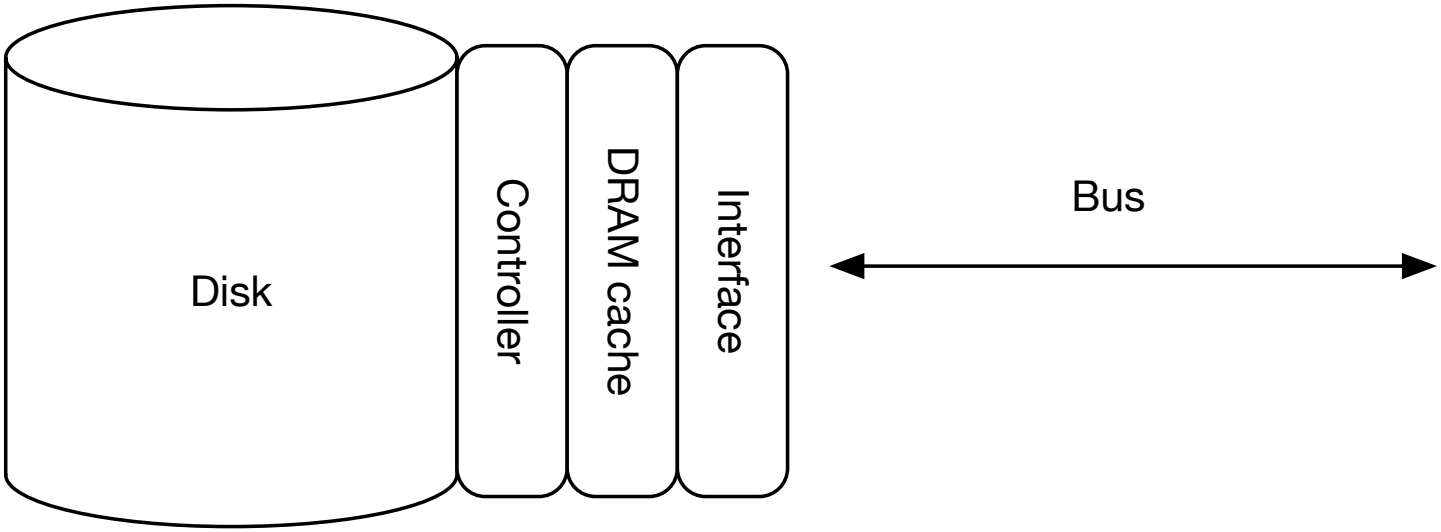
Storage Devices: HDD & Block Request Scheduling

Azza Abouzied

Storage Devices

Components of a storage device

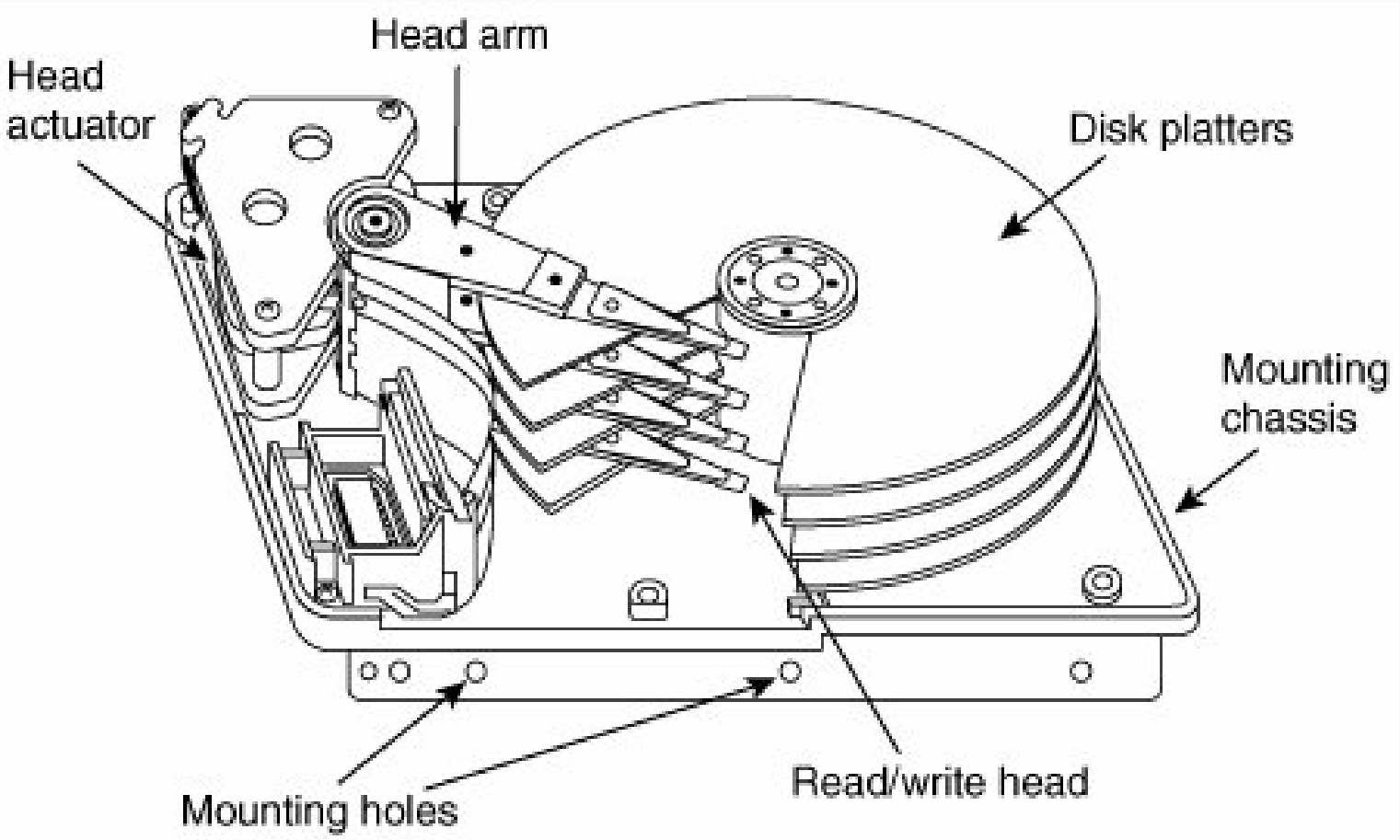
8-64 MB disk cache/buffer
Cache's Block Replacement Algorithm: LRU
Firmware/embedded OS for the controller



Read/Write
Caching
Bad Blocks and Error Correction
Scheduling
Interleaving

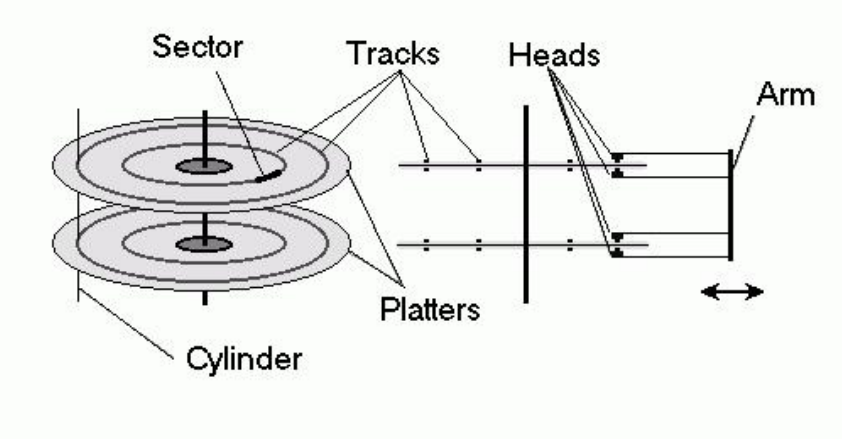
IDE/ATA
SATA
SCSI
Ultra-SCSI
PCI/PCI-e

Hard Disk Drives



There are two heads per platter. As you can write to both sides of a platter.

HDD Terms



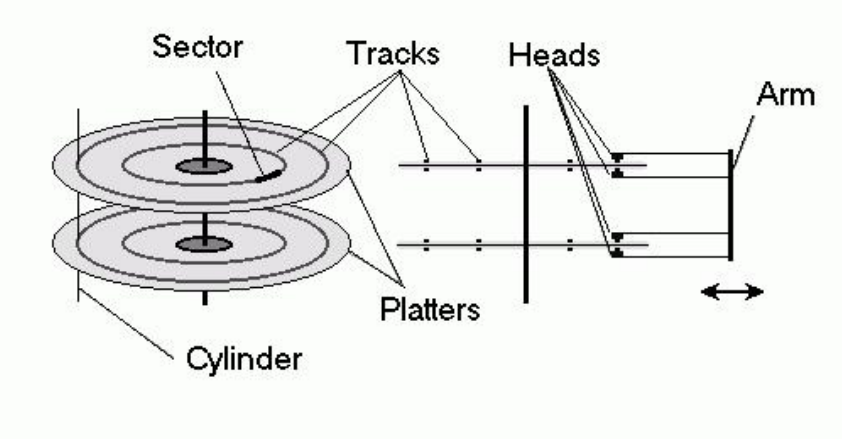
How to read or write from a sector?

1. The disk controller gets a Cylinder-Head-Sector (CHS)[a] address.
2. The arm seeks to the right cylinder.
3. Waits until the sector comes below the head. Reads from / Writes to it.

---

[a]

HDD Terms



How to read or write from a sector?

1. The disk controller gets a Cylinder-Head-Sector (CHS)[a] address.
2. The arm seeks to the right cylinder.
3. Waits until the sector comes below the head. Reads from / Writes to it.

[a] Mostly obsolete, we use Logical Block Addressing (LBA) and the controller does the translation. Why?

Disk Sectors

Disk Formatting organizes the sector

1. Preamble/Header
2. 512 bytes of data
3. ECC $\approx$16 bytes

After formatting the size of the disk is less than the advertised size, also depends on how you define Gigabytes

Each Disk ships with a bunch of spare sectors. If a sector is corrupt, use the spare. Skip bad sectors in the future. Spares are not included in capacity computation

50 Years of Change

	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 / 100,000
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

Disk's Today

	Cheetah 15k.7	Barracuda XT
Capacity		
Formatted capacity (GB)	600	2000
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)	3.5/3.9	8.5/9.5
Track-to-track read/write (ms)	0.2-0.4	0.8/1.0
Internal transfer (MB/sec)	1,450-2,370	600
Transfer rate (MB/sec)	122-204	138
Cache size (MB)	16	64
Reliability		
Recoverable read errors	1 per 10^{12} bits read	1 per 10^{10} bits read
Non-recoverable read errors	1 per 10^{16} bits read	1 per 10^{14} bits read

The Limiting Factor

Why are disks not fast enough?

1. At 7200 RPM, a full rotation is 8ms and expected rotation time is 4ms.

2. The seek cost is 4-10 ms.

3. Our transfer bandwidth is 40-125MB/s

Transfer 1KB

Seek + Rotational Delay + Transfer

4ms + 4ms + 1KB/125MB/s = 8ms + 0.007ms = 8.007ms

Our effective transfer rate is 1KB/8.007ms = 125KB/s = 1/1000 of 125MB/s!

How can we maximize performance?

Can we get an effective transfer rate that is 9/10 of disk bandwidth (bw) instead of 1/1000?

Amortization!

$$bw \times \frac{9}{10} = \frac{size}{size/bw + seek + rotation}$$

$$size = 9 * bw \times (seek + rotation)$$

$$size = 9 * 125MB/s \times (4ms + 4ms) = 9MB$$

FIFO

Assume you have the following track/cylinder requests:
98, 183, 37, 122, 14, 124, 65, 67

Pros:

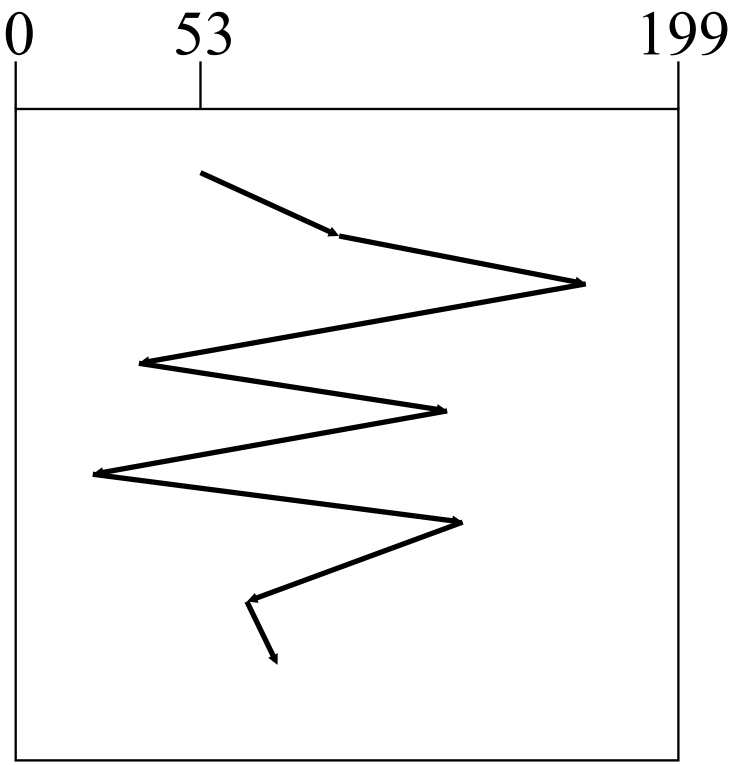
1. Fairness: Blocks arrive in the order requested

Cons:

1. Long seeks

2. Wild swings

How many tracks are visited?
640 tracks!



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

Shortest Seek Time First (SSTF)

Pick track closest on disk to the current head position.

Pros:

1. Minimize seek time

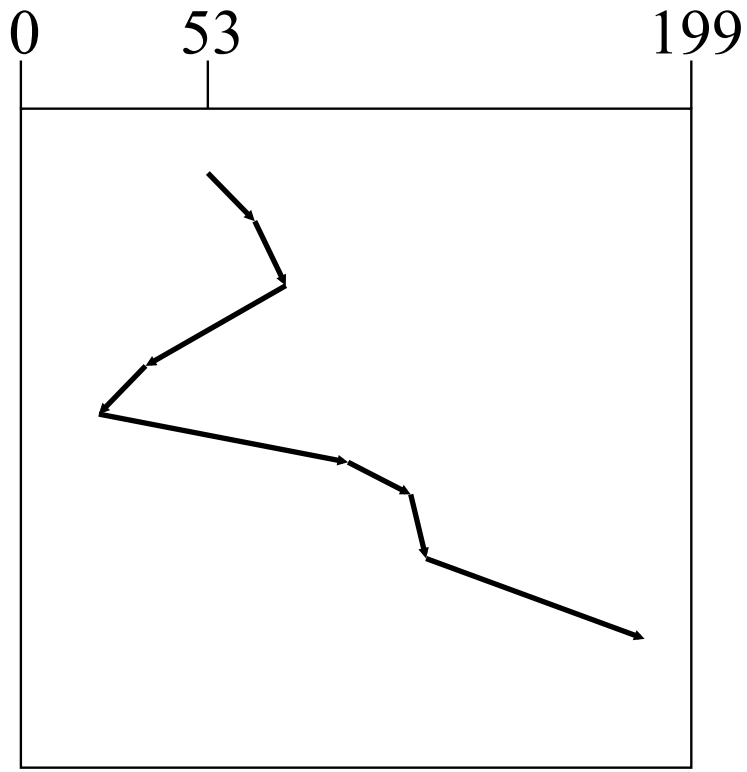
Cons:

1. Starvation

2. Ignore rotation**

How many tracks are visited?

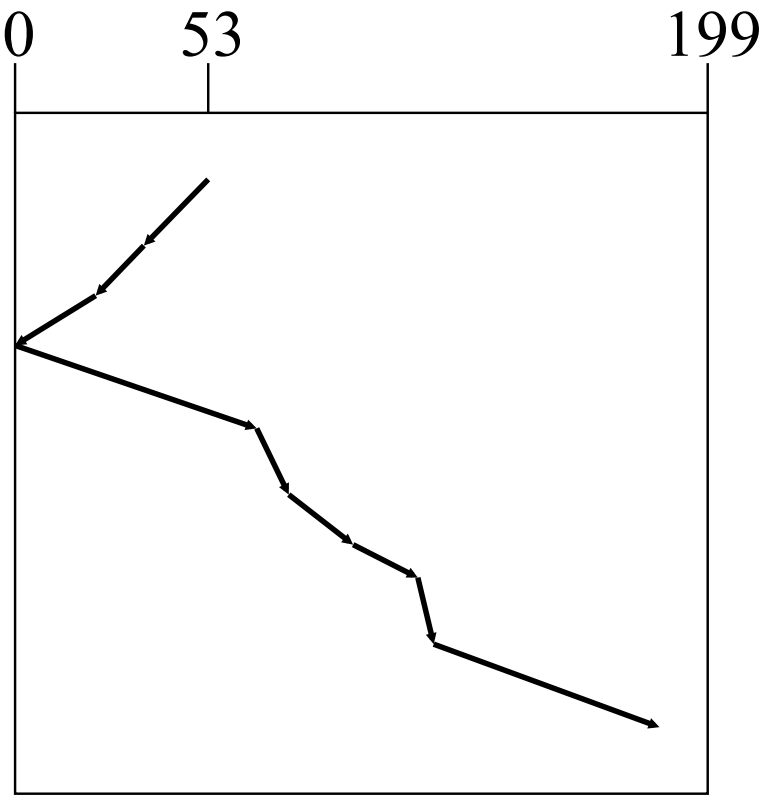
236 tracks



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

Elevator

Pick the closest in the direction of head (So no back and forth head movement)



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

Pros:

1. No Starvation

Cons:

1. Can still do better!

How many tracks are visited?
230 tracks

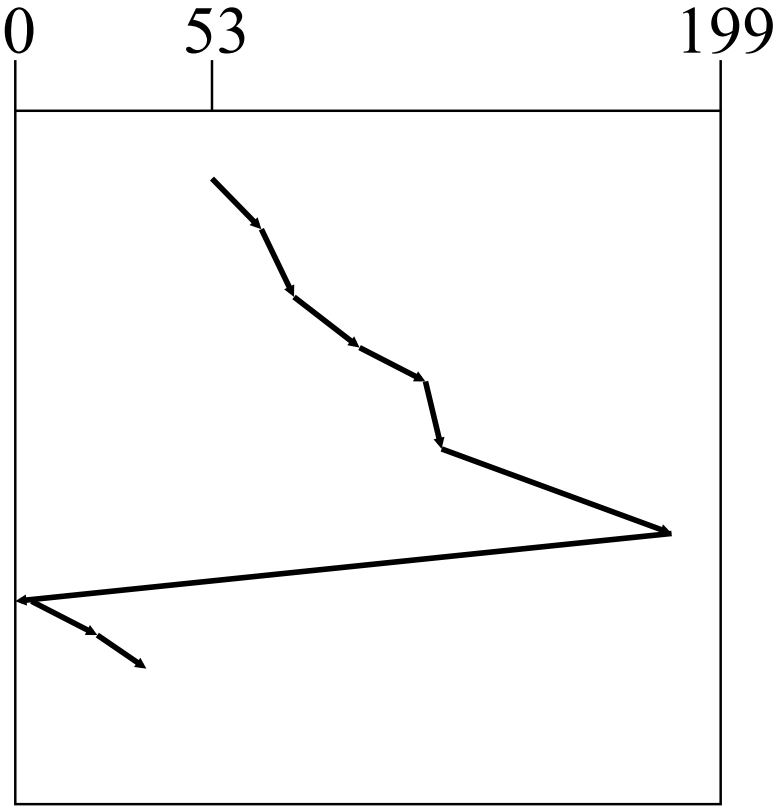
Elevator

Like an elevator, except once it reaches the end it jumps to the other end. Always moves in one direction.

Pros:

1. Uniform Service time

How many tracks are visited?
187 (a jump is not counted as a scan!) An optimization where you jump to the furthest track request instead of track 0 lowers the cost to 157 tracks.



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

Questions?