Operating Systems
ECE344
Disks and RAID

Ashvin Goel
ECE
University of Toronto
Outline

- Disks
  - Disk scheduling algorithms
- Redundancy in storage systems, RAID
Disk Geometry

- Disks have multiple platters

- Each platter has an arm, and a head for accessing data:
  - The head accesses data while the platter is rotating
  - The arm moves about the pivot to access different parts of the disk
Each platter has multiple tracks
Each track has multiple sectors
A sector has a header, 512 bytes of data and 16 bytes of ECC
Head reads/writes one sector at a time
A cylinder consists of the same track across different platters
  - The platter arms move together
  - The different heads access data in parallel
Seagate Mach.2 (480MB/s)
Distance Between Head and Platter

Comparative Sizes of Potential Contaminants
(Size numbers are fractions of an inch)

- Read/Write Head Clearance: 0.00015
- Disk: 0.0015
- Dust Particle: 0.0025
- Recording Media Layer: 0.00025
- Smoke Particle: 0.00625
- Finger Print: 0.000625
- Human Hair: 0.003
Disk Access Delays

- Time to access a disk sector is determined by 3 delays
  - **Seek time**
    - Time to move head to correct track
  - **Rotational delay**
    - Time for disk to rotate to correct sector
  - **Transfer time**
    - Time to read/write the bits of sector
Disk Performance Trends

- **Capacity**
  - Increases 50-100% per year (2X every two years)
  - Multi-terabyte disks available today

- **Transfer rate (BW)**
  - Increases 40% per year (2X every two years)
  - Typically, 100-200 MB/s
  - Sector transfer time is 100-200 microseconds

- **Seek time and rotation time**
  - Decreases 8% per year (becomes 1/2 every 10 years)
  - Seek and rotation time are typically 4-8 ms today
Disk Performance

- Highest bandwidth for sequential access
  - No seek time, no rotational delay, only transfer time

- Worst bandwidth for random access
  - Seek time due to head movement, rotational delay, transfer time

- Disk scheduling aims to minimize seek time

- Hard to control rotational delay
Addressing Disks

- Older disks required OS to specify all parameters for transferring data
  - E.g., cylinder #, track #, sector #, transfer size
- Modern disks are more complicated
  - Not all sectors are the same size, sectors are remapped, etc.
- Current disks provide a higher-level interface
  - Disk exports its data as a logical array of sectors
  - Disk maps logical sectors to its surface
  - OS code is simpler but disk parameters are hidden
Disk Errors

- Lots of errors possible
  - E.g., latent sector errors, mis-directed writes, etc.
  - Transient vs. hard errors
  - Some errors can be masked by ECC

- Bad sectors
  - Disk allocate spare sectors per track
  - Bad sector can be mapped to spare in factory or by disk controller
Disk Scheduling Algorithms

- Applications use read/write system calls to issue concurrent read and write requests for disk sectors
- OS adds these requests to a disk scheduler queue
- Disk scheduler issues these requests to the disk
- Scheduler aims is to improve disk performance:
  - Reduces seek time
  - Reads several consecutive sectors of data together
- Algorithms
  - First-come, first served (FCFS)
    - Simple, fair, slow
  - Shortest seek time first (SSF)
  - SCAN (Elevator)
Shortest Seek First (SSF)

- Cylinder number not available, so use sector number
- Shortest seek first minimizes arm motion
- Unlike FCFS, starvation is possible
SCAN (Elevator)
SCAN (Elevator)

- Use a bit to track outward or inward arm direction
- Service the next pending request in same direction
- When there are no more requests in the current direction, reverse direction
- Increases seek time compared to SSF but ensures no starvation

- A variant is called C-SCAN
  - SCAN, but request go in one direction (typewriter)
  - What are its benefits/drawbacks vs. SCAN?
Redundancy in Storage Systems

- Redundant Array of Inexpensive Disks (RAID)
  - Idea: Use many disks in parallel
  - Increases storage bandwidth, improves reliability

- RAID organization
  - A chunk (or strip or block) is a unit of read or write on a disk
    - It consists of one or more sectors
  - Consecutive chunks are placed on different disks and can be read or written in parallel, thus improving performance
    - For example, Chunks 4, 5, 6 and 7 are placed on Disks 0, 1, 2, 3
RAID Level 0

- RAID level 0: disk striping
  - Distributes data across several disks for speed
  - No redundancy
RAID Level 1

- RAID level 1: mirroring
  - Backup solution
  - Write both, read either
  - 50% utilization
  - Handles up to 50% disk failures
RAID Level 4

- RAID level 4: Use a dedicated parity disk
  - Calculate XOR value of chunks and store on parity disk
    - E.g., $P_{03} = S_0 + S_1 + S_2 + S_3$, where $+$ is XOR
  - Updating parity efficiently
    - Say $S_0$ is being updated from $S_{0\_old}$ to $S_{0\_new}$, then
      - $P_{03\_new} = P_{03\_old} + S_{0\_old} + S_{0\_new}$
  - Uses one extra disk for storing the XOR parity value
    - Utilization: $(N-1)/N$, $N$ is the number of disks
  - Any one single disk (including parity disk) can fail, and its data can be recovered
RAID Level 5

- RAID level 5: striping with distributed parity
  - Similar to 4 but parity information is distributed across all disks
  - Avoids bottleneck for parity disk
Think Time

- What are the main delays associated with disks?
- Disk scheduling algorithms optimize for which of these delays? Why?
What are the main delays associated with disks?

- Seek time: head move between tracks, this time depends on the arm movement and is the most significant delay.
- Rotational time: head need to wait for the correct sector on the track, this time depends on the rotational speed (10K rpm, 15K rpm), and is the next most significant delay.
- Transfer time: head reads data from the sector, this time is a small fraction of the rotational time since there are many sectors in a track.
Think Time Answers

Disk scheduling algorithms optimize for which of these delays? Why?

- OS-level disk scheduling algorithms mainly optimize for seeks because they are the most expensive. It is hard to optimize for rotational delay directly because the exact location of sectors and the current position of head is not known to OS.

- However, the disk controller h/w may implement its own scheduling algorithm, which could optimize for all disk h/w delays.
Think Time (RAID)

- What are the benefits and drawbacks of RAID Level 5 over RAID Level 1?
- Can RAID Level 5 handle two full disk errors?
- Can RAID Level 5 handle sector errors?
What are the benefits and drawbacks of RAID Level 5 over RAID Level 1?

- Benefits of RAID 5
  - Lower space overhead
- Drawbacks of RAID 5
  - Lrites are more expensive because parity updates require reading parity data
  - More complex and expensive recovery on disk failure

Can RAID Level 5 handle two full disk errors?

- No, that requires RAID Level 6 (look it up)
Can RAID Level 5 handle sector errors?

- RAID is primarily designed for whole disk failures. Sector errors are handled using ECC.

If RAID can detect a sector error on a disk and the entire disk is replaced, then RAID can handle a disk sector error.

However, the problem is that sector errors are latent (i.e., they don't show up unless one accesses a sector). So, unless you are scanning disks in the background for sector errors (which is called scrubbing), these errors are not known.

Now say a disk fails and RAID recovery is used to recover from this failure. The problem is that this involves scanning the rest of the disks, which may have sector errors, and they become visible during recovery. At this point, two or more disks have errors (the original failed disk + the disk with sector errors), and so RAID5 can't recover from such errors.