

COS 318: Operating Systems

Disks

Kai Li
Computer Science Department
Princeton University



(<http://www.cs.princeton.edu/courses/cos318/>)

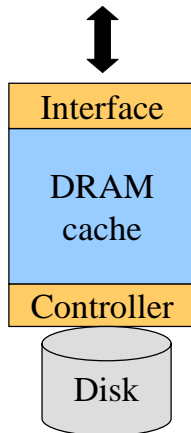
Today's Topics

- ◆ Disk controller
- ◆ Disk layout
- ◆ Technology trend
- ◆ Disk performance
- ◆ Disk scheduling
- ◆ RAID

A Typical Disk Controller

- ◆ External connection
 - IDE/ATA, SATA
 - SCSI, SCSI-2, Ultra SCSI, Ultra-160 SCSI, Ultra-320 SCSI
 - Fibre channel
- ◆ Cache
 - Buffer data between disk and interface
- ◆ Controller
 - Read/write operation
 - Cache replacement
 - Failure detection and recovery

External connection

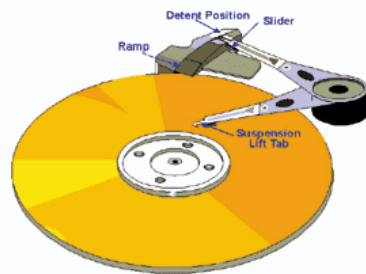


Disk Caching

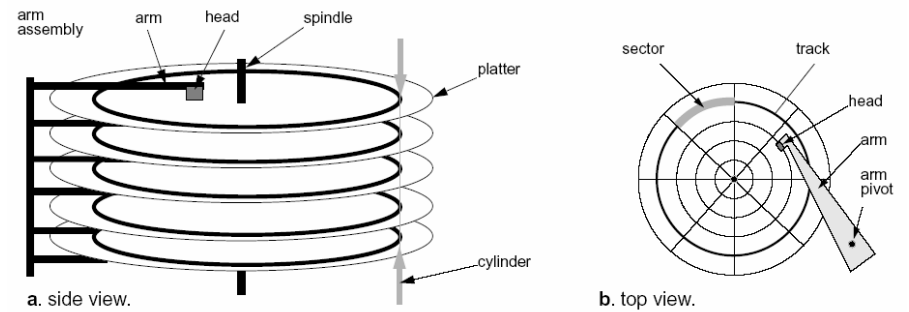
- ◆ Method
 - Use DRAM to cache recently accessed blocks
 - Most disk has 16MB
 - Some of the RAM space stores “firmware” (an embedded OS)
 - Blocks are replaced usually in an LRU order
- ◆ Pros
 - Good for reads if accesses have locality
- ◆ Cons
 - Cost
 - Need to deal with reliable writes

Disk Arm and Head

- ◆ Disk arm
 - A disk arm carries disk heads
- ◆ Disk head
 - Mounted on an actuator
 - Read and write on disk surface
- ◆ Read/write operation
 - Disk controller receives a command with <track#, sector#>
 - Seek the right cylinder (tracks)
 - Wait until the right sector comes
 - Perform read/write



Mechanical Component of A Disk Drive

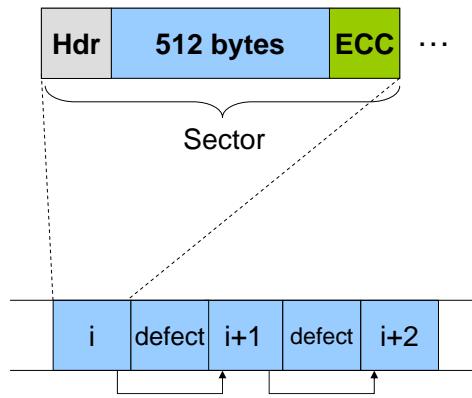


- ◆ Tracks
 - Concentric rings around disk surface, bits laid out serially along each track
- ◆ Cylinder
 - A track of the platter, 1000-5000 cylinders per zone, 1 spare per zone
- ◆ Sectors
 - Each track is split into arc of track (min unit of transfer)



Disk Sectors

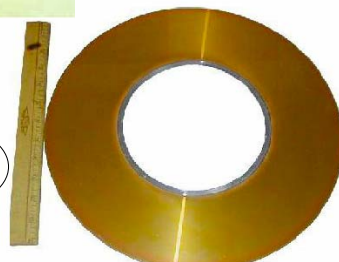
- ◆ Where do they come from?
 - Formatting process
 - Logical maps to physical
- ◆ What is a sector?
 - Header (ID, defect flag, ...)
 - Real space (e.g. 512 bytes)
 - Trailer (ECC code)
- ◆ What about errors?
 - Detect errors in a sector
 - Correct them with ECC
 - If not recoverable, replace it with a spare
 - Skip bad sectors in the future



Disks Were Large



First Disk:
IBM 305 RAMAC (1956)
5MB capacity
50 disks, each 24"



Today's Disks Are Smaller



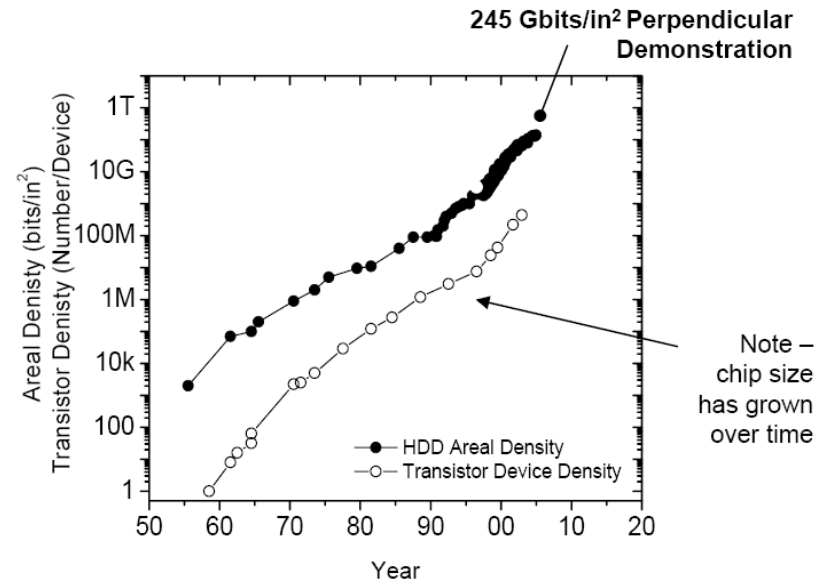
Form factor:
.5-1" x 4" x 5.7"
Storage:
120-750GB

Form factor:
.4-.7" x 2.7" x 3.9"
Storage:
60-200GB

Form factor:
.2-.4" x 2.1" x 3.4"
Storage:
1GB-8GB



Areal Density vs. Moore's Law



(Mark Kryder at SNW 2006)

50 Years Later (Mark Kryder at SNW 2006)

	IBM RAMAC (1956)	Seagate Momentus (2006)	Difference
Capacity	5MB	160GB	32,000
Areal Density	2K bits/in ²	130 Gbits/in ²	65,000,000
Disks	50 @ 24" diameter	2 @ 2.5" diameter	1 / 2,300
Price/MB	\$1,000	\$0.01	1 / 3,200,000
Spindle Speed	1,200 RPM	5,400 RPM	5
Seek Time	600 ms	10 ms	1 / 60
Data Rate	10 KB/s	44 MB/s	4,400
Power	5000 W	2 W	1 / 2,500
Weight	~ 1 ton	4 oz	1 / 9,000



Sample Disk Specs (from Seagate)

	Cheetah 15k.5	Barracuda ES
Capacity		
Formatted capacity (GB)	300	750
Discs	4	4
Heads	8	8
Sector size (bytes)	512	512
Performance		
External interface	Ultra320 SCSI, FC, S. SCSI	SATA
Spindle speed (RPM)	15,000	7,200
Average latency (msec)	2.0	4.16
Seek time, read/write (ms)	3.5/4.0	8.5/9.5
Track-to-track read/write (ms)		0.8/1.0
External transfer (MB/sec)	300-400	350
Max transfer (MB/sec)	150	120
Max sustained transfer (MB/sec)	125	78
Cache size (MB)	16	16
Reliability		
Recoverable read errors	1 per 10 ¹² bits read	
Non-recoverable read errors	1 per 10 ¹⁶ bits read	1 sector per 10 ¹⁴



Disk Performance

- ◆ Seek
 - Position heads over cylinder, typically 3.5-9.5 ms
- ◆ Rotational delay
 - Wait for a sector to rotate underneath the heads
 - Typically 8 – 4 ms (7,200 – 15,000RPM) or ½ rotation takes 4 - 2ms
- ◆ Transfer bytes
 - Transfer bandwidth is typically 30-125 Mbytes/sec
- ◆ Performance of transfer 1 Kbytes
 - Seek (4 ms) + half rotational delay (2ms) + transfer (0.017 ms)
 - Total time is 6.02 ms or 167 Kbytes/sec (1/360 of 60MB/sec)!



More on Performance

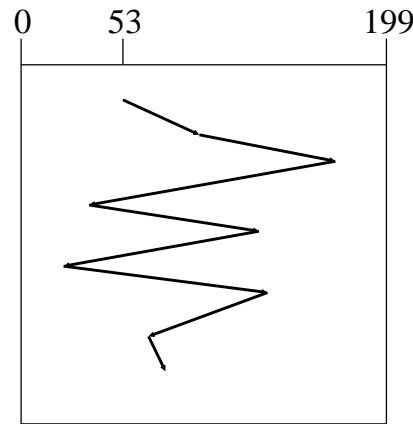
- ◆ Speed depends on sectors
 - Accesses to sectors on outer tracks are faster, why?
 - There are more sectors on outer tracks than inner tracks
- ◆ What block size can get 90% of the disk transfer bandwidth?
 - Use previous assumptions
 - $B = (54MB * 0.006) / 0.1 = 3.24MB$
- ◆ Seek time and rotational latency dominate the cost of small reads
 - Disk transfer bandwidth are wasted
 - Need algorithms to reduce seek time

Block Size (Kbytes)	% of Disk Transfer Bandwidth
1Kbytes	0.28%
8Kbytes	2.17%
256Kbytes	41.56%
1Mbytes	73.99%
2Mbytes	85.05%
3.24Mbytes	90%



FIFO (FCFS) order

- ◆ Method
 - First come first serve
- ◆ Pros
 - Fairness among requests
 - In the order applications expect
- ◆ Cons
 - Arrival may be on random spots on the disk (long seeks)
 - Wild swing can happen

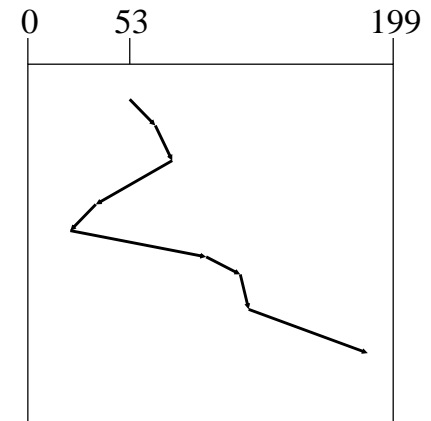


98, 183, 37, 122, 14, 124, 65, 67



SSTF (Shortest Seek Time First)

- ◆ Method
 - Pick the one closest on disk
 - Rotational delay is in calculation
- ◆ Pros
 - Try to minimize seek time
- ◆ Cons
 - Starvation
- ◆ Question
 - Is SSTF optimal?
 - Can we avoid the starvation?



98, 183, 37, 122, 14, 124, 65, 67
(65, 67, 37, 14, 98, 122, 124, 183)



Elevator (SCAN)

Method

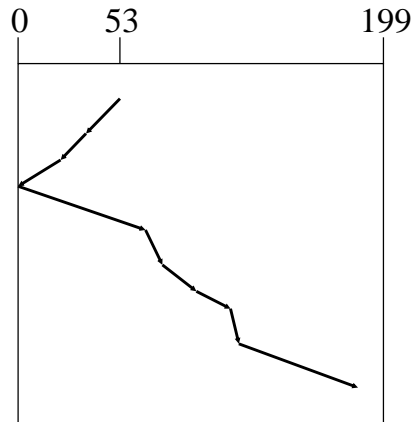
- Take the closest request in the direction of travel
- Real implementations do not go to the end (called LOOK)

Pros

- Bounded time for each request

Cons

- Request at the other end will take a while



98, 183, 37, 122, 14, 124, 65, 67
(37, 14, 65, 67, 98, 122, 124, 183)

17

C-SCAN (Circular SCAN)

Method

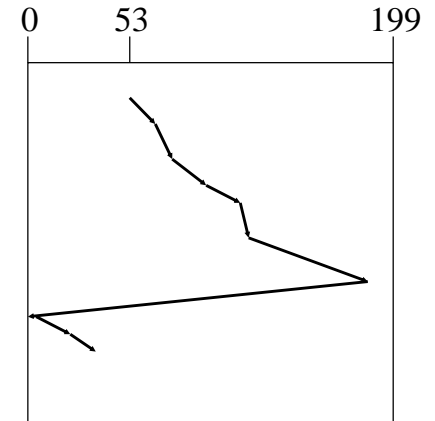
- Like SCAN
- But, wrap around
- Real implementation doesn't go to the end (C-LOOK)

Pros

- Uniform service time

Cons

- Do nothing on the return



98, 183, 37, 122, 14, 124, 65, 67
(65, 67, 98, 122, 124, 183, 14, 37)

18

Discussions

Which is your favorite?

- FIFO
- SSTF
- SCAN
- C-SCAN

Disk I/O request buffering

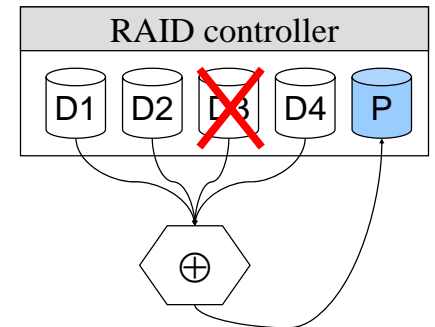
- Where would you buffer requests?
- How long would you buffer requests?

19

RAID (Redundant Array of Independent Disks)

Main idea

- Store the error correcting codes on other disks
- General error correcting codes are too powerful
- Use XORs or single parity
- Upon any failure, one can recover the entire block from the spare disk (or any disk) using XORs



$$P = D1 \oplus D2 \oplus D3 \oplus D4$$

$$D3 = D1 \oplus D2 \oplus P \oplus D4$$

Pros

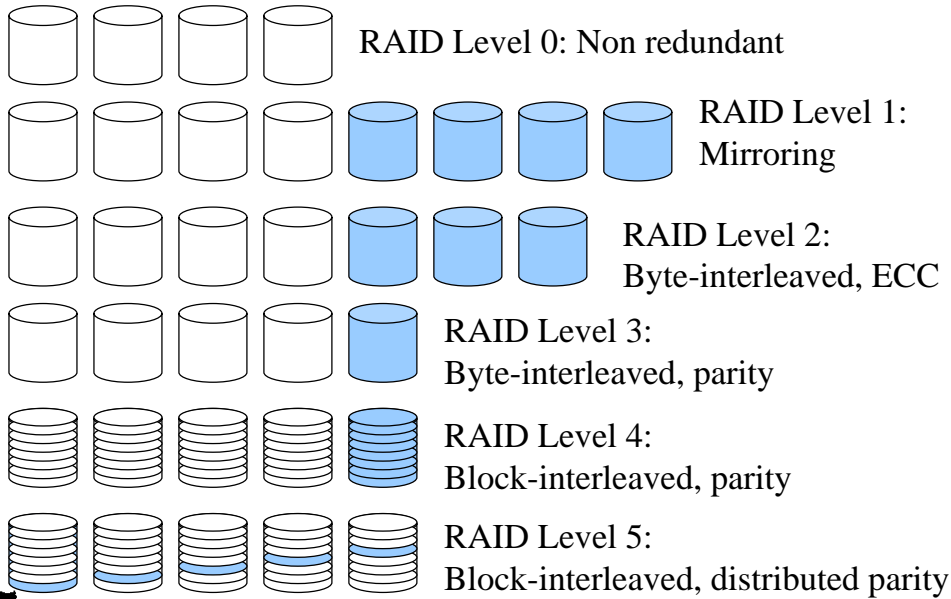
- Reliability
- High bandwidth

Cons

- The controller is complex

20

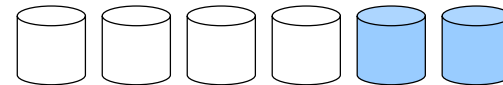
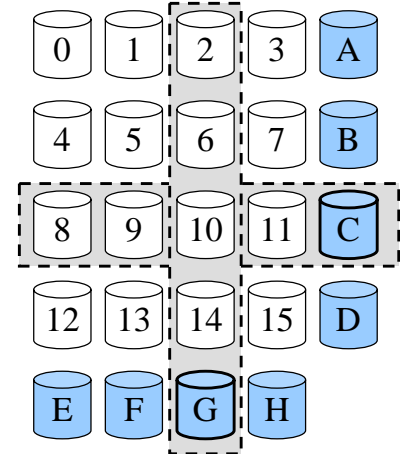
Synopsis of RAID Levels



21

RAID Level 6 and Beyond

- ◆ Goals
 - Less computation and fewer updates per random writes
 - Small amount of extra disk space
- ◆ Extended Hamming code
 - Remember Hamming code?
- ◆ Specialized Eraser Codes
 - IBM Even-Odd, NetApp RAID-DP, ...
- ◆ Beyond RAID-6
 - Reed-Solomon codes, using MOD 4 equations
 - Can be generalized to deal with $k (>2)$ disk failures



22

Dealing with Disk Failures

- ◆ What failures
 - Power failures
 - Disk failures
 - Human failures
- ◆ What mechanisms required
 - NVRAM for power failures
 - Hot swappable capability
 - Monitoring hardware
- ◆ RAID reconstruction
 - Reconstruction during operation
 - What happens if a reconstruction fail?
 - What happens if the OS crashes during a reconstruction

23

Summary

- ◆ Disk is complex
- ◆ Disk real density is on Moore's law curve
- ◆ Need large disk blocks to achieve good throughput
- ◆ OS needs to perform disk scheduling
- ◆ RAID improves reliability and high throughput at a cost
- ◆ Careful designs to deal with disk failures

24