Overview of Data Storage

Physical Characteristics of Disks

A *disk drive* consists of

- disk assembly: part of the disk drive that contains the physical data storage.
- head assembly: part of the disk drive that contains *heads* for accessing data.

Disk assembly consists of disk platters. Each platter has two surfaces.

Disk surfaces are partitioned into circular *track*, each consisting of a number of *sectors*.

A collection of tracks equidistant from the center on all surfaces is called a *cylinder*. (in other words: the tracks that are under the disk head at the same time).

Disk Access

A disk **block** or **page** is a logical unit of data storage which ***can be read from disk in a single disk access command***.

Often a disk block is equal in size to a disk sector, or a small number of disk sectors (2,3,4).

Disk latency: time between a disk access command is issued and the data is delivered from disk to main memory. Disk latency consists of the following components:

- disk controller processing time (typically, fraction of a ms.);
- *seek time*: time it takes to move disk head to the correct cylinder (10-40ms);
- *rotational latency*: time it takes to rotate the disk (10ms for a full rotation);
- *transfer time*: the time it takes to read in all the sectors of the block. ($\sim$ 10Mb/sec).

I/O model of Computation

Observations:

- In database applications, the amount of data exceeds main memory capacity of the computers.
- When data is stored in secondary storage (on disks), **data access time dominates all other computation times during query execution**.

Conclusions:

- Efficiency of query processing in DBMS needs to be measured w.r.t. disk access it requires, rather than w.r.t. computational complexity of the algorithms.
- It is of utmost importance to implement efficient disk access in DBMS.
- Efficient disk access can be implemented as follows:
  - *Confine all disk access in DBMS to **block/page** access.* That is, all disk access commands in DBMS must access a full disk block.
  - Develop and implement techniques for efficient storage and access data stored in disk blocks.

Data Storage Techniques

Cylinder-based Organization. Store consecutive blocks on a single cylinder (on different surfaces). Improves seek time for consecutive read operations. But if access requests are "random", will not help.

Use of Multiple Disks. Multiple disks mean multiple disk controllers, which means, more disk access requests per time unit can be satisified. But requires extra investment in the disk infrastructure.

Mirroring. Here, multiple disks are used again, each repeating the data stored on other disks. This improves the response times and the number of requests processed per time unit, but has higher cost and poses issues when data is updated.

Elevator Algorithm. If multiple disk access requests come to disk controller at the same time, we can schedule them using the same principles as are used in elevator operation - assuming the disk head is our "elevator" and tracks are "floors". This can streamline processing of multiple requests, but is not as efficient for non-busy request schedules.

Prefetching/Double Buffering. Sometimes it is possible to "know" which blocks will be needed and schedule their optimal retrieval before the actual access requests arrive. But, prefetching requires extra main memory.