CS315: Database Systems
Physical Design

Arnab Bhattacharya
arnabb@cse.iitk.ac.in

Computer Science and Engineering,
Indian Institute of Technology, Kanpur
http://web.cse.iitk.ac.in/~cs315/

2nd semester, 2022-23

Tue 10:30-11:45, Thu 12:00-13:15

Physical Storage Media

- Cache (primary storage)
  - Fastest
  - Most costly
  - Volatile: contents vanish once power is off

Physical Storage Media

- Cache (primary storage)
  - Fastest
  - Most costly
  - Volatile: contents vanish once power is off
- Main memory (primary storage)
  - Fast
  - May not be enough to hold a database
  - Volatile

Physical Storage Media

- Cache (primary storage)
  - Fastest
  - Most costly
  - Volatile: contents vanish once power is off
- Main memory (primary storage)
  - Fast
  - May not be enough to hold a database
  - Volatile
- Flash memory (secondary storage, online storage)
  - SSD
  - Non-volatile
  - Read is quite fast
  - Write is slower due to erase
  - Supports a fixed number of write/erase cycles
  - Cheaper than main memory

Physical Storage Media (contd.)

- Magnetic disk (secondary storage, online storage)
    - Large
    - Direct-access: can read and write any location
    - Data needs to be brought to memory
    - Slower
    - Non-volatile

Physical Storage Media (contd.)

- Magnetic disk (secondary storage, online storage)
    - Large
    - Direct-access: can read and write any location
    - Data needs to be brought to memory
    - Slower
    - Non-volatile
- Optical storage (tertiary storage, offline storage)
    - CD, DVD, etc.
    - Non-volatile
    - Write-once, read-many
    - Slower
    - Re-writable also available

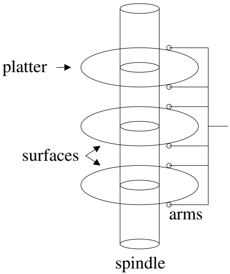
Physical Storage Media (contd.)

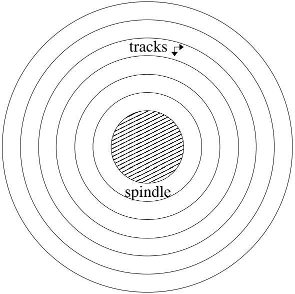
- Magnetic disk (secondary storage, online storage)
  - Large
  - Direct-access: can read and write any location
  - Data needs to be brought to memory
  - Slower
  - Non-volatile
- Optical storage (tertiary storage, offline storage)
  - CD, DVD, etc.
  - Non-volatile
  - Write-once, read-many
  - Slower
  - Re-writable also available
- Magnetic tape (tertiary storage, offline storage)
  - Sequential access
  - Much slower
  - Very high capacity
  - Much cheaper

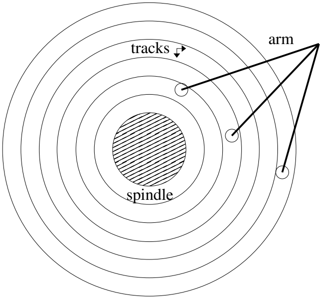
Disks

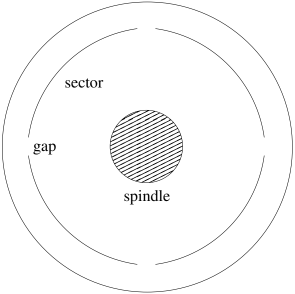
- Physically, disks consist of circular platters
- Both surfaces of a platter can be accessed
- Each surface contains concentric tracks
- Tracks are divided into sectors separated by gaps
- Aligned tracks form a cylinder

Disk Access


platter
surfaces
arms
spindle


tracks
spindle


tracks
arm
spindle


sector
gap
spindle

Disk Access Time

- *Smallest* unit of information that can be read from or written to disk is a sector
- Block or page is a logical unit read from or written to by O/S
  - Block consists of a contiguous sequence of sectors

Disk Access Time

- *Smallest* unit of information that can be read from or written to disk is a sector
- Block or page is a logical unit read from or written to by O/S
  - Block consists of a contiguous sequence of sectors
- Access time T_{access}: Time to access a particular sector

$$T_{access} = T_{seek} + T_{rotation} + T_{transfer}$$

- Seek time T_{seek}: Time to position arm heads over cylinder containing the target sector
  - Typical seek time: 8 ms
- Rotational latency $T_{rotation}$: (Average) time to rotate r/w head to the first bit of the sector
  - $T_{rotation}$ = (1 / 2) × (1 / rpm) × (60 s / 1 min)
- Transfer time $T_{transfer}$: Time to read bits from the sector
  - $T_{transfer}$ = (1 / (#sectors / track)) × (60 / rpm)

Typical Disk Parameters

- Average seek times from 4-10 ms
- Rotational speeds are 60, 90, 120, 250 revolutions per second, i.e., 3600, 5400, 7200, 15000 rpm respectively
- Sector sizes vary between 512 bytes and 1024 bytes
- 400 to 1000 sectors per track
- 20,000 to 50,000 tracks per surface
- 1 to 5 platters per disk

Typical Disk Parameters

- Average seek times from 4-10 ms
- Rotational speeds are 60, 90, 120, 250 revolutions per second, i.e., 3600, 5400, 7200, 15000 rpm respectively
- Sector sizes vary between 512 bytes and 1024 bytes
- 400 to 1000 sectors per track
- 20,000 to 50,000 tracks per surface
- 1 to 5 platters per disk
- Example: To access one sector, it requires
  - Rotational speed = 7200 rpm
  - Average seek time T_{seek} = 8 ms
  - Average #sectors / track = 400
  - $T_{rotation} = (1 / 2) \times (1 / 7200) \times 60 = 4.17$ ms
  - $T_{transfer} = (1 / 400) \times (1 / 7200) \times 60 = 0.02$ ms
  - $\therefore T_{access} = 8 + 4.17 + 0.02 = 12$ ms

Access Times

- This disk access time is for random I/O

Access Times

- This disk access time is for random I/O
- Once the first bit is read, the rest (sequential I/O) is almost free (only 0.02 ms)
- Data transfer rates or bulk transfer rates are calculated more precisely using gaps

Access Times

- This disk access time is for random I/O
- Once the first bit is read, the rest (sequential I/O) is almost free (only 0.02 ms)
- Data transfer rates or bulk transfer rates are calculated more precisely using gaps
- Time for memory access is multiple orders of magnitude lesser

Access Times

- This disk access time is for random I/O
- Once the first bit is read, the rest (sequential I/O) is almost free (only 0.02 ms)
- Data transfer rates or bulk transfer rates are calculated more precisely using gaps
- Time for memory access is multiple orders of magnitude lesser
- Disk access time is dominated by seek time and rotational latency
- Sequential access algorithms exploit the (almost) free access time of later bits heavily
- Most algorithms aim to avoid random I/Os

Optimization of Disk Block Access

- *Disk arm scheduling*: schedule such that movement of disk arm head is minimized
  - Elevator algorithm: move arm in one direction, process all requests in that order, and then move arm back in reverse direction

Optimization of Disk Block Access

- *Disk arm scheduling*: schedule such that movement of disk arm head is minimized
  - Elevator algorithm: move arm in one direction, process all requests in that order, and then move arm back in reverse direction
- *File organization*: organize blocks of a file to minimize random I/Os
  - Defragmention: put all blocks contiguously, and reduce fragmentation

Optimization of Disk Block Access

- *Disk arm scheduling*: schedule such that movement of disk arm head is minimized
  - Elevator algorithm: move arm in one direction, process all requests in that order, and then move arm back in reverse direction
- *File organization*: organize blocks of a file to minimize random I/Os
  - Defragmention: put all blocks contiguously, and reduce fragmentation
- *Deferred writes*: Postpone and perform writes batchwise
  - Use non-volatile write buffers, e.g., flash memory
  - Maintain logs for correctness

Data Redundancy and Parallelism

- Redundancy improves reliability
- RAID: Redundant arrays of independent disks
- Uses mirroring or shadowing
  - Failure only if both fail
- Mean time to data loss depends on mean time to failure for each disk and mean time to repair

Data Redundancy and Parallelism

- Redundancy improves reliability
- RAID: Redundant arrays of independent disks
- Uses mirroring or shadowing
  - Failure only if both fail
- Mean time to data loss depends on mean time to failure for each disk and mean time to repair
- Parallelism reduces mean response time

File Organization

- Records in a file can be organized differently
- *Heap*: A record is placed anywhere where there is space
- *Sequential*: Records are placed sequentially in the order of the search key
- *Hashing*: Records are put in the block where they hash to

Storage of Special Data

- Data dictionary or system catalog stores metadata
    - *Data about data*

Storage of Special Data

- Data dictionary or system catalog stores metadata
  - *Data about data*
- Information about relations
  - Name of relation
  - Name and type of attributes
  - Name and definition of views
  - Constraints

Storage of Special Data

- Data dictionary or system catalog stores metadata
  - *Data about data*
- Information about relations
  - Name of relation
  - Name and type of attributes
  - Name and definition of views
  - Constraints
- User information including password and access priviledge

Storage of Special Data

- Data dictionary or system catalog stores metadata
  - *Data about data*
- Information about relations
  - Name of relation
  - Name and type of attributes
  - Name and definition of views
  - Constraints
- User information including password and access priviledge
- Statistics about relations
  - Number of tuples
  - Histograms of values

Storage of Special Data

- Data dictionary or system catalog stores metadata
  - *Data about data*
- Information about relations
  - Name of relation
  - Name and type of attributes
  - Name and definition of views
  - Constraints
- User information including password and access priviledge
- Statistics about relations
  - Number of tuples
  - Histograms of values
- Organization of relations
  - Storage organization
  - Physical address

Storage of Special Data

- Data dictionary or system catalog stores metadata
  - *Data about data*
- Information about relations
  - Name of relation
  - Name and type of attributes
  - Name and definition of views
  - Constraints
- User information including password and access priviledge
- Statistics about relations
  - Number of tuples
  - Histograms of values
- Organization of relations
  - Storage organization
  - Physical address
- Information about indices
  - Name of attribute and relation
  - Physical address of index

Storage of Special Data

- Data dictionary or system catalog stores metadata
    - *Data about data*
- Information about relations
    - Name of relation
    - Name and type of attributes
    - Name and definition of views
    - Constraints
- User information including password and access priviledge
- Statistics about relations
    - Number of tuples
    - Histograms of values
- Organization of relations
    - Storage organization
    - Physical address
- Information about indices
    - Name of attribute and relation
    - Physical address of index
- Large objects with pointers and buffer management