Overview

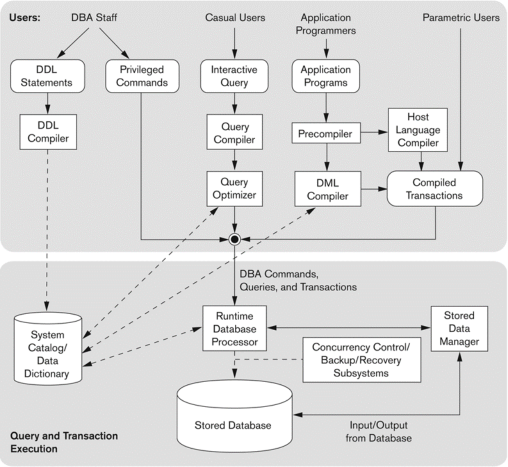


Figure 2.3
Component modules of a DBMS and their interactions.

In the following weeks, we will cover:

- File storage and indexing
- Query processing and optimization
- Transaction processing
- Concurrency control
- Database recovery

Picture from Chapter 2, Fundamentals of Database Systems

THE UNIVERSITY OF AUCKLAND
Te Whare Wananga o Tamaki Makaurau
NEW ZEALAND
SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

Database Systems

File Storage

Jing Sun and Miao Qiao
The University of Auckland

Overview

1. Physical storage hierarchy
2. Secondary storage
3. Storage access
4. Organize records in a file
5. Redundant arrays of independent disks (RAID)

Reading materials:

- Chapter 12, Database System Concepts, Seventh Edition (`http://codex.cs.yale.edu/avi/db-book/`)
- Chapter 16, Fundamentals of Database Systems

Storage Hierarchy

Various storage media are measured by their access time, price per bit, and reliability.

Capacity:

- bit: 0 or 1
- byte: 8 bits
- kilobytes (KB) $=$ 1000 bytes
- megabytes (MB), gigabytes (GB), terabytes (TB), petabytes (PB)

Reliability:

- volatile : Lose contents when power is switched off.
- non-volatile : Persistent contents .

Storage Hierarchy

THE UNIVERSITY OF AUCKLAND Te Whare Wānanga o Tāmaki Makaurau NEW ZEALAND
SCIENCE DEPARTMENT OF COMPUTER SCIENCE

storage media	access time[1]	affordable size	reliability
L1 cache reference	0.5 ns	most expensive	volatile
L2 cache reference	7 ns	most expensive	volatile
main memory	10^{-7} sec	4GB - 16GB	volatile
flash memory	read: 10^{-7} sec write: 10^{-7} sec	1GB - 1TB	non-volatile 10^3 — 10^5 write cycles
magnetic-disk	10^{-2} sec	500GB - 10TB	non-volatile
optical disc	0.1 sec	large	non-volatile
magnetic tapes	very slow	very large	non-volatile

Video on magnetic-disk, optical disc and magnetic tapes, edited on source.

[1] Numbers that everyone should know from Jeff Dean.

Jim Gray's Storage Latency Analogy: How Far Away is the Data?

10**9 tape	Andromeda		2,000yr
10**6 disk	Pluto		2yr
100 Memory	Pittsburgh		1.5h
10 On board cache	This building		10min
2 on chip cache	This room		1min
1 registers	In my head		

Storage Hierarchy

Reference: More specific numbers (from Fundamentals of Databases).

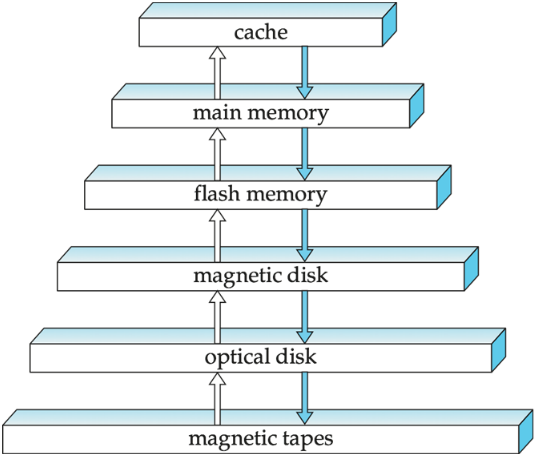
Type	Capacity*	Access Time	Max Bandwidth	Commodity Prices (2014)**
Main Memory- RAM	4GB–1TB	30ns	35GB/sec	$100–$20K
Flash Memory- SSD	64 GB–1TB	50μs	750MB/sec	$50–$600
Flash Memory- USB stick	4GB–512GB	100μs	50MB/sec	$2–$200
Magnetic Disk	400 GB–8TB	10ms	200MB/sec	$70–$500
Optical Storage	50GB–100GB	180ms	72MB/sec	$100
Magnetic Tape	2.5TB–8.5TB	10s–80s	40–250MB/sec	$2.5K–$30K
Tape jukebox	25TB–2,100,000TB	10s–80s	250MB/sec–1.2PB/sec	$3K–$1M+

*Capacities are based on commercially available popular units in 2014.

**Costs are based on commodity online marketplaces.

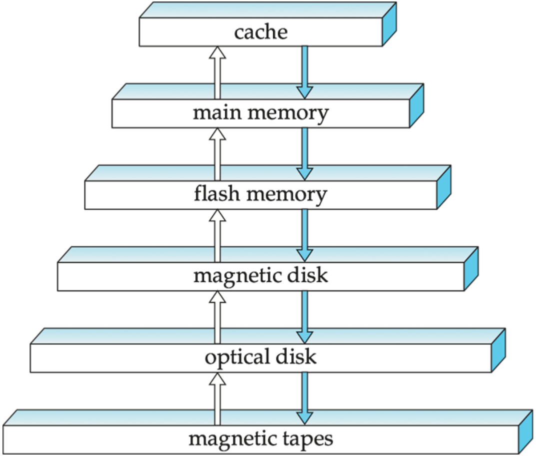
Figure: Types of Storage with Capacity, Access Time, Max Bandwidth (Transfer Speed), and Commodity Cost

Storage Hierarchy



- primary storage: fastest access but expensive and volatile.
- secondary storage: moderately fast access, moderately expensive, non-volatile.
- tertiary storage: slow access but cheap and non-volatile.

Storage Hierarchy



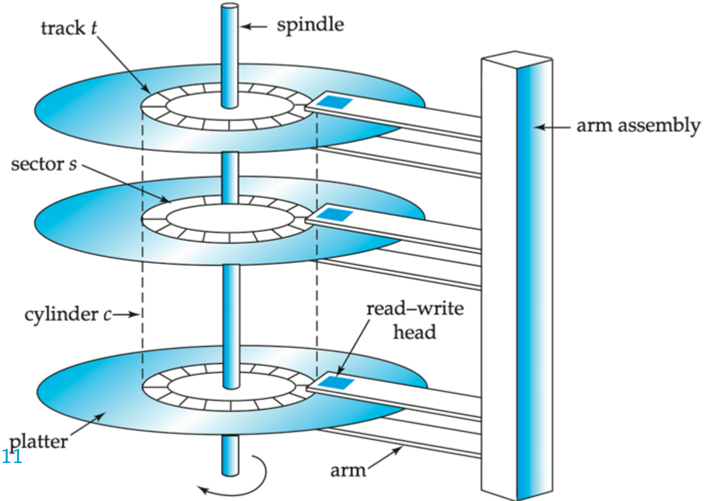
The data of a database is, typically,

- stored on magnetic disks and
- and partially copied to the main memory at running time.

Overview

1. Physical storage hierarchy
2. Secondary storage
3. Storage access
4. Redundant arrays of independent disks (RAID)
5. Organize records in a file

Magnetic Disk



- A platter surface consists of circular tracks
- A circular track consists of sectors
- A read-write head is close to a platter surface

Magnetic Disk

Time of moving data from sector x, track y to the memory:

- Access time: from an I/O request to seeing the first byte of data in the memory
  - Seek time (4-15 ms): receive an I/O request and move the disk arm to y-th track
  - Rotate delay (4-11 ms): rotate the spindle to align the read/write head to sector x
- Transfer delay (transfer rate: 50-130MB/s): rotate the spindle so that the data that stored consecutively on the disk can be transferred

Sequential access is far more efficient than random access! — why?

Consider: Cost of accessing 64 bytes of data in two cases:

- The data is stored sequentially.
- The data is stored randomly in 8 different places. $8\times$ slower

How to reconcile random access and sequential access? — Enforce each random access to read/write at least a certain amount of sequentially stored data.

Magnetic Disk

Blocks (or Pages)

The operating system (OS) formats/initialize the tracks of the disk into equal-sized blocks (or pages). Block size is fixed during initialization and cannot be changed dynamically. The OS transfers data between the main memory and the disk in blocks, one transfer of a block is called an I/O.

The block size ranges from 512 bytes to 8 KB.

Pros and cons of small and large block sizes, respectively.

- Small blocks: access time wasted
- Large blocks: space wasted due to partially filled blocks

Overview

1. Physical storage hierarchy
2. Secondary storage
3. Storage access
4. Redundant arrays of independent disks (RAID)
5. Organize records in a file

Storage access

- Buffer (or buffer pool)
  - Definition: the part of the available main memory used for storing copies of disk blocks.
  - Aim: to reduce the number of block transfers (I/O) between the disk and memory.

- Buffer manager: the subsystem which allocates the buffer space and manage the pages in the buffer pool. It maintains, for each page in the buffer,
  - a pin count: the number of current users (e.g., transactions or programs) of the page; if the pin-count goes to 0, the page is unpinned
  - a dirty bit: is set to 1 whenever the page is updated by any application program.

Storage access

The buffer manager handles read / write requests.

Read. Upon receiving a request of reading an item from a specific disk block X is posed, the buffer manager

- check if the disk block already resides in the buffer pool,
- if yes, return the memory address of the block without spending an I/O
- otherwise
  - allocate buffer pool space for the block and
  - if pool is full before the allocation, perform **buffer replacement**:
    - choose an unpinned page Y if there is any, otherwise wait under such a page Y exists
    - release the space of Y

      — if Y is dirty, **write** Y **back** to the disk which overwrites the original disk block of Y
  - load block X from the disk to the buffer and return the memory address of the block

Storage access

Write. Upon receiving a request of updating a data item on a specific disk block Z, the buffer manager

- check if Z is already in the buffer; if not, read Z from the disk to the buffer pool — as before, perform buffer replacement if necessary
- update the data item and mark the page (in the buffer pool) as dirty
- write the page back when a dirty page receives a **force output** instruction.

A force output is posed when the database expects some updates to the database to be persistent after a point of time.

Storage access

THE UNIVERSITY OF AUCKLAND Te Whare Wānanga o Tāmaki Makaurau NEW ZEALAND | SCIENCE DEPARTMENT OF COMPUTER SCIENCE

Buffer replacement. Recall that if the buffer is full when before allocating the space for a page, the buffer replaces one block W in the buffer:

- if W is clean, throw out W.
- if W is dirty, write W back to the disk.

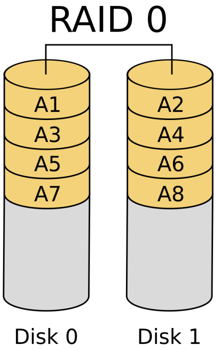
The block R that shall be replaced is chosen by polices

- Least recently used (LRU)
- Clock policy
- First-in-first-out (FIFO)
- Most recently used (MRU)
- ...

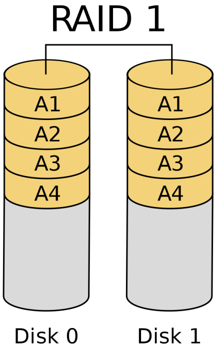
Overview

1. Physical storage hierarchy
2. Secondary storage
3. Storage access
4. Redundant arrays of independent disks (RAID)
5. Organize records in a file

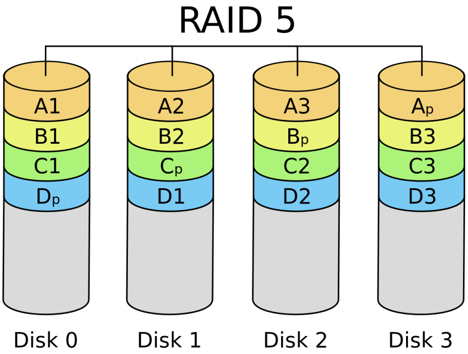
Redundant arrays of independent disks (RAID)



(a) Striping across 2 disks: adds performance but not reliability



(b) Mirroring across 2 disks: adds reliability but not performance (except for reads)



(c) Striping +1 parity disk: adds performance and reliability at lower storage cost

Overview

1. Physical storage hierarchy
2. Secondary storage
3. Storage access
4. Redundant arrays of independent disks (RAID)
5. Organize records in a file

Thank you!

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

SCIENCE
DEPARTMENT OF COMPUTER SCIENCE