What’s Inside A Disk Drive?

- Spindle
- Arm
- Actuator
- Platters
- Electronics (including a processor and memory!)
- SCSI connector

Image courtesy of Seagate Technology
Disk Geometry

- Disks consist of **platters**, each with two **surfaces**.
- Each surface consists of concentric rings called **tracks**.
- Each track consists of **sectors** separated by **gaps**.
Disk Geometry (Multiple-Platter View)

- Aligned tracks form a cylinder.
### Disk Capacity

**Capacity**: maximum number of bits that can be stored.
- Vendors express capacity in units of gigabytes (GB), where 1 GB = \(10^9\) Bytes.

**Capacity is determined by these technology factors:**
- **Recording density** (bits/in): number of bits that can be squeezed into a 1 inch segment of a track.
- **Track density** (tracks/in): number of tracks that can be squeezed into a 1 inch radial segment.
- **Areal density** (bits/in\(^2\)): product of recording and track density.

\[
6B = 2^{30}B
\]

\[
10^9 < 2^{30}
\]
Computing Disk Capacity

Capacity = (# bytes/sector) \times (\text{avg.} \ # \ sectors/track) \times \\
(\# \ tracks/surface) \times (\# \ surfaces/platter) \times \\
(\# \ platters/disk)

Example:
- 512 bytes/sector
- 300 sectors/track (on average)
- 20,000 tracks/surface
- 2 surfaces/platter
- 5 platters/disk

\begin{align*}
\text{Capacity} &= 512 \times 300 \times 20000 \times 2 \times 5 \\
&= 30,720,000,000 \\
&= 30.72 \text{ GB}
\end{align*}
Disk Operation (Single-Platter View)

The disk surface spins at a fixed rotational rate

The read/write head is attached to the end of the arm and flies over the disk surface on a thin cushion of air.

By moving radially, the arm can position the read/write head over any track.
Disk Operation (Multi-Platter View)

Read/write heads move in unison from cylinder to cylinder

Spindle

Arm
Disk Structure - top view of single platter

- Surface organized into tracks
- Tracks divided into sectors
Disk Access

Head in position above a track
Disk Access

Rotation is counter-clockwise
Disk Access – Read

About to read blue sector
Disk Access – Read

After BLUE read

After reading blue sector
Disk Access – Read

After **BLUE** read

Red request scheduled next
Disk Access – Seek

After **BLUE** read

Seek for **RED**

Seek to red’s track
Disk Access – Rotational Latency

After **BLUE** read

Seek for **RED**

Rotational latency

Wait for red sector to rotate around
Disk Access – Read

After BLue read  
Seek for RED  
Rotational latency  
After RED read

Complete read of red
Disk Access – Service Time Components

After BLUE read

Seek for RED

Rotational latency

After RED read

Data transfer

Seek

Rotational latency

Data transfer
Disk Access Time

- **Average time to access some target sector approximated by:**
  - $T_{\text{access}} = T_{\text{avg seek}} + T_{\text{avg rotation}} + T_{\text{avg transfer}}$

- **Seek time** ($T_{\text{avg seek}}$)
  - Time to position heads over cylinder containing target sector.
  - Typical $T_{\text{avg seek}}$ is 3—9 ms

- **Rotational latency** ($T_{\text{avg rotation}}$)
  - Time waiting for first bit of target sector to pass under r/w head.
  - $T_{\text{avg rotation}} = \frac{1}{2} \times \frac{1}{\text{RPMs}} \times 60 \text{ sec/1 min}$
  - Typical $T_{\text{avg rotation}} = 7200 \text{ RPMs} \approx 2-5 \text{ ms}$

- **Transfer time** ($T_{\text{avg transfer}}$)
  - Time to read the bits in the target sector.
  - $T_{\text{avg transfer}} = \frac{1}{\text{RPM}} \times \frac{1}{(\text{avg # sectors/track})} \times 60 \text{ secs/1 min.}$
  - Really small
Disk Performance

- **Two scenarios:**
  - **Random Access:** no locality in sectors accessed
    - $T_{access} = T_{avg\ seek} + T_{avg\ rotation} + T_{avg\ transfer}$
  - **Sequential Access:** accessing consecutive sectors
    - No seek time or rotational delay!! Just transfer time.

- **Data from a representative disk:**

<table>
<thead>
<tr>
<th>Access Type</th>
<th>Throughput (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Access, 8kB blocks</td>
<td>1.26</td>
</tr>
<tr>
<td>Random Access, 64kB blocks</td>
<td>9.18</td>
</tr>
<tr>
<td>Sequential Access, 64kB blocks</td>
<td>111</td>
</tr>
</tbody>
</table>
Recording Zones:

- Modern disks partition tracks into disjoint subsets called *recording zones*
  - Each track in a zone has the same number of sectors, determined by the circumference of innermost track.
  - Each zone has a different number of sectors/track

- Outside tracks have more sectors
  - Same rotation speed
  - Higher MB/sec
Logical Disk Blocks

- Modern disks present a simpler abstract view of the complex sector geometry:
  - The set of available sectors is modeled as a sequence of b-sized logical blocks (0, 1, 2, ...)

- Mapping between logical blocks and actual (physical) sectors
  - Maintained by hardware/firmware device called disk controller.
  - Converts requests for logical blocks into (surface, track, sector) triples.

- Allows controller to set aside spare cylinders for each zone.
  - Accounts for the difference in “formatted capacity” and “maximum capacity”.
Solid State Disks (SSDs)

- Pages: 512KB to 4KB, Blocks: 32 to 128 pages
- Data read/written in units of pages.
- Page can be written only after its block has been erased
- A block wears out after 100,000 repeated writes.
SSD Performance Characteristics

<table>
<thead>
<tr>
<th></th>
<th>Sequential read throughput</th>
<th>Sequential write throughput</th>
<th>Random read throughput</th>
<th>Random write throughput</th>
<th>Random read access</th>
<th>Random write access</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>250 MB/s</td>
<td>170 MB/s</td>
<td>140 MB/s</td>
<td>14 MB/s</td>
<td>30 us</td>
<td>300 us</td>
</tr>
</tbody>
</table>

- Why are random writes so slow?
  - Erasing a block is slow (around 1 ms)
  - Write to a page triggers a copy of all useful pages in the block
    - Find an used block (new block) and erase it
    - Write the page into the new block
    - Copy other pages from old block to the new block
SSD Tradeoffs vs Rotating Disks

- **Advantages**
  - No moving parts → faster, less power, more rugged

- **Disadvantages**
  - Have the potential to wear out
    - Mitigated by “wear leveling logic” in flash translation layer
    - E.g. Intel X25 guarantees 1 petabyte ($10^{15}$ bytes) of random writes before they wear out
  - In 2010, about 100 times more expensive per byte

- **Applications**
  - MP3 players, smart phones, laptops
  - Beginning to appear in desktops and servers
## Storage Trends

### SRAM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>19,200</td>
<td>2,900</td>
<td>320</td>
<td>256</td>
<td>100</td>
<td>75</td>
<td>60</td>
<td>320</td>
</tr>
<tr>
<td>access (ns)</td>
<td>300</td>
<td>150</td>
<td>35</td>
<td>15</td>
<td>3</td>
<td>2</td>
<td>1.5</td>
<td>200</td>
</tr>
</tbody>
</table>

### DRAM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>8,000</td>
<td>880</td>
<td>100</td>
<td>30</td>
<td>1</td>
<td>0.1</td>
<td>0.06</td>
<td>130,000</td>
</tr>
<tr>
<td>access (ns)</td>
<td>375</td>
<td>200</td>
<td>100</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>9</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>0.064</td>
<td>0.256</td>
<td>4</td>
<td>16</td>
<td>64</td>
<td>2,000</td>
<td>8,000</td>
<td>125,000</td>
</tr>
</tbody>
</table>

### Disk

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$/MB$</td>
<td>500</td>
<td>100</td>
<td>8</td>
<td>0.30</td>
<td>0.01</td>
<td>0.005</td>
<td>0.0003</td>
<td>1,600,000</td>
</tr>
<tr>
<td>access (ns)</td>
<td>87</td>
<td>75</td>
<td>28</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>29</td>
</tr>
<tr>
<td>typical size (MB)</td>
<td>1</td>
<td>10</td>
<td>160</td>
<td>1,000</td>
<td>20,000</td>
<td>160,000</td>
<td>1,500,000</td>
<td>1,500,000</td>
</tr>
</tbody>
</table>
RAID: Redundant Array of Inexpensive Disks

- **Problem:** Disks fail -> total data loss
  - Improving reliability of a disk is expensive.
  - Cheaper just to buy a few extra disks.

- **Idea:** ECC for your disks
  - Files are “striped” across multiple disks
  - Redundancy yields high data availability
    - 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

- A multi-billion industry 80% non-PC disks sold in RAIDs
Common RAID configurations

**RAID 0**
No redundancy, Fast access

**RAID 1**
Mirror Data, most expensive sol’n

**RAID 3/4**
Parity drive protects against 1 failure

**RAID 5**
Rotated parity across all drives
Summary

- I/O devices are much slower than processors.
  - Engineered to be accessible, but to not slow down computation

- Spindle-based devices:
  - Access time = seek time + rotational delay + transfer time
  - Lay files out contiguously!

- RAID: Redundant Array of Inexpensive Disks
  - Achieve reliable storage, but not by making reliable disks
  - Use redundancy (e.g., parity) to reconstruct lost disk