Storage Systems

Main Points

- File systems
  - Useful abstractions on top of physical devices
- Storage hardware characteristics
  - Disks and flash memory
- File system usage patterns

File System Abstraction

- File system
  - Persistent, named data
  - Hierarchical organization (directories, subdirectories)
  - Access control on data
- File: named collection of data
  - Linear sequence of bytes (or a set of sequences)
  - Read/write or memory mapped
- Crash and storage error tolerance
  - Operating system crashes (and disk errors) leave file system in a valid state
- Performance
  - Achieve close to the hardware limit in the average case

Storage Devices

- Magnetic disks
  - Storage that rarely becomes corrupted
  - Large capacity at low cost
  - Block level random access
  - Slow performance for random access
  - Better performance for streaming access
- Flash memory
  - Storage that rarely becomes corrupted
  - Capacity at intermediate cost (50x disk)
  - Block level random access
  - Good performance for reads; worse for random writes
### Magnetic Disk

- Disk: Contains information that can be read and written.
- Tracks: Lines on the disk where data is stored.
- Sectors: Small areas on the tracks that store data.

### Disk Tracks

- ~1 micron wide
  - Wavelength of light is ~0.5 micron
  - Resolution of human eye: 50 microns
  - ~0.01K on a typical 2.5" disk
- Separated by unused guard regions
  - Reduces likelihood neighboring tracks are corrupted during writes (still a small non-zero chance)
- Track length varies across disk
  - Outside: More sectors per track, higher bandwidth
  - Disk is organized into regions of tracks with same # of sectors/track
  - Only outer half of radius is used
    - Most of the disk area in the outer regions of the disk

### Sectors

Sectors contain sophisticated error correcting codes
- Disk head magnet has a field wider than track
- Hide corruptions due to neighboring track writes

- Sector sparing
  - Remap bad sectors transparently to spare sectors on the same surface

- Slip sparing
  - Remap all sectors (when there is a bad sector) to preserve sequential behavior

- Track skewing
  - Sector numbers offset from one track to the next, to allow for disk head movement for sequential ops
Disk Performance

Disk Latency =
Seek Time + Rotation Time + Transfer Time
Seek Time: time to move disk arm over track (1-20ms)
Fine-grained position adjustment necessary for head to “settle”
Head switch time ~ track switch time (on modern disks)
Rotation Time: time to wait for disk to rotate under disk head
Disk rotation: 4 – 15ms (depending on price of disk)
Transfer Time: time to transfer data onto/off of disk
Disk head transfer rate: 50-100MB/s (5-10 usec/sector)
Host transfer rate dependent on I/O connector (USB, SATA, ...)

Toshiba Disk (2008)

<table>
<thead>
<tr>
<th>Size</th>
<th>Platters/Heads</th>
<th>Capacity</th>
<th>Spindle speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>2/4</td>
<td>320 GB</td>
<td>7200 RPM</td>
</tr>
<tr>
<td>Spindle speed</td>
<td></td>
<td></td>
<td>10.5 ms/ 12.0 ms</td>
</tr>
<tr>
<td>Average seek</td>
<td></td>
<td></td>
<td>19 ms</td>
</tr>
<tr>
<td>Maximum seek</td>
<td></td>
<td></td>
<td>1 ms</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>54–128 MB/s</td>
</tr>
<tr>
<td>Track-to-track</td>
<td></td>
<td></td>
<td>375 MB/s</td>
</tr>
<tr>
<td>Seek time</td>
<td></td>
<td></td>
<td>16 MB</td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td></td>
<td>16.35 W</td>
</tr>
<tr>
<td>Typical</td>
<td></td>
<td></td>
<td>11.68 W</td>
</tr>
<tr>
<td>Idle</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question

• How long to complete 500 random disk reads, in FIFO order?

Question

• How long to complete 500 random disk reads, in FIFO order?
  – Seek: average 10.5 msec
  – Rotation: average 4.15 msec
  – Transfer: 5-10 usec
  
  \[ 500 \times (10.5 + 4.15 + 0.01)/1000 = 7.3 \text{ seconds} \]
Question

• How long to complete 500 sequential disk reads?

Seek Time: 10.5 ms (to reach first sector)
Rotation Time: 4.15 ms (to reach first sector)
Transfer Time: (outer track)
500 sectors * 512 bytes / 128MB/sec = 2 ms
Total: 10.5 + 4.15 + 2 = 16.7 ms
Might need an extra head or track switch (+1 ms)
Track buffer may allow some sectors to be read off disk out of order (-2 ms)

Question

• How large a transfer is needed to achieve 80% of the max disk transfer rate?

Assume x rotations are needed, then solve for x:
0.8 (10.5 ms + (1 ms + 8.4 ms) x) = 8.4 ms · x
Total: x = 9.1 rotations, 9.8 MB
Disk Scheduling

- FIFO
  - Schedule disk operations in order they arrive
  - Downsides?

Disk Scheduling

- Shortest seek time first
  - Not optimal!
  - Suppose cluster of requests at far end of disk
  - Downsides?

Disk Scheduling

- SCAN: move disk arm in one direction, until all requests satisfied, then reverse direction
Disk Scheduling

- **CSCAN**: move disk arm in one direction, until all requests satisfied, then start again from farthest request.

Disk Scheduling

- **R-CSCAN**: CSCAN but take into account that short track switch is < rotational delay.

**Question**

- How long to complete 500 random disk reads, in any order?

**Question**

- How long to complete 500 random disk reads, in any order?
  - Disk seek: 1ms (most will be short)
  - Rotation: 4.15ms
  - Transfer: 5-10usec
- Total: 500 * (1 + 4.15 + 0.01) = 2.2 seconds
  - May be a bit shorter with R-CSCAN
**Flash Memory**

- Writes require large erasure block first
  - No update in place
  - 128 – 512 KB
  - Several milliseconds
- Write/read page (2-4KB)
  - 10s of microseconds

**Flash Translation Layer**

- Flash device firmware maps logical page # to a physical location
  - Allows firmware to move pages as needed
  - Wear-levelling (can only write a physical page a limited number of times)
  - Avoid pages that no longer work
  - Coalesce live pages during erasure
- TRIM command
  - File system tells device when pages are no longer in use

**File System Workload**

- File sizes
  - Are most files small or large?
  - Which accounts for more total storage: small or large files?
- File access
  - Are most accesses to small or large files?
  - Which accounts for more total I/O bytes: small or large files?