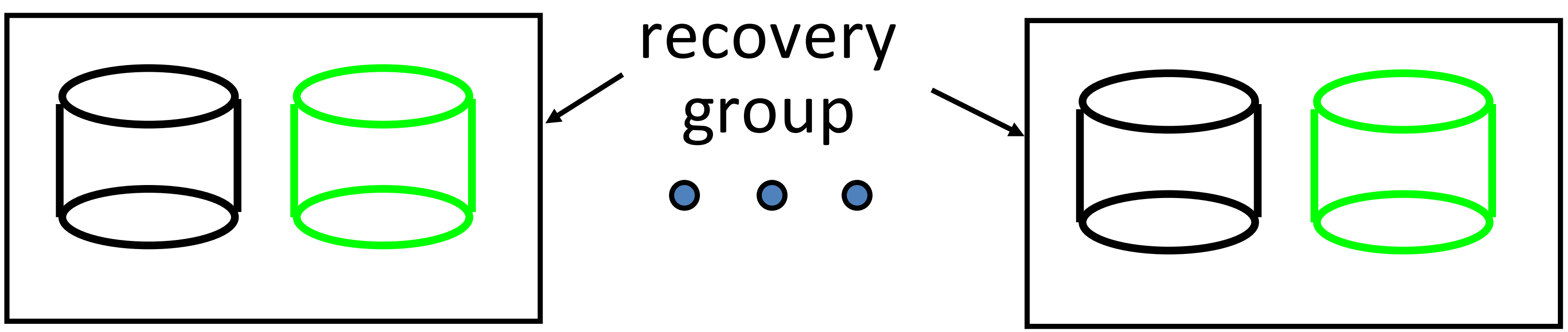
	IBM 3390K	IBM 3.5" 0061	x70	
Capacity	20 GBytes	320 MBytes	23 GBytes	
Volume	97 cu. ft.	0.1 cu. ft.	11 cu. ft.	9X
Power	3 KW	11 W	1 KW	3X
Data Rate	15 MB/s	1.5 MB/s	120 MB/s	8X
I/O Rate	600 I/Os/s	55 I/Os/s	3900 IOs/s	6X
MTTF	250 KHrs	50 KHrs	??? Hrs	
Cost	$250K	$2K	$150K	

Disk Arrays have potential for large data and I/O rates, high MB per cu. ft., high MB per KW, but what about reliability?

RAID: Redundant Arrays of (Inexpensive) Disks

- Files are "striped" across multiple disks
- Redundancy yields high data availability
  - Availability: service still provided to user, even if some components failed
- Disks will still fail
- Contents reconstructed from data redundantly stored in the array
  - Capacity penalty to store redundant info
  - Bandwidth penalty to update redundant info

Redundant Arrays of Inexpensive Disks RAID 1: Disk Mirroring/Shadowing



- Each disk is fully duplicated onto its "mirror"
  Very high availability can be achieved
- Writes limited by single-disk speed
- Reads may be optimized

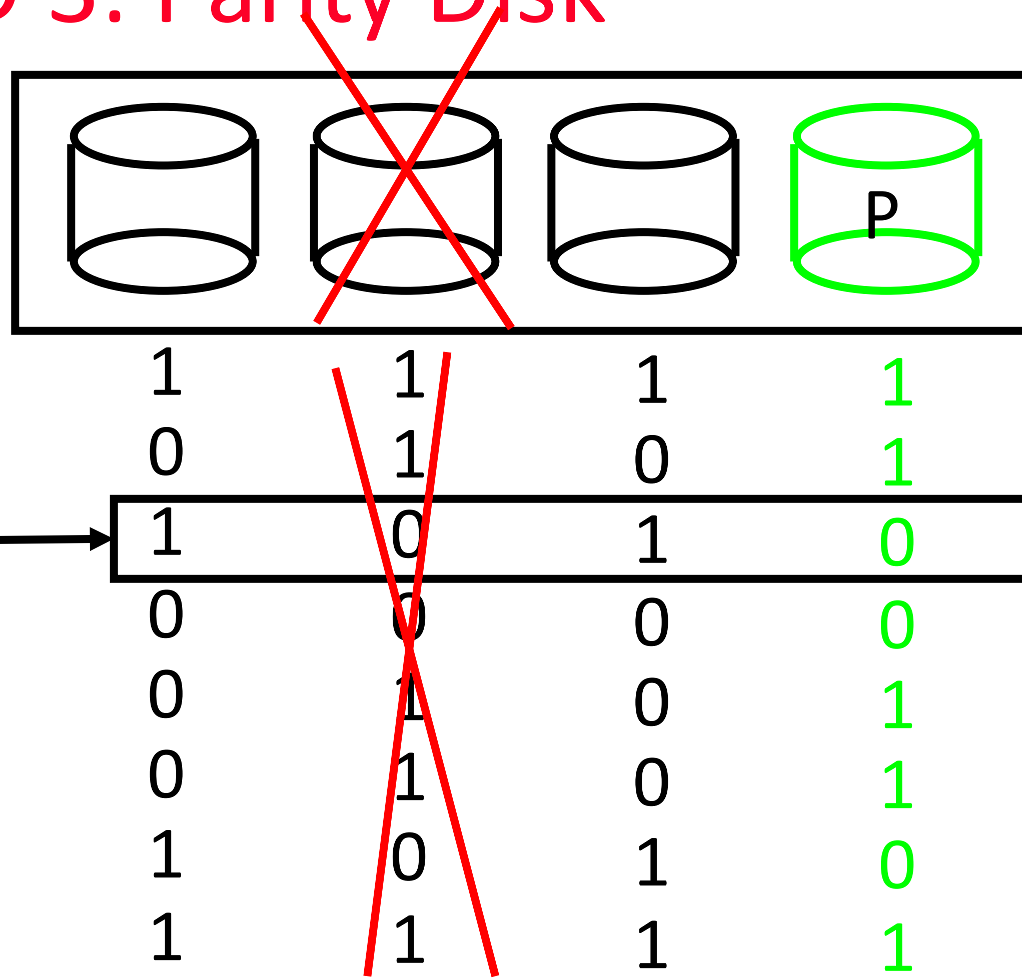
Most expensive solution: 100% capacity overhead

Redundant Array of Inexpensive Disks
RAID 3: Parity Disk

10010011
11001101
10010011
. . .

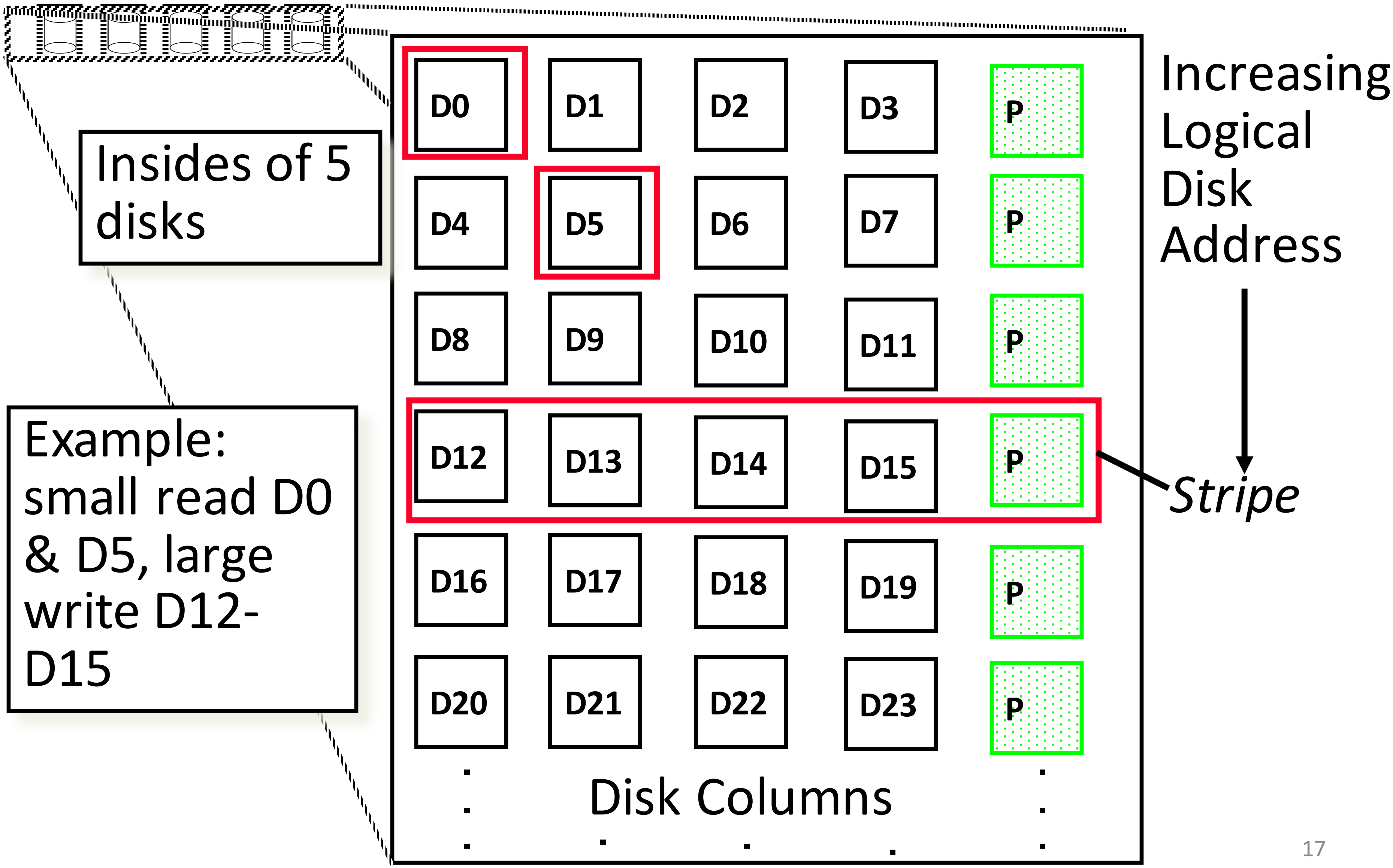
logical record

Striped physical records



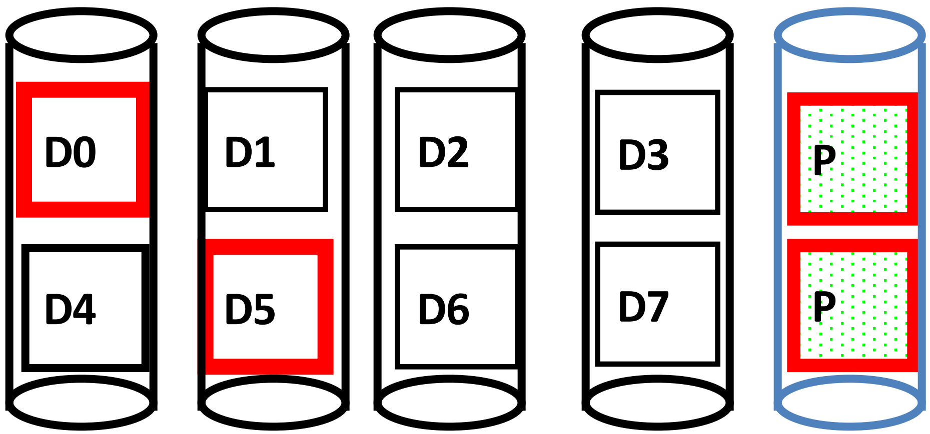
P contains sum of other disks per stripe mod 2 ("parity")
If disk fails, subtract P from sum of other disks to find missing information

Redundant Arrays of Inexpensive Disks RAID 4: High I/O Rate Parity



Inspiration for RAID 5

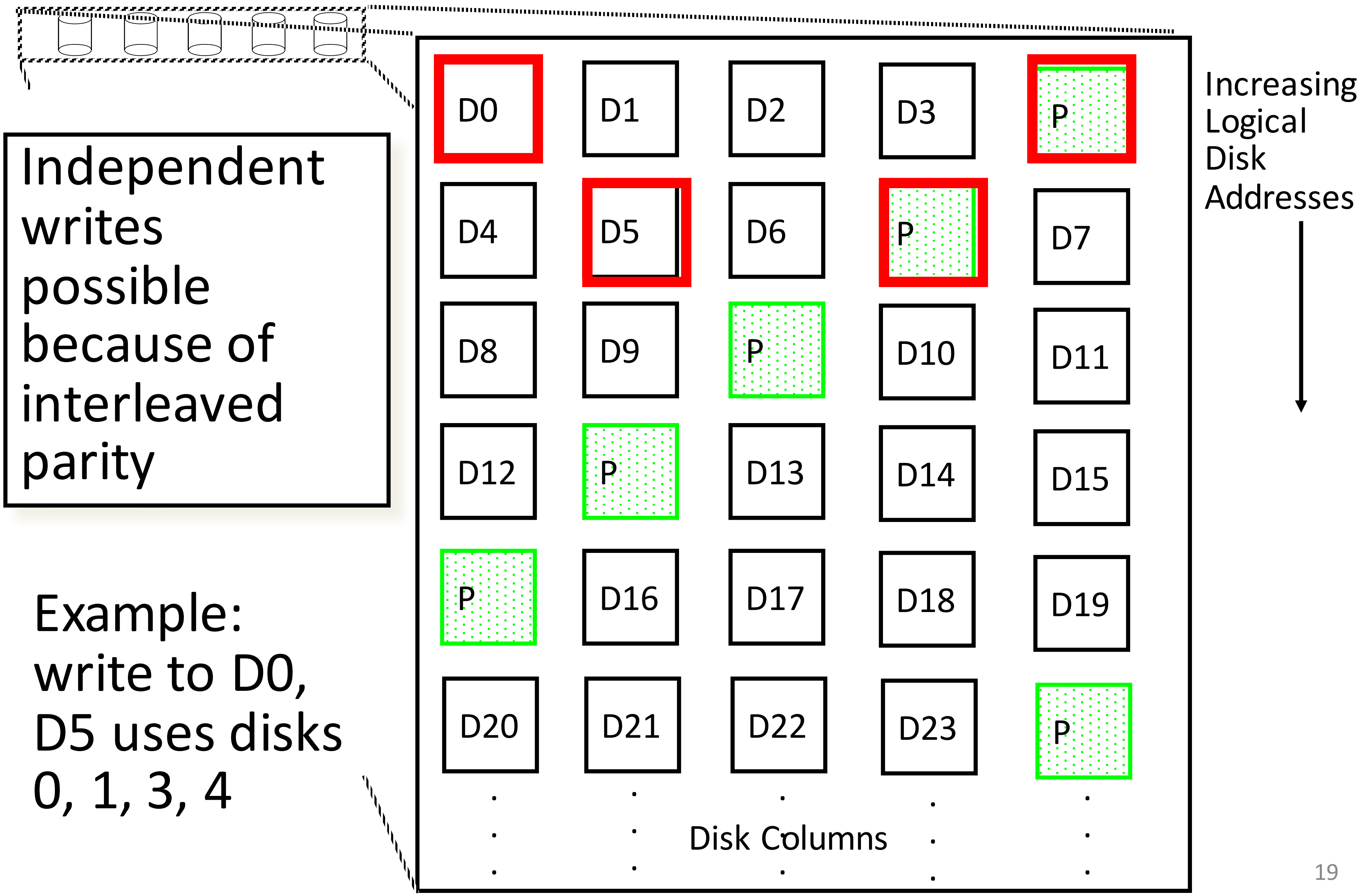
- RAID 4 works well for small reads
- Small writes (write to one disk):
  - Option 1: read other data disks, create new sum and write to Parity Disk
  - Option 2: since P has old sum, compare old data to new data, add the difference to P
- Small writes are limited by Parity Disk: Write to D0, D5 both also write to P disk



RAID 5: High I/O Rate Interleaved Parity

Independent writes possible because of interleaved parity

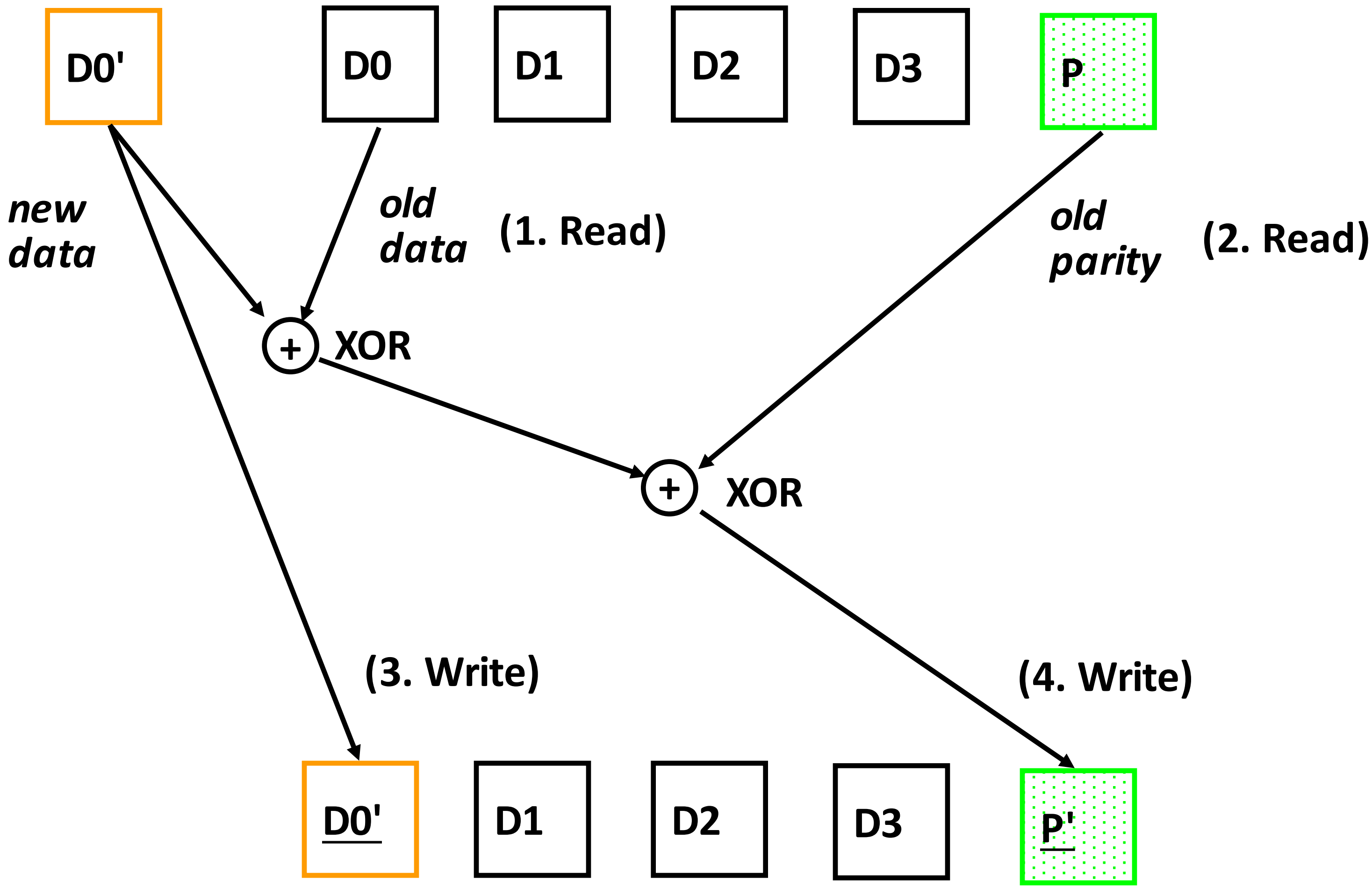
Example: write to D0, D5 uses disks 0, 1, 3, 4



Problems of Disk Arrays: Small Writes

RAID-5: Small Write Algorithm

1 Logical Write = 2 Physical Reads + 2 Physical Writes



Tech Report Read 'Round the World (December 1987)

A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A. Patterson, Garth Gibson, and Randy H. Katz

Google | Case for Raid

Scholar | About 138,000 results (**0.08** sec)

Articles

Legal documents

Any time

[BOOK] A **case** for redundant arrays of inexpensive disks (**RAID**)
DA Patterson, G Gibson, RH Katz - 1988 - dl.acm.org
Abstract Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O. While the capacity of Single Large Expensive Disks (SLED) has grown rapidly, the performance improvement of SLED has been modest. ...
Cited by 2814 Related articles All 239 versions Cite More

Expensive Disk (SLED) has grown rapidly, the performance improvement of SLED has been modest. Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability.

This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAIDs to an IBM 3380 and a Fujitsu Super Eagle.

RAID-I

- RAID-I (1989)
  - Consisted of a Sun 4/280 workstation with 128 MB of DRAM, four dual-string SCSI controllers, 28 5.25-inch SCSI disks and specialized disk striping software

RAID II

- 1990-1993
- Early Network Attached Storage (NAS) System running a Log Structured File System (LFS)
- Impact:
  - $25 Billion/year in 2002
  - Over $150 Billion in RAID device sold since 1990-2002
  - 200+ RAID companies (at the peak)
  - Software RAID a standard component of modern OSs

And, in Conclusion, …

- Memory
  - Hamming distance 2: Parity for Single Error Detect
  - Hamming distance 3: Single Error Correction Code + encode bit position of error
- Treat disks like memory, except you know when a disk has failed—erasure makes parity an Error Correcting Code
- RAID-2, -3, -4, -5: Interleaved data and parity