



Storage Hierarchy

Primary Storage consists of main memory and processor cache

Very fast, e.g. for DDR4-3200 memory:

- sequential read: 50 GByte/s
- latency: 100ns

Data can be directly processed by CPU

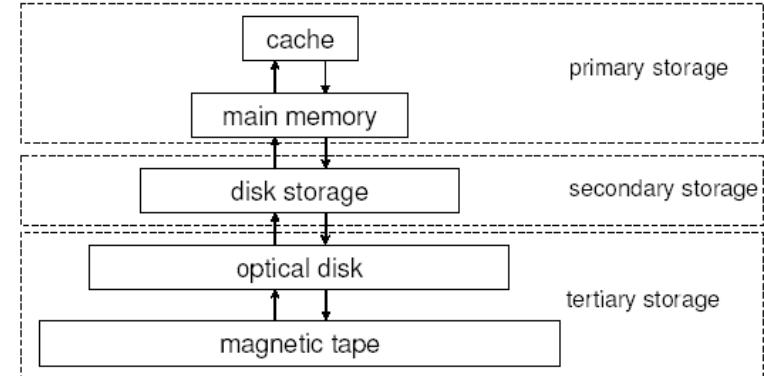
Access to data with fine granularity: each byte can be addressed

Number of accessible bytes depends on address scheme:
32-bit address scheme implies that only 2^{32} bytes are addressable

Volatile, non-reliable storage media

Storage Hierarchy

Primary, secondary and tertiary storage



Primary Storage

Secondary Storage

hard disk storage or SSD

stable, non-volatile, reliable

much larger, e.g. 10 TByte per medium

By orders of magnitude cheaper

Data can not be directly processed

Access granularity is coarse: blocks of e.g., 4 KBytes

Much slower, even for modern drives:

- for HDD: sequential read: up to 280 MByte/s, latency: 1-30 ms
- for SSD: sequential read: up to 7000 MByte/s, latency: 10-100 µs

Necessary:

- good buffer management (high hit ratio)
- good query management

SDD versus HDD

Advantages SSD:

- faster
- physically smaller

Advantages HDD:

- larger capacity per drive
- cheaper

In particular the cost limits the use of SSDs to store large amounts of data

- SSDs dominate for storing boot-partitions and small amounts of data
- HDDs still frequently for storing large databases

Sales figures for 2020:

- the number of sold drives is quite similar for HDDs and SSDs
- the sold capacity of HDDs is around five times the capacity of SSDs

Performance Metrics (1)

Crucial for the performance of a DBMS is the transition between main memory (primary storage) and secondary storage

- to work with data, it has to be in primary storage (in a region called buffer), but:
- main memory is not persistent
- the main memory buffer is typically much smaller than the database. Consequence: We have to discard data from the buffer to load other parts of the database into the buffer; modified data has to transferred to secondary storage before it can be discarded
- Result: Data has to be transferred frequently between main memory and secondary storage.

Data is transferred between main memory and secondary storage in units called blocks (size depends on media, e.g., 4KByte).

Tertiary Storage

Usage:

- for long-term archival storage or
- short-term logging (journals) of database updates

Secondary storage is too expensive for this purpose

Media: optical discs, magnetic tapes

Size of a magnetic tape: several terabytes

Medium is typically switched: “offline storage”

Disadvantage: access gap extremely large: manually access medium, insert medium

Performance Metrics (2)

A disk I/O (read or write of a block) is very expensive compared with what is likely to be done with the block once it arrives in main memory.

- Perhaps 1,000,000 machine instructions in the time to do one random disk I/O.
- Random block accesses is the norm if there are several processes accessing disks, and the disk controller does not schedule accesses carefully.

Reasonable model of computation that requires secondary storage: count only the disk I/O's.

Good DBMS algorithms

- a) Try to make sure that if we read a block, we use much of the data on the block.
- b) Try to put blocks that are accessed together in physically adjacent locations.
- c) Try to buffer commonly used blocks in main memory.

Sorting Example

Setup:

- 10^7 records of 100 bytes = 10^9 bytes file.
 - Stored on a disk with 4KByte blocks, each holding 40 records + header information.
 - Entire file takes 250,000 blocks.
- 50MByte available main memory = 12,800 blocks \approx 1/20th of file.
- Task: Sort records of file by primary key field.