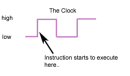


24

Clock rate on CPU

- Instruction execution synchronized to clock frequency
- Limited by how fast electrons move
- Smaller built \Rightarrow faster signal
- Higher frequency \Rightarrow faster computer

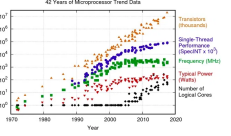


Power Wall
Energy dissipation \sim frequency³

25

"Moore's Law" and frequency

CPU frequency not increasing anymore
Increase in number of cores!




Age of parallelization

26

What makes computers fast?

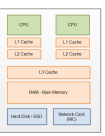
1. Optimized instruction sets
2. Reducing bottlenecks (caching...)
3. CPU clock frequency
4. Parallelization/Distributed



27

Multiprocessor

- Need to split tasks!
- CPUs can share memory or not
- Not everything can run in parallel
- Parallelization on multiple levels
 - Easy: run multiple scripts
 - Hard: inside your code




Generally, use other computing devices

28

Accelerators

Specialized ("hardwired" optimized connections) on specific computation

- Multiple CPUs
- GPU (Graphical PU)
- TPU (Tensorinatra PU) by Google
- FPU (Intelligence PU) by graphics

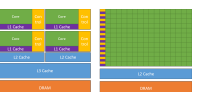


ASIC (Application Specific Integrated Circuit)
- FPGA (Field Programmable Gate Array)

29

GPUs & Vectorization

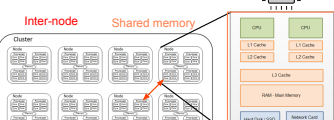
Single Instruction Multiple Data — SIMD
Apply same instruction to multiple values ("working with vectors")
Graphical Processing Unit (GPU) contains many SIMD units
- 500-5000 (simple) ALU



More in tomorrow's lecture!

30


Real world layout



31

Real world layout

How to add three arrays? 10, 1000, 10 mio entries?
What if we need to sum the entries up afterward?



32

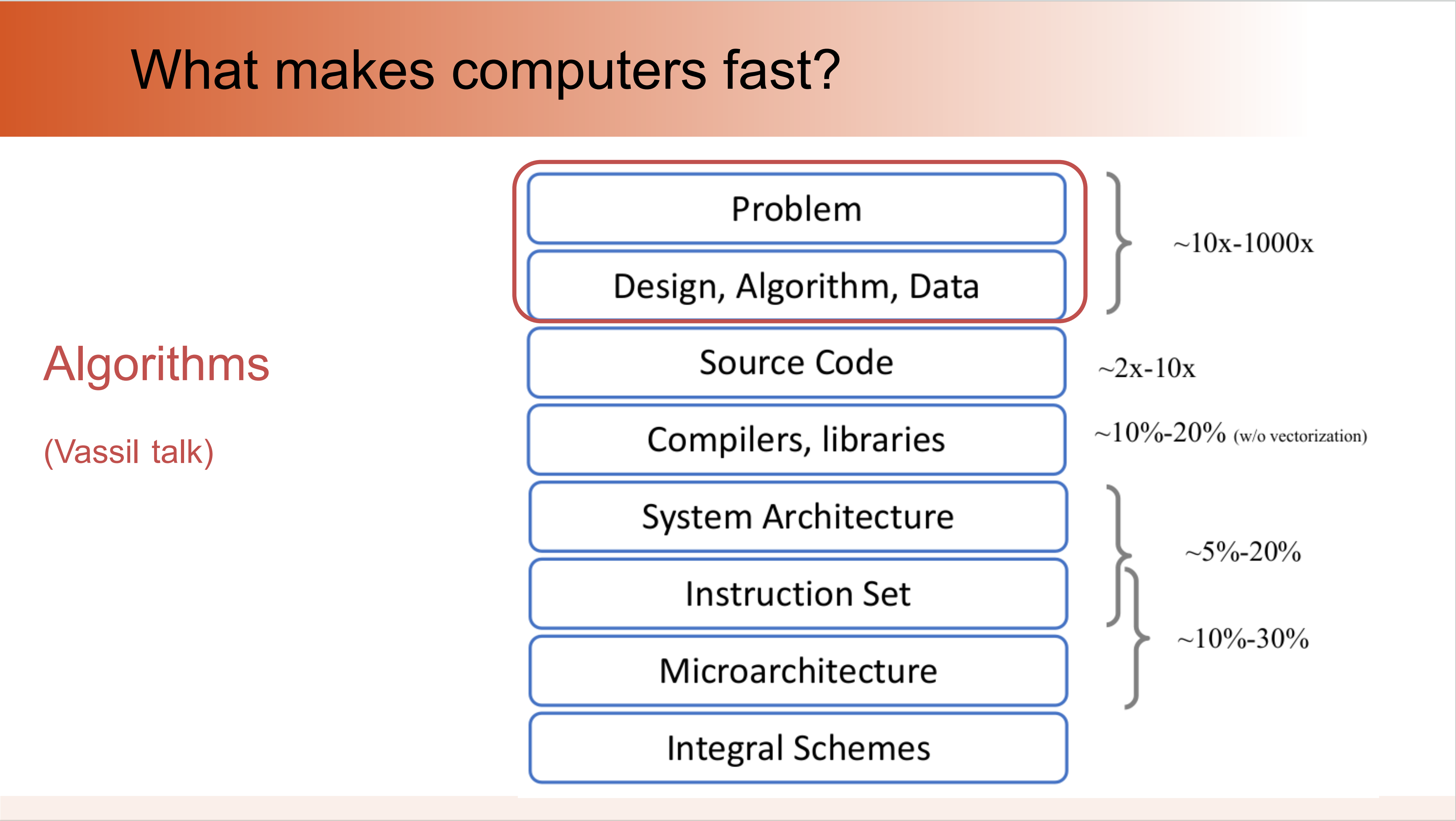
Real world layout

How to add three arrays? 10, 1000, 10 mio entries?
What if we need to sum the entries up afterward,
can we optimize the chain?
Doesn't fit into the memory?

Generalization \leftrightarrow Specialization

33

What makes computers fast?



Algorithms
(Vassil talk)