

CSE 451: Operating Systems

Winter 2024

Module 16

Spinning Disk drives

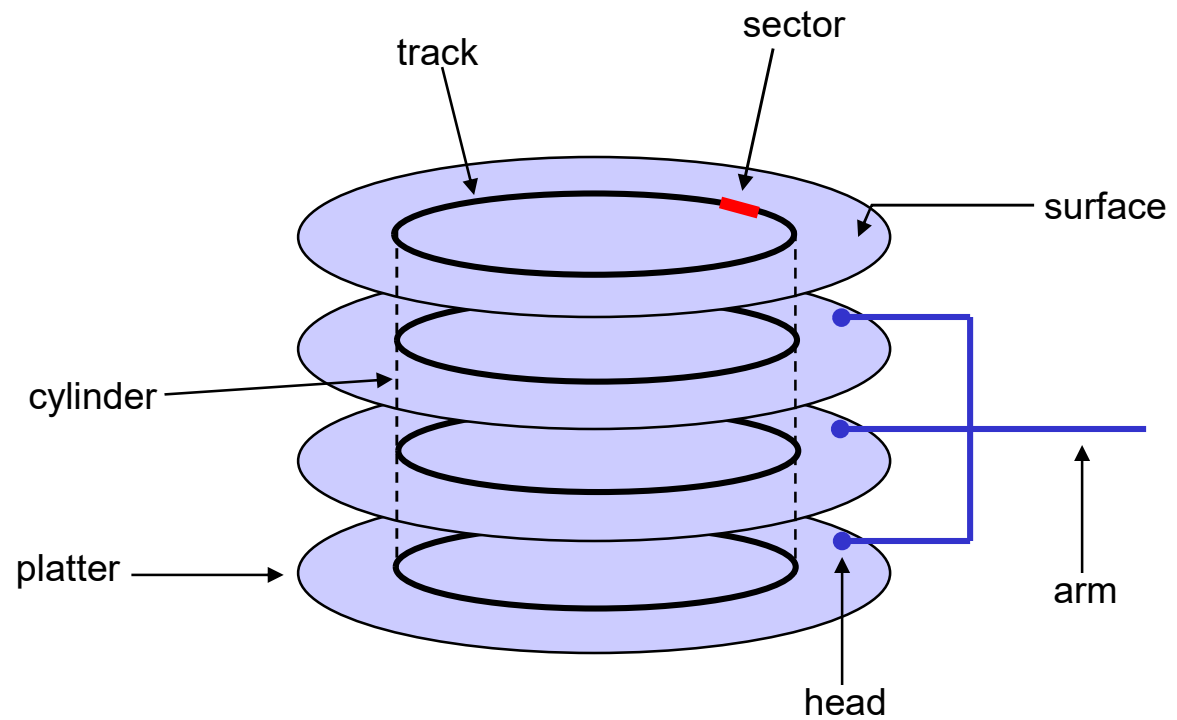
Are they still being used?

Gary Kimura

Physical disk structure

- Disk components

- platters
- surfaces
- tracks
- sectors
- cylinders
- arm
- heads



Disk performance

- Performance depends on a number of steps
 - **seek**: moving the disk arm to the correct cylinder
 - depends on how fast disk arm can move
 - seek times aren't diminishing very quickly (why?)
 - **rotation (latency)**: waiting for the sector to rotate under head
 - depends on rotation rate of disk
 - rates are increasing, but slowly (why?)
 - **transfer**: transferring data from surface into disk controller, and from there sending it back to host
 - depends on density of bytes on disk
 - increasing, relatively quickly
- When the OS uses the disk, it tries to minimize the cost of all of these steps
 - particularly seeks and rotation

Performance via disk layout

- OS may increase file block size in order to reduce seeking
- OS may seek to co-locate “related” items in order to reduce seeking
 - blocks of the same file
 - data and metadata for a file

Performance via caching, pre-fetching

- Keep data or metadata in memory to reduce physical disk access
 - problem?
- If file access is sequential, fetch blocks into memory before requested

Performance via disk scheduling

- Seeks are very expensive, so the OS attempts to schedule disk requests that are queued waiting for the disk
 - FCFS (do nothing)
 - reasonable when load is low
 - long waiting time for long request queues
 - SSTF (shortest seek time first)
 - minimize arm movement (seek time), maximize request rate
 - unfairly favors middle blocks
 - SCAN (elevator algorithm)
 - service requests in one direction until done, then reverse
 - skews wait times non-uniformly (why?)
 - C-SCAN
 - like scan, but only go in one direction (typewriter)
 - uniform wait times

Interacting with disks

- In the old days...
 - OS would have to specify cylinder #, sector #, surface #, transfer size
 - i.e., OS needs to know all of the disk parameters
- Modern disks are even more complicated
 - not all sectors are the same size, sectors are remapped, ...
 - disk provides a higher-level interface, e.g., SCSI
 - exports data as a logical array of blocks [0 ... N]
 - maps **logical blocks** to cylinder/surface/sector
 - OS only needs to name logical block #, disk maps this to cylinder/surface/sector
 - on-board cache
 - as a result, physical parameters are hidden from OS
 - both good and bad

Solid state drives: disruption

- Hard drives are based on spinning magnetic platters
 - *mechanics* of drives determine performance characteristics
 - sector addressable, not byte addressable
 - capacity improving exponentially
 - sequential bandwidth improving reasonably
 - random access latency improving very slowly
 - cost dictated by massive economies of scale, and many decades of commercial development and optimization

- Solid state drives are based on NAND flash memory
 - no moving parts; performance characteristics driven by electronics and physics – more like RAM than spinning disk
 - relative technological newcomer, so costs are still quite high in comparison to hard drives, but dropping fast



SSD performance: reads

- Reads
 - unit of read is a *page*, typically 4KB large
 - SSD can typically handle 10,000 – 100,000 reads/s
 - 0.01 – 0.1 ms read latency (50-1000x better than disk seeks)
 - 40-400 MB/s read throughput (1-3x better than disk seq. thpt)

SSD performance: writes

- Writes
 - flash media must be *erased* before it can be written to
 - unit of erase is a block, typically 64-256 pages long
 - usually takes 1-2ms to erase a block
 - blocks can only be erased a certain number of times before they become unusable – typically 10,000 – 1,000,000 times
 - unit of write is a page
 - writing a page can be 2-10x slower than reading a page
- Writing to an SSD is complicated
 - random write to existing block: read block, erase block, write back modified block
 - leads to hard-drive like performance (300 random writes / s)
 - sequential writes to erased blocks: fast!
 - SSD-read like performance (100-200 MB/s)

SSDs: dealing with erases, writes

- Lots of higher-level strategies can help hide the warts of an SSD
 - many of these work by virtualizing pages and blocks on the drive (i.e., exposing logical pages, not physical pages, to the rest of the computer)
 - wear-leveling: when writing, try to spread erases out evenly across physical blocks of the SSD
 - Intel promises 100GB/day x 5 years for its SSD drives
 - log-structured filesystems: convert random writes within a filesystem to log appends on the SSD (more later)
 - build drives out of arrays of SSDs, add lots of cache