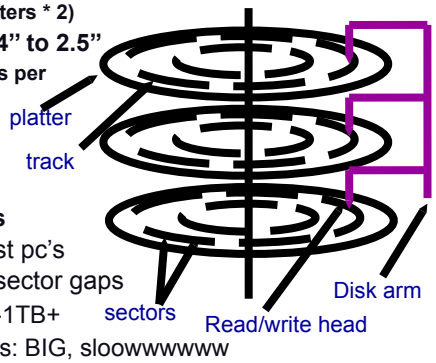
CS 140 - Summer 2008 - Handout #15: Disks

What do disks look like?

- **2-30 heads (platters * 2)**
  diameter 14" to 2.5"
- **700-20480 tracks per surface**
- **16-1600 sectors per track**
- **sector size:**
  64-8k bytes
  512 for most pc's
  note: inter-sector gaps
- capacity: 20M-1TB+
- main adjectives: BIG, sloowwwwww



Some typical disk drives

	DeskStar HDS5725050	Cheetah ST373454LW
Capacity	500GB	73GB
Disk/Heads	5/10	2/4
Cylinders	103,182	50,864
Sectors/track	~946	~921
Speed	7200RPM	15000RPM
Latency (ms)	4.2	2.0
Avg seek (ms)	8.2/9.2	3.4/4.0
Track-2-track(ms)	0.8/1.3	0.2/0.4

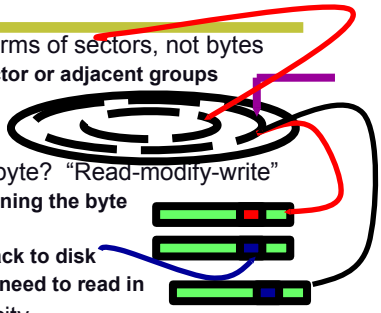
Disk vs. Memory

- Smallest write: sector
- Atomic write = sector
- Random access: 5ms
  not on a good curve
- Sequential access: 200MB/s
- Cost $.20/GB
- Crash: doesn't matter ("non-volatile")

- (usually) bytes
- byte, word
- 50 ns
  faster all the time
- 200-1000MB/s
- $20/GB
- contents gone ("volatile")

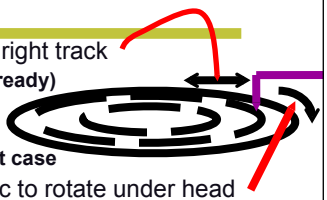
Some useful facts

- Disk reads/writes in terms of sectors, not bytes
  Read/write single sector or adjacent groups



- How to write a single byte? "Read-modify-write"
  Read in sector containing the byte
  Modify that byte
  Write entire sector back to disk
  Key: if cached, don't need to read in
- Sector = unit of atomicity.
  Sector write done completely, even if crash in middle
  (disk saves up enough momentum to complete)
  Larger atomic units have to be synthesized by OS

Some useful costs



- Seek: move disk arm to the right track
  Best case: 0ms (on track already)
  Worst: ~30-50ms (move over entire disk)
  Average: 10-20ms, 1/3 worst case
- Rotational delay: wait for sec to rotate under head
  Best: 0ms (over sector)
  Worst: ~16ms (entire rotation)
  Average: ~8ms (1/2 worst case)
- Transfer bandwidth: suck bits off of device
- Cost of disk access? Seek + rotation + transfer time
  Read a single sector: 10ms + 8ms + 50us ~= 18ms
  Cool: read an entire track? Seek + transfer! (why?)

Some useful trends

- Disk bandwidth and cost/bit improving exponentially
  Similar to CPU speed, memory size, etc.
- Seek time and rotational delay improving *very* slowly
  Why? require moving physical object (disk arm)
- Some implications:
  Disk accesses a huge system bottleneck & getting worse
  Bandwidth increase lets system (pre-)fetch large chunks for about the same cost as small chunk.
    Result? trade bandwidth for latency if you can get lots of related stuff.
    How to get related stuff? Cluster together on disk
  Memory size increasing faster than typical workload size
    More and more of workload fits in file cache
    disk traffic changes: mostly writes and new data

Flash RAM disks

- Disk storage devices made from Flash RAM
  - **Non-volatile random access memory**
  - **Read/Erase/Write operations**
  - **~$4/GB for flash itself, more for "drive"**
- Issues for file systems:
  - **No-seek or rotational delays.**
  - **Currently large transfer delays.**
  - **Durability issues (limited number of writes per block)**