ECE 5730
Memory Systems
Spring 2009

Redundant Array of Inexpensive (Independent) Disks (RAID)

Cornell University

Announcements

- No office hours today

- Make-up class #2
  - Thursday, April 30, 6:00-7:15pm, PH 403
  - Pizza and soda

- Exam II
  - May 7, 7:00-10:00pm, Hollister 314
  - Covers material from 3/10-4/28 but excluding 4/22 (Lectures 14-21, 23-24)

- Final report (15-25 double-spaced pages)
  - Email Word or PDF to me by 11:59pm on May 1
  - 20 points off final project grade if late

Defective Sectors and Wasted Space

- answer from Spencer

- Since we waste space at the end of tracks, map that wasted space where the defects are found

- i.e. remap drive to consider the defects as "spare" sectors

- Theoretically doable but adds complexity
  - Unique formatting info would be required for every track on the drive
  - Amount of wasted space varies from track to track within a zone
  - Size of the defect must be determined

- too annoying for industry

Storage Subsystems ("RAID")

- **Multiple disk drives managed as a single unit to improve performance and reliability**

re-mapped

- **Appear as one or more *logical drives* to the user**

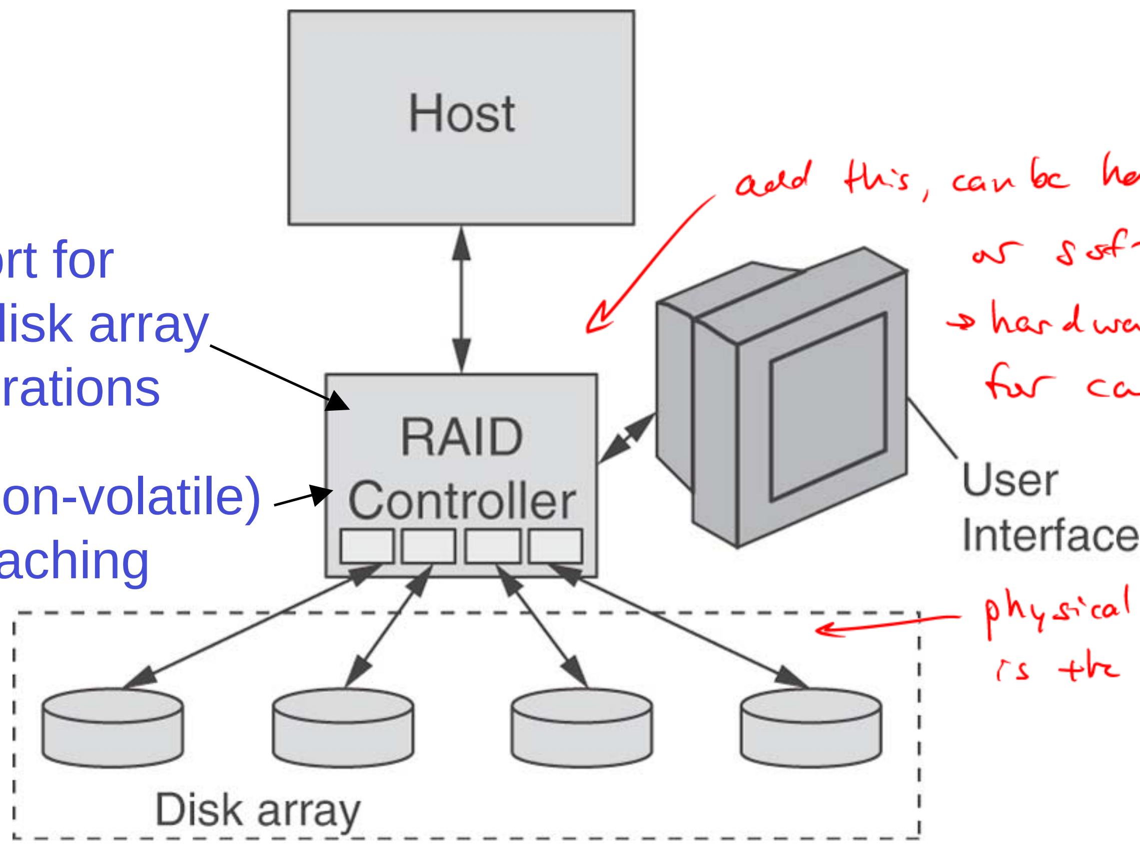
$100's in 2009

add this, can be hardware or software

→ hardware allows for caching

support for multiple disk array configurations

read and (non-volatile) write caching



physical interface is the same

Why Storage Subsystems ("RAID")?

- **Performance**
  - RAID 1
  - Read performance improves by duplicating, alternating, or distributing reads among N drives
  - RAID 0
  - RAID 2-5
  - Writes must occur to duplicate drives and may become a bottleneck

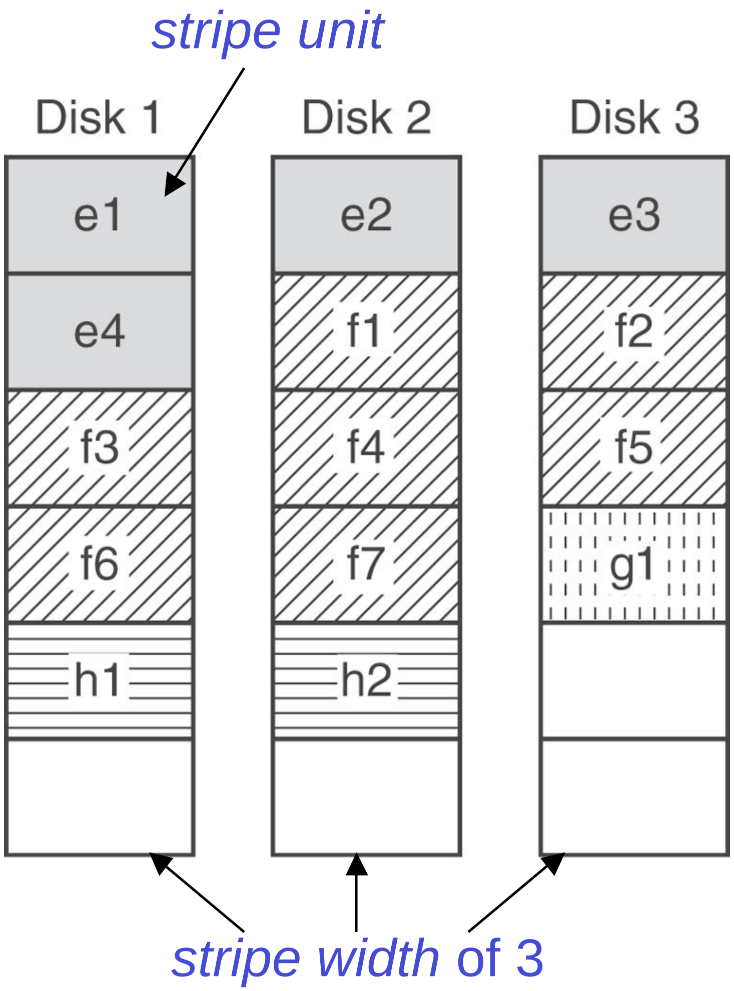
- **Reliability**
  - Redundancy permits the system to remain operational upon drive failures (w/ degraded performance)
  - Data can be copied from the redundant drive(s) to a replacement while the system remains operational

RAID Operating Modes

- **Normal mode**
  - All drives are operational
- **Degraded mode**
  - One (two in some RAIDs) drive is out of service
  - System is still operational
  - performance suffers
- **Rebuild mode**
  - Replacement disk is being populated with data
  - replaced failed disk, rebuilding from parity

Data Striping (RAID 0)

- **User data is interleaved across a group of drives**
- **Improves performance by distributing the load among the drives**
  - parallel access, more bandwidth
- **No protection against disk failures (misnomer)**
  - silly to call it RAID



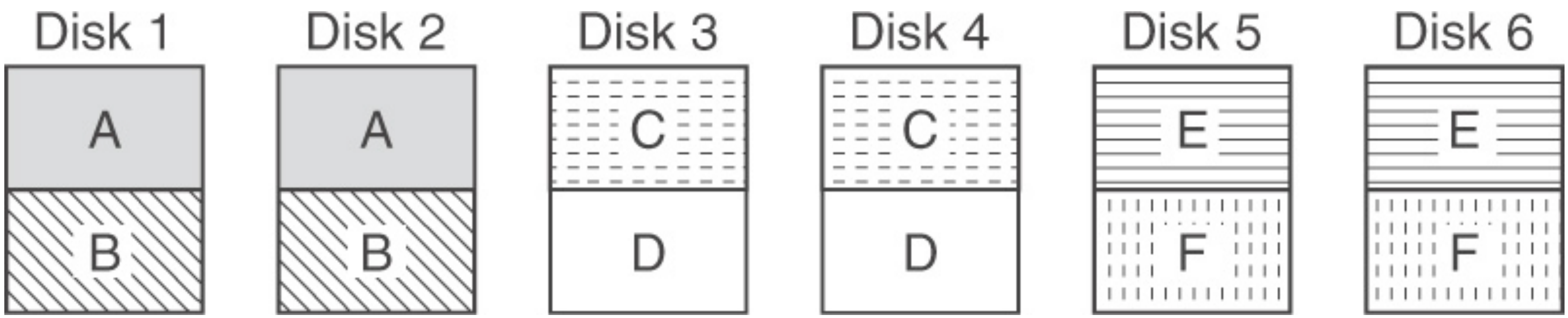
[24.1]

Data Mirroring

- Two copies of the same data are kept on two different drives
  - or more
- Data still available in the case of a disk failure
  - if only 1 of 2 fails.
- Writes must be performed to both drives
  - done in parallel
- Reads can be performed to one or both drives
  - parallelism

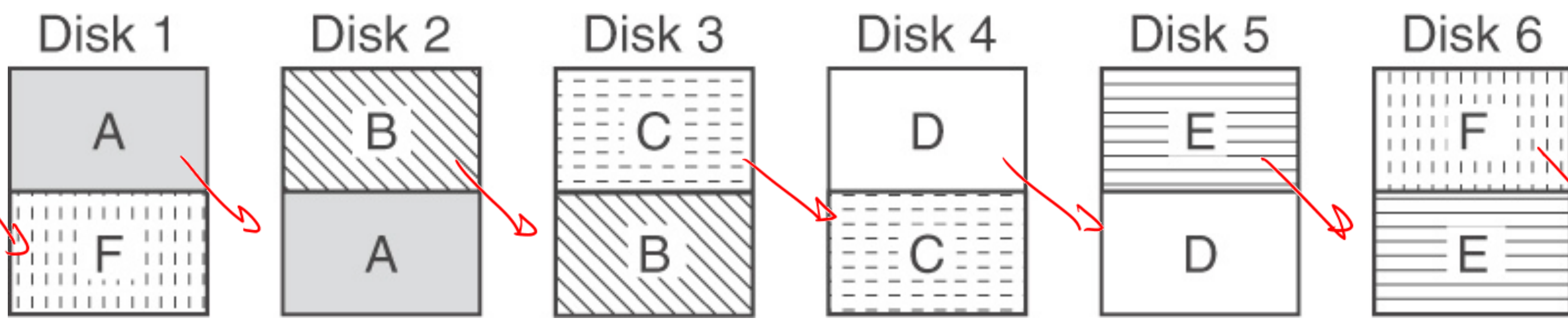
Mirroring Alternatives

- **Basic mirroring (RAID 1)** → straight duplication in drive pairs



↑ disk 2 goes down, only disk 1 has copies

- **Chained decluster mirroring** → offset by 1

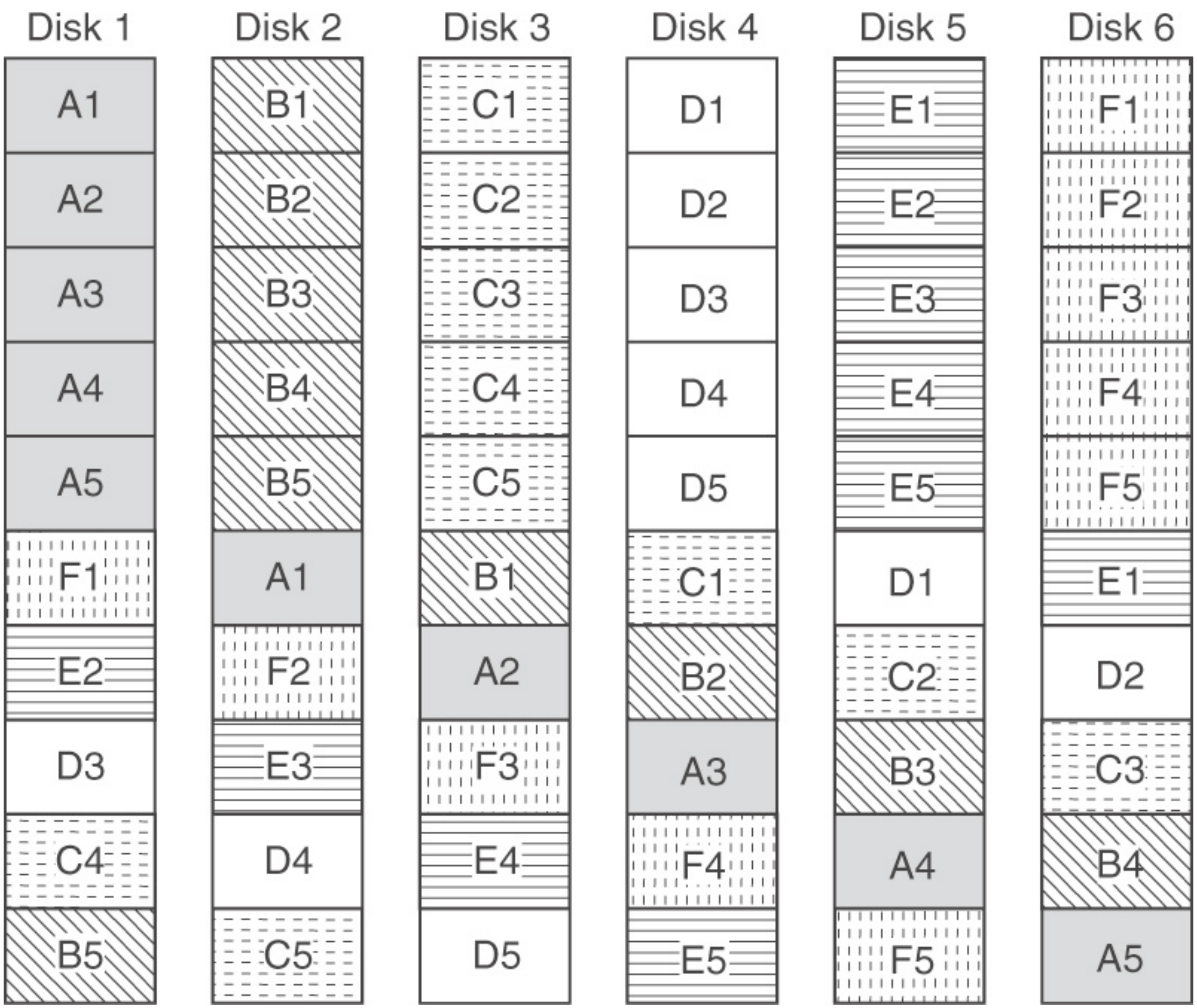


↑ if disk 2 goes down, disk 1 and 3 have the data, spreading out the data locations for reliability

[24.2,24.3]

Mirroring Alternatives

- **Interleaved decluster mirroring**



[24.4]

if we lose disk 2, B is distributed across all drives

How Do We Handle Reads?

- **Send reads to both drives**

  - whichever comes back first wins

- **Alternate reads between the two drives**

  - load balancing

- **Send reads for 1/2 address space to one drive, and the other 1/2 to the other drive**
  - **Reduces average seek distance for cylinder format**
  - **May create load imbalance**

    - may have heavy access on 1/2 of the address space

- **Maintain a single queue and schedule the next read based on which drive is available**

  - schedule accesses (more complexity (overhead))

Mirroring Performance

- **Load balance in normal mode**
  - **Basic the worst, interleaved decluster the best**

2 drives only

full distribution

- **Degraded operation**
  - **Basic: one drive has to carry the load**
  - **Chained decluster: two drives carry the load, and controller can offset load even further**
  - **Interleaved decluster: N-1 drives to carry the load**

- **Rebuild**
  - **Similar to degraded**

Mirroring Reliability

- Tolerates a single drive failure among N drives

- Mean time to data loss (MTTDL) depends on
  - Mean time to failure (MTTF) of each drive → doesn't change
  - Mean time to repair (MTTR) a failed drive → how long to replace / rebuild
  - Type of mirroring

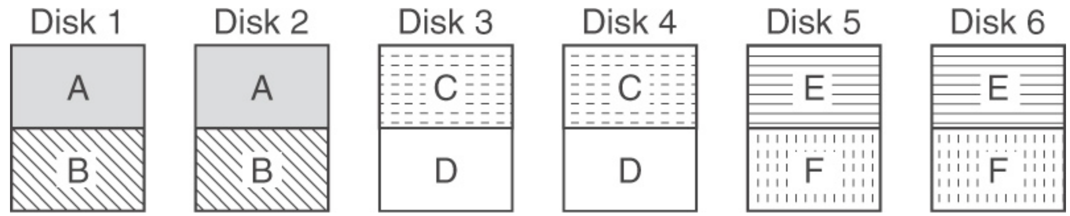
- Basic MTTF calculations (assuming CFR)
  - MTTF of one drive = MTTF
  - MTTF of M drives = MTTF/M
  - Mean time to one failure in M drives, then another specific drive failing = (MTTF/M)×(MTTF/MTTR)

Mirroring Reliability

- **Basic**

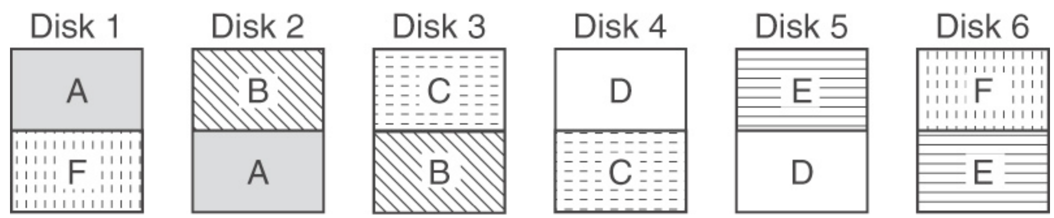
mean time to data loss

$$MTTDL = \frac{MTTF^2}{M \times MTTR}$$



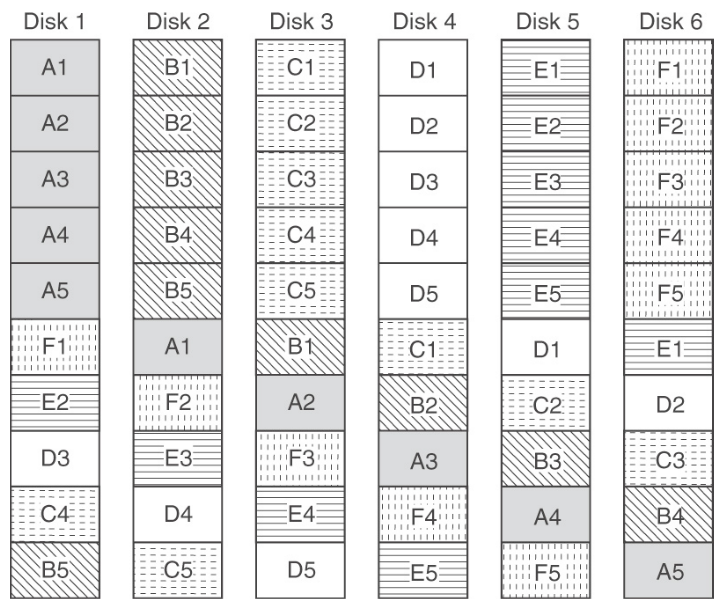
- **Chained decluster**

$$MTTDL = \frac{MTTF^2}{2M \times MTTR}$$



- **Interleaved decluster**

$$MTTDL = \frac{MTTF^2}{M \times (M-1) \times MTTR}$$



[24.2,24.3,24.4]

Bit-Level Striping + ECC (RAID 2)

- Each disk holds one of the data + ECC bits
- Example
  - (7,4) ECC code has 4 data bits + 3 ECC bits
  - 4 data drives and 3 redundant (ECC) drives
- ECC generated on writes, checked on reads

  and correct
- Provides online correction of single drive error
  - detect 2-bit errors, fix 1-bit error
- Not used due to lower cost alternatives
  - proposed in a paper, never used

Byte-Level Striping + Parity (RAID 3)

- Striped data drives + 1 parity drive
- Parity generated on writes, checked on reads
- Since faulty drive is known, can correct error from parity information
  - Written info: $P = A \oplus B \oplus C \oplus D \overset{\text{xor}}{\oplus} E$ ← 5 data drives
  - If B bad, can recover: $B = P \oplus A \oplus C \oplus D \oplus E$
- Good for sequential, bad for random accesses

Block-Level Striping + Parity (RAID 4)

- Striping unit = one or more sectors
- Parity generated on writes, checked on reads
- Can perform writes to individual drives
  - Written info: $P = A \oplus B \oplus C \oplus D \oplus E$ ← xor ← again can reconstruct w/ parity
  - If only C is to be written: $P' = P \oplus C \oplus C'$

new old new

only have to read old values of P and C

- Parity drive is a bottleneck

Block Level

BL Striping + Dist Parity (RAID 5)

- **Parity is distributed evenly among the drives**

Disk 1	Disk 2	Disk 3	Disk 4	Disk 5	Disk 6
$D_{1}1$	$D_{1}2$	$D_{1}3$	$D_{1}4$	$D_{1}5$	P_1
$D_{2}1$	$D_{2}2$	$D_{2}3$	$D_{2}4$	P_2	$D_{2}6$
$D_{3}1$	$D_{3}2$	$D_{3}3$	P_3	$D_{3}5$	$D_{3}6$
$D_{4}1$	$D_{4}2$	P_4	$D_{4}4$	$D_{4}5$	$D_{4}6$
$D_{5}1$	P_5	$D_{5}3$	$D_{5}4$	$D_{5}5$	$D_{5}6$
P_6	$D_{6}2$	$D_{6}3$	$D_{6}4$	$D_{6}5$	$D_{6}6$

parity for drive 6

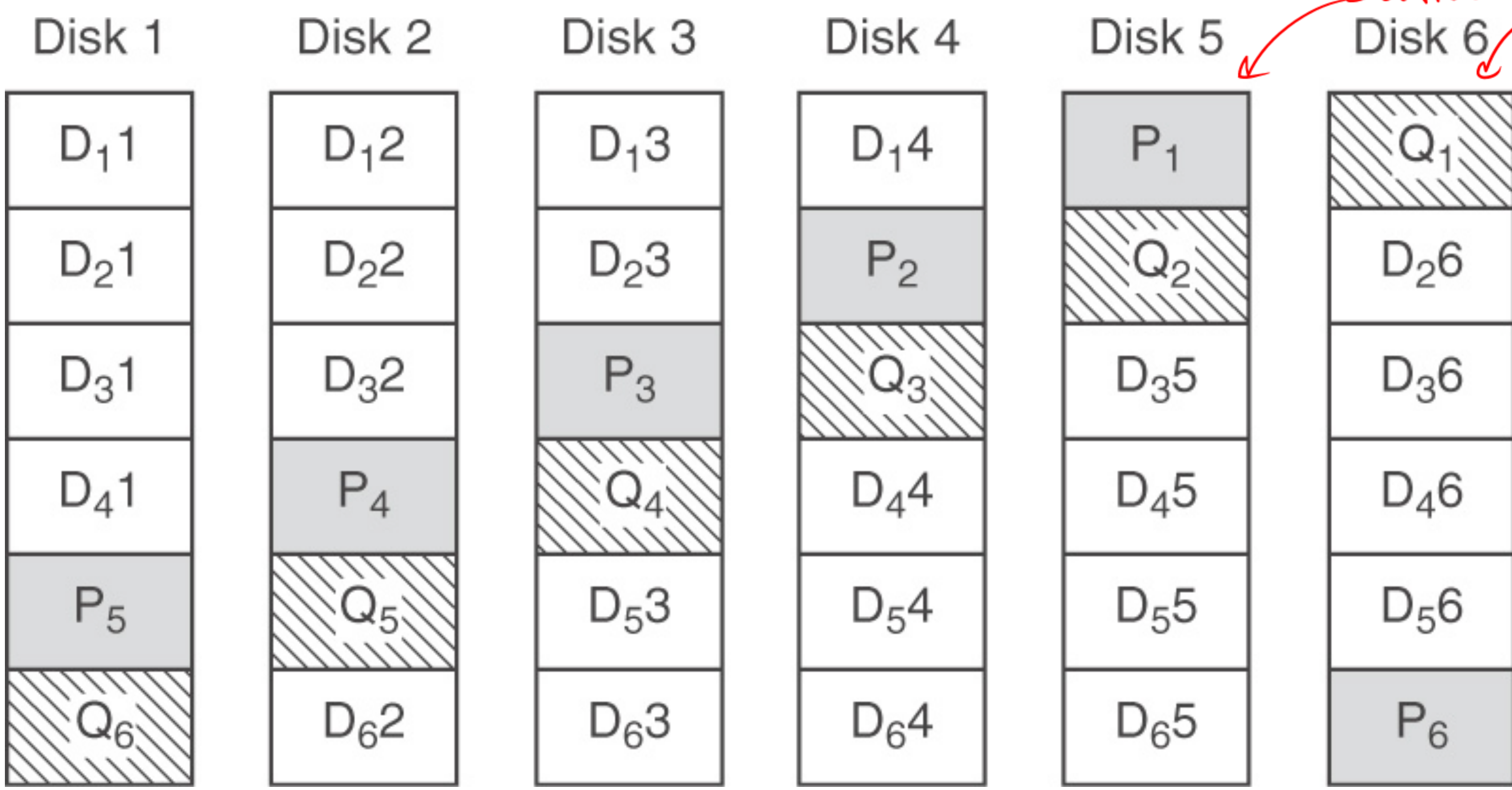
- **Eliminates parity drive bottleneck**

[24.6] - better performance

BL Striping + 2 Dist Parity (RAID 6)

- **Corrects two errors with a 2nd parity calculation**

2 different parities for disk 1



- **Guards against a 2nd error during rebuild**

[24.8] – can protect against second drive failure during rebuild

Normal Mode Performance

- Reads
  - Mirrored performs the best
  - RAID 3 good for large sequential requests
  - RAID 5/6 same as JBODs (Just a Bunch of Disks)

- Writes
  - Mirrored requires same data be written to both disks
  - RAID 3 performs same as for reads
  - RAID 5 requires 2 RMWs for small writes
  - RAID 6 requires 3 RMWs for small writes

Read Modify write

⇒ read 2 drives to get parity modify

then write

Normal Mode Performance

- **Assume random workload of R reads and W small writes with N data disks**

	JBOD	Mirroring	RAID-3	RAID-5	RAID-6
Total number of disks	N	2N	N + 1	N + 1	N + 2
Workload per disk	(R + W)/N	(R + 2W)/2N	R + W	(R + 2RMW)/(N + 1)	(R + 3RMW)/(N + 2)

[24.1]

Degraded Mode Performance

- **Reads from failed drive**
  - Mirroring: discussed earlier
  - RAID 3: data reconstructed on the fly
  - RAID 5: have to read every disk to reconstruct data
  - RAID 6: have to read N+1 disks to reconstruct single error, N+2 disks for double

- **Writes to failed drive**
  - Mirroring: single write to good drive
  - RAID 3: writes done as usual
  - RAID 5/6: have to read every disk to create parity for every write

Rebuild Mode Performance

- Mirroring: data copied from disk(s)

- RAID 5/6: have to read entire content of all disks
  - Done in small chunks to allow servicing user requests

RAID 3/5/6 Reliability

- **RAID 3/5**

$$MTTDL = \frac{MTTF^2}{(N+1) \times N \times MTTR}$$

longer than
for mirroring

- **RAID 6**

$$MTTDL = \frac{MTTF^3}{(N+2) \times (N+1) \times N \times MTTR^2}$$

number of
data disks

RAID Reliability Comparison

- Assume
  - MTTF = 1 million hours
  - $MTTR_{mirror}$ = 1 hour
  - $MTTR_{RAID\ 5/6}$ = 4 hours

way better as it can tolerate 2x errors

	Basic Mirroring (RAID-1)	Chained Decluster Mirroring	Interleave Decluster Mirroring	RAID-4/5	RAID-6
Equal total number of disks = 8	1.25×10^5 (4 data disks)	6.25×10^4 (4 data disks)	1.79×10^4 (4 data disks)	4.46×10^3 (7 data disks)	1.86×10^8 (6 data disks)
Equal number of data disks = 4	1.25×10^5 (8 total disks)	6.25×10^4 (8 total disks)	1.79×10^4 (8 total disks)	1.25×10^4 (5 total disks)	5.21×10^8 (6 total disks)

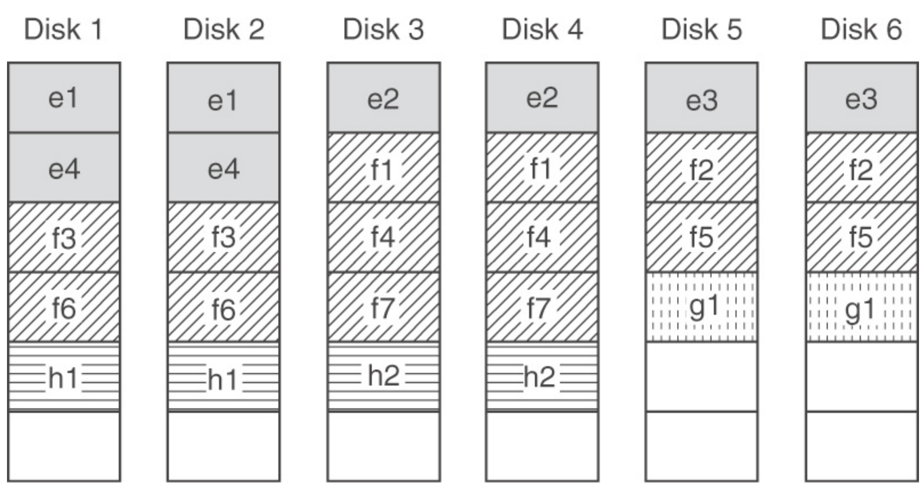
- BUT, a lot more overhead with mirroring

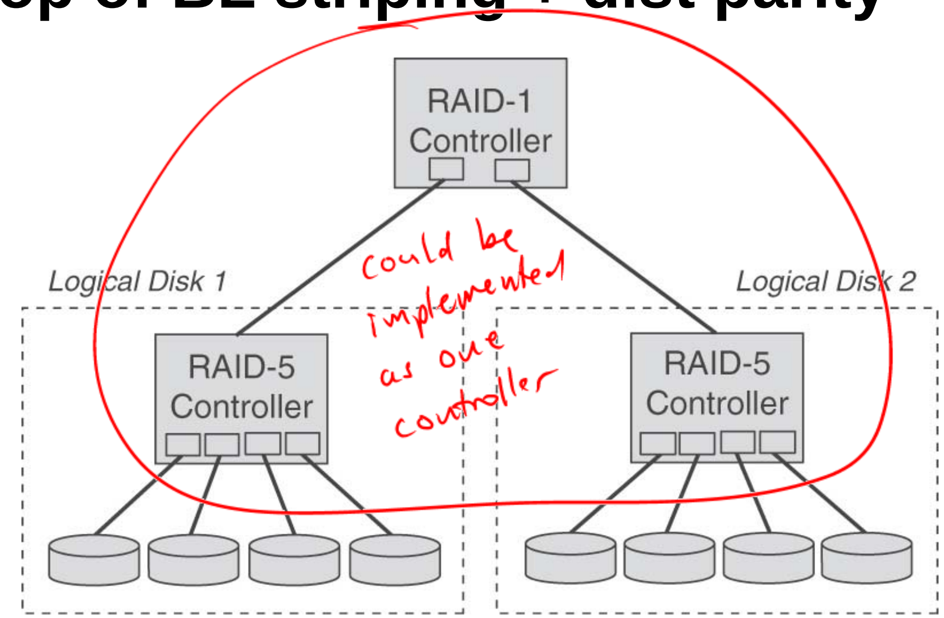
[24.2]

Hierarchical RAID

- **Different RAID levels can be combined**

- **Examples**
  - **RAID 01: striping on top of mirroring**
  - **RAID 10: mirroring on top of striping**
  - **RAID 15: mirroring on top of BL striping + dist parity**





[24.5,24.12]

Next Time

Disk Case Study

Disk Power Management