External Memory Algorithms and Data Structures

Gerth S. Brodal, Rolf Fagerberg

Daimi

Fall 2000

Course

Lectures:

- Based on articles.
- Theoretical.
- New stuff: 1995-2000.

Evaluation:

- Programming projects (a few small and one large).
- Aim: Hands-on.

Homepage:

- `www.daimi.au.dk/~gerth/emF00`

Course

Literature:

- Papers.

Prerequisites:

- dADS and dAlg.

Credits:

- 2 points.

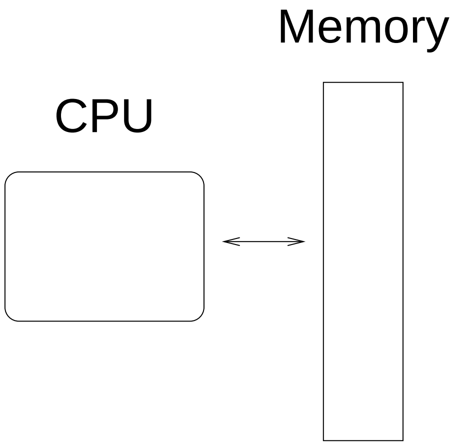
Course

Contents (approximate):

- The I/O model.
- Generic principles for designing I/O-efficient algorithms.
- Algorithms and lower bounds for basic problems (sorting, selection, permuting).
- I/O-efficient basic data structures (search trees, priority queues).
- I/O efficient algorithms and algorithms for problems from
  - computational geometry,
  - strings,
  - graphs.

Analysis of algorithms

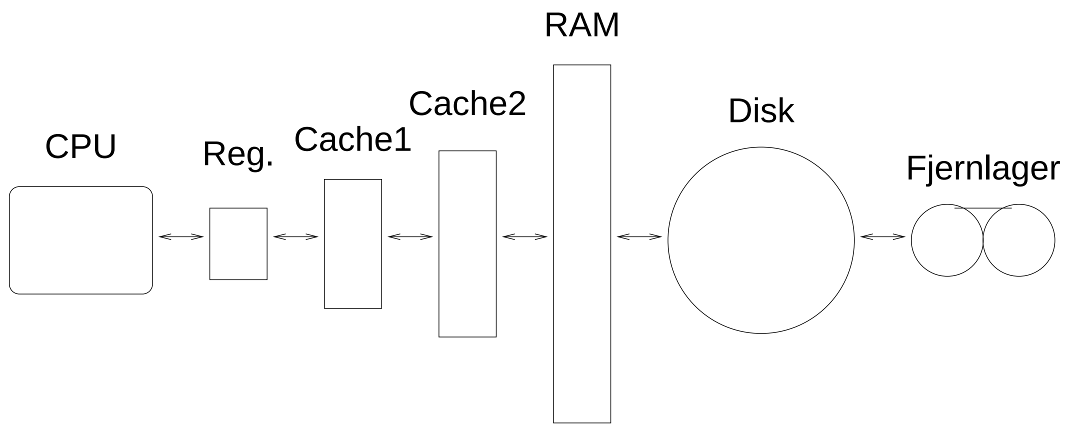
The standard model:



- Add: 1 unit of time
- Mult: 1 unit of time
- Branch: 1 unit of time
- MemAccess: 1 unit of time

Reality

Memory hierarchy:



	Access time	*Volume*
Registers	1 cycle	1 Kb
Cache	5 cycles	512 Kb
RAM	50 cycles	64 Mb
Disk	2,000,000 cycles	10 Gb

(Source: Data Structures and Algorithms in Java, *Goodrich and Tamassia*, 1998)

I/O bottleneck

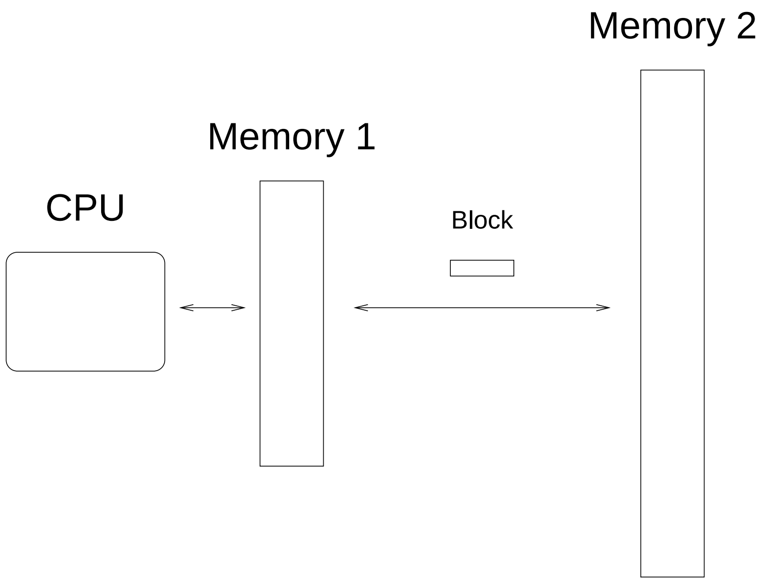
Many real-life problems of Gigabyte and Terabyte size:

- Geographic Information Systems (GIS).
- Computer graphics, animation.
- VLSI design.
- Databases (weather, geology, financial, WWW, phone companies, banks).

I/O bottleneck $\Rightarrow$ faster algorithms requires fewer I/O's.

Analysis of algorithms

New I/O-model:



Parameters:

$$
\begin{aligned}
N &= \text{no. of elements in problem.} \\
M &= \text{no. of elements that fits in RAM.} \\
B &= \text{no. of elements in a block on disk.} \\
D &= \text{no. of disks.}
\end{aligned}
$$

Cost: Number of I/O's (block transfers) between Memory 1 and Memory 2.

Generic Example

Consider two $O(n)$ algorithms:

1. Memory accessed randomly $\Rightarrow$ page fault at each memory access. One element used per I/O.
2. Memory accessed so that all elements of a block on disk are used right after another. B elements used per I/O.

$$O(n) \text{ I/O operations}$$
$$\text{vs.}$$
$$O(n/B) \text{ I/O operations}$$

Typically, $B \sim 1000$.