

RAID

ECE595
Nov 29

Y. Charlie Hu



Logic 101

- $0 \oplus 0 =$
- $0 \oplus 1 =$
- $1 \oplus 0 =$
- $1 \oplus 1 =$
- $0 \oplus a =$
- $b \oplus b =$



2

[lec26] Observations

- Getting first byte from disk read is slow
 - Access time + seek time + rotational delay + read time
- Peak disk bandwidth good, but rarely achieved
- Towards mitigate disk performance impact
 - Move some disk data into main memory – file caching
 - Do extra calculations to speed up disk access
 - There are often multiple disk requests outstanding
 - *Schedule requests to shorten seeks!*

3

Review: Disk Scheduling

- Problem statement:
 - Given the mapping of 1-D array of logical blocks to the sectors of disk, and disk requests keep arriving, schedules disk requests currently in the queue to maximize the disk I/O **throughput**
 - *Simplification:* the OS knows above which cylinder the disk head is, but not which sector
- Disk scheduling vs. CPU scheduling
- Disk scheduling algorithms
 - FCFS, SSTF, SCAN, C-SCAN
 - Why is SSTF not optimal?

4



Practice questions



- Does SSTF scheduling tend to favor middle cylinders or the innermost/outmost cylinders?
- Why is rotational delay often not considered in disk scheduling?
 - Disk do not export rotational position
 - Even if they did, the delay makes it imprecise
 - Adding disk head movement, even more imprecise

5

Practice questions



- Requests are not usually uniformly distributed. For example, a cylinder containing the file system FAT or inodes can be expected to be accessed more frequently than a cylinder that contains only files.

Suppose you know that 50 percent of the requests are for a small, fixed number of cylinders, would any of the scheduling algorithms we discussed be particularly good for this case? Explain your answer.

6

[Ice26] Observations



- Getting first byte from disk read is slow
 - high **latency**
- Peak disk bandwidth good, but rarely achieved
- Towards mitigate disk performance impact
 - Move some disk data into main memory – file caching
 - Disk scheduling
 - There are often multiple disk requests outstanding
 - Schedule requests to shorten seeks!
 - **What else can we try?**
 - **Adding multiple disk arms to the disk?**

7

A startup company idea?



- **Random block access time = seek time + rotational delay + reading time**

- Performance of transfer 1 Kbytes (7200 RPM)
 - Seek (5.3 ms) + half rotational delay (4.15ms) + transfer (0.04 ms)
 - Total time is 8.34ms → **120 Kbytes/sec!**

- If you have double degrees in EE and ME, can you think of a revolutionary idea?
- Do you think you will get investors excited?

8

Exploiting parallelism?

- Goal: improve performance under heavy load
- Suitability
 - do we have parallelism semantically?
- Feasibility
 - how to realize parallelism?

9

RAID (1987, UC Berkeley)

- Two motivations
 - (in the past) Operating in parallel can increase disk throughput
 - RAID = Redundant Array of Inexpensive Disks
 - (today) Redundancy can increase reliability
 - RAID = Redundant Array of Independent Disks

10

RAID -- Two main ideas (1)

- Parallel reading (for performance)
 - Splitting bits of a byte across multiple disks
 - 8 disks (bit-level striping)
 - Logically acts like single disk with sector size $X \cdot 8$ and transfer time $/ 8$
 - Reduce the transfer time of large access (e.g. 1 4K block)
 - Alternatively, block-level striping
 - Increases the throughput of multiple small accesses (e.g. 8 512-byte blocks)

11

RAID -- Two main ideas (2)

- Mirroring or shadowing (for reliability)
 - Logical disk consists of 2 physical disks in parallel
 - Every write performed on both disks
 - Can read from either disk
 - Probability of both fail at the same time?

12

RAID – combining the two ideas!

- Mirroring gives reliability, but expensive
- Striping gives high data-transfer rate, but not reliability
- Challenge: can you provide redundancy at low cost?

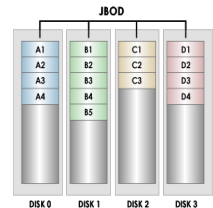
13

JBOD (non-RAID architecture)

- Definition: concatenation of disks
 - E.g. combine 3GB, 15GB, 5.5GB drives into 23.5GB logical drive
 - Opposite of disk partitioning
 - Also uses an array of independent disks

- No performance advantage
- Failure of one disk?

- Use:
 - convenience



Synopsis of RAID Levels



RAID Level 0: Not redundant
Block-level striping

- Performance?
 - Read/write?
- If one disk fails?
 - FS MTTF?
- Use
 - Large read-only NFS servers (quick mounting)
 - Create a small number of large logical disks
 - Microsoft windows servers: # of drive letters ≤ 24

15

Synopsis of RAID Levels



RAID Level 0: Not redundant
Block-level striping

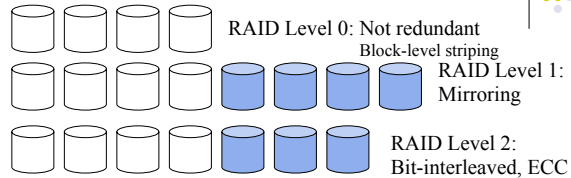


RAID Level 1:
Mirroring

- Performance? (Independent controllers)
 - Read?
 - Seek time?
- Reliability (failure probability) ?
- Interesting Use
 - Online snapshot (split the mirror and backup)

16

Synopsis of RAID Levels



ECC can detect/correct which bit went wrong

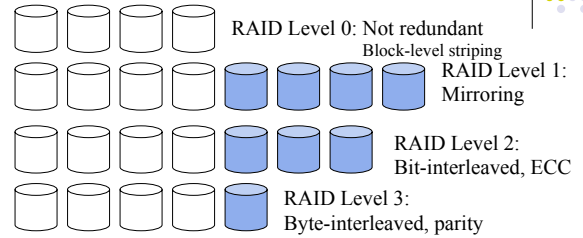
Low-level format of modern drives already contains ECC → controller can catch errors in a sector

Level 2 is never used.

Instead,

17

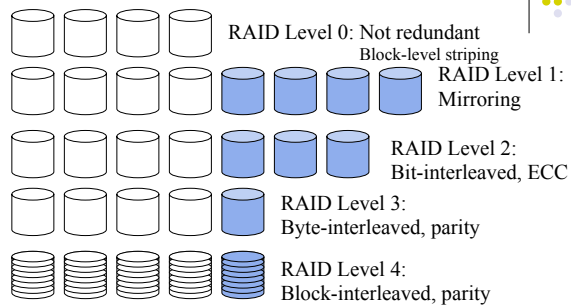
Synopsis of RAID Levels



Compared to RAID Level 1:
High transfer rate for each block access
A req involves all drives + seek/rotational delay → cannot service multiple requests simultaneously
Computing/writing parity expensive → Dedicated parity hardware (offload CPU)

18

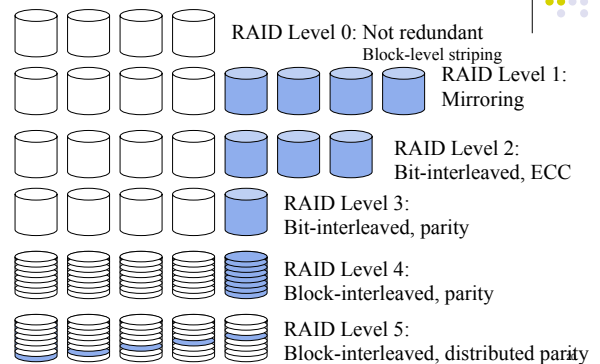
Synopsis of RAID Levels



Lower transfer rate for each block (by single disk)
Higher overall rate (many small files, or a large file)
Large writes → parity blocks can be written in parallel
Small writes → 2 reads / 2 writes !

19

Synopsis of RAID Levels

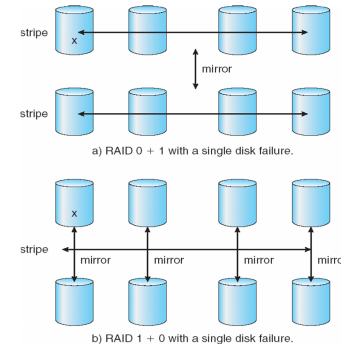


Later additions

- RAID 6: RAID 5 + additional parity block
- Nested RAID levels
 - RAID 01 (0+1): mirrored RAID 0 (stripe)
 - RAID 10 (1+0): stripe of RAID 1 (mirror)
 - RAID 50 (5+0): stripe of RAID 5
 - RAID 100 (10+0): stripe of RAID 10
- Proprietary RAID levels
 - Double parity
 - RAID 1.5
 - RAID 7
 - ...

21

RAID (0 + 1) and (1 + 0)



RAID Implementation

- Typically in hardware
 - Special-purpose RAID controller (PCI card)
 - Manages disks
 - Performs parity calculation
- Can be in software (by OS)
 - Can be fast
 - At the cost of CPU time

23

Practice question

- Consider a RAID Level 5 organization comprising five disks, with the parity for sets of four blocks on four disks stored on the fifth disk. How many blocks are accessed in order to perform the following?
 - A write of one block of data
 - A write of seven continuous blocks of data

24

Practice question



- True-or-False:

It is faster to write 1 block in a RAID Level 5 organization (with 4+1 disks) than in a RAID Level 1 organization (with 2 disks)

25

Practice question



- Assume that

- you have a mixed configuration comprising disks organized as RAID Level 1 and as RAID Level 5 disks;
- the system has flexibility in deciding which disk organization to use for storing a particular file.
- you have a mixed workload of *frequently-read* and *frequently-written* files (implying small updates)

Q: Which files should be stored in the RAID Level 1 disks and which in the RAID Level 5 disks in order to optimize performance?

26

Reading



- Chapter 12

27