I/O Systems (2): Disks

CSE 2431: Introduction to Operating Systems
Instructor: Adam C. Champion, Ph.D.
Reading: Chap. 11, §§13.5, 14.5.6 [OSC]
Outline

• Disk Organization
• Disk Performance
• Disk Scheduling Algorithms
Persistent Storage Devices

- Hard disk (Magnetic disks)
  - IDE disks (ATA disks): cheaper, lower performance
  - SCSI disks: high-performance, expensive
- Solid-state disk
- Floppy disk
- Optical disk: DVD, CD, Blu-ray
- Flash memory (USB sticks)
# Hard Disk Drive Examples

<table>
<thead>
<tr>
<th>Form Factor (in)</th>
<th>Dimensions (in)</th>
<th>Storage (GB)</th>
<th># Platters</th>
<th>Capacity per Platter (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.25</td>
<td>5.75 × 3.25 × 8</td>
<td>47 (1998)</td>
<td>14</td>
<td>3.36</td>
</tr>
<tr>
<td>3.5</td>
<td>4 × 1 × 5.75</td>
<td>14,000 (2017)</td>
<td>9</td>
<td>2,000</td>
</tr>
<tr>
<td>2.5</td>
<td>2.75 × 0.5 × 3.945</td>
<td>5,000 (2016)</td>
<td>5</td>
<td>1,000</td>
</tr>
</tbody>
</table>

New 2.5-inch 6,495 MB HDD vs. old 5.25-inch 110 MB HDD

Hard Disk Capacity (Areal Density)

Disk Technology Trends

• Disks are getting *smaller* for similar capacity
  – Spin faster, less rotational delay, higher bandwidth
  – Less distance for head to travel (faster seeks)
  – Lighter weight (for portables)

• Disk data is getting *denser*
  – More bits/square inch
  – Tracks are closer together
  – Doubles density every 18 months

• Disks are getting *cheaper* ($/GB)
  – Factor of $2 \times$ per year since 1991
  – Head close to surface
Disk Internals

- Platter: 1–12 per disk
- Cylinder
  - Certain track of the platters
  - 3.5-inch disk has ~2,000 cylinders
- Disk arm: seeks correct cylinder
Detailed View of a Disk

Figure 1: the mechanical components of a disk drive.
Disk Organization

• Disk surface
  – Circular disk coated with magnetic material

• Tracks
  – Concentric rings around disk surface, bits laid out serially along each track

• Sectors
  – Each track is split into arc of track (min transfer unit)
Disk Organization As Fiction

• Fixed arc implies inefficiency
  – Short inner sectors, long outer sectors

• Reality
  – More sectors on outer tracks (zones)
  – Disks map transparently
Examples of Disk Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>IBM 360-KB floppy disk</th>
<th>WD 3000 HLFS hard disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of cylinders</td>
<td>40</td>
<td>36,481</td>
</tr>
<tr>
<td>Tracks per cylinder</td>
<td>2</td>
<td>255</td>
</tr>
<tr>
<td>Sectors per track</td>
<td>9</td>
<td>63 (avg)</td>
</tr>
<tr>
<td>Sectors per disk</td>
<td>720</td>
<td>586,072,368</td>
</tr>
<tr>
<td>Bytes per sector</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>Disk capacity</td>
<td>360 KB</td>
<td>300 GB</td>
</tr>
<tr>
<td>Seek time (adjacent cylinders)</td>
<td>6 msec</td>
<td>0.7 msec</td>
</tr>
<tr>
<td>Seek time (average case)</td>
<td>77 msec</td>
<td>4.2 msec</td>
</tr>
<tr>
<td>Rotation time</td>
<td>200 msec</td>
<td>6 msec</td>
</tr>
<tr>
<td>Time to transfer 1 sector</td>
<td>22 msec</td>
<td>1.4 µsec</td>
</tr>
</tbody>
</table>
Disk Zones

- Physical geometry of a disk with two zones
- A possible virtual geometry for this disk
Disk RAID (1)

- RAID levels 0 through 2
- Backup and parity drives are shaded
Disk RAID (2)

- RAID levels 3 through 5
- Backup and parity drives are shaded
Solid State Drive (1)

- Popular storage device
  - No moving parts!
  - Writes data to NAND/NOR memory cells (1–4 bytes/cell)
- Form factors:
  - Hard disk
  - mSATA (51 × 30 × 5 mm)
  - M.2 (22 × 42–80 × 12–30 mm)
- Capacity: 128 GB – 1.5 TB
- Performance (Intel Optane):
  - Sequential: ≤ 2.2 GB/sec
  - Random: ≤ 550,000 I/O ops/sec

Source: [https://www.intel.com/](https://www.intel.com/)
Solid State Drive (2)

• I/O performed in *pages*, can’t overwrite in place
  – Must erase first at block increments
  – Can only erase about 100,000× before wear out
• Flash controller does *garbage collection* to free page space
• *Wear leveling* writes to all flash cells equally

<table>
<thead>
<tr>
<th>Valid page</th>
<th>Valid page</th>
<th>Invalid page</th>
<th>Invalid page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Invalid page</strong></td>
<td>Valid page</td>
<td>Invalid page</td>
<td><strong>Valid page</strong></td>
</tr>
</tbody>
</table>

NAND flash memory block with valid and invalid pages.

*Source: Silberschatz et al., Operating Systems Concepts, 10th ed., Fig. 11-4.*
CD Structure

Recording structure of a CD or CD-ROM
Logical Data Layout on CD-ROM

Symbols of 14 bits each

42 Symbols make 1 frame

Frames of 588 bits, each containing 24 data bytes

Preamble

98 Frames make 1 sector

Mode 1 sector (2352 bytes)

Bytes 16

Data

ECC

2048

288
CD-ROM Structure

- Cross section of CD-R disk and laser (not to scale)
- Silver CD-ROM has similar structure
  - Without dye layer
  - With pitted aluminum layer instead of gold
A double sided, dual layer DVD disk
Blu-ray Optical Disk

- Typically stores movies (1920 × 1020, 25 GB)
- Ultra HD disks available (4K movies; 50–100 GB)
- Similar structure to CDs, DVDs
- Sophisticated copy protection schemes

Source: Wikimedia Commons (public domain)
Comparison of Optical Media

Source: https://en.wikipedia.org/wiki/File:Comparison_CD_DVD_HDDVD_BD.svg
Disk Formatting (1)

A disk sector

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Data</th>
<th>ECC</th>
</tr>
</thead>
</table>
Disk Formatting (2)

An illustration of cylinder skew
Disk Formatting (3)

- No interleaving
- Single interleaving
- Double interleaving
Contents

• Disk Organization
• Disk Performance
• Disk Arm Scheduling Algorithms
Disk Performance Factor: Seeking

• Seeking: position the head to the desired cylinder (2–5 ms)

• Seek speed depends on:
  – Available power for pivot motor (0.5× seek time needs 4× power)
  – Arm stiffness (30–40g acceleration required for short seek time; flexible arms can twist ⇒ crash head into platter surface)

• A seek is composed of
  • A speedup, a coast, a slowdown, a settle
  • For very short seeks, the settle time dominates (1-3ms)

• Real-life analogy?
Disk Performance: Other Factors

• Rotational delay
  – Wait for a sector to rotate underneath the heads
  – Typically 6.0–8.3 ms (7,200–10,000 RPM) or ½ rotation takes 3–4.15 ms

• Transfer bytes
  – Average transfer bandwidth (15–37 Mbytes/sec)

• *Real-life analogy?*

• Performance of transfer 1 Kbytes
  – Seek (5.3 ms) + *half* rotational delay (3 ms) + transfer (0.04 ms)
  – Total time is 8.34ms or 120 Kbytes/sec!

• What block size can get 90% of the disk transfer bandwidth?
# Disk Examples (Summarized Specs)

<table>
<thead>
<tr>
<th>Capacity, Interface &amp; Configuration</th>
<th>Seagate Barracuda</th>
<th>IBM Ultrastar 72ZX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formatted Gbytes</strong></td>
<td>28</td>
<td>73.4</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>Ultra ATA/66</td>
<td>Ultra160 SCSI</td>
</tr>
<tr>
<td><strong>Platters / Heads</strong></td>
<td>4 / 8</td>
<td>11/22</td>
</tr>
<tr>
<td><strong>Bytes per sector</strong></td>
<td>512</td>
<td>512-528</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Performance</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max Internal transfer rate (Mbytes/sec)</strong></td>
<td>40</td>
<td>53</td>
</tr>
<tr>
<td><strong>Max external transfer rate (Mbytes/sec)</strong></td>
<td>66.6</td>
<td>160</td>
</tr>
<tr>
<td><strong>Avg Transfer rate (Mbytes/sec)</strong></td>
<td>&gt; 15</td>
<td>22.1-37.4</td>
</tr>
<tr>
<td><strong>Multisegmented cache (Kbytes)</strong></td>
<td>512</td>
<td>16,384</td>
</tr>
<tr>
<td><strong>Average seek, read/write (msec)</strong></td>
<td>8</td>
<td>5.3</td>
</tr>
<tr>
<td><strong>Average rotational latency (msec)</strong></td>
<td>4.16</td>
<td>2.99</td>
</tr>
<tr>
<td><strong>Spindle speed (RPM)</strong></td>
<td>7,200</td>
<td>10,000</td>
</tr>
</tbody>
</table>
Disk Behaviors

- There are more sectors on outer tracks than inner tracks
  - Read outer tracks: 37.4 MB/s
  - Read inner tracks: 22 MB/s
- Seek time and rotational latency dominates the cost of small reads
  - Much disk transfer bandwidth wasted!

<table>
<thead>
<tr>
<th>Block Size (Kbytes)</th>
<th>% of Disk Transfer Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.5%</td>
</tr>
<tr>
<td>8</td>
<td>3.7%</td>
</tr>
<tr>
<td>256</td>
<td>55%</td>
</tr>
<tr>
<td>1,024</td>
<td>83%</td>
</tr>
<tr>
<td>2,048</td>
<td>90%</td>
</tr>
</tbody>
</table>
Questions

• Since seek time and rotation delay dominate, how do you improve I/O performance?
Observations

• Getting first byte from disk read is slow (high latency)

• Peak bandwidth high, but rarely achieved

• Need to mitigate disk performance impact
  – Do extra calculations to speed up disk access
    • Schedule requests to shorten seeks
  – Move some disk data into main memory: file system caching
Contents

• Disk Organization
• Disk Performance
• Disk Scheduling Algorithms
Hard Disk Scheduling

• Which disk request is serviced first?
  – FCFS
  – Shortest seek time first
  – Elevator (SCAN)
  – C-SCAN (Circular SCAN)

• Look familiar?
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

• Analogy:
  – Can elevator scheduling use FCFS?

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

- **Pro:** Aims to minimize seek time

- **Cons:** Starvation

- **Question**
  - Is SSTF optimal?
  - Can we avoid 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
  – Travel to end, change direction

• Pros: Bounded time for each request

• Cons: Request at the other end will take time

• **LOOK algorithm**
  – Do not go to the end
  – Service last request, then change direction

98, 183, 37, 122, 14, 124, 65, 67
(37, 14, 65, 67, 98, 122, 124, 183)
C-SCAN (Circular SCAN)

- Method: like SCAN, but wrap around
- Pros: Uniform service time
- Cons: Does nothing on return

C-LOOK
- Do not go to the end
- Service the last request, then wrap around

98, 183, 37, 122, 14, 124, 65, 67
(65, 67, 98, 122, 124, 183, 14, 37)
Solid-State Disk Scheduling

- SSD scheduling algorithms simpler than HDDs
- Linux uses *NOOP* (no scheduling), but nearby logical block requests combined
  - *Write amplification* (one write leads to garbage collection, many I/O ops.) hurts performance
  - File system can notify about empty blocks to be erased (TRIM on SATA SSDs)
History of Disk-related Concerns

• When memory was expensive, minimize bookkeeping
• When disks were expensive, maximize number of usable sectors
• When disks became more common, increase reliability
• When processors got faster, make them appear faster
• Key point: find, improve *bottlenecks, single points of failure*
Summary

• Disk Organization
• Disk Performance
• Disk Scheduling Algorithms